Oracle® Exadata System Software

User's Guide

19.3.0
E93159-09
February 2020

Oracle Exadata System Software User's Guide, 19.3.0

E93159-09

Primary Author: Janet Stern

Contributors: Ravikiran Akkayajhula, Andrew Babb, Nilesh Choudhury, Siddhartha Datta, Boris Erlikhman, Jaime Figueroa, Gurmeet Goindi, Cecilia Gervasio Grant, Roger Hansen, Mark Hollinger, Iori Honda, Kevin Jernigan, Yiliang Jin, Rakesh Kashyap, Frank Kobylanski, Kishy Kumar, Yang Liu, Juan Loaiza, Scott Martin, Krishnan Meiyyappan, Adrian Ng, Chien Nguyen, Michael Nowak, Dmitry Potapov, Darryl Presley, Ashish Ray, Akshay Shah, Vivek S. Sharma, Jia Shi, Kesavan Srinivasan, Mahesh Subramaniam, Alex Tsukerman, Shreyas Udgaonkar, Kothanda Umamageswaran, Doug Utzig, Zheren Zhang

Contents

Preface

1 Introducing Oracle Exadata System Software

2 Configuring Oracle Exadata System Software

3 Administering Oracle ASM Disk Groups on Oracle Exadata Storage Servers

5 Managing I/O Resources

6 Monitoring and Tuning Oracle Exadata System Software

8 Using the dcli Utility

9 Setting up Oracle Exadata Storage Snapshots

A Upgrading Oracle Exadata System Software

B Installation Information

C Automatic Workload Repository Views for Oracle Exadata System Software

D Oracle Exadata System Software Accessibility Recommendations

List of Examples

List of Figures

List of Tables

Preface

Oracle Exadata System Software User's Guide describes how to initialize and administer Oracle Exadata System Software. This guide describes the Oracle Exadata System Software product and its components, as well as Oracle Exadata System Software administrative and deployment procedures. This preface contains the following topics:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

Oracle Exadata System Software User's Guide is intended for Oracle Database and storage administrators who perform the following tasks:

- Configure Oracle Exadata System Software
- Manage Oracle Exadata System Software
- Troubleshoot Oracle Exadata System Software

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For additional information, see the following Oracle resources:

- *Oracle Exadata Database Machine System Overview*
- *Oracle Exadata Database Machine Installation and Configuration Guide*
- *Oracle Exadata Database Machine Maintenance Guide*
- *Oracle Engineered Systems Extending and Multi-Rack Cabling Guide*
- *Oracle Exadata Database Machine Security Guide*
- *Oracle Database 2 Day DBA*
- *Oracle Database Administrator’s Guide*
- *Oracle Database Concepts*
- *Oracle Automatic Storage Management Administrator's Guide*
- *Oracle Database Error Messages Reference*
- *Oracle Database 2 Day + Real Application Clusters Guide*
- *Oracle Clusterware Administration and Deployment Guide*
- *Oracle Real Application Clusters Administration and Deployment Guide*
- Platform-specific Oracle Database, Oracle Clusterware, and Oracle Real Application Clusters installation guides

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, emphasis, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.
$ prompt	The dollar sign ($) prompt indicates a command run as the `oracle` user.
# prompt	The pound (#) prompt indicates a command that is run as the `root` user.

1 Introducing Oracle Exadata System Software

This chapter introduces Oracle Exadata System Software.

- Overview of Oracle Exadata System Software
  Oracle Exadata Storage Server is a highly optimized storage server that runs Oracle Exadata System Software to store and access Oracle Database data.
- Key Features of Oracle Exadata System Software
  This section describes the key features of Oracle Exadata System Software.
- Oracle Exadata System Software Components
  This section provides a summary of the following Oracle Exadata System Software components.

1.1 Overview of Oracle Exadata System Software

Oracle Exadata Storage Server is a highly optimized storage server that runs Oracle Exadata System Software to store and access Oracle Database data.

With traditional storage, data is transferred to the database server for processing. In contrast, Oracle Exadata System Software provides database-aware storage services, such as the ability to offload SQL and other database processing from the database server, while remaining transparent to the SQL processing and database applications. Oracle Exadata Database Machine storage servers process data at the storage level, and pass only what is needed to the database servers.

Oracle Exadata System Software is installed on both the storage servers and the database servers. Oracle Exadata System Software offloads some SQL processing from the database server to the storage servers. Oracle Exadata System Software enables function shipping between the database instance and the underlying storage, in addition to traditional data shipping. Function shipping greatly reduces the amount of data processing that must be done by the database server. Eliminating data transfers and database server workload can greatly benefit query processing operations that often become bandwidth constrained. Eliminating data transfers can also provide a significant benefit to online transaction processing (OLTP) systems that include large batch and report processing operations.

The hardware components of Oracle Exadata Storage Server are carefully chosen to match the needs of high performance processing. The Oracle Exadata System Software is optimized to maximize the advantage of the hardware components. Each storage server delivers outstanding processing bandwidth for data stored on disk, often several times better than traditional solutions.

Oracle Exadata Database Machine storage servers use state-of-the-art RDMA Network Fabric interconnections between servers and storage. Each RDMA Network Fabric link provides bandwidth of 40 GB for RDMA over InfiniBand Network Fabric or 100 GB for RDMA over RoCE Network Fabric. Additionally, the interconnection protocol uses direct data placement, also referred to as direct memory access (DMA), to ensure low CPU overhead by directly moving data from the wire to database buffers with no extra copies. The RDMA Network Fabric has the flexibility of a LAN network with the efficiency of a storage area network (SAN). With an RDMA Network Fabric network, Oracle Exadata Database Machine eliminates network bottlenecks that could reduce performance. This RDMA Network Fabric network also provides a high performance cluster interconnection for Oracle Real Application Clusters (Oracle RAC) servers.

The Oracle Exadata Database Machine architecture scales to any level of performance. To achieve higher performance or greater storage capacity, you add more storage servers (cells) to the configuration. As more storage servers are added, capacity and performance increase linearly. Data is mirrored across storage servers to ensure that the failure of a storage server does not cause loss of data or availability. The scale-out architecture achieves near infinite scalability, while lowering costs by allowing storage to be purchased incrementally on demand.

> **Note:**
>
> Oracle Exadata System Software must be used with Oracle Exadata Database Machine storage server hardware, and only supports Oracle databases on the database servers of Oracle Exadata Database Machines. Information is available on My Oracle Support at
>
> `http://support.oracle.com`
>
> and on the Products page of Oracle Technology Network at
>
> `http://www.oracle.com/technetwork/index.html`

1.2 Key Features of Oracle Exadata System Software

This section describes the key features of Oracle Exadata System Software.

- Reliability, Modularity, and Cost-Effectiveness
  Oracle Exadata System Software enables cost-effective modular storage hardware to be used in a scale-out architecture while providing a high level of availability and reliability.
- Compatibility with Oracle Database
  When the minimum required versions are met, all Oracle Database features are fully supported with Oracle Exadata System Software.
- Smart Flash Technology
  The Exadata Smart Flash Cache feature of the Oracle Exadata System Software intelligently caches database objects in flash memory, replacing slow, mechanical I/O operations to disk with very rapid flash memory operations.
- Persistent Memory Accelerator and RDMA
  Persistent Memory (PMEM) Accelerator provides direct access to persistent memory using remote direct memory access (RDMA), enabling faster response times and lower read latencies.
- Centralized Storage
  You can use Oracle Exadata Storage Server to consolidate your storage requirements into a central pool that can be used by multiple databases.

- I/O Resource Management (IORM)
  I/O Resource Management (IORM) and the Oracle Database Resource Manager enable multiple databases and pluggable databases to share the same storage while ensuring that I/O resources are allocated across the various databases.
- In-Memory Columnar Format Support
  You can store data in the In-Memory columnar format in the flash cache in an Oracle Exadata Database Machine environment.
- Offloading of Data Search and Retrieval Processing
  One of the most powerful features of Oracle Exadata System Software is that it offloads the data search and retrieval processing to the storage servers.
- Offloading of Incremental Backup Processing
  To optimize the performance of incremental backups, the database can offload block filtering to Oracle Exadata Storage Server.
- Protection Against Data Corruption
  Data corruptions, while rare, can have a catastrophic effect on a database, and therefore on a business.
- Fast File Creation
  File creation operations are offloaded to Oracle Exadata Storage Servers.
- Storage Index
  Oracle Exadata Storage Servers maintain a storage index which contains a summary of the data distribution on the disk.

1.2.1 Reliability, Modularity, and Cost-Effectiveness

Oracle Exadata System Software enables cost-effective modular storage hardware to be used in a scale-out architecture while providing a high level of availability and reliability.

All single points of failure are eliminated in the Oracle Exadata Storage Server architecture by data mirroring, fault isolation technology, and protection against disk and other storage hardware failure.

In the Oracle Exadata Storage Server architecture, one or more storage cells can support one or more databases. The placement of data is transparent to database users and applications. Storage cells use Oracle Automatic Storage Management (Oracle ASM) to distribute data evenly across the cells. Because Oracle Exadata Storage Servers support dynamic disk insertion and removal, the online dynamic data redistribution feature of Oracle ASM ensures that data is appropriately balanced across the newly added, or remaining, disks without interrupting database processing. Oracle Exadata Storage Servers also provides data protection from disk and cell failures.

1.2.2 Compatibility with Oracle Database

When the minimum required versions are met, all Oracle Database features are fully supported with Oracle Exadata System Software.

Oracle Exadata System Software works equally well with single-instance or Oracle Real Application Clusters (Oracle RAC) deployments of Oracle Database. Oracle Data Guard, Oracle Recovery Manager (RMAN), Oracle GoldenGate, and other database features are managed the same with Exadata storage cells as with traditional storage.

This enables database administrators to use the same tools with which they are familiar.

Refer to My Oracle Support Doc ID 888828.1 for a complete list of the minimum required software versions.

Related Topics

- Exadata Database Machine and Exadata Storage Server Supported Versions (My Oracle Support Doc ID 888828.1)

1.2.3 Smart Flash Technology

The Exadata Smart Flash Cache feature of the Oracle Exadata System Software intelligently caches database objects in flash memory, replacing slow, mechanical I/O operations to disk with very rapid flash memory operations.

- Flash Cache
  Oracle has implemented smart flash cache directly in Oracle Exadata Storage Server.
- Flash Logging
  Oracle Exadata Smart Flash technology is also used to reduce the latency of log write I/O operations by eliminating performance bottlenecks that might occur due to database logging.
- WriteBack Flash Cache
  WriteBack flash cache provides the ability to cache write I/Os directly to PCI flash in addition to read I/Os.

1.2.3.1 Flash Cache

Oracle has implemented smart flash cache directly in Oracle Exadata Storage Server.

Oracle Exadata Smart Flash Cache holds frequently-accessed data in very fast flash storage while most data is kept in very cost-effective disk storage. This happens automatically without the user having to take any action.

Oracle Exadata Smart Flash Cache is smart because it knows when to avoid trying to cache data that will never be reused or will not fit in the cache. Oracle Database and Oracle Exadata System Software allow the user to provide directives at the database table, index and segment level to ensure that specific data is retained in flash. Tables can be moved in and out of flash with a simple command, without the need to move the table to different tablespaces, files or LUNs as is done with traditional storage using flash disks.

1.2.3.2 Flash Logging

Oracle Exadata Smart Flash technology is also used to reduce the latency of log write I/O operations by eliminating performance bottlenecks that might occur due to database logging.

The time to commit user transactions is very sensitive to the latency of log write operations. In addition, many performance-critical database algorithms, such as space management and index splits, are very sensitive to log write latency.

Although the disk controller has a large battery-backed DRAM cache that can accept writes very quickly, some write operations to disk can still be slow during periods of high I/O. Even with relatively few redo log write operations that are slow, these write operations can cause performance issues. It is these situations that Oracle Exadata Smart Flash Log is designed to alleviate.

The goal of the Oracle Exadata Smart Flash Log is to perform redo write operations simultaneously to both flash memory and disk, and complete the write operation when the first of the two completes. This gives Oracle Exadata Database Machine the best of both worlds by avoiding problems due to latency spikes on either type of media. Smart Flash Logging is most beneficial during busy periods when the disk controller cache occasionally becomes filled with blocks that have not been written to disk and therefore degrades to real disk performance versus disk cache performance. It is important to note that Smart Flash Logging improves latency of log write operations, but it does not improve total disk throughput. If an application is bottle-necked on disk throughput, then Smart Flash Logging can provide little benefit because log response time is not the limiting factor to performance.

It is also crucial to note that the purpose of Smart Flash Logging is not to use flash to consistently beat disk controller performance. It is used as an auxiliary destination that provides low latency when disks occasionally become slow, thus avoiding a negative impact on database performance.

Oracle Exadata Smart Flash Log improves user transaction response time, and increases overall database throughput for I/O intensive workloads by accelerating performance critical database algorithms.

Related Topics

- ALTER CELL

1.2.3.3 WriteBack Flash Cache

WriteBack flash cache provides the ability to cache write I/Os directly to PCI flash in addition to read I/Os.

The Flash Cache component on the Oracle Exadata storage cells can be configured in two ways: WriteThrough or WriteBack. WriteThrough cache reads IOs on the flash cache. In WriteBack mode, introduced with Oracle Exadata System Software release 11.2.3.2.0, all I/Os (reads/writes) are cached into the flash cache, boosting the performance of the databases.

WriteBack flash cache significantly improves the write intensive operations because writing to flash cache is faster than writing to hard disks. If your application writes intensively and if you find significant waits for "free buffer waits" or high I/O times, then you should consider using WriteBack flash cache.

Related Topics

- ALTER CELL

1.2.4 Persistent Memory Accelerator and RDMA

Persistent Memory (PMEM) Accelerator provides direct access to persistent memory using remote direct memory access (RDMA), enabling faster response times and lower read latencies.

Starting with Oracle Exadata System Software release 19.3.0, workloads that require ultra low response time such as stock trades and IOT devices can take advantage of PMEM and RDMA in the form of a PMEM Cache and PMEM Logging. PMEM is a new, persistent memory tier available on Oracle Exadata Storage Servers X8M-2 EF and HC and newer generations of Exadata Storage Servers with Persistent Memory (X*M). When clients read from the PMEM cache, Oracle Exadata System Software can do an RDMA read of the cached data, with much faster results compared to Flash Cache.

PMEM Cache can be used in the following configurations:

PMEM Cache Mode	Flash Cache Mode	Supported Configuration?
Write Through	Write Through	Yes. This is the default configuration for High Capacity servers with Normal Redundancy.
Write Through	Write Back	Yes. This is the default configuration for High Capacity servers with High Redundancy. This is also the default configuration for Extreme Flash servers.
Write Back	Write Back	Yes.
Write Back	Write Through	No. Write-intensive workloads can overload the write-back PMEM Cache.

PMEM Logging uses PMEM and RDMA to provide substantially lower redo log write latency. If the redo log resides on PMEM, clients can do an RDMA-write directly into the redo log. While this solution can provide low latencies, placing all redo log files on PMEM is cost prohibitive. If the redo log does not reside on PMEM, the Oracle Exadata System Software uses shared receive queues (SRQs) to send I/O buffers from the client to cellsrv via RDMA, and this reduces transport latencies. Cellsrv still performs the writes of the redo log data to disk (and flash, if Flash Logging is enabled). In cases when the `PMEMLOG` is bypassed due to lack of buffers or because it is disabled for a given database, Flash Logging is used instead.

Administrators can monitor PMEM Logging performance along with the appropriate Automatic Workload Repository (AWR) statistics either through reports or the `DBA_HIST_CELL_GLOBAL` and `DBA_HIST_CELL_GLOBAL_SUMMARY` views. Administrators can dynamically configure an IORM plan such that use of PMEM for the redo log is disabled for a given database or Oracle Automatic Storage Management (Oracle ASM) cluster. You can control PMEM Logging usage through the dbPlan and a new `pmemlog` attribute. The value for the `pmemlog` attribute must a percentage value in the range 0-100.

1.2.5 Centralized Storage

You can use Oracle Exadata Storage Server to consolidate your storage requirements into a central pool that can be used by multiple databases.

Oracle Exadata System Software with Oracle Automatic Storage Management (Oracle ASM) evenly distributes the data and I/O load for every database across available disks in the storage pool. Every database can use all of the available disks to achieve superior I/O rates. Oracle Exadata Storage Servers can provide higher efficiency and performance at a lower cost while also lowering your storage administration overhead.

1.2.6 I/O Resource Management (IORM)

I/O Resource Management (IORM) and the Oracle Database Resource Manager enable multiple databases and pluggable databases to share the same storage while ensuring that I/O resources are allocated across the various databases.

Oracle Exadata System Software works with IORM and Oracle Database Resource Manager to ensure that customer-defined policies are met, even when multiple databases share the grid. As a result, one database cannot monopolize the I/O bandwidth and degrade the performance of the other databases.

IORM enables storage cells to service I/O resources among multiple applications and users across all databases in accordance with sharing and prioritization levels established by the administrator. This improves the coexistence of online transaction processing (OLTP) and reporting workloads, because latency-sensitive OLTP applications can be given a larger share of disk and flash I/O bandwidth than throughput-sensitive batch applications. Oracle Database Resource Manager enables the administrator to control processor utilization on the database host on a per-application basis. Combining IORM and Oracle Database Resource Manager enables the administrator to establish more accurate policies.

IORM also manages the space utilization for Exadata Smart Flash Cache. Critical OLTP workloads can be guaranteed space in Exadata Smart Flash Cache to provide consistent performance.

IORM for a database or pluggable database (PDB) is implemented and managed from the Oracle Database Resource Manager. Oracle Database Resource Manager in the database instance communicates with the IORM software in the storage cell to manage user-defined service-level targets. Database resource plans are administered from the database, while interdatabase plans are administered on the storage cell.

Related Topics

- Managing I/O Resources

1.2.7 In-Memory Columnar Format Support

You can store data in the In-Memory columnar format in the flash cache in an Oracle Exadata Database Machine environment.

Oracle Exadata Database Machine supports all of the In-Memory optimizations, such as accessing only the compressed columns required, SIMD vector processing, storage indexes, and so on.

If you set the `INMEMORY_SIZE` database initialization parameter to a non-zero value (requires the Oracle Database In-Memory option), then objects accessed using a Smart Scan are brought into the flash cache and are automatically converted into the In-Memory columnar format. The data is converted initially into a columnar cache format, which is different from Oracle Database In-Memory's columnar format. The data is rewritten in the background into Oracle Database In-Memory columnar format. As a result, all subsequent accesses to the data benefit from all of the In-Memory optimizations when that data is retrieved from the flash cache.

A new segment-level attribute, `CELLMEMORY`, has also been introduced to help control which objects should not be populated into flash using the In-Memory columnar format and which type of compression should be used. Just like the `INMEMORY` attribute, you can specify different compression levels as sub-clauses to the CELLMEMORY attribute. However, not all of the INMEMORY compression levels are available; only MEMCOMPRESS FOR QUERY LOW and MEMCOMPRESS FOR CAPACITY LOW (default). You specify the CELLMEMORY attribute using a SQL command, such as the following:

```
ALTER TABLE trades CELLMEMORY MEMCOMPRESS FOR QUERY LOW
```

The PRIORTY sub-clause available with Oracle Database In-Memory is not available on Oracle Exadata Database Machine because the process of populating the flash cache on Exadata storage servers if different from populating DRAM in the In-Memory column store on Oracle Database servers.

1.2.8 Offloading of Data Search and Retrieval Processing

One of the most powerful features of Oracle Exadata System Software is that it offloads the data search and retrieval processing to the storage servers.

Oracle Exadata System Software does this by performing predicate filtering, which entails evaluating database predicates to optimize the performance of certain classes of bulk data processing.

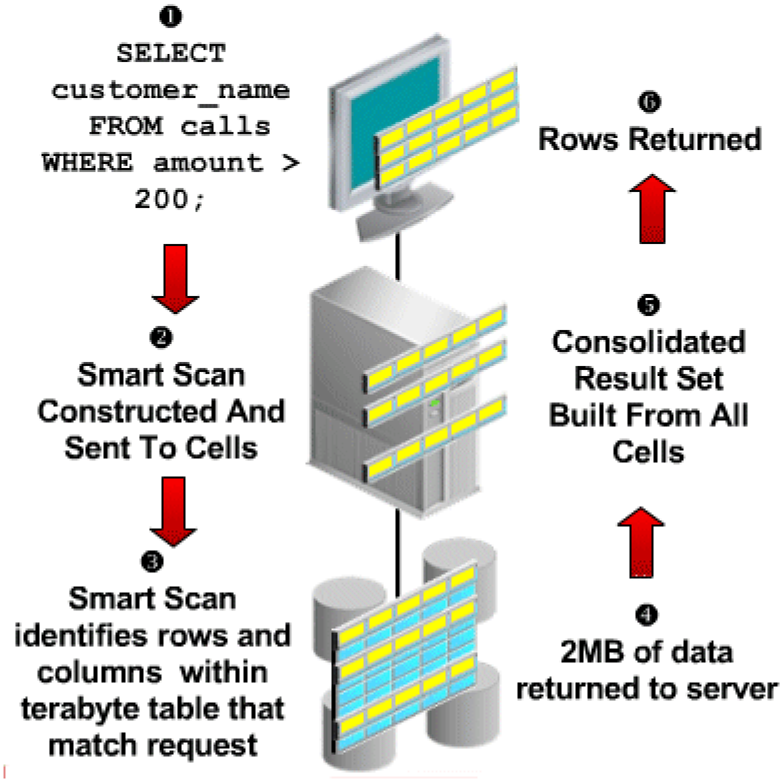
Oracle Database can optimize the performance of queries that perform table and index scans to evaluate selective predicates in Oracle Exadata Storage Server. The database can complete these queries faster by pushing the database expression evaluations to the storage cell. These expressions include simple SQL command predicates, such as amount > 200, and column projections, such as SELECT customer_name. For example:

```
SQL> SELECT customer_name FROM calls WHERE amount > 200;
```

In the preceding example, only rows satisfying the predicate, specified columns, and predicated columns are returned to the database server, eliminating unproductive data transfer to the database server.

Oracle Exadata System Software uses storage-side predicate evaluation that transfers simplified, predicate evaluation operations for table and index scans to the storage cell. This brings the table scan closer to the disk to enable a higher bandwidth, and prevents sending unmatched rows to hosts.

Figure 1-1   Offloading Data Search and Retrieval


❶
SELECT customer_name FROM calls WHERE amount > 200;
❷
Smart Scan Constructed And Sent To Cells
❸
Smart Scan identifies rows and columns within terabyte table that match request
❹
2MB of data returned to server
❺
Consolidated Result Set Built From All Cells
❻
Rows Returned

1.2.9 Offloading of Incremental Backup Processing

To optimize the performance of incremental backups, the database can offload block filtering to Oracle Exadata Storage Server.

This optimization is only possible when taking backups using Oracle Recovery Manager (RMAN). The offload processing is done transparently without user intervention. During offload processing, Oracle Exadata System Software filters out the blocks that are not required for the incremental backup in progress. Therefore, only the blocks that are required for the backup are sent to the database, making backups significantly faster.

Related Topics

- Using V$BACKUP_DATAFILE with Oracle Exadata Storage Server
  The `V$BACKUP_DATAFILE` view contains columns relevant to Oracle Exadata Storage Server during Oracle Recovery Manager (RMAN) incremental backups.
- Making Incremental Backups: Quick Start

1.2.10 Protection Against Data Corruption

Data corruptions, while rare, can have a catastrophic effect on a database, and therefore on a business.

Oracle Exadata System Software takes data protection to the next level by protecting business data, not just the physical bits.

The key approach to detecting and preventing corrupted data is block checking in which the storage subsystem validates the Oracle block contents. Oracle Database validates and adds protection information to the database blocks, while Oracle Exadata System Software detects corruptions introduced into the I/O path between the database and storage. It stops corrupted data from being written to disk, and validates data when reading the disk. This eliminates a large class of failures that the database industry had previously been unable to prevent.

Unlike other implementations of corruption checking, checks with Oracle Exadata System Software operate completely transparently. No parameters need to be set at the database or storage tier. These checks transparently handle all cases, including Oracle Automatic Storage Management (Oracle ASM) disk rebalance operations and disk failures.

1.2.11 Fast File Creation

File creation operations are offloaded to Oracle Exadata Storage Servers.

Operations such as `CREATE TABLESPACE`, which can create one or more files, have a significant increase in speed due to file creation offload.

1.2.12 Storage Index

Oracle Exadata Storage Servers maintain a storage index which contains a summary of the data distribution on the disk.

The storage index is maintained automatically, and is transparent to Oracle Database. It is a collection of in-memory region indexes, and each region index stores summaries for up to eight columns. There is one region index for each 1 MB of disk space. Storage indexes work with any non-linguistic data type, and work with linguistic data types similar to non-linguistic indexes.

Each region index maintains the minimum and maximum values of the columns of the table. The minimum and maximum values are used to eliminate unnecessary I/O, also known as **I/O filtering**. The cell physical IO bytes saved by storage index statistic, available in the `V$SYS_STAT` view, shows the number of bytes of I/O saved using storage index. The content stored in one region index is independent of the other region indexes. This makes them highly scalable, and avoids latch contention.

Queries using the following comparisons are improved by the storage index:

- Equality (=)
- Inequality (<, !=, or >)
- Less than or equal (<=)
- Greater than or equal (>=)
- IS NULL
- IS NOT NULL

Storage indexes are built automatically after Oracle Exadata System Software receives a query with a comparison predicate that is greater than the maximum or less than the minimum value for the column in a region, and would have benefited if a storage index had been present. Oracle Exadata System Software automatically learns which storage indexes would have benefited a query, and then creates the storage index automatically so that subsequent similar queries benefit.

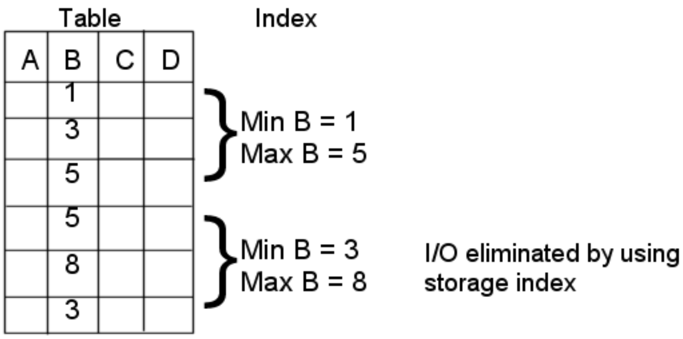
> **Note:**
>
> The effectiveness of storage indexes can be improved by ordering the rows based on columns that frequently appear in WHERE query clauses.

> **Note:**
>
> The storage index is maintained during write operations to uncompressed blocks and OLTP compressed blocks. Write operations to Exadata Hybrid Columnar CompressionExadata Hybrid Columnar Compression compressed blocks or encrypted tablespaces invalidate a region index, but not the storage index. The storage index for Exadata Hybrid Columnar Compression is rebuilt on subsequent scans.

> **See Also:**
>
> Using V$SYSSTAT with Oracle Exadata System Software

Example 1-1 Elimination of Disk I/O with Storage Index

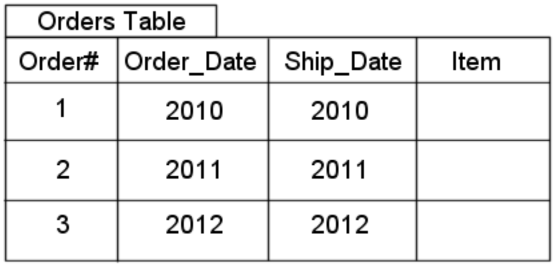
The following figure shows a table and region indexes. The values in the table range from one to eight. One region index stores the minimum 1, and the maximum of 5. The other region index stores the minimum of 3, and the maximum of 8.



For a query such as SELECT * FROM TABLE WHERE B<2, only the first set of rows match. Disk I/O is eliminated because the minimum and maximum of the second set of rows do not match the WHERE clause of the query.

Example 1-2 Partition Pruning-like Benefits with Storage Index

In the following figure, there is a table named Orders with the columns Order_Number, Order_Date, Ship_Date, and Order_Item. The table is range partitioned by Order_Date column.



Orders Table

Order#	Order_Date	Ship_Date	Item
1	2010	2010	
2	2011	2011	
3	2012	2012	

The following query looks for orders placed since January 1, 2015:

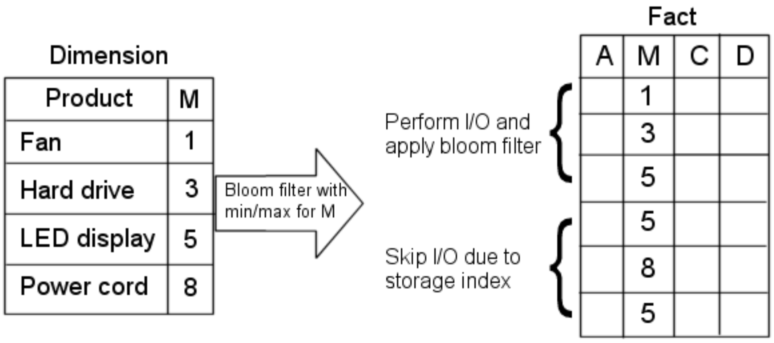
```
SELECT count (*) FROM Orders WHERE Order_Date >= to_date ('2015-01-01', \
'YYY-MM-DD')
```

Because the table is partitioned on the `Order_Date` column, the preceding query avoids scanning unnecessary partitions of the table. Queries on `Ship_Date` do not benefit from `Order_Date` partitioning, but `Ship_Date` and `Order_Number` are highly correlated with `Order_Date`. Storage indexes take advantage of ordering created by partitioning or sorted loading, and can use it with the other columns in the table. This provides partition pruning-like performance for queries on the `Ship_Date` and `Order_Number` columns.

Example 1-3 Improved Join Performance Using Storage Index

Using storage index allows table joins to skip unnecessary I/O operations. For example, the following query would perform an I/O operation and apply a Bloom filter to only the first block of the fact table.

```
SELECT count(*) FROM fact, dim WHERE fact.m=dim.m AND dim.product="Hard
drive"
```



Dimension

Product	M
Fan	1
Hard drive	3
LED display	5
Power cord	8

Fact

A	M	C	D
	1		
	3		
	5		
	5		
	8		
	5		

Bloom filter with min/max for M
Perform I/O and apply bloom filter
Skip I/O due to storage index

The I/O for the second block of the fact table is completely eliminated by storage index as its minimum/maximum range (5,8) is not present in the Bloom filter.

1.3 Oracle Exadata System Software Components

This section provides a summary of the following Oracle Exadata System Software components.

- About Oracle Exadata System Software
  Unique software algorithms in Oracle Exadata System Software implement database intelligence in storage, PCI-based flash, and RDMA Network Fabric networking to deliver higher performance and capacity at lower costs than other platforms.
- About Oracle Automatic Storage Management
  Oracle Automatic Storage Management (Oracle ASM) is the cluster volume manager and file system used to manage Oracle Exadata Storage Server resources.
- About Grid RAID
  A grid Redundant Array of Independent Disks (RAID) configuration uses Oracle ASM mirroring capabilities.
- About Storage Server Security
  Security for Exadata Storage Servers is enforced by identifying which clients can access storage servers and grid disks.
- About iDB Protocol
  The *i*DB protocol is a unique Oracle data transfer protocol that serves as the communications protocol among Oracle ASM, database instances, and storage cells.
- About Oracle Exadata System Software Processes
  Oracle Exadata System Software uses its own set of background processes.
- About Cell Management
  Each cell in the Oracle Exadata Storage Server grid is individually managed with Cell Control Command-Line Interface (CellCLI).
- About Database Server Software
  Oracle software is installed on the Exadata database servers.
- About Oracle Enterprise Manager for Oracle Exadata Database Machine
  Oracle Enterprise Manager provides a complete target that enables you to monitor Oracle Exadata Database Machine, including configuration and performance, in a graphical user interface (GUI).

1.3.1 About Oracle Exadata System Software

Unique software algorithms in Oracle Exadata System Software implement database intelligence in storage, PCI-based flash, and RDMA Network Fabric networking to deliver higher performance and capacity at lower costs than other platforms.

Oracle Exadata Storage Server is a network-accessible storage device with Oracle Exadata System Software installed on it. The software communicates with the database using a specialized *i*DB protocol, and provides both simple I/O functionality, such as block-oriented reads and writes, and advanced I/O functionality, including predicate offload and I/O Resource Management (IORM). Each storage server has physical disks. The physical disk is an actual device within the storage server that constitutes a single disk drive spindle.

Within the storage servers, a logical unit number (LUN) defines a logical storage resource from which a single cell disk can be created. The LUN refers to the access point for storage resources presented by the underlying hardware to the upper software layers. The precise attributes of a LUN are configuration-specific. For example, a LUN could be striped, mirrored, or both striped and mirrored.

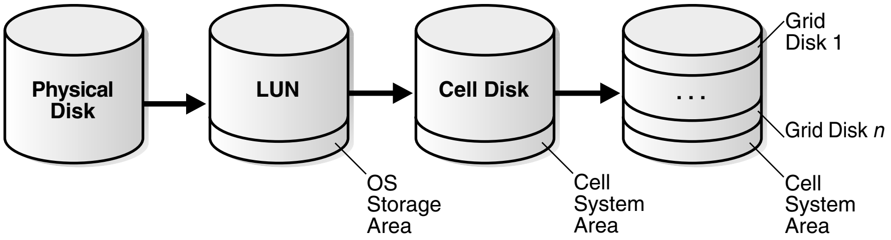
A cell disk is an Oracle Exadata System Software abstraction built on the top of a LUN. After a cell disk is created from the LUN, it is managed by Oracle Exadata System Software and can be further subdivided into grid disks, which are directly exposed to the database and Oracle Automatic Storage Management (Oracle ASM) instances. Each grid disk is a potentially non-contiguous partition of the cell disk that is directly exposed to Oracle ASM to be used for the Oracle ASM disk group creations and expansions.

This level of virtualization enables multiple Oracle ASM clusters and multiple databases to share the same physical disk. This sharing provides optimal use of disk capacity and bandwidth. Various metrics and statistics collected on the cell disk level enable you to evaluate the performance and capacity of storage servers. IORM schedules the cell disk access in accordance with user-defined policies.

The following image illustrates how the components of a storage server (also called a **cell**) are related to grid disks.

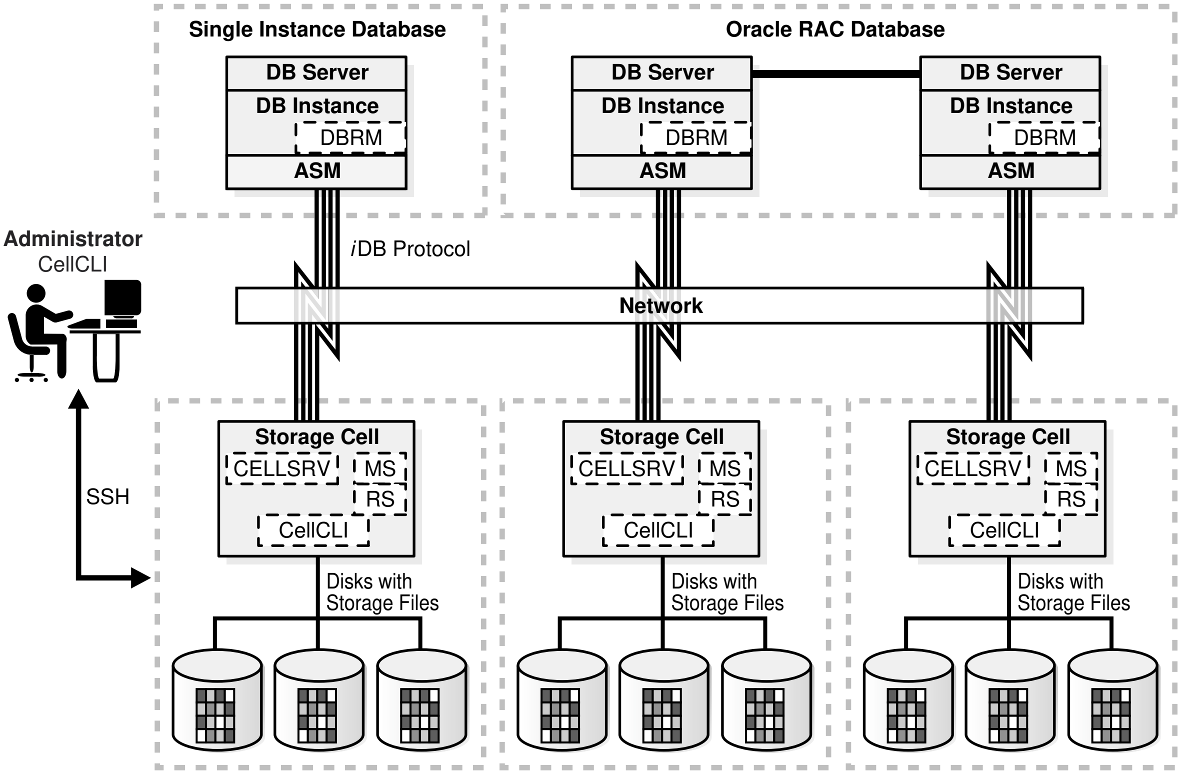
- A LUN is created from a physical disk.
- A cell disk is created on a LUN. A segment of cell disk storage is used by the Oracle Exadata System Software system, referred to as the cell system area.
- Multiple grid disks can be created on a cell disk.

Figure 1-2 Oracle Exadata Storage Server Components



The following image illustrates software components in the Oracle Exadata Storage Server environment.

Figure 1-3 Software Components in the Oracle Exadata Database Machine Environment



The figure illustrates the following environment:

- Single-instance or Oracle RAC databases access storage servers using the *i*DB protocol over a RDMA Network Fabric network. Each database server runs the Oracle Database and Oracle Grid Infrastructure software. Resources are managed for each database instance by Oracle Database Resource Manager (shown as DBRM).
- The database servers include Oracle Exadata System Software functionality, such as a Management Server (MS) and command-line interface (DBMCLI).
- Storage servers contain cell-based utilities and processes from Oracle Exadata System Software, including:
  - Cell Server (CELLSRV)—the primary component of the Oracle Exadata System Software running in the storage server, which provides the majority of the storage server services. CELLSRV services database requests for disk I/O and provides the advanced SQL offload capabilities. CELLSRV implements the I/O Resource Management (IORM) functionality to meter out I/O bandwidth to the various databases and consumer groups issuing I/O calls on the storage server.
  - Management Server (MS)—the primary interface to administer, manage and query the status of the storage server. It works in cooperation with the Cell Control Command-Line Interface (CellCLI) and processes most of the commands from CellCLI.

  - Restart Server (RS)—monitors the heartbeat with the MS and the CELLSRV processes, and restarts the servers if they fail to respond within the allowable heartbeat period.
- Storage cells are configured on the network, and are managed by the Oracle Exadata System Software CellCLI utility.
- Each storage server contains multiple disks which store the data for the database instances on the database servers. The data is stored in disks managed by Oracle ASM.

1.3.2 About Oracle Automatic Storage Management

Oracle Automatic Storage Management (Oracle ASM) is the cluster volume manager and file system used to manage Oracle Exadata Storage Server resources.

Oracle ASM provides enhanced storage management by:

- Striping database files evenly across all available storage cells and disks for optimal performance.
- Using mirroring and failure groups to avoid any single point of failure.
- Enabling dynamic add and drop capability for non-intrusive cell and disk allocation, deallocation, and reallocation.
- Enabling multiple databases to share storage cells and disks.

The following topics provide a brief overview of Oracle ASM:

- Oracle ASM Disk Groups
  An Oracle Automatic Storage Management (Oracle ASM) disk group is the primary storage abstraction within Oracle ASM, and is composed of one or more grid disks.
- Oracle ASM Failure Group
  An Oracle ASM failure group is a subset of disks in an Oracle ASM disk group that can fail together because they share the same hardware.
- Maximum Availability with Oracle ASM
  Oracle recommends high redundancy Oracle ASM disk groups, and file placement configuration which can be automatically deployed using Oracle Exadata Deployment Assistant.

Related Topics

- Overview of Oracle Automatic Storage Management

1.3.2.1 Oracle ASM Disk Groups

An Oracle Automatic Storage Management (Oracle ASM) disk group is the primary storage abstraction within Oracle ASM, and is composed of one or more grid disks.

Oracle Exadata Storage Server grid disks appear to Oracle ASM as individual disks available for membership in Oracle ASM disk groups. Whenever possible, grid disk names should correspond closely with Oracle ASM disk group names to assist in problem diagnosis between Oracle ASM and Oracle Exadata System Software.

The Oracle ASM disk groups are as follows:

- DATA is the data disk group.

- RECO is the recovery disk group.
- DBFS (Oracle Database File System) is the file system disk group.
- SPARSE is a sparse disk group to keep snapshot files.

To take advantage of Oracle Exadata System Software features, such as predicate processing offload, the disk groups must contain only Oracle Exadata Storage Server grid disks, and the tables must be fully inside these disk groups.

> **Note:**
>
> The Oracle Database and Oracle Grid Infrastructure software must be release 12.1.0.2.0 BP3 or later when using sparse grid disks.

Related Topics

- *Oracle Automatic Storage Management Administrator's Guide*

1.3.2.2 Oracle ASM Failure Group

An Oracle ASM failure group is a subset of disks in an Oracle ASM disk group that can fail together because they share the same hardware.

Oracle ASM considers failure groups when making redundancy decisions.

For Oracle Exadata Storage Servers, all grid disks, which consist of the Oracle ASM disk group members and candidates, can effectively fail together if the storage cell fails. Because of this scenario, all Oracle ASM grid disks sourced from a given storage cell should be assigned to a single failure group representing the cell.

For example, if all grid disks from two storage cells, A and B, are added to a single Oracle ASM disk group with normal redundancy, then all grid disks on storage cell A are designated as one failure group, and all grid disks on storage cell B are designated as another failure group. This enables Oracle Exadata System Software and Oracle ASM to tolerate the failure of either storage cell.

Failure groups for Oracle Exadata Storage Server grid disks are set by default so that the disks on a single cell are in the same failure group, making correct failure group configuration simple for Oracle Exadata Storage Servers.

You can define the redundancy level for an Oracle ASM disk group when creating a disk group. An Oracle ASM disk group can be specified with normal or high redundancy. Normal redundancy double mirrors the extents, and high redundancy triple mirrors the extents. Oracle ASM normal redundancy tolerates the failure of a single cell or any set of disks in a single cell. Oracle ASM high redundancy tolerates the failure of two cells or any set of disks in two cells. Base your redundancy setting on your desired protection level. When choosing the redundancy level, ensure the post-failure I/O capacity is sufficient to meet the redundancy requirements and performance service levels. Oracle recommends using three cells for normal redundancy. This ensures the ability to restore full redundancy after cell failure. Consider the following:

- If a cell or disk fails, then Oracle ASM automatically redistributes the cell or disk contents across the remaining disks in the disk group as long as there is enough space to hold the data. For an existing disk group using Oracle ASM redundancy, the `USABLE_FILE_MB` and `REQUIRED_FREE_MIRROR_MB` columns in the

V$ASM_DISGKROUP view give the amount of usable space and space for redundancy, respectively.

- If a cell or disk fails, then the remaining disks should be able to generate the IOPS necessary to sustain the performance service level agreement.

After a disk group is created, the redundancy level of the disk group cannot be changed. To change the redundancy of a disk group, you must create another disk group with the appropriate redundancy, and then move the files.

Each Exadata Cell is a failure group. A normal redundancy disk group must contain at least two failure groups. Oracle ASM automatically stores two copies of the file extents, with the mirrored extents placed in different failure groups. A high redundancy disk group must contain at least three failure groups. Oracle ASM automatically stores three copies of the file extents, with each file extent in separate failure groups.

System reliability can diminish if your environment has an insufficient number of failure groups. A small number of failure groups, or failure groups of uneven capacity, can lead to allocation problems that prevent full use of all available storage.

Related Topics

- Administering Oracle ASM Disk Groups Using Oracle Exadata Storage Servers

1.3.2.3 Maximum Availability with Oracle ASM

Oracle recommends high redundancy Oracle ASM disk groups, and file placement configuration which can be automatically deployed using Oracle Exadata Deployment Assistant.

High redundancy can be configured for DATA, RECO or any other Oracle ASM group with a minimum of 3 storage cells. Starting with Exadata Software release 12.1.2.3.0, the voting disks can reside in a high redundancy disk group, and additional quorum disks (essentially equivalent to voting disks) can reside on database servers if there are fewer than 5 Exadata storage cells.

Maximum availability architecture (MAA) best practice uses three Oracle ASM disk groups, DATA, RECO, and DBFS. The disk groups are located as follows:

- The disk groups are striped across all disks and Oracle Exadata Storage Servers to maximize I/O bandwidth and performance, and simplify management.
- The DATA disk group is located on the outer section of all disks. This is true only for physical deployments. This is not applicable to Oracle VM deployments.
- The RECO disk group is located on the outer/inner section of all disks. This is true only for physical deployments. This is not applicable to Oracle VM deployments.
- The DBFS disk group is located on the inner section of all disks.
- The DATA and RECO disk groups are configured for high redundancy.

The preceding attributes ensure optimal file placement in the different Oracle ASM disk groups. In addition, all operations have access to full I/O bandwidth, when needed. To avoid excessive resource consumption, use I/O Resource Management, Oracle Database Resource Manager, and instance caging.

The benefits of high redundancy disk groups are illustrated by the following outage scenarios:

- Double partner disk failure: Protection against loss of the database and Oracle ASM disk group due to a disk failure followed by a second partner disk failure.
- Disk failure when Oracle Exadata Storage Server is offline: Protection against loss of the database and Oracle ASM disk group when a storage server is offline and one of the storage server's partner disks fails. The storage server may be offline because of Exadata storage planned maintenance, such as Exadata rolling storage server patching.
- Disk failure followed by disk sector corruption: Protection against data loss and I/O errors when latent disk sector corruptions exist and a partner storage disk is unavailable either due to planned maintenance or disk failure.

If the voting disk resides in a high redundancy disk group that is part of the default Exadata high redundancy deployment, the cluster and database will remain available for the above failure scenarios. If the voting disk resides on a normal redundancy disk group, then the database cluster will fail and the database has to be restarted. You can eliminate that risk by moving the voting disks to a high redundancy disk group and creating additional quorum disks on database servers.

Oracle recommends High Redundancy for ALL (DATA and RECO) disk groups because it provides maximum application availability against storage failures and operational simplicity during a storage outage. In contrast, if all disk groups were configured with normal redundancy and two partner disks fail, all clusters and databases on Exadata will fail and you will lose all your data (normal redundancy does not survive double partner disk failures). Other than better storage protection, the major difference between high redundancy and normal redundancy is the amount of usable storage and write I/Os. High redundancy requires more space, and has three write I/Os instead of two. The additional write I/O normally has negligible impact with Exadata smart write-back flash cache.

The following table describes that redundancy option, as well as others, and the relative availability trade-offs. The table assumes that voting disks reside in high redundancy disk group. Refer to *Oracle Exadata Database Machine Maintenance Guide* to migrate voting disks to high redundancy disk group for existing high redundancy disk group configurations.

Redundancy Option	Availability Implications	Recommendation
High Redundancy for ALL (DATA and RECO)	Zero application downtime and zero data loss for the preceding storage outage scenarios if voting disks reside in high redundancy disk group. If voting disks currently reside in normal redundancy disk group, refer to *Oracle Exadata Database Machine Maintenance Guide* to migrate them to high redundancy disk group.	Use this option for best storage protection and operational simplicity for mission-critical applications. Requires more space for higher redundancy.

Redundancy Option	Availability Implications	Recommendation
High Redundancy for DATA only	Zero application downtime and zero data loss for preceding storage outage scenarios. This option requires an alternative archive destination.	New default deployment configuration with 8 TB disks. Use this option for best storage protection for DATA with slightly higher operational complexity. More available space than High Redundancy for ALL. Refer to My Oracle Support note 2059780.1 for details.
High Redundancy for RECO only	Zero data loss for the preceding storage outage scenarios.	Use this option when longer recovery times are acceptable for the preceding storage outage scenarios. Recovery options include the following: • Restore and recover: - Recreate DATA disk group - Restore from RECO and tape-based backups, if required - Recover database • Switch and recover: - Use RMAN switch to copy Recover database
Normal Redundancy for ALL (DATA and RECO) **Note:** Cross-disk mirror isolation by using ASM disk group content type limits an outage to a single disk group when two disk partners are lost in a normal redundancy group that share physical disks and storage servers.	The preceding storage outage scenarios resulted in failure of all Oracle ASM disk groups. However, using cross-disk group mirror isolation the outage is limited to one disk group. **Note:** This option is not available for eighth or quarter racks.	Oracle recommends a minimum of High Redundancy for DATA only. Use the Normal Redundancy for ALL option when the primary database is protected by an Oracle Data Guard standby database deployed on a separate Oracle Exadata Database Machine or when the Exadata Database Machine is servicing only development or test databases. Oracle Data Guard provides real-time data protection and fast failover for storage failures. If Oracle Data Guard is not available and the DATA or RECO disk groups are lost, then leverage recovery options described in My Oracle Support note 1339373.1.

The optimal file placement for setup for MAA is:

- **Oracle Database files** — DATA disk group

- **Flashback log files, archived redo files, and backup files** — RECO disk group
- **Redo log files** — First high redundancy disk group. If no high redundancy disk group exists, then redo log files are multiplexed across the DATA and RECO disk groups.
- **Control files** — First high redundancy disk group. If no high redundancy disk groups exist, the use one control file in the DATA disk group. The backup control files should reside in the RECO disk group, and `RMAN CONFIGURE CONTROLFILE AUTOBACKUP ON` should be set.
- **Server parameter files (SPFILE)** — First high redundancy disk group. If no high redundancy disk group exists, then SPFILE should reside in the DATA disk group. SPFILE backups should reside in the RECO disk group.
- **Oracle Cluster Registry (OCR) and voting disks for Oracle Exadata Database Machine Full Rack and Oracle Exadata Database Machine Half Rack** — First high redundancy disk group. If no high redundancy disk group exists, then the files should reside in the DATA disk group.
- **Voting disks for Oracle Exadata Database Machine Quarter Rack or Eighth Rack** — First high redundancy disk group, otherwise in normal redundancy disk group. If there are fewer than 5 Exadata storage cells with high redundancy disk group, additional quorum disks will be stored on Exadata database servers during OEDA deployment. Refer to *Oracle Exadata Database Machine Maintenance Guide* to migrate voting disks to high redundancy disk group for existing high redundancy disk group configurations.
- **Temporary files** — First normal redundancy disk group. If the High Redundancy for ALL option is used, then the use the first high redundancy disk group.
- **Staging and non-database files** — DBFS disk group

Related Topics

- Database High Availability Checklist
- Configuration Prerequisites and Operational Steps for Higher Availability for a RECO disk group or Fast Recovery Area Failure (My Oracle Support Doc ID 2059780.1)
- Operational Steps for Recovery after Losing a Disk Group in an Exadata Environment (My Oracle Support Doc ID 1339373.1)

1.3.3 About Grid RAID

A grid Redundant Array of Independent Disks (RAID) configuration uses Oracle ASM mirroring capabilities.

To use grid RAID, you place grid disks in an Oracle ASM disk group with a normal or high redundancy level, and set all grid disks in the same cell to be in the same Oracle ASM failure group. This ensures that Oracle ASM does not mirror data extents using disks within the cell. Using disks from different cells ensures that an individual cell failure does not cause the data to be unavailable.

Grid RAID also provides simplified creation of cell disks. With grid RAID, LUNs are automatically created from available physical disks because Oracle software automatically creates the required LUNs.

1.3.4 About Storage Server Security

Security for Exadata Storage Servers is enforced by identifying which clients can access storage servers and grid disks.

Clients include Oracle ASM instances, database instances, and clusters. When creating or modifying grid disks, you can configure the Oracle ASM owner and the database clients that are allowed to use those grid disks.

Related Topics

- About Exadata Storage Server Security Modes

1.3.5 About iDB Protocol

The *i*DB protocol is a unique Oracle data transfer protocol that serves as the communications protocol among Oracle ASM, database instances, and storage cells.

General-purpose data transfer protocols operate only on the low-level blocks of a disk. In contrast, the *i*DB protocol is aware of the Oracle internal data representation and is the necessary complement to Exadata storage server specific features, such as predicate processing offload.

In addition, the *i*DB protocol provides interconnection bandwidth aggregation and failover.

1.3.6 About Oracle Exadata System Software Processes

Oracle Exadata System Software uses its own set of background processes.

Oracle Exadata System Software includes the following software processes:

- Cell Server (CELLSRV) services *i*DB requests for disk I/O and advanced Oracle Exadata Storage Server services, such as predicate processing offload. CELLSRV is implemented as a multithreaded process and should be expected to use the largest portion of processor cycles on a storage cell.
- Management Server (MS) provides standalone storage cell management and configuration. MS runs on an Oracle WebLogic Server.
- Restart Server (RS) monitors the CELLSRV and MS processes and restarts them, if necessary.

1.3.7 About Cell Management

Each cell in the Oracle Exadata Storage Server grid is individually managed with Cell Control Command-Line Interface (CellCLI).

The CellCLI utility provides a command-line interface to the cell management functions, such as cell initial configuration, cell disk and grid disk creation, and performance monitoring. The CellCLI utility runs on the cell, and is accessible from a client computer that has network access to the storage cell or is directly connected to the cell. The CellCLI utility communicates with Management Server to administer the storage cell.

To access the cell, you should either use Secure Shell (SSH) access, or local access, for example, through a KVM switch (keyboard, video or visual display unit, mouse) switch. SSH allows remote access, but local access might be necessary during the initial configuration when the cell is not yet configured for the network. With local access, you have access to the cell operating system shell prompt and use various tools, such as the CellCLI utility, to administer the cell.

You can run the same CellCLI commands remotely on multiple cells with the dcli utility.

To manage a cell remotely from a compute node, you can use the ExaCLI utility. ExaCLI enables you to run most CellCLI commands on a cell. This is necessary if you do not have direct access to a cell to run CellCLI, or if SSH service on the cell has been disabled. To run commands on multiple cells remotely, you can use the `exadcli` utility.

Related Topics

- Using the CellCLI Utility
  You use the Cell Control Command-Line Interface (CellCLI) utility to manage Oracle Exadata System Software.
- Using the dcli Utility
  The dcli utility facilitates centralized management across an Oracle Exadata System Software realm.

> **See Also:**
>
> - Using the ExaCLI Utility in *Oracle Exadata Database Machine Maintenance Guide*, for additional information about managing cells remotely
> - Using the exadcli Utility in *Oracle Exadata Database Machine Maintenance Guide*, for additional information about managing multiple cells remotely

1.3.8 About Database Server Software

Oracle software is installed on the Exadata database servers.

Oracle Exadata System Software works seamlessly with Oracle Database. The software on the database servers includes:

- Oracle Database instance, which contains the set of Oracle Database background processes that operate on the stored data and the shared allocated memory that those processes use to do their work.
- Oracle Automatic Storage Management (Oracle ASM), which provides storage management optimized for the database and Oracle Exadata Storage Servers. Oracle ASM is part of Oracle Grid Infrastructure.

  The Oracle ASM instance handles placement of data files on disks, operating as a metadata manager. The Oracle ASM instance is primarily active during file creation and extension, or during disk rebalancing following a configuration change. Run-time I/O operations are sent directly from the database to storage cells without passing through an Oracle ASM instance.

- The Oracle Database Resource Manager, which ensures that I/O resources are properly allocated within a database.
- The *i*DB protocol is used by the database instance to communicate with cells, and is implemented in an Oracle-supplied library statically linked with the database server.

Related Topics

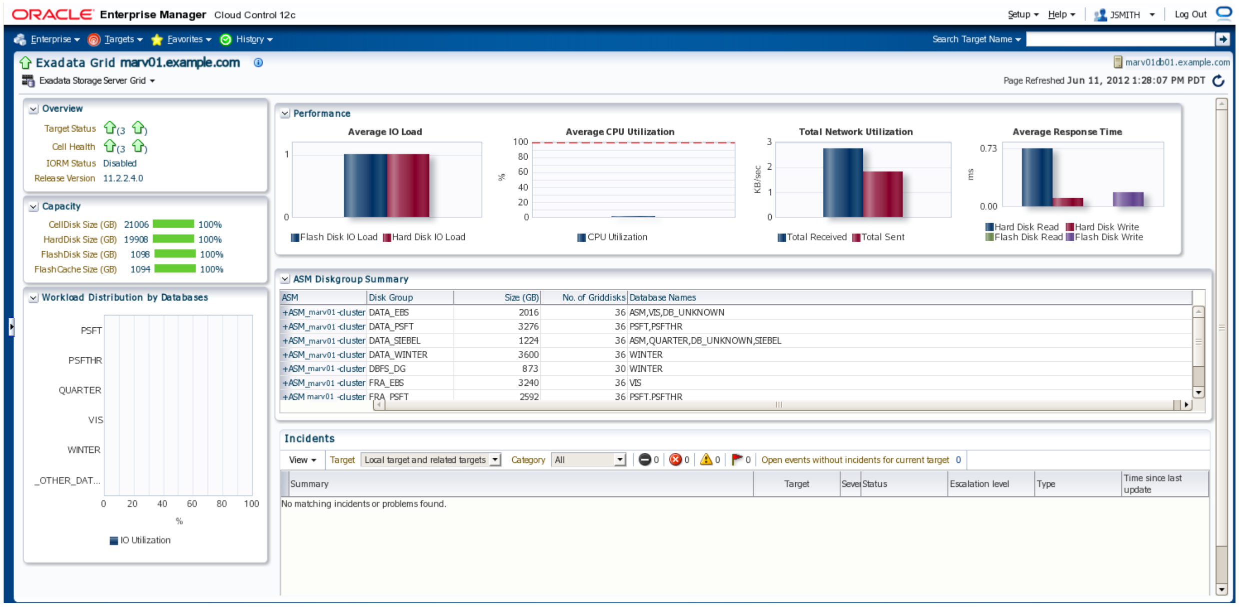
- Managing I/O Resources
  I/O Resource Management (IORM) is a tool for managing how multiple databases and the workloads within the databases share the I/O resources of Oracle Exadata System Software.
- About iDB Protocol
  The *i*DB protocol is a unique Oracle data transfer protocol that serves as the communications protocol among Oracle ASM, database instances, and storage cells.

1.3.9 About Oracle Enterprise Manager for Oracle Exadata Database Machine

Oracle Enterprise Manager provides a complete target that enables you to monitor Oracle Exadata Database Machine, including configuration and performance, in a graphical user interface (GUI).

The following figure shows Oracle Exadata Storage Grid Home Page. Viewing this page, you can quickly see the health of the storage servers, key storage performance characteristics and resource utilization of storage by individual databases.

Figure 1-4 Oracle Exadata Storage Server Home Page in Oracle Enterprise Manager



Page Refreshed Jun 11, 2012 1:28:07 PM PDT

Target Status (3)
Cell Health (3)
IORM Status Disabled
Release Version 11.2.2.4.0

CellDisk Size (GB) 21006 100%
HardDisk Size (GB) 19908 100%
FlashDisk Size (GB) 1098 100%
FlashCache Size (GB) 1094 100%

ASM	Disk Group	Size (GB)	No. of Griddisks	Database Names
+ASM_marv01-cluster	DATA_EBS	2016	36	ASM,VIS,DB_UNKNOWN
+ASM_marv01-cluster	DATA_PSFT	3276	36	PSFT,PSFTHR
+ASM_marv01-cluster	DATA_SIEBEL	1224	36	ASM,QUARTER,DB_UNKNOWN,SIEBEL
+ASM_marv01-cluster	DATA_WINTER	3600	36	WINTER
+ASM_marv01-cluster	DBFS_DG	873	30	WINTER
+ASM_marv01-cluster	FRA_EBS	3240	36	VIS
+ASM_marv01-cluster	FRA_PSFT	2592	36	PSFT,PSFTHR

Target Local target and related targets Category All ⊖ 0 ⊗ 0 ⚠ 0 ⚑ 0 Open events without incidents for current target 0

Summary	Target	Sever	Status	Escalation level	Type	Time since last update

No matching incidents or problems found.

In addition to reports, Oracle Enterprise Manager for Oracle Exadata Storage Server enables you to set metric thresholds for alerts and monitor metric values to determine the health of a storage cell.

Related Topics

- *Oracle Enterprise Manager Exadata Management Getting Started Guide*

2 Configuring Oracle Exadata System Software

This chapter describes the major steps to configure Oracle Exadata System Software.

You determine the number of disks and cells needed in the grid based on your requirements for capacity, performance, and redundancy.

Hardware and software have already been installed for the storage servers. The procedures in this chapter describe how to configure a storage cell for use with the Oracle Database and Oracle Automatic Storage Management (Oracle ASM) instances.

> **Note:**
>
> Modifications to the Oracle Exadata Storage Server hardware or software are not supported. Only the documented network interfaces on the Oracle Exadata Storage Server should be used for all connectivity including management and storage traffic. Additional network interfaces should not be used.

- Understanding Oracle Exadata System Software Release Numbering
  The Oracle Exadata System Software release number is related to the Oracle Database release number.
- Understanding Oracle Exadata Storage Server Configuration
  Oracle Exadata Storage Server ships with all hardware and software pre-installed, however you must configure Oracle Exadata System Software for your environment.
- Network Configuration and IP Addresses Recommendations
  Follow the recommendation network configuration for Oracle Exadata Storage Server.
- Assigning IP Addresses for Oracle Exadata Database Machine
  This topic summarizes the Oracle Exadata Database Machine network preparation before installing the new storage server.
- Configuring Oracle Exadata System Software for Your Location
  Customize the software installation for your environment.
- Configuring Cells, Cell Disks, and Grid Disks with CellCLI
  Configure the cells, cell disks and grid disks for each new Oracle Exadata Storage Server.
- Creating Flash Cache and Flash Grid Disks
  Oracle Exadata Storage Servers are equipped with flash disks. These flash disks can be used to create flash grid disks to store frequently accessed data.

- Setting Up Configuration Files for a Database Server Host
  After Oracle Exadata Storage Server is configured, the database server host must be configured with the `cellinit.ora` and the `cellip.ora` files to use the cell.
- Understanding Automated Cell Maintenance
  The Management Server (MS) includes a file deletion policy based on the date.

Related Topics

- Using Oracle Exadata Deployment Assistant
- Optimizing Performance
  You can use various settings to optimize the performance of Oracle Exadata System Software.
- Installation Information

2.1 Understanding Oracle Exadata System Software Release Numbering

The Oracle Exadata System Software release number is related to the Oracle Database release number.

The Oracle Exadata System Software release number matches the highest Oracle Grid Infrastructure and Oracle Database version it supports. For example, the highest version Oracle Exadata System Software release 18 supports is Oracle Grid Infrastructure and Oracle Database release 18. The highest version Oracle Exadata System Software release 12.2 supports is Oracle Grid Infrastructure and Oracle Database release 12.2.0.1.

Release 18c and Later Numbering

The Oracle Exadata System Software release that followed release 12.2.1.1.8 was renamed to 18.1.0 and a new numbering scheme for the Oracle Exadata System Software was implemented. Instead of a legacy nomenclature such as 12.2.1.1.5, a three field format consisting of: *Year*.*Update*.*Revision* is used, for example 18.1.0. This new numbering scheme allows you to clearly determine:

- The annual release designation of the software
- The latest software update, which can contain new features
- The latest software revision, which includes security and software fixes

If there are new features or new hardware supported, a new software update will be release during the year, for example, 19.2. To allow you to keep current on just security-related and other software fixes after your feature environment becomes stable, software revisions are made available approximately once a month, for example 19.1.3.

Numbering for Releases Prior to 18c

- The first two digits of the Oracle Exadata System Software release number represent the major Oracle Database release number, such as Oracle Database 12c Release 1 (12.1). Oracle Exadata System Software release 12.1 is compatible with all Oracle Database 12c Release 1 (12.1) releases.

- The third digit usually represents the component-specific Oracle Database release number. This digit usually matches the fourth digit of the complete release number, such as 12.1.0.1.0 for the current release of Oracle Database.
- The last two digits represent the Oracle Exadata System Software release.

Related Topics

- *Oracle Database Administrator's Guide*
- Release Update Introduction and FAQ (My Oracle Support Doc ID 2285040.1)

2.2 Understanding Oracle Exadata Storage Server Configuration

Oracle Exadata Storage Server ships with all hardware and software pre-installed, however you must configure Oracle Exadata System Software for your environment.

This topic provides a general overview of the configuration tasks. Subsequent topics describe the actual procedures.

- Assign IP Addresses for the Storage Servers
  As part of configuring the storage servers, you assign IP addresses to connect the storage to the various networks.
- Configure the Storage Server for Your Location
  Configure the storage server for use within your company.
- Configure the Storage Cell
  Use the `ALTER CELL` command to configure the cell.
- Verify Storage Cell Attributes
  Use the `LIST CELL DETAIL` command to verify the storage cell attributes.
- Create the Storage Cell Disks
  Use the `CREATE CELLDISK` command to create the cell disks.
- Create the Grid Disks
  Use the `CREATE GRIDDISK` command to create the grid disks. The size of the disks depends on your requirements.
- Create the PMEM Cache
  By default, the `CREATE CELL` command creates the cell disks, which are used when creating the PMEM Cache.
- Create the Flash Disks and Flash Cache
  By default, the `CREATE CELL` command creates flash cell disks on all flash disks. The command then creates Exadata Smart Flash Cache on the flash cell disks.
- Configure Oracle Auto Service Request (ASR)
  Oracle Auto Service Request (ASR) for Oracle Exadata Database Machine automatically creates service requests by detecting common hardware faults.

2.2.1 Assign IP Addresses for the Storage Servers

As part of configuring the storage servers, you assign IP addresses to connect the storage to the various networks.

Assign IP addresses for each storage server for the following ports:

- Network access port
- Remote management port
- RDMA Network Fabric port

2.2.2 Configure the Storage Server for Your Location

Configure the storage server for use within your company.

1. Power on the storage server.
2. Change the default passwords.
3. Set the time zone on the storage server to make the local time.
4. Configure other information as needed, such as NTP and DNS servers.

Related Topics

- Understanding User Accounts
- Change the DNS Server on Exadata Storage Server
- Change Time Zone Settings on Exadata Storage Servers
- Set the NTP Server on Exadata Storage Servers

2.2.3 Configure the Storage Cell

Use the `ALTER CELL` command to configure the cell.

In Example 2-1, e-mail notification is configured to send e-mail messages to the administrator of the storage cell. The hyphen (-) at the end of each line of the `ALTER CELL` command allows the command to continue to additional lines before pressing Enter. As an alternative, you can run the command using a text file.

Example 2-1 Configuring a New Cell

```
CellCLI> ALTER CELL                                                       -
         smtpServer='my_mail.example.com',                                -
         smtpFromAddr='john.doe@example.com',                             -
         smtpPwd=email_address_password,                                  -
         smtpToAddr='jane.smith@example.com',                             -
         notificationPolicy='clear',                                      -
         notificationMethod='mail,snmp'
```

2.2.4 Verify Storage Cell Attributes

Use the `LIST CELL DETAIL` command to verify the storage cell attributes.

Example 2-2 Viewing Storage Cell Details

This example shows how to view the storage cell attributes.

```
CellCLI> LIST CELL DETAIL
         name:                   cell01
         accessLevelPerm:        remoteLoginEnabled
         bbuStatus:              normal
         cellVersion:            OSS_19.3.0.0.0_LINUX.X64_190910
         cpuCount:               64/64
         diagHistoryDays:        7
         doNotServiceLEDStatus:  off
         fanCount:               12/12
         fanStatus:              normal
         flashCacheMode:         WriteBack
         httpsAccess:            ALL
         id:                     1904XCA016
         interconnectCount:      2
         interconnect1:          ib0
         interconnect2:          ib1
         iormBoost:              0.0
         ipaddress1:             192.168.41.245/21
         ipaddress2:             192.168.41.246/21
         kernelVersion:          4.14.35-1902.5.0.el7uek.x86_64
         locatorLEDStatus:       off
         makeModel:              Oracle Corporation ORACLE SERVER X8-2L
High Capacity
         memoryGB:               188
         metricHistoryDays:      7
         offloadGroupEvents:
         pmemCacheMode:          WriteThrough
         powerCount:             2/2
         powerStatus:            normal
         ramCacheMaxSize:        0
         ramCacheMode:           On
         ramCacheSize:           0
         releaseImageStatus:     success
         releaseVersion:         19.3.0.0.0.190824
         rpmVersion:             cell-19.3.0.0.0_LINUX.X64_190824-1.x86_64
         releaseTrackingBug:     29344484
         smtpFrom:               "Exadata Admins"
         smtpFromAddr:           exa_admins@example.com
         smtpServer:             my_mail_svr.example.com
         smtpToAddr:             jane.smith@example.com
         snmpSubscriber:
host=host1,port=162,community=public,type=asr,asrPort=16161
         status:                 online
         temperatureReading:     29.0
         temperatureStatus:      normal
         upTime:                 2 days, 7:05
         usbStatus:              normal
         cellsrvStatus:          stopped
         msStatus:               running
         rsStatus:               running
```

2.2.5 Create the Storage Cell Disks

Use the `CREATE CELLDISK` command to create the cell disks.

In Example 2-3, the `ALL` option creates all the cell disks using the default names.

The cell disks are created with names in the form CD_*lunID_cellname*. The *lunID* and *cellname* values correspond to the id attribute of the LUN and name attribute of the cell. You can specify other disk names if you create single cell disks.

On Oracle Exadata Storage Server with flash disks, the CREATE CELLDISK ALL command also creates cell disks on the flash disks.

```
CellCLI> CREATE CELLDISK ALL
CellDisk FD_01_cell01 successfully created
CellDisk FD_02_cell01 successfully created
CellDisk FD_03_cell01 successfully created
CellDisk FD_04_cell01 successfully created
CellDisk FD_05_cell01 successfully created
CellDisk FD_06_cell01 successfully created
CellDisk FD_07_cell01 successfully created
CellDisk FD_08_cell01 successfully created
CellDisk FD_09_cell01 successfully created
CellDisk FD_10_cell01 successfully created
CellDisk FD_11_cell01 successfully created
CellDisk FD_12_cell01 successfully created
CellDisk FD_13_cell01 successfully created
CellDisk FD_14_cell01 successfully created
CellDisk FD_15_cell01 successfully created
```

> **Note:**
>
> The CREATE CELLDISK command creates cell disks on flash disks if they do not currently exist. If there are cell disks on the flash disks, then they are not created again.

Example 2-3 Creating Cell Disks

```
CellCLI> CREATE CELLDISK ALL
CellDisk CD_00_cell01 successfully created
CellDisk CD_01_cell01 successfully created
CellDisk CD_02_cell01 successfully created
...
CellDisk CD_10_cell01 successfully created
CellDisk CD_11_cell01 successfully created
```

2.2.6 Create the Grid Disks

Use the CREATE GRIDDISK command to create the grid disks. The size of the disks depends on your requirements.

1. Determine the naming format for the grid disks or use the ALL PREFIX option.

   Grid disk names must be unique across all cells within a single deployment. By following the recommended naming conventions for naming the grid and cell disks you automatically get unique names. If you do not use the default generated name when creating grid disks, then you must ensure that the grid disk name is unique across all storage cells. If the disk name is not unique, then it might not be possible to add the grid disk to an Oracle Automatic Storage Management (Oracle ASM) disk group.

When the `ALL PREFIX` option is used, the generated grid disk names are composed of the grid disk prefix followed by an underscore (_) and then the cell disk name.

2. Use the `CREATE GRIDDISK` command to create the grid disks.

   When creating a grid disk:

   - You do not have to specify the `size` attribute. The maximum size possible is automatically chosen if the `size` attribute is omitted.
   - Offset determines the position on the disk where the grid disk is allocated. The outermost tracks have lower offset values, and these tracks have greater speed and higher bandwidth. Offset can be explicitly specified to create grid disks that are relatively higher performing than other grid disks. If offset is not specified, then the best (warmest) available offset is chosen automatically in chronological order of grid disk creation. You should first create those grid disks expected to contain the most frequently accessed (hottest) data, and then create the grid disks that contain the relatively colder data.
   - Sparse grid disks only need to be created when using snapshots. The sparse disk stores the files generated by the snapshot. All standard grid disk operations are supported for sparse grid disks. Sparse grid disks have an additional attribute, `virtualsize`. The attribute configures the maximum virtual space the grid disk must provide. The attribute can be resized if the configuration runs out of virtual space on the sparse grid disk and there is physical space available.

     The maximum allowed size of a sparse disk is the size of free space on the cell disk. The maximum allowed virtual size is 100 TB.

     Oracle Exadata System Software monitors physical space used by sparse grid disks, and generates an alert when most of the space is used. To avoid out-of-space errors, add more physical space to the grid disk using the `ALTER GRIDDISK` command, or delete some of the Oracle ASM files to free space on the grid disk.

Example 2-4 Creating Grid Disks

This example shows how to create grid disks. In this example, the `ALL HARDDISK PREFIX` option creates one grid disk on each cell disk of the storage cell. The Oracle ASM disk group name is used with `PREFIX` to identify which grid disk belongs to the disk group. Prefix values `data` and `reco` are the names of the Oracle ASM disk groups that are created.

```
CellCLI> CREATE GRIDDISK ALL HARDDISK PREFIX=data, size=300G
GridDisk data_CD_00_cell01 successfully created
GridDisk data_CD_01_cell01 successfully created
GridDisk data_CD_02_cell01 successfully created
...
GridDisk data_CD_11_cell01 successfully created

CellCLI> CREATE GRIDDISK ALL HARDDISK PREFIX=reco, size=600G
GridDisk reco_CD_00_cell01 successfully created
GridDisk reco_CD_01_cell01 successfully created
GridDisk reco_CD_02_cell01 successfully created
```

```
...
GridDisk reco_CD_11_cell01 successfully created
```

The LIST GRIDDISK command shows the grid disks that are created.

```
CellCLI> LIST GRIDDISK
         data_CD_00_cell01       active
         data_CD_01_cell01       active
         data_CD_02_cell01       active
         ...
         data_CD_11_cell01       active

         reco_CD_00_cell01      active
         reco_CD_01_cell01      active
         reco_CD_02_cell01      active
         ...
         reco_CD_11_cell01      active
```

Example 2-5   Creating a Sparse Grid Disk

In this example, the sparse grid disk uses up to 300 GB from the physical cell disk size, but it exposes 20000 GB virtual space for the Oracle ASM files.

```
CellCLI> CREATE GRIDDISK ALL HARDDISK PREFIX=sp, size=300G,
virtualsize=20000G
GridDisk sp_CD_00_cell01 successfully created
GridDisk sp_CD_01_cell01 successfully created
GridDisk sp_CD_02_cell01 successfully created
...
GridDisk sp_CD_11_cell01 successfully created
```

2.2.7 Create the PMEM Cache

By default, the CREATE CELL command creates the cell disks, which are used when creating the PMEM Cache.

- Use the CREATE PMEMCACHE ALL to create the PMEM Cache on all PMEM cell disks.

You can use the ALTER PMEMCACHE command to alter the set of cell disks used by PMEM cache, flush dirty blocks from PMEM cache, or cancel a previous flush operation on the specified cell disks to re-enable caching.

2.2.8 Create the Flash Disks and Flash Cache

By default, the CREATE CELL command creates flash cell disks on all flash disks. The command then creates Exadata Smart Flash Cache on the flash cell disks.

- Use the CREATE GRIDDISK ALL FLASHDISK PREFIX='FLASH' and CREATE FLASHCACHE commands to create the flash disks and flash cache.

To change the size of the Exadata Smart Flash Cache or create flash grid disks it is necessary to remove the flash cache, and then create the flash cache with a different size, or create the flash grid disks.

2.2.9 Configure Oracle Auto Service Request (ASR)

Oracle Auto Service Request (ASR) for Oracle Exadata Database Machine automatically creates service requests by detecting common hardware faults.

ASR support covers selected components, such as disks and flash cards, in Oracle Exadata Storage Servers and Oracle Exadata Database Servers.

- If you did not elect to configure Oracle Auto Service Request (ASR) when using Oracle Exadata Deployment Assistant (OEDA) to configure your Oracle Exadata Rack, then refer to *Oracle Auto Service Request Quick Installation Guide for Oracle Exadata Database Machine* for configuration instructions.

Related Topics

- *Oracle Exadata Database Machine Maintenance Guide*
- About Oracle Auto Service Request

2.3 Network Configuration and IP Addresses Recommendations

Follow the recommendation network configuration for Oracle Exadata Storage Server.

- If your network is not already configured, then set up a fault-tolerant, private network subnet for Oracle Exadata Database Machine storage servers and database servers with multiple switches to eliminate the switch as a single point of failure. If all the interconnections in the storage network are connected through a single switch, then that switch can be a single point of failure.

  If you are using a managed switch, then ensure that the switch VLAN configuration isolates storage server network traffic from all other network traffic.

- Allocate a block of IP addresses for the storage server general administration and the Integrated Lights Out Manager (ILOM) interfaces. Typically, these interfaces are on the same subnet, and may share the subnet with other hosts. For example, on the 192.168.200.0/24 subnet, you could assign the block of IP addresses between 192.168.200.31 and 192.168.200.100 for storage server general administration and ILOM interfaces. Other hosts sharing the subnet would be allocated IP addresses outside the block. The general administration and ILOM interfaces can be placed on separate subnets, but this is not required.

  The following is a sample of four non-overlapping blocks of addresses. One set of addresses should be assigned to the normal Gigabit Ethernet interface for the storage servers. The other blocks can be assigned for the ILOM port for the storage servers. The third set can be used for the database server Gigabit Ethernet ports, and the fourth for the database server ILOM ports.

  ```
  192.168.200.0/21 (netmask 255.255.248.0)
  192.168.208.0/21 (netmask 255.255.248.0)
  192.168.216.0/21 (netmask 255.255.248.0)
  192.168.224.0/21 (netmask 255.255.248.0)
  ```

- The RDMA Network Fabric network should be a private network for use by the database server hosts and storage servers, and can have private local network addresses. These addresses must also be allocated in non-overlapping blocks.

  The following example has 2 blocks of local RDMA Network Fabric addresses. Both the database server RDMA Network Fabric addresses and the storage server RDMA Network Fabric addresses must be in the same subnet to communicate with each other. With bonding, only one subnet is necessary for RDMA Network Fabric addresses.

  ```
  192.168.50.0/24 (netmask 255.255.255.0)
  192.168.51.0/24 (netmask 255.255.255.0)
  ```

  The preceding subnet blocks do not conflict with each other, and do not conflict with the current allocation to any of the hosts. It is a good practice to allocate the subnet blocks so that they have an identical netmask, which helps to simplify network administration.

- For Oracle Exadata Database Machine, the maximum allowed number of hosts in an RDMA Network Fabric network is 4096. Therefore, the network prefix value for the RDMA Network Fabric network must be equal to or greater than 20. This means the netmask must be between 255.255.240.0 and 255.255.255.254 inclusive.

  You can determine the network prefix value for a given host IP address and its netmask using the `ipcalc` utility on any Oracle Linux machine, as follows:

  ```
  ipcalc <host ip address> -m <netmask for the host>  -p
  ```

- Do not allocate addresses that end in `.0`, `.1`, or `.255`, or those that would be used as broadcast addresses for the specific netmask that you have selected. For example, avoid addresses such as 192.168.200.0, 192.168.200.1, and 192.168.200.255.

- Ensure the network allows for future expansion. For example, 255.255.255.254 is valid network (prefix /31) but it only allows 1 host.

- If a domain name system (DNS) is required, then set up your DNS to help reference storage servers and interconnections. Oracle Exadata Database Machine storage servers do not require DNS. However, if DNS is required, then set up your DNS with the appropriate IP address and host name of the storage servers.

- The RDMA Network Fabric network should be used for network and storage communication when using Oracle Clusterware. Use the following command to verify the private network for Oracle Clusterware communication is using the RDMA Network Fabric:

  ```
  oifcfg getif -type cluster_interconnect
  ```

- The Reliable Data Socket (RDS) protocol should be used over the RDMA Network Fabric network for database server to storage server communication and Oracle Real Application Clusters (Oracle RAC) communication. Check the alert log to verify the private network for Oracle RAC is running the RDS protocol over the

RDMA Network Fabric network. The following message should be in the alert logs for the instances:

```
cluster interconnect IPC version: Oracle RDS/IP (generic)
```

If the RDS protocol is not being used over the RDMA Network Fabric network, then perform the following procedure:

1. Shut down any processes that are using the Oracle software.
2. Change to the `ORACLE_HOME/rdbms/lib` directory.
3. Relink the Oracle Database software with the RDS protocol.

   ```
   make -f ins_rdbms.mk ipc_rds ioracle
   ```

> **Note:**
>
> If Oracle ASM uses a separate Oracle home from the database instance, then RDS should be enabled for the binaries in both homes.

2.4 Assigning IP Addresses for Oracle Exadata Database Machine

This topic summarizes the Oracle Exadata Database Machine network preparation before installing the new storage server.

Each storage server contains the following network ports:

- One dual-port RDMA Network Fabric card

  Oracle Exadata Storage Servers are designed to be connected to two separate RDMA Network Fabric switches for high availability. The dual port card is only for availability. Each port of the RDMA Network Fabric card is capable of transferring the full data bandwidth generated by the storage server. The loss of one network connection does not impact the performance of the storage server.
- Ethernet ports for normal network access, depending on the platform
  - Oracle Exadata Database Machine X6-2 and earlier storage servers come with four Ethernet ports. However, only connect one port to a switch, and configure it for network access.
- One Ethernet port is exposed by the Baseboard Management Controller (BMC), or Management Controller (MC) on the storage server. This port is used for Integrated Lights Out Manager (ILOM).

To prepare the storage server for network access, you must perform the following steps:

1. Assign one address to the bonded RDMA Network Fabric port. When you first set up the storage server, you are prompted for the RDMA Network Fabric configuration information. This information is used automatically during the `CREATE CELL` command on first boot, and provides the data path for communication between the storage server and the database servers.

To change the IP address after the initial configuration, use the following command, where *interface_name* is the interface name for the RDMA Network Fabric:

```
CREATE CELL interconnect1=interface_name, interconnect2=interface_name
```

For RDMA over InfiniBand Network Fabric networks, the interface names are `ib0` and `ib1`. For RDMA over RoCE Network Fabric networks, the interface names are `re0` and `re1`.

2. Assign an IP address to the storage server for network access.
3. Assign an IP address to the storage server for ILOM.

   You can access the remote management functionality with a Java-enabled Web browser at the assigned IP address.

> **See Also:**
>
> Oracle Integrated Lights Out Manager (ILOM) Documentation at `http://www.oracle.com/goto/ilom/docs`

2.5 Configuring Oracle Exadata System Software for Your Location

Customize the software installation for your environment.

- Configuring ILOM With Static IP for Oracle Exadata Storage Servers
  Basic lights-out remote management configuration is done during the first boot.
- Preparing the Servers
  Use the following steps to prepare the database servers and storage servers for use.

2.5.1 Configuring ILOM With Static IP for Oracle Exadata Storage Servers

Basic lights-out remote management configuration is done during the first boot.

Refer to Preparing the Servers for Integrated Lights Out Manager (ILOM) configuration information.

> **Caution:**
>
> Do not enable the sideband management available in ILOM. Doing so disables all the SNMP agent reporting and monitoring functionality for the server.

2.5.2 Preparing the Servers

Use the following steps to prepare the database servers and storage servers for use.

1. Configure Integrated Lights Out Manager (ILOM).
2. Power on the storage server to boot its operating system.
3. Respond to the prompts to configure the system, after the storage server boots.

   Press **y** to confirm, or **n** to retry or terminate when you are prompted for a yes or no response during the configuration steps. The yes or no prompt shows the default choice in brackets. If you do not enter a response, then the default choice is selected when you press **Enter**.
4. Check the network connections.

   The list of all discovered interfaces displays, and you are prompted to check the cables for those interfaces that do not have an active network cable connection. You can retry the configuration steps after connecting the cables, or ignore the unconnected interfaces. Only connected interfaces can be configured.
5. Enter the DNS server IP addresses, if needed.

   A DNS is not needed for a standalone, private storage environment.
6. Enter the time preference.

   - Choose the local time region number from the displayed list of available time regions.
   - Choose the location within the time region number from the displayed list of locations.
7. Enter the Network Time Protocol (NTP) servers.

   These servers are required to maintain the time on the system correctly, and are synchronized to your local time source.
8. Enter the Ethernet addresses, RDMA Network Fabric IP addresses and interfaces.

   A list of all Ethernet and RDMA Network Fabric interfaces that have an active network connection is displayed with the name of the interface on the extreme left. The RDMA over InfiniBand Network Fabric interface is named `BONDIB0` and uses bonding between physical RDMA over InfiniBand Network Fabric interfaces `ib0` and `ib1`. Bonding provides the ability to transparently fail over from `ib0` to `ib1` or from `ib1` to `ib0` if connectivity is lost to `ib0` or `ib1`, respectively.

   For each Ethernet and RDMA Network Fabric interface you configure, you are prompted for the following that apply to the interface:

   - IP address
   - Netmask
   - Gateway IP address
   - Fully-qualified domain name

   If you choose not to configure each interface in the list, then that interface is not configured, and it does not start at system startup. After the configuration of the IP addresses, the system completes the startup process. At the end of the process, additional packages are installed.

9. Select the canonical, fully-qualified domain name from the list.

   This host name is the primary public host name for the server, and is part of the `/etc/sysconfig/network` file.

   If more than one Ethernet interface was configured with the gateway, then select the line number for the default gateway. This gateway is in the `/etc/sysconfig/network` file, and is used as the default gateway.

10. Provide the following information when prompted for it:

    - ILOM full, domain-qualified host name
    - ILOM IP address
    - ILOM netmask
    - ILOM gateway
    - ILOM NTP servers
    - ILOM DNS server

11. (Oracle Exadata Storage Server only) Change the initial passwords for the `root`, `celladmin`, and `cellmonitor` users to more secure passwords.

    > **Note:**
    >
    > If you do not have the password for the `root` user, then contact Oracle Support Services.

    To change the passwords, log in as the `root` user, then use the `passwd` command to change the passwords, such as the following:

    ```
    # passwd
    # passwd celladmin
    # passwd cellmonitor
    ```

    To verify the changed passwords, log in and out of the server using each of the user names.

    > **Note:**
    >
    > The `cellmonitor` user is set up with privileges that enable you to only view Oracle Exadata Storage Server objects. You must be logged in as the `celladmin` user to perform administrative tasks.

12. Check for any failures reported in the `/var/log/cellos/vldrun.first_boot.log` file after the first boot configuration.

    For each failed validation, perform the following procedure:

    a. Look for the `/var/log/cellos/validations/`*`failed_validation_name`*`.SuggestedRemedy` file.

       The file exists only if the validation process has identified some corrective action. Follow the suggestions in the file to correct the cause of the failure.

   b. If the SuggestedRemedy file does not exist, then examine the log file for the failed validation in /var/log/cellos/validations to track down the cause, and correct it as needed.

13. (Oracle Exadata Storage Server only) Use the following commands to verify acceptable performance levels:

```
# cellcli -e "alter cell shutdown services cellsrv"

# cellcli -e "calibrate"
```

Related Topics

- Configuring ILOM With Static IP for Oracle Exadata Storage Servers
  Basic lights-out remote management configuration is done during the first boot.
- Using the ipconf Utility
  The ipconf utility is used to set and change the following parameters on Oracle Exadata Storage Servers.

2.6 Configuring Cells, Cell Disks, and Grid Disks with CellCLI

Configure the cells, cell disks and grid disks for each new Oracle Exadata Storage Server.

The steps described in Preparing the Servers must be completed before starting this task.

During the procedure, you can display help using the HELP command, and object attributes using the DESCRIBE command. Example 2-6 shows how to display help and a list of attributes for Oracle Exadata Database Machine CELL objects.

1. Log in as the celladmin user.
2. Use the cellcli command to start Cell Control Command-Line Interface (CellCLI) to connect to the storage cell.

   The required cell services, Restart Server (RS) and Management Server (MS), should be running after the binary has been installed. If not, then an error message displays when using the CellCLI utility. If an error message displays, then run the following commands to start Oracle Exadata System Software RS and MS services:

```
CellCLI> ALTER CELL STARTUP SERVICES RS
CellCLI> ALTER CELL STARTUP SERVICES MS
```

3. Configure the cell using the CellCLI ALTER CELL command. During first boot, the cell is created, and the flash cell disks and flash cache defined automatically.

```
CellCLI> ALTER CELL
name=cell_name,                                       -

smtpServer='my_mail.example.com',                              -

smtpFromAddr='john.doe@example.com',                           -
```

```
           smtpPwd=email_address_password,                           -

smtpToAddr='jane.smith@example.com',                          -

notificationPolicy='clear',                                   -
           notificationMethod='mail,snmp'
```

4. Use the following command to check the storage cell attributes, and to verify the current configuration:

   ```
   CellCLI> LIST CELL DETAIL
   ```

5. Create the cell disks, using the `CREATE CELLDISK` command. In most cases, you can use the default cell disk names and LUN IDs. Use the following command to create cell disks and LUN IDs with the default values.

   ```
   CellCLI> CREATE CELLDISK ALL
   ```

6. Create grid disks on each cell disk of the storage cell, using the `CREATE GRIDDISK` command.
7. Exit the CellCLI utility after setting up the storage cell using the following command:

   ```
   CellCLI> EXIT
   ```

8. Repeat the configuration process for each new storage cell. This procedure must be done on each new cell before configuring the Exadata Cell realm, the database server hosts, or the database and Oracle ASM instances.

Example 2-6   Displaying Help Information

```
CellCLI> HELP
CellCLI> HELP CREATE CELL
CellCLI> HELP ALTER CELL
CellCLI> DESCRIBE CELL
```

After you complete the cell configuration, you can perform the following optional steps on the storage cell:

- Add the storage cell to the Exadata Cell realm.
- Configure security on the Oracle Exadata Storage Server grid disks, as described in Configuring Security for Oracle Exadata Storage Server Software.
- Configure an inter-database plan for a cell rather than using the default plans, as described in Managing I/O Resources .

For database server hosts other than those in Oracle Exadata Database Machine, refer to release notes for enabling them to work with Oracle Exadata Storage Servers.

Related Topics

- ALTER CELL
- Creating Flash Cache and Flash Grid Disks
  Oracle Exadata Storage Servers are equipped with flash disks. These flash disks can be used to create flash grid disks to store frequently accessed data.

- CREATE GRIDDISK

2.7 Creating Flash Cache and Flash Grid Disks

Oracle Exadata Storage Servers are equipped with flash disks. These flash disks can be used to create flash grid disks to store frequently accessed data.

Alternatively, all or part of the flash disk space can be dedicated to Exadata Smart Flash Cache. In this case, the most frequently-accessed data is cached in Exadata Smart Flash Cache.

The `ALTER CELLDISK ... FLUSH` command must be run before exporting a cell disk to ensure that the data not synchronized with the disk (dirty data) is flushed from flash cache to the grid disks.

- By default, the `CREATE CELL` command creates flash cell disks on all flash disks, and then creates Exadata Smart Flash Cache on the flash cell disks.

  To change the size of the Exadata Smart Flash Cache or create flash grid disks it is necessary to remove the flash cache, and then create the flash cache with a different size, or create the flash grid disks.
- To change the amount of flash cache allocated, use the `flashcache` attribute with the `CREATE CELL` command.

  If the `flashcache` attribute is not specified, then all available flash space is allocated for flash cache.
- To explicitly create the Exadata Smart Flash Cache, use the `CREATE FLASHCACHE` command. Use the `celldisk` attribute to specify which flash cell disks contain the Exadata Smart Flash Cache.

  Alternatively, you can specify `ALL` instead of `celldisk` to use all flash cell disks. Use the `size` attribute to specify the total size of the flash cache to allocate. The allocation is evenly distributed across all flash cell disks.

Example 2-7 Using the CREATE FLASHCACHE Command

This example shows how to create the Exadata Smart Flash Cache. The entire size of the flash cell disk is not used because the `size` attribute has been set.

```
CellCLI> CREATE FLASHCACHE ALL size=100g
Flash cache cell01_FLASHCACHE successfully created
```

Example 2-8 Using the CREATE GRIDDISK Command to Create Flash Grid Disks

This example shows how to use the remaining space on the flash cell disks to create flash grid disks.

```
CellCLI> CREATE GRIDDISK ALL FLASHDISK PREFIX='FLASH'
GridDisk FLASH_FD_00_cell01 successfully created
GridDisk FLASH_FD_01_cell01 successfully created
GridDisk FLASH_FD_02_cell01 successfully created
GridDisk FLASH_FD_03_cell01 successfully created
GridDisk FLASH_FD_04_cell01 successfully created
GridDisk FLASH_FD_05_cell01 successfully created
GridDisk FLASH_FD_06_cell01 successfully created
GridDisk FLASH_FD_07_cell01 successfully created
GridDisk FLASH_FD_08_cell01 successfully created
GridDisk FLASH_FD_09_cell01 successfully created
GridDisk FLASH_FD_10_cell01 successfully created
GridDisk FLASH_FD_11_cell01 successfully created
GridDisk FLASH_FD_12_cell01 successfully created
GridDisk FLASH_FD_13_cell01 successfully created
GridDisk FLASH_FD_14_cell01 successfully created
GridDisk FLASH_FD_15_cell01 successfully created

CellCLI> LIST GRIDDISK
         FLASH_FD_00_cell01      active
         FLASH_FD_01_cell01      active
         FLASH_FD_02_cell01      active
         FLASH_FD_03_cell01      active
         FLASH_FD_04_cell01      active
         FLASH_FD_05_cell01      active
         FLASH_FD_06_cell01      active
         FLASH_FD_07_cell01      active
         FLASH_FD_08_cell01      active
         FLASH_FD_09_cell01      active
         FLASH_FD_10_cell01      active
         FLASH_FD_11_cell01      active
         FLASH_FD_12_cell01      active
         FLASH_FD_13_cell01      active
         FLASH_FD_14_cell01      active
         FLASH_FD_15_cell01      active
```

Example 2-9 Displaying the Exadata Smart Flash Cache Configuration for a Cell

Use the `LIST FLASHCACHE` command to display the Exadata Smart Flash Cache configuration for the cell, as shown in this example.

```
CellCLI> LIST FLASHCACHE DETAIL
         name:                   cell01_FLASHCACHE
         cellDisk:               FD_00_cell01, FD_01_cell01,FD_02_cell01,
                                 FD_03_cell01, FD_04_cell01, FD_05_cell01,
                                 FD_06_cell01, FD_07_cell01, FD_08_cell01,
                                 FD_09_cell01, FD_10_cell01, FD_11_cell01,
                                 FD_12_cell01, FD_13_cell01, FD_14_cell01,
                                 FD_15_cell01
         creationTime:           2009-10-19T17:18:35-07:00
         id:                     b79b3376-7b89-4de8-8051-6eefc442c2fa
         size:                   365.25G
         status:                 normal
```

Example 2-10 Dropping Exadata Smart Flash Cache from a Cell

To remove Exadata Smart Flash Cache from a cell, use the `DROP FLASHCACHE` command.

```
CellCLI> DROP FLASHCACHE
Flash cache cell01_FLASHCACHE successfully dropped
```

Related Topics

- CREATE CELL
- CREATE GRIDDISK
- LIST GRIDDISK
- LIST FLASHCACHE
- DROP FLASHCACHE

2.8 Setting Up Configuration Files for a Database Server Host

After Oracle Exadata Storage Server is configured, the database server host must be configured with the `cellinit.ora` and the `cellip.ora` files to use the cell.

The files are located in the `/etc/oracle/cell/network-config` directory.

- The `cellinit.ora` file contains the database IP addresses.
- The `cellip.ora` file contains the storage cell IP addresses.

Both files are located on the database server host. These configuration files contain IP addresses, not host names.

The `cellinit.ora` file is host-specific, and contains all database IP addresses that connect to the storage network used by Oracle Exadata Storage Servers. This file must exist for each database that connect to Oracle Exadata Storage Servers. The IP addresses are specified in Classless Inter-Domain Routing (CIDR) format. The first IP address must be designated as `ipaddress1`, the second IP address as `ipaddress2`, and so on.

The following list is an example of the IP address entry for a single database server in Oracle Exadata Database Machine:

- Oracle Exadata Database Server in Oracle Exadata Database Machine X4-2
  - `ipaddress1=192.168.10.1/22`
  - `ipaddress2=192.168.10.2/22`
- Oracle Exadata Database Server in Oracle Exadata Database Machine X3-2 or Oracle Exadata Database Machine X2-2
  - `ipaddress1=192.168.50.23/24`
- Oracle Exadata Database Server in Oracle Exadata Database Machine X3-8 or Oracle Exadata Database Machine X2-8
  - `ipaddress1=192.168.41.111/21`
  - `ipaddress2=192.168.41.112/21`
  - `ipaddress3=192.168.41.113/21`
  - `ipaddress4=192.168.41.114/21`

The IP addresses should not be changed after this file is created.

> **Note:**
>
> At boot time on an 8-socket system, each database server generates a `cellaffinity.ora` configuration file. The `cellaffinity.ora` file resides in the `/etc/oracle/cell/network-config` directory, and must be readable by Oracle Database.
>
> The file contains a mapping between the NUMA node numbers and the IP address of the network interface card closest to each server. Oracle Database uses the file to select the closest network interface card when communicating with Oracle Exadata Storage Servers, thereby optimizing performance.
>
> This file is only generated and used on an 8-socket system. On a 2-socket system, there is no performance to be gained in this manner, and no `cellaffinity.ora` file. The file is not intended to be directly edited with a text editor.

To configure a database server host for use with a cell, refer to Maintaining the RDMA Network Fabric for RoCE Network or Maintaining the RDMA Network Fabric for InfiniBand Network.

2.9 Understanding Automated Cell Maintenance

The Management Server (MS) includes a file deletion policy based on the date.

When there is a shortage of space in the Automatic Diagnostic Repository (ADR) directory, then MS deletes the following files:

- All files in the ADR base directory older than 7 days.
- All files in the `LOG_HOME` directory older than 7 days.
- All metric history files older than 7 days.

The retention period of seven days is the default. The retention period can be modified using the `metricHistoryDays` and `diagHistoryDays` attributes with the `ALTER CELL` command. The `diagHistoryDays` attribute controls the ADR files, and the `metricHistoryDays` attribute controls the other files.

If there is sufficient disk space, then trace files are not purged. This can result in files persisting in the ADR base directory past the time limit specified by `diagHistoryDays`.

In addition, the `alert.log` file is renamed if it is larger than 10 MB, and versions of the file that are older than 7 days are deleted if their total size is greater than 50 MB.

MS includes a file deletion policy that is triggered when file system utilization is high. Deletion of files in the `/` (root) directory and the `/var/log/oracle` directory is triggered when file utilization is 80 percent. Deletion of files in the `/opt/oracle` file system is triggered when file utilization reaches 90 percent, and the alert is cleared when utilization is below 85 percent. An alert is sent before the deletion begins. The alert includes the name of the directory, and space usage for the subdirectories. In particular, the deletion policy is as follows:

- The /var/log/oracle file systems, files in the ADR base directory, metric history directory, and LOG_HOME directory are deleted using a policy based on the file modification time stamp.
  - Files older than the number of days set by the metricHistoryDays attribute value are deleted first
  - Successive deletions occur for earlier files, down to files with modification time stamps older than or equal to 10 minutes, or until file system utilization is less than 75 percent.
  - The renamed alert.log files and ms-odl generation files that are over 5 MB, and older than the successively-shorter age intervals are also deleted.
  - Crash files in the /var/log/oracle/crashfiles directory that are more than one day old can be deleted. If the space pressure is not heavy, then the retention time for crash files is the same as for other files. If there are empty directories under /var/log/oracle/crashfiles, these directories are also deleted.
- For the /opt/oracle file system, the deletion policy is similar to the preceding settings. However, the file threshold is 90 percent, and files are deleted until the file system utilization is less than 85 percent.
- When file system utilization is full, the files controlled by the diagHistoryDays and metricHistoryDays attributes are purged using the same purging policy.
- For the / file system, files in the home directories (cellmonitor and celladmin), /tmp, /var/crash, and /var/spool directories that are over 5 MB and older than one day are deleted.

Every hour, MS deletes eligible alerts from the alert history using the following criteria. Alerts are considered eligible if they are stateless or they are stateful alerts which have been resolved.

- If there are less than 500 alerts, then alerts older than 100 days are deleted.
- If there are between 500 and 999 alerts, then the alerts older than 7 days are deleted.
- If there are 1,000 or more alerts, then all eligible alerts are deleted every minute.

> **Note:**
>
> Any directories or files with SAVE in the name are not deleted.

Related Topics

- ALTER CELL
- DESCRIBE CELL

3

Administering Oracle ASM Disk Groups on Oracle Exadata Storage Servers

You can administer Oracle Automatic Storage Management (Oracle ASM) disk groups with Oracle Exadata Storage Server grid disks

- Overview of Oracle Exadata Database Machine Storage
  Storage in Oracle Exadata Database Machine consists of servers, cell disks, grid disks, Oracle ASM disk groups, and Oracle ASM failure groups.
- Administering Oracle ASM Disk Groups Using Oracle Exadata Storage Servers
  There are basic Oracle ASM tasks needed to use Oracle Exadata Storage Servers.
- Administering Oracle Exadata Storage Server Grid Disks with Oracle ASM
  Use the following procedures for managing grid disks used with Oracle ASM.

Related Topics

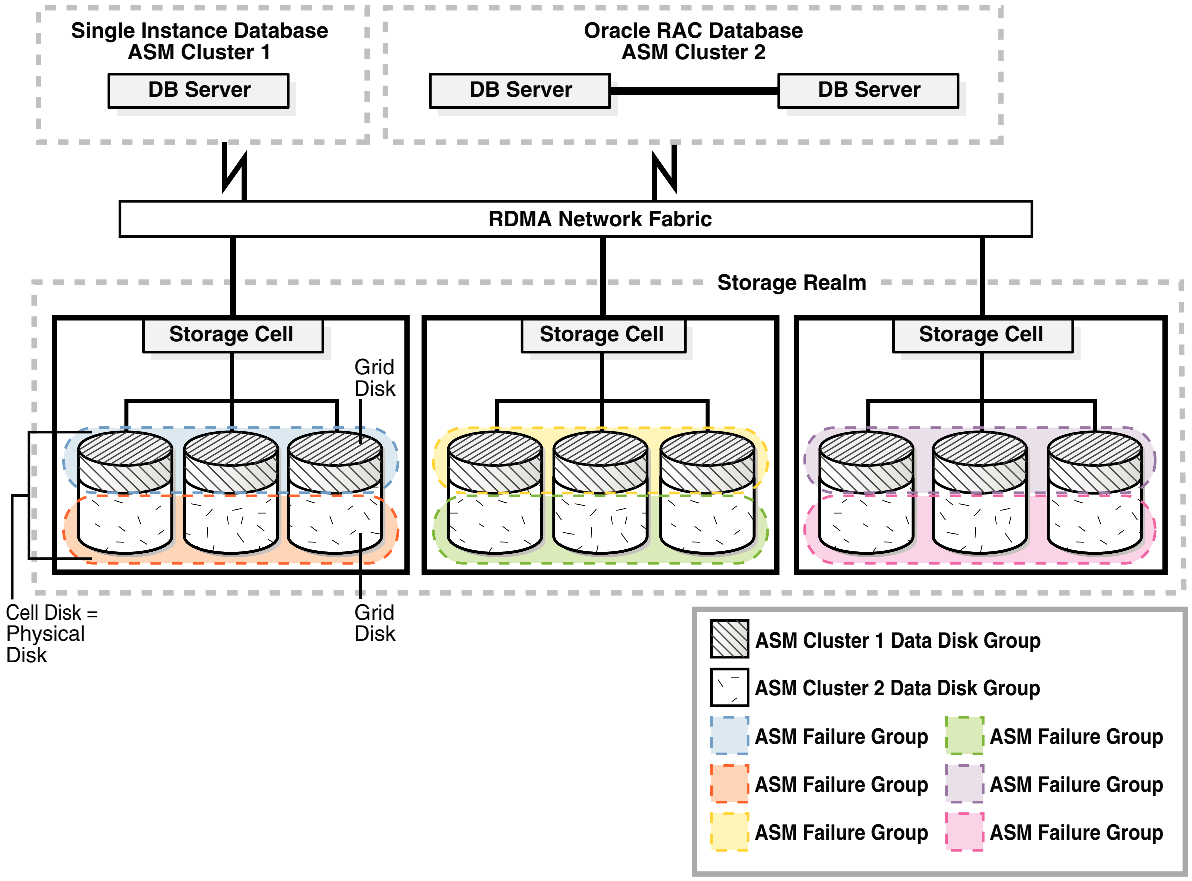
- Maintaining the Hard Disks of Exadata Storage Servers

3.1 Overview of Oracle Exadata Database Machine Storage

Storage in Oracle Exadata Database Machine consists of servers, cell disks, grid disks, Oracle ASM disk groups, and Oracle ASM failure groups.

The following image shows Oracle ASM disk groups created from Oracle Exadata Storage Server grid disks. It represents a typical, but simplified configuration, that can be used as a model for building larger storage grids with additional storage servers and disks.

Figure 3-1 Sample Oracle Exadata Storage Server Grid



This example storage grid illustrates the following:

- The storage servers in the grid use an RDMA Network Fabric network to connect to the database servers that have a single-instance database or Oracle Real Application Clusters (Oracle RAC) database installation.
- Each storage server contains multiple physical disks.
- Each cell disk represents a physical disk and a LUN.
- Each cell disk is partitioned into grid disks.
- Oracle ASM disk groups are created using the grid disks.

Oracle ASM failure groups are created to ensure that files are not mirrored on the same storage server, enabling the system to tolerate the failure of a storage server. The number of failure groups equals the number of storage servers. Each failure group is composed of a subset of grid disks in the Oracle ASM disk group that belong to a single storage server.

3.2 Administering Oracle ASM Disk Groups Using Oracle Exadata Storage Servers

There are basic Oracle ASM tasks needed to use Oracle Exadata Storage Servers.

- Understanding Oracle ASM Disk Groups for Oracle Exadata Storage Servers
  This topic explains Oracle Automatic Storage Management (Oracle ASM) disk groups, and how to create an Oracle ASM disk group for Oracle Exadata System Software using the `CREATE DISKGROUP` SQL command.
- Adding a Disk to an Oracle ASM Disk Group
  You can add a disk to an Oracle ASM disk group.
- Mounting or Dismounting an Oracle ASM Disk Group
  A disk group must be mounted by an Oracle ASM instance before database instances can access the files in the disk group.
- Changing a Disk to Offline or Online
  You can change an Oracle ASM disk to `INACTIVE` or `ACTIVE`.
- Dropping a Disk from an Oracle ASM Disk Group
  You can drop a grid disk from a disk group.
- Dropping an Oracle ASM Disk Group
  You can drop an Oracle ASM disk group.
- Enabling the Oracle ASM appliance.mode Attribute
  The Oracle ASM `appliance.mode` attribute improves disk rebalance completion time when dropping one or more Oracle ASM disks.

3.2.1 Understanding Oracle ASM Disk Groups for Oracle Exadata Storage Servers

This topic explains Oracle Automatic Storage Management (Oracle ASM) disk groups, and how to create an Oracle ASM disk group for Oracle Exadata System Software using the `CREATE DISKGROUP` SQL command.

Before creating an Oracle ASM disk group, determine which grid disks belong to the Oracle ASM disk group. It is recommended that you choose similar names for the Oracle ASM disk group and its grid disks whenever possible.

The Oracle Exadata Storage Server grid disks are specified with the following pattern:

```
o/cell_IPaddress/griddisk_name
```

In the preceding syntax, *cell_IPaddress* is the IP address of Oracle Exadata Storage Server, and *griddisk_name* is the name of the grid disk.

The cell discovery strings begin with the `o/` prefix.

When specifying the grid disks to be added to the disk group, consider the following:

- The default Oracle ASM disk name is the grid disk name. Oracle recommends using the default name.
- The default failure group name is the cell name. Oracle recommends using the default name.

When a failure group is not specified, Oracle ASM adds each disk within its own failure group. However, when the disks are stored on Oracle Exadata Storage Servers and a failure group is not specified, Oracle ASM adds a disk to the failure group for that cell. The failure group name is the cell name.

Note:

If a cell is renamed, and a disk from that cell is added to an existing disk group that has disks from that cell, then Oracle ASM adds the new disk to a failure group using the new cell name. To ensure all the disks from the cell are in one failure group, add the disk to the disk group and specify the original failure group name.

To enable Smart Scan predicate offload processing, all disks in a disk group must be Oracle Exadata Storage Server grid disks. You cannot include conventional disks with Oracle Exadata Storage Server grid disks.

- Setting the Oracle ASM Content Type When Using Normal Redundancy
  Setting the `content.type` attribute for disk groups provides better recovery time objective (RTO) and recovery point objective (RPO or data loss tolerance).
- Creating Oracle ASM Disk Groups
  You can create Oracle ASM disk groups on Oracle Exadata Storage Server grid disks.

Related Topics

- Naming Conventions for Oracle Exadata Storage Server Grid Disks
  Using a consistent naming convention helps to identify Exadata components.
- Oracle ASM Disk Groups
  An Oracle Automatic Storage Management (Oracle ASM) disk group is the primary storage abstraction within Oracle ASM, and is composed of one or more grid disks.

3.2.1.1 Setting the Oracle ASM Content Type When Using Normal Redundancy

Setting the `content.type` attribute for disk groups provides better recovery time objective (RTO) and recovery point objective (RPO or data loss tolerance).

When using normal redundancy with Oracle Grid Infrastructure release 11.2.0.3 or later, and the `compatible.asm` value is `11.2.0.3` or later, then set the `content.type` attribute for the DATA, RECO and DBFS_DG disk groups.

The value of `content.type` attribute should be set as follows:

- DATA and SPARSE disk groups — `data`
- RECO disk group — `recovery`
- DBFS_DG disk group — `system`

Note:

- Do not use the `content.type` attribute to distinguish the availability characteristics of disk groups that are used for a different purpose, such as those created to support a particular service.
- The Oracle Database and Oracle Grid Infrastructure software must be release 12.1.0.2.0 BP5 or later when using sparse grid disks.

1. Use the ALTER DISKGROUP command to set the content.type attribute for an existing disk group, and then rebalance the disk group.

   ```
   ALTER DISKGROUP reco SET ATTRIBUTE 'content.type'='recovery';
   ALTER DISKGROUP reco REBALANCE POWER preferred_power_setting ;
   ```

   The rebalance operation can take a long time, but the data in the disk group is fully redundant throughout the operation. Oracle ASM monitors the rebalance operation, and Oracle Exadata System Software sends an e-mail message when the operation completes.

2. Check the content.type attributes using the following query:

   ```
   SQL> SELECT dg.name,a.value FROM v$asm_diskgroup dg,        \
        v$asm_attribute a WHERE dg.group_number=a.group_number \
        AND a.name='content.type' AND (dg.name LIKE 'DATA%'    \
        OR dg.name LIKE 'RECO%' OR dg.name LIKE 'DBFS_DG%');

   NAME                 VALUE
   -------------------- --------------------
   DATA                 data
   RECO                 recovery
   DBFS_DG              system
   ```

Example 3-1 Adding content type while creating a disk group

In this example, the compatible.rdbms attribute is set to 11.2.0.2 in order to support both Oracle Database release 11.2.0.2 and release 11.2.0.3 databases in a consolidated environment.

```
CREATE DISKGROUP data NORMAL REDUNDANCY
DISK
'o/*/DATA*'
ATTRIBUTE 'content.type' = 'DATA',
'content.type' = 'DATA',
'AU_SIZE' = '4M',
'cell.smart_scan_capable'='TRUE',
'compatible.rdbms'='11.2.0.2',
'compatible.asm'='11.2.0.3';
```

3.2.1.2 Creating Oracle ASM Disk Groups

You can create Oracle ASM disk groups on Oracle Exadata Storage Server grid disks.

To create an Oracle ASM disk group to use Oracle Exadata Storage Server grid disks, perform the following procedure:

1. Connect to the Oracle ASM instance.
2. Ensure that the ORACLE_SID environment variable is set to the Oracle ASM instance using a command similar to the following:

   ```
   $ setenv ORACLE_SID ASM_instance_SID
   ```

3. Start SQL*Plus on the Oracle ASM instance, and log in as a user with SYSASM administrative privileges.

   ```
   $ sqlplus / AS SYSASM
   ```

4. Determine which Oracle Exadata Storage Server grid disks are available by querying the V$ASM_DISK view on the Oracle ASM instance, using the following syntax:

   ```
   SQL> SELECT path, header_status STATUS FROM V$ASM_DISK WHERE path LIKE
   'o/%';
   ```

5. Create an Oracle ASM disk group to include disks on the cells.

   In this example, the ALTER command is needed to change compatible.rdbms for the disk group created during installation to hold the OCR and voting disks. The compatible.rdbms attribute is set to 11.2.0.2 in order to support both Oracle Database release 11.2.0.2 and release 11.2.0.3 databases in a consolidated environment.

   ```
   CREATE DISKGROUP data HIGH REDUNDANCY
   DISK 'o/*/DATA*'
   ATTRIBUTE 'AU_SIZE' = '4M',
             'content.type' = 'data',
             'compatible.rdbms'='11.2.0.4',
             'compatible.asm'='19.0.0.0';

   SQL> CREATE DISKGROUP reco HIGH REDUNDANCY
   DISK 'o/*/RECO*'
   ATTRIBUTE 'AU_SIZE' = '4M',
             'content.type' = 'recovery',
             'compatible.rdbms'='11.2.0.4',
             'compatible.asm'='19.0.0.0';

   REM for Exadata systems prior to X7
   SQL> ALTER DISKGROUP dbfs_dg SET ATTRIBUTE
        'content.type' = 'system',
        'compatible.rdbms' = '11.2.0.4';
   ```

   When creating sparse disk groups, use a command similar to the following:

   ```
   SQL> CREATE DISKGROUP sparsedg NORMAL REDUNDANCY
   DISK 'o.*/sparse_*'
   ATTRIBUTE 'AU_SIZE' = '4M',
             'content.type' = 'data',
             'cell.smart_scan_capable'='TRUE',
             'compatible.rdbms' = '12.1.0.2',
             'compatible.asm' = '19.0.0.0',
             'cell.sparse_dg' = 'allsparse';
   ```

   In the preceding command, the cell.sparse_dg attribute defines the disk group as a sparse disk group. The attribute does not have to included if the disk group is not a sparse disk group.

> **Note:**
>
> - When defining sparse grid disks, the `compatible.asm` and `compatible.rdbms` attributes must be at least `12.1.0.2.0`.
> - The Oracle ASM disk group `compatible` attributes take precedence over the `COMPATIBLE` initialization parameter for the Oracle ASM instance.
> - The Oracle Database and Oracle Grid Infrastructure software must be release 12.1.0.2.0 BP5 or later when using sparse grid disks.

6. View the Oracle ASM disk groups and associated attributes with a SQL query on `V$ASM` dynamic views.

```
SQL> SELECT dg.name AS diskgroup, SUBSTR(a.name,1,24) AS name,
     SUBSTR(a.value,1,24) AS value FROM V$ASM_DISKGROUP dg,
V$ASM_ATTRIBUTE a
     WHERE dg.group_number = a.group_number;

DISKGROUP                    NAME                       VALUE
---------------------------- ------------------------
------------------------
DATA                         compatible.rdbms           11.2.0.4
DATA                         compatible.asm             19.0.0.0
DATA                         au_size                    4194304
DATA                         disk_repair_time           3.6h
DATA                         cell.smart_scan_capable    TRUE
...
```

7. Create a tablespace in the disk group to take advantage of Oracle Exadata System Software features, such as offload processing. The tablespace should contain the tables that you want to query with offload processing.

```
SQL> CREATE TABLESPACE tablespace_name DATAFILE '+DATA';
```

   In the preceding command, `+DATA` is the name of the Oracle ASM disk group.

8. Verify that the tablespace is in an Oracle Exadata Storage Server disk group. The `PREDICATE_EVALUATION` column of the `DBA_TABLESPACES` view indicates whether predicates are evaluated by host (`HOST`) or by storage (`STORAGE`).

```
SQL> SELECT tablespace_name, predicate_evaluation FROM dba_tablespaces
     WHERE tablespace_name = 'DATA_TB';

TABLESPACE_NAME                PREDICA
------------------------------ -------
DATA_TB                        STORAGE
```

Example 3-2   Creating Oracle ASM disk groups on grid disks

This example shows the use of a `CREATE DISKGROUP` command to create a disk group. The names of the disk groups are shown in bold. The `compatible.rdbms` attribute is set to 11.2.0.2 in order to support both Oracle Database release 11.2.0.2 and release 11.2.0.3 databases in a consolidated environment.

```
SQL> CREATE DISKGROUP data HIGH REDUNDANCY

-- These grid disks are on cell01
   DISK
   'o/*/data_CD_00_cell01',
   'o/*/data_CD_01_cell01',
   'o/*/data_CD_02_cell01',

-- These grid disks are on cell02
   DISK
   'o/*/data_CD_00_cell02',
   'o/*/data_CD_01_cell02',
   'o/*/data_CD_02_cell02',

-- These disk group attributes must be set for cell access
-- Note that this disk group is set for cell only
   ATTRIBUTE 'compatible.rdbms' = '11.2.0.4',
             'content.type' = 'data',
             'compatible.asm' = '19.0.0.0',
             'au_size' = '4M',
             'cell.smart_scan_capable' = 'TRUE';

SQL> CREATE DISKGROUP reco HIGH REDUNDANCY

-- These grid disks are on cell01
   DISK
   'o/*/reco_CD_00_cell01',
   'o/*/reco_CD_01_cell01',
   'o/*/reco_CD_02_cell01'

-- These grid disks are on cell02
   DISK
   'o/*/reco_CD_00_cell02',
   'o/*/reco_CD_01_cell02',
   'o/*/reco_CD_02_cell02'

-- These disk group attributes must be set for cell access
-- Note that this disk group is set for cell only
   ATTRIBUTE 'compatible.rdbms' = '11.2.0.4',
             'content.type' = 'recovery',
             'compatible.asm' = '19.0.0.0',
             'au_size' = '4M',
             'cell.smart_scan_capable' = 'TRUE';
```

Related Topics

- Introducing Oracle Automatic Storage Management
- Administration of Oracle ASM Instances
- ALTER DISKGROUP
- CREATE DISKGROUP

- CREATE TABLESPACE

3.2.2 Adding a Disk to an Oracle ASM Disk Group

You can add a disk to an Oracle ASM disk group.

Do not add Oracle Exadata Storage Server grid disks to an Oracle ASM disk group that is not on an Oracle Exadata Storage Server unless you are planning to migrate the disk group to an Oracle Exadata Storage Server disk group.

1. Determine which disks are available by querying the `V$ASM_DISK` view on the Oracle ASM instance.

   If the header status is set to `CANDIDATE`, then the disk is a candidate for a disk group.

2. Use the SQL command `ALTER DISKGROUP` with the `ADD DISK` clause to add the disk to the Oracle ASM disk group.

   ```
   SQL> ALTER DISKGROUP disk_group_name
   ADD DISK 'o/cell_IPaddress/data*';
   ```

After the disk is added, Oracle ASM rebalances the disk group. Oracle ASM monitors the rebalance operation, and Oracle Exadata System Software sends an e-mail message when the operation is complete.

You can query the `V$ASM_OPERATION` view for the status of the rebalance operation.

Related Topics

- Naming Conventions for Oracle Exadata Storage Server Grid Disks
  Using a consistent naming convention helps to identify Exadata components.

3.2.3 Mounting or Dismounting an Oracle ASM Disk Group

A disk group must be mounted by an Oracle ASM instance before database instances can access the files in the disk group.

Mounting the disk group requires discovering all of the disks and locating the files in the disk group that is being mounted.

- To mount a disk group, use the SQL `ALTER DISKGROUP` command with the `MOUNT` option.
- To dismount a disk group, use the SQL `ALTER DISKGROUP` command with the `DISMOUNT` option.
- You can use the `FORCE` option of the `ALTER DISKGROUP` command `MOUNT` clause to mount disk groups if their components are unavailable.

  This results in a loss of full redundancy.

Related Topics

- *Oracle Automatic Storage Management Administrator's Guide*

3.2.4 Changing a Disk to Offline or Online

You can change an Oracle ASM disk to `INACTIVE` or `ACTIVE`.

1. Determine which disk you want offline or online in the Oracle ASM disk group.

   Query the V$ASM_DISK and V$ASM_DISKGROUP views on the Oracle ASM instance.

2. Use one of the following commands:

   - To make a disk inactive, use the following command:

     ```
     CellCLI> ALTER GRIDDISK gdisk_name INACTIVE
     ```

   - To make a disk active, use the following command:

     ```
     CellCLI> ALTER GRIDDISK gdisk_name ACTIVE
     ```

   As soon as the disk is online, the disk group is rebalanced.

Oracle ASM monitors the rebalance operation, and Oracle Exadata System Software sends an e-mail message when the operation is complete.

You can query the V$ASM_OPERATION view for the status of the rebalance operation.

Related Topics

- Determining Which Oracle ASM Disk Group Contains an Oracle Exadata Storage Server Grid Disk
  If a grid disk name matches the Oracle ASM disk name, and the name contains the Oracle ASM disk group name, then you can determine the Oracle ASM disk group to which the grid disk belongs.
- ALTER GRIDDISK

3.2.5 Dropping a Disk from an Oracle ASM Disk Group

You can drop a grid disk from a disk group.

1. Determine which disks you want to drop from the Oracle ASM disk group.

   Query the V$ASM_DISK and V$ASM_DISKGROUP views on the Oracle ASM instance.

   If you are removing an Oracle Exadata Storage Server grid disk, then ensure that you identify the grid disks that are mapped to each Oracle ASM disk group.

2. Use the SQL ALTER DISKGROUP command with the DROP DISK clause to drop the disks from the Oracle ASM disk group.

   ```
   SQL> ALTER DISKGROUP disk_group_name
   DROP DISK data_CD_11_cell01 NORMAL;
   ```

   Do not use the FORCE option when dropping the disk from the Oracle ASM disk group. If you use the FORCE option, Oracle Exadata System Software will attempt to add the disk back to the disk group if the disk online automation operation is triggered, by rebooting the storage server, for example. See Enhanced Manageability Features in *Oracle Exadata Database Machine System Overview*.

When the disk is dropped from the Oracle ASM disk group, Oracle ASM rebalances the disk group. Oracle ASM monitors the rebalance operation, and Oracle Exadata System Software sends an e-mail message when the operation is complete.

You can query the V$ASM_OPERATION view for the status of the rebalance operation.

After an Oracle Exadata Storage Server grid disk is dropped from the Oracle ASM disk group, you can drop the grid disk from the cell.

Related Topics

- Determining Which Oracle ASM Disk Group Contains an Oracle Exadata Storage Server Grid Disk
  If a grid disk name matches the Oracle ASM disk name, and the name contains the Oracle ASM disk group name, then you can determine the Oracle ASM disk group to which the grid disk belongs.
- Determining Which Oracle Exadata Storage Server Grid Disks Belong to an Oracle ASM Disk Group
  If a grid disk name contains the Oracle ASM disk group name, then you can use SQL commands on the Oracle ASM instance to list the Oracle ASM disk group names.
- Dropping an Oracle Exadata Storage Server Grid Disk
  To drop an Oracle Exadata Storage Server grid disk, use the CellCLI `DROP GRIDDISK` command.

3.2.6 Dropping an Oracle ASM Disk Group

You can drop an Oracle ASM disk group.

If you cannot mount a disk group but must drop it, then use the `FORCE` option with the `DROP DISKGROUP` command.

1. Determine the disk group that you want to drop.

   Query the `V$ASM_DISKGROUP` view on the Oracle ASM instance.

2. Use the SQL `DROP DISKGROUP` command to drop the Oracle ASM disk group.

3.2.7 Enabling the Oracle ASM appliance.mode Attribute

The Oracle ASM `appliance.mode` attribute improves disk rebalance completion time when dropping one or more Oracle ASM disks.

Setting the `appliance.mode` attribute helps restore redundancy faster after a failure. The attribute can only be enabled on disk groups that meet the following requirements:

- The Oracle ASM disk group attribute `compatible.asm` is set to release 11.2.0.4, or 12.1.0.2 or later.
- The `cell.smart_scan_capable` attribute is set to `TRUE`.
- All disks in the disk group are the same type; for example, all disks are hard disks or all disks are flash disks.
- All disks in the disk group are the same size.
- All failure groups in the disk group have an equal number of disks:
  - For eighth rack configurations, all failure groups have 4 disks, or all failure groups have 6 disks.
  - For all other rack configurations, all failure groups have 10 disks, or all failure groups have 12 disks.
- There are at least 3 failure groups in the disk group.

- No disk in the disk group is offline.

> **Note:**
>
> Enabling the `appliance.mode` attribute for existing disk groups may cause an increase of data movement during the next rebalance operation.

The `appliance.mode` attribute is automatically enabled when creating a new disk group. Existing disk groups must explicitly set the attribute using the `ALTER DISKGROUP` command.

```
SQL> ALTER DISKGROUP disk_group SET ATTRIBUTE 'appliance.mode'='TRUE';
```

To disable the `appliance.mode` attribute during disk group creation, set the attribute to `FALSE`.

```
SQL> CREATE DISKGROUP data NORMAL REDUNDANCY
DISK
'o/*/DATA*'
ATTRIBUTE 'content.type' = 'data',
          'au_size' = '4M',
          'cell.smart_scan_capable'='TRUE',
          'compatible.rdbms'='11.2.0.3',
          'compatible.asm'='11.2.0.4',
          'appliance.mode'='FALSE';
```

3.3 Administering Oracle Exadata Storage Server Grid Disks with Oracle ASM

Use the following procedures for managing grid disks used with Oracle ASM.

- Naming Conventions for Oracle Exadata Storage Server Grid Disks
  Using a consistent naming convention helps to identify Exadata components.
- Changing an Oracle Exadata Storage Server Grid Disk That Belongs to an Oracle ASM Disk Group
  Before you change a grid disk that belongs to an Oracle ASM disk group, you must consider how the change might affect the Oracle ASM disk group to which the grid disk belongs.
- Resizing Grid Disks
  You can resize grid disks and Oracle ASM disk groups to shrink one with excess free space and increase the size of another that is near capacity.
- Determining Which Oracle ASM Disk Group Contains an Oracle Exadata Storage Server Grid Disk
  If a grid disk name matches the Oracle ASM disk name, and the name contains the Oracle ASM disk group name, then you can determine the Oracle ASM disk group to which the grid disk belongs.

- Determining Which Oracle Exadata Storage Server Grid Disks Belong to an Oracle ASM Disk Group
  If a grid disk name contains the Oracle ASM disk group name, then you can use SQL commands on the Oracle ASM instance to list the Oracle ASM disk group names.
- Handling Disk Replacement
  If a disk has a problem, the physical disk status changes.

3.3.1 Naming Conventions for Oracle Exadata Storage Server Grid Disks

Using a consistent naming convention helps to identify Exadata components.

The name of the grid disk should contain the cell disk name to make it easier to determine which grid disks belong to a cell disk. To help determine which grid disks belong to an Oracle ASM disk group, a subset of the grid disk name should match all or part of the name of the Oracle ASM disk group to which the grid disk will belong.

For example, if a grid disk is created on the cell disk `CD_03_cell01`, and that grid disk belongs to an Oracle ASM disk group named `data0`, then the grid disk name should be `data0_CD_03_cell01`.

When you use the `ALL PREFIX` option with `CREATE GRIDDISK`, a unique grid disk name is automatically generated that includes the prefix and cell name. If you do not use the default generated name when creating grid disks, then you must ensure that the grid disk name is unique across all cells. If the disk name is not unique, then it might not be possible to add the grid disk to an Oracle ASM disk group.

Related Topics

- Determining Which Oracle ASM Disk Group Contains an Oracle Exadata Storage Server Grid Disk
  If a grid disk name matches the Oracle ASM disk name, and the name contains the Oracle ASM disk group name, then you can determine the Oracle ASM disk group to which the grid disk belongs.
- Determining Which Oracle Exadata Storage Server Grid Disks Belong to an Oracle ASM Disk Group
  If a grid disk name contains the Oracle ASM disk group name, then you can use SQL commands on the Oracle ASM instance to list the Oracle ASM disk group names.
- CREATE CELLDISK
- CREATE GRIDDISK

3.3.2 Changing an Oracle Exadata Storage Server Grid Disk That Belongs to an Oracle ASM Disk Group

Before you change a grid disk that belongs to an Oracle ASM disk group, you must consider how the change might affect the Oracle ASM disk group to which the grid disk belongs.

- Changing an Oracle Exadata Storage Server Grid Disk Name
  Use the CellCLI interface to change the name of a grid disk.

- Dropping an Oracle Exadata Storage Server Grid Disk
  To drop an Oracle Exadata Storage Server grid disk, use the CellCLI DROP GRIDDISK command.

3.3.2.1 Changing an Oracle Exadata Storage Server Grid Disk Name

Use the CellCLI interface to change the name of a grid disk.

- To change attributes of a grid disk, use the CellCLI ALTER GRIDDISK command.

  Use the DESCRIBE GRIDDISK command to determine which Oracle Exadata Storage Server grid disk attributes can be modified.

> **Caution:**
>
> Before changing the name of a grid disk that belongs to an Oracle ASM disk group, ensure that the corresponding Oracle ASM disk is offline.

Example 3-3 Changing an Oracle Exadata Storage Server Grid Disk Name

Use the ALTER GRIDDISK command to rename a grid disk.

```
CellCLI> ALTER GRIDDISK data011 name='data0_CD_03_cell04'
```

Related Topics

- Changing a Disk to Offline or Online
  You can change an Oracle ASM disk to INACTIVE or ACTIVE.
- ALTER GRIDDISK
- Naming Conventions for Oracle Exadata Storage Server Grid Disks
  Using a consistent naming convention helps to identify Exadata components.

3.3.2.2 Dropping an Oracle Exadata Storage Server Grid Disk

To drop an Oracle Exadata Storage Server grid disk, use the CellCLI DROP GRIDDISK command.

Make the grid disk inactive before dropping the grid disk to ensure that the grid disk is not in use. The FORCE option can be used to force the grid disk that is in use to be dropped.

> **Caution:**
>
> - Before dropping a grid disk that belongs to an Oracle ASM disk group, ensure that the corresponding Oracle ASM disk was dropped from the disk group.
> - Before dropping a grid disk using the FORCE option, ensure that the Oracle ASM disk was dropped from the disk group.

1. Drop the Oracle ASM disk from the disk group.

   ```
   SQL> ALTER DISKGROUP disk_group_name DROP DISK disk_name;
   ```

2. Make the corresponding grid disk inactive.

   ```
   CellCLI> ALTER GRIDDISK disk_name INACTIVE
   ```

3. Drop the grid disk.

   ```
   CellCLI> DROP GRIDDISK disk_name
   ```

Example 3-4 Dropping a specific grid disk

After you have dropped the Oracle ASM disk from the disk group, you can drop the related grid disk.

```
CellCLI> ALTER GRIDDISK data0_CD_03_cell04 INACTIVE
CellCLI> DROP GRIDDISK data0_CD_03_cell04
```

Example 3-5 Dropping all grid disks

After you have dropped the Oracle ASM disks from the disk group, you can drop multiple grid disks using a single command.

```
CellCLI> ALTER GRIDDISK ALL INACTIVE
CellCLI> DROP GRIDDISK ALL PREFIX=data0
```

Example 3-6 Using the FORCE option when dropping a grid disk

If you cannot make a grid disk `INACTIVE`, but must drop the grid disk, use the `FORCE` option. The `FORCE` option forces the grid disk that is in use to be dropped

```
CellCLI> DROP GRIDDISK data02_CD_04_cell01 FORCE
```

Related Topics

- DROP GRIDDISK

3.3.3 Resizing Grid Disks

You can resize grid disks and Oracle ASM disk groups to shrink one with excess free space and increase the size of another that is near capacity.

Initial configuration of Oracle Exadata Database Machine disk group sizes is based on Oracle best practices and the location of the backup files.

- For internal backups: allocation of available space is 40% for the DATA disk groups, and 60% for the RECO disk groups.
- For external backups: allocation of available space is 80% for the DATA disk group, and 20% for the RECO disk group.

The disk group allocations can be changed after deployment. For example, the DATA disk group allocation may be too small at 60%, and need to be resized to 80%.

If your system has no free space available on the cell disks and one disk group, for example RECO, has plenty of free space, then you can resize the RECO disk group to a smaller size and reallocate the free space to the DATA disk group. The free space available after shrinking the RECO disk group is at a non-contiguous offset from the existing space allocations for the DATA disk group. Grid disks can use space anywhere on the cell disks and do not have to be contiguous.

If you are expanding the grid disks and the cell disks already have sufficient space to expand the existing grid disks, then you do not need to first resize an existing disk group. You would skip steps 2 and 3 below where the example shows the RECO disk group and grid disks are shrunk (you should still verify the cell disks have enough free space before growing the DATA grid disks). The amount of free space the administrator should reserve depends on the level of failure coverage.

If you are shrinking the size of the grid disks, you should understand how space is reserved for mirroring. Data is protected by Oracle ASM using normal or high redundancy to create one or two copies of data, which are stored as file extents. These copies are stored in separate failure groups. A failure in one failure group does not affect the mirror copies, so data is still accessible.

When a failure occurs, Oracle ASM re-mirrors, or **rebalances**, any extents that are not accessible so that redundancy is reestablished. For the re-mirroring process to succeed, sufficient free space must exist in the disk group to allow creation of the new file extent mirror copies. If there is not enough free space, then some extents will not be re-mirrored and the subsequent failure of the other data copies will require the disk group to be restored from backup. Oracle ASM sends an error when a re-mirror process fails due to lack of space.

You must be using Oracle Exadata System Software release 12.1.2.1.0 or higher, or have the patch for bug 19695225 applied to your software.

This procedure for resizing grid disks applies to bare metal and virtual machine (VM) deployments.

- Determine the Amount of Available Space
  To increase the size of the disks in a disk group you must either have unallocated disk space available, or you have to reallocate space currently used by a different disk group.
- Shrink the Oracle ASM Disks in the Donor Disk Group
  If there is no free space available on the cell disks, you can reduce the space used by one disk group to provide additional disk space for a different disk group.
- Shrink the Grid Disks in the Donor Disk Group
  After shrinking the disks in the Oracle ASM disk group, you then shrink the size of the grid disks on each cell.
- Increase the Size of the Grid Disks Using Available Space
  You can increase the size used by the grid disks if there is unallocated disk space either already available, or made available by shrinking the space used by a different Oracle ASM disk group.
- Increase the Size of the Oracle ASM Disks
  You can increase the size used by the Oracle ASM disks after increasing the space allocated to the associated grid disks.

Related Topics

- Understanding ASM Capacity and Reservation of Free Space in Exadata (My Oracle Support Doc ID 1551288.1)

- Bug 19695225 - Running Many Create or Alter Griddisk Commands Over Time Causes Cell Disk Metadata Corruption (ORA-600 [addNewSegmentsToGDisk_2]) and Loss of Cell Disk Content (My Oracle Support Doc ID 1991445.1)

3.3.3.1 Determine the Amount of Available Space

To increase the size of the disks in a disk group you must either have unallocated disk space available, or you have to reallocate space currently used by a different disk group.

You can also use a script available in "Script to Calculate New Grid Disk and Disk Group Sizes in Exadata (My Oracle Support Doc ID 1464809.1)" to assist in determining how much free space is available to shrink a disk group.

1. View the space currently used by the disk groups.

```
SELECT name, total_mb, free_mb, total_mb - free_mb used_mb,
round(100*free_mb/total_mb,2) pct_free
FROM v$asm_diskgroup
ORDER BY 1;

NAME                             TOTAL_MB    FREE_MB    USED_MB
PCT_FREE
------------------------------ ---------- ---------- ----------
----------
DATAC1                           68812800    9985076   58827724
14.51
RECOC1                           94980480   82594920   12385560
86.96
```

   The example above shows that the DATAC1 disk group has only about 15% of free space available while the RECOC1 disk group has about 87% free disk space. The PCT_FREE displayed here is raw free space, not usable free space. Additional space is needed for rebalancing operations.

2. For the disk groups you plan to resize, view the count and status of the failure groups used by the disk groups.

```
SELECT dg.name, d.failgroup, d.state, d.header_status, d.mount_mode,
 d.mode_status, count(1) num_disks
FROM V$ASM_DISK d, V$ASM_DISKGROUP dg
WHERE d.group_number = dg.group_number
AND dg.name IN ('RECOC1', 'DATAC1')
GROUP BY dg.name, d.failgroup, d.state, d.header_status, d.mount_status,
  d.mode_status
ORDER BY 1, 2, 3;

NAME       FAILGROUP      STATE      HEADER_STATU MOUNT_S  MODE_ST
NUM_DISKS
---------- -------------  ---------- ------------ -------- -------
---------
DATAC1     EXA01CELADM01  NORMAL     MEMBER        CACHED  ONLINE   12
DATAC1     EXA01CELADM02  NORMAL     MEMBER        CACHED  ONLINE   12
DATAC1     EXA01CELADM03  NORMAL     MEMBER        CACHED  ONLINE   12
DATAC1     EXA01CELADM04  NORMAL     MEMBER        CACHED  ONLINE   12
```

```
DATAC1     EXA01CELADM05  NORMAL     MEMBER        CACHED  ONLINE   12
DATAC1     EXA01CELADM06  NORMAL     MEMBER        CACHED  ONLINE   12
DATAC1     EXA01CELADM07  NORMAL     MEMBER        CACHED  ONLINE   12
DATAC1     EXA01CELADM08  NORMAL     MEMBER        CACHED  ONLINE   12
DATAC1     EXA01CELADM09  NORMAL     MEMBER        CACHED  ONLINE   12
DATAC1     EXA01CELADM10  NORMAL     MEMBER        CACHED  ONLINE   12
DATAC1     EXA01CELADM11  NORMAL     MEMBER        CACHED  ONLINE   12
DATAC1     EXA01CELADM12  NORMAL     MEMBER        CACHED  ONLINE   12
DATAC1     EXA01CELADM13  NORMAL     MEMBER        CACHED  ONLINE   12
DATAC1     EXA01CELADM14  NORMAL     MEMBER        CACHED  ONLINE   12
RECOC1     EXA01CELADM01  NORMAL     MEMBER        CACHED  ONLINE   12
RECOC1     EXA01CELADM02  NORMAL     MEMBER        CACHED  ONLINE   12
RECOC1     EXA01CELADM03  NORMAL     MEMBER        CACHED  ONLINE   12
RECOC1     EXA01CELADM04  NORMAL     MEMBER        CACHED  ONLINE   12
RECOC1     EXA01CELADM05  NORMAL     MEMBER        CACHED  ONLINE   12
RECOC1     EXA01CELADM06  NORMAL     MEMBER        CACHED  ONLINE   12
RECOC1     EXA01CELADM07  NORMAL     MEMBER        CACHED  ONLINE   12
RECOC1     EXA01CELADM08  NORMAL     MEMBER        CACHED  ONLINE   12
RECOC1     EXA01CELADM09  NORMAL     MEMBER        CACHED  ONLINE   12
RECOC1     EXA01CELADM10  NORMAL     MEMBER        CACHED  ONLINE   12
RECOC1     EXA01CELADM11  NORMAL     MEMBER        CACHED  ONLINE   12
RECOC1     EXA01CELADM12  NORMAL     MEMBER        CACHED  ONLINE   12
RECOC1     EXA01CELADM13  NORMAL     MEMBER        CACHED  ONLINE   12
RECOC1     EXA01CELADM14  NORMAL     MEMBER        CACHED  ONLINE   12
```

The above example is for a full rack, which has 14 cells and 14 failure groups for DATAC1 and RECOC1. Verify that each failure group has at least 12 disks in the NORMAL state (num_disks). If you see disks listed as MISSING, or you see an unexpected number of disks for your configuration, then do not proceed until you resolve the problem.

Extreme Flash systems should see a disk count of 8 instead of 12 for num_disks.

3. List the corresponding grid disks associated with each cell and each failure group, so you know which grid disks to resize.

```
SELECT dg.name, d.failgroup, d.path
FROM V$ASM_DISK d, V$ASM_DISKGROUP dg
WHERE d.group_number = dg.group_number
AND dg.name IN ('RECOC1', 'DATAC1')
ORDER BY 1, 2, 3;

NAME        FAILGROUP      PATH
----------- -------------
----------------------------------------------
DATAC1      EXA01CELADM01  o/192.168.74.43/DATAC1_CD_00_exa01celadm01
DATAC1      EXA01CELADM01  o/192.168.74.43/DATAC1_CD_01_exa01celadm01
DATAC1      EXA01CELADM01  o/192.168.74.43/DATAC1_CD_02_exa01celadm01
DATAC1      EXA01CELADM01  o/192.168.74.43/DATAC1_CD_03_exa01celadm01
DATAC1      EXA01CELADM01  o/192.168.74.43/DATAC1_CD_04_exa01celadm01
DATAC1      EXA01CELADM01  o/192.168.74.43/DATAC1_CD_05_exa01celadm01
DATAC1      EXA01CELADM01  o/192.168.74.43/DATAC1_CD_06_exa01celadm01
DATAC1      EXA01CELADM01  o/192.168.74.43/DATAC1_CD_07_exa01celadm01
DATAC1      EXA01CELADM01  o/192.168.74.43/DATAC1_CD_08_exa01celadm01
DATAC1      EXA01CELADM01  o/192.168.74.43/DATAC1_CD_09_exa01celadm01
```

```
DATAC1      EXA01CELADM01  o/192.168.74.43/DATAC1_CD_10_exa01celadm01
DATAC1      EXA01CELADM01  o/192.168.74.43/DATAC1_CD_11_exa01celadm01
DATAC1      EXA01CELADM02  o/192.168.74.44/DATAC1_CD_00_exa01celadm01
DATAC1      EXA01CELADM02  o/192.168.74.44/DATAC1_CD_01_exa01celadm01
DATAC1      EXA01CELADM02  o/192.168.74.44/DATAC1_CD_02_exa01celadm01
...
RECOC1      EXA01CELADM13  o/192.168.74.55/RECOC1_CD_00_exa01celadm13
RECOC1      EXA01CELADM13  o/192.168.74.55/RECOC1_CD_01_exa01celadm13
RECOC1      EXA01CELADM13  o/192.168.74.55/RECOC1_CD_02_exa01celadm13
...
RECOC1      EXA01CELADM14  o/192.168.74.56/RECOC1_CD_09_exa01celadm14
RECOC1      EXA01CELADM14  o/192.168.74.56/RECOC1_CD_10_exa01celadm14
RECOC1      EXA01CELADM14  o/192.168.74.56/RECOC1_CD_11_exa01celadm14

168 rows returned.
```

4. Check the cell disks for available free space.

   Free space on the cell disks can be used to increase the size of the DATAC1 grid disks. If there is not enough available free space to expand the DATAC1 grid disks, then you must shrink the RECOC1 grid disks to provide the additional space for the desired new size of DATAC1 grid disks.

   ```
   [root@exa01adm01 tmp]# dcli -g ~/cell_group -l root "cellcli -e list
   celldisk \
     attributes name,freespace"
   exa01celadm01: CD_00_exa01celadm01 0
   exa01celadm01: CD_01_exa01celadm01 0
   exa01celadm01: CD_02_exa01celadm01 0
   exa01celadm01: CD_03_exa01celadm01 0
   exa01celadm01: CD_04_exa01celadm01 0
   exa01celadm01: CD_05_exa01celadm01 0
   exa01celadm01: CD_06_exa01celadm01 0
   exa01celadm01: CD_07_exa01celadm01 0
   exa01celadm01: CD_08_exa01celadm01 0
   exa01celadm01: CD_09_exa01celadm01 0
   exa01celadm01: CD_10_exa01celadm01 0
   exa01celadm01: CD_11_exa01celadm01 0
   ...
   ```

   In this example, there is no free space available, so you must shrink the RECOC1 grid disks first to provide space for the DATAC1 grid disks. In your configuration there might be plenty of free space available and you can use that free space instead of shrinking the RECOC1 grid disks.

5. Calculate the amount of space to shrink from the RECOC1 disk group and from each grid disk.

   The minimum size to safely shrink a disk group and its grid disks must take into account the following:

   - Space currently in use (`USED_MB`)
   - Space expected for growth (`GROWTH_MB`)
   - Space needed to rebalance in case of disk failure (`DFC_MB`), typically 15% of total disk group size

The minimum size calculation taking the above factors into account is:

```
Minimum DG size (MB) = (USED_MB + GROWTH_MB ) * 1.15
```

- USED_MB can be derived from V$ASM_DISKGROUP by calculating TOTAL_MB - FREE_MB
- GROWTH_MB is an estimate specific to how the disk group will be used in the future and should be based on historical patterns of growth

For the RECOC1 disk group space usage shown in step 1, we see the minimum size it can shrink to assuming no growth estimates is:

Minimum RECOC1 size = (TOTAL_MB - FREE_MB + GROWTH_MB) * 1.15

= (94980480 - 82594920 + 0) * 1.15 = 14243394 MB = 13,910 GB

In the example output shown in Step 1, RECOC1 has plenty of free space and DATAC1 has less than 15% free. So, you could shrink RECOC1 and give the freed disk space to DATAC1. If you decide to reduce RECOC1 to half of its current size, the new size is 94980480 / 2 = 47490240 MB. This size is significantly above the minimum size we calculated for the RECOC1 disk group above, so it is safe to shrink it down to this value.

The query in Step 2 shows that there are 168 grid disks for RECOC1, because there are 14 cells and 12 disks per cell (14 * 12 = 168). The estimated new size of each grid disk for the RECOC1 disk group is 47490240 / 168, or 282,680 MB.

Find the closest 16 MB boundary for the new grid disk size. If you do not perform this check, then the cell will round down the grid disk size to the nearest 16 MB boundary automatically, and you could end up with a mismatch in size between the Oracle ASM disks and the grid disks.

```
SQL> SELECT 16*TRUNC(&new_disk_size/16) new_disk_size FROM dual;
Enter value for new_disk_size: 282680

NEW_DISK_SIZE
-------------
       282672
```

Based on the above result, you should choose 282672 MB as the new size for the grid disks in the RECOC1 disk group. After resizing the grid disks, the size of the RECOC1 disk group will be 47488896 MB.

6. Calculate how much to increase the size of each grid disk in the DATAC1 disk group.

   Ensure the Oracle ASM disk size and the grid disk sizes match across the entire disk group. The following query shows the combinations of disk sizes in each disk group. Ideally, there is only one size found for all disks and the sizes of both the Oracle ASM (total_mb) disks and the grid disks (os_mb) match.

```
SELECT dg.name, d.total_mb, d.os_mb, count(1) num_disks
FROM v$asm_diskgroup dg, v$asm_disk d
WHERE dg.group_number = d.group_number
GROUP BY dg.name, d.total_mb, d.os_mb;

NAME                             TOTAL_MB      OS_MB   NUM_DISKS
```

```
------------------------------ ---------- ---------- ----------
DATAC1                             409600     409600        168
RECOC1                             565360     565360        168
```

After shrinking RECOC1's grid disks, the following space is left per disk for DATAC1:

Additional space for DATAC1 disks = *RECOC1_current_size - RECOC1_new_size*
= 565360 - 282672 = 282688 MB

To calculate the new size of the grid disks for the DATAC1 disk group, use the following:

DATAC1's disks new size = *DATAC1_ disks_current_size + new_free_space_from_RECOC1*
= 409600 + 282688 = 692288 MB

Find the closest 16 MB boundary for the new grid disk size. If you do not perform this check, then the cell will round down the grid disk size to the nearest 16 MB boundary automatically, and you could end up with a mismatch in size between the Oracle ASM disks and the grid disks.

```
SQL> SELECT 16*TRUNC(&new_disk_size/16) new_disk_size FROM dual;
Enter value for new_disk_size: 692288

NEW_DISK_SIZE
-------------
       692288
```

Based on the query result, you can use the calculated size of 692288 MB for the disks in the DATAC1 disk groups because the size is on a 16 MB boundary. If the result of the query is different from the value you supplied, then you must use the value returned by the query because that is the value to which the cell will round the grid disk size.

The calculated value of the new grid disk size will result in the DATAC1 disk group having a total size of 116304384 MB (168 disks * 692288 MB).

3.3.3.2 Shrink the Oracle ASM Disks in the Donor Disk Group

If there is no free space available on the cell disks, you can reduce the space used by one disk group to provide additional disk space for a different disk group.

This task is a continuation of a example where space in the RECOC1 disk group is being reallocated to the DATAC1 disk group.

Before resizing the disk group, make sure the disk group you are taking space from has sufficient free space.

1. Shrink the Oracle ASM disks for the RECO disk group down to the new desired size for all disks.

   Use the new size for the disks in the RECO disk group that was calculated in Step 5 of Determine the Amount of Available Space.

   ```
   SQL> ALTER DISKGROUP recoc1 RESIZE ALL SIZE 282672M REBALANCE POWER 64;
   ```

> **Note:**
>
> The ALTER DISKGROUP command may take several minutes to complete. The SQL prompt will not return until this operation has completed.
>
> If the specified disk group has quorum disks configured within the disk group, then the ALTER DISKGROUP ... RESIZE ALL command could fail with error ORA-15277. Quorum disks are configured if the requirements specified in Managing Quorum Disks for High Redundancy Disk Groups are met. As a workaround, you can specify the storage server failure group names (for the ones with a FAILURE_TYPE of REGULAR, not QUORUM) explicitly in the SQL command, for example:
>
> ```
> SQL> ALTER DISKGROUP recoc1 RESIZE DISKS IN FAILGROUP
> exacell01 SIZE 282672M,
> exacell02 SIZE 282672M, exacell03 SIZE 282672M REBALANCE POWER
> 64;
> ```

Wait for rebalance to finish by checking the view GV$ASM_OPERATION.

```
SQL> set lines 250 pages 1000
SQL> col error_code form a10
SQL> SELECT dg.name, o.*
  2  FROM gv$asm_operation o, v$asm_diskgroup dg
  3  WHERE o.group_number = dg.group_number;
```

Proceed to the next step ONLY when the query against GV$ASM_OPERATION shows no rows for the disk group being altered.

2. Verify the new size of the ASM disks using the following queries:

```
SQL> SELECT name, total_mb, free_mb, total_mb - free_mb used_mb,
  2   ROUND(100*free_mb/total_mb,2) pct_free
  3  FROM v$asm_diskgroup
  4  ORDER BY 1;

NAME                             TOTAL_MB    FREE_MB    USED_MB
PCT_FREE
------------------------------ ---------- ---------- ----------
----------
DATAC1                           68812800    9985076   58827724
14.51
RECOC1                           47488896   35103336   12385560
73.92

SQL> SELECT dg.name, d.total_mb, d.os_mb, COUNT(1) num_disks
  2  FROM v$asm_diskgroup dg, v$asm_disk d
  3  WHERE dg.group_number = d.group_number
  4  GROUP BY dg.name, d.total_mb, d.os_mb;

NAME                             TOTAL_MB      OS_MB  NUM_DISKS
------------------------------ ---------- ---------- ----------
```

```
DATAC1                                409600      409600         168
RECOC1                                282672      565360         168
```

The above query example shows that the disks in the RECOC1 disk group have been resized to a size of 282672 MG each, and the total disk group size is 47488896 MB.

3.3.3.3 Shrink the Grid Disks in the Donor Disk Group

After shrinking the disks in the Oracle ASM disk group, you then shrink the size of the grid disks on each cell.

This task is a continuation of a example where space in the RECOC1 disk group is being reallocated to the DATAC1 disk group.

You must have first completed the task Shrink the Oracle ASM Disks in the Donor Disk Group.

1. Shrink the grid disks associated with the RECO disk group on all cells down to the new, smaller size.

   For each storage cell identified in Determine the Amount of Available Space in Step 3, shrink the grid disks to match the size of the Oracle ASM disks that were shrunk in the previous task. Use commands similar to the following:

   ```
   dcli -c exa01celadm01 -l root "cellcli -e alter griddisk
   RECOC1_CD_00_exa01celadm01 \
   ,RECOC1_CD_01_exa01celadm01 \
   ,RECOC1_CD_02_exa01celadm01 \
   ,RECOC1_CD_03_exa01celadm01 \
   ,RECOC1_CD_04_exa01celadm01 \
   ,RECOC1_CD_05_exa01celadm01 \
   ,RECOC1_CD_06_exa01celadm01 \
   ,RECOC1_CD_07_exa01celadm01 \
   ,RECOC1_CD_08_exa01celadm01 \
   ,RECOC1_CD_09_exa01celadm01 \
   ,RECOC1_CD_10_exa01celadm01 \
   ,RECOC1_CD_11_exa01celadm01 \
   size=282672M "

   dcli -c exa01celadm02 -l root "cellcli -e alter griddisk
   RECOC1_CD_00_exa01celadm02 \
   ,RECOC1_CD_01_exa01celadm02 \
   ,RECOC1_CD_02_exa01celadm02 \
   ,RECOC1_CD_03_exa01celadm02 \
   ,RECOC1_CD_04_exa01celadm02 \
   ,RECOC1_CD_05_exa01celadm02 \
   ,RECOC1_CD_06_exa01celadm02 \
   ,RECOC1_CD_07_exa01celadm02 \
   ,RECOC1_CD_08_exa01celadm02 \
   ,RECOC1_CD_09_exa01celadm02 \
   ,RECOC1_CD_10_exa01celadm02 \
   ,RECOC1_CD_11_exa01celadm02 \
   size=282672M "

   ...
   ```

```
dcli -c exa01celadm14 -l root "cellcli -e alter griddisk
RECOC1_CD_00_exa01celadm14 \
,RECOC1_CD_01_exa01celadm14 \
,RECOC1_CD_02_exa01celadm14 \
,RECOC1_CD_03_exa01celadm14 \
,RECOC1_CD_04_exa01celadm14 \
,RECOC1_CD_05_exa01celadm14 \
,RECOC1_CD_06_exa01celadm14 \
,RECOC1_CD_07_exa01celadm14 \
,RECOC1_CD_08_exa01celadm14 \
,RECOC1_CD_09_exa01celadm14 \
,RECOC1_CD_10_exa01celadm14 \
,RECOC1_CD_11_exa01celadm14 \
size=282672M "
```

2. Verify the new size of the grid disks using the following command:

```
[root@exa01adm01 tmp]# dcli -g cell_group -l root "cellcli -e list griddisk
attributes name,size where name like \'RECOC1.*\' "

exa01celadm01: RECOC1_CD_00_exa01celadm01 276.046875G
exa01celadm01: RECOC1_CD_01_exa01celadm01 276.046875G
exa01celadm01: RECOC1_CD_02_exa01celadm01 276.046875G
exa01celadm01: RECOC1_CD_03_exa01celadm01 276.046875G
exa01celadm01: RECOC1_CD_04_exa01celadm01 276.046875G
exa01celadm01: RECOC1_CD_05_exa01celadm01 276.046875G
exa01celadm01: RECOC1_CD_06_exa01celadm01 276.046875G
exa01celadm01: RECOC1_CD_07_exa01celadm01 276.046875G
exa01celadm01: RECOC1_CD_08_exa01celadm01 276.046875G
exa01celadm01: RECOC1_CD_09_exa01celadm01 276.046875G
exa01celadm01: RECOC1_CD_10_exa01celadm01 276.046875G
exa01celadm01: RECOC1_CD_11_exa01celadm01 276.046875G
...
```

The above example shows that the disks in the RECOC1 disk group have been resized to a size of 282672 MB each (276.046875 * 1024).

3.3.3.4 Increase the Size of the Grid Disks Using Available Space

You can increase the size used by the grid disks if there is unallocated disk space either already available, or made available by shrinking the space used by a different Oracle ASM disk group.

This task is a continuation of a example where space in the RECOC1 disk group is being reallocated to the DATAC1 disk group. If you already have sufficient space to expand an existing disk group, then you do not need to reallocate space from a different disk group.

1. Check that the cell disks have the expected amount of free space.

After completing the tasks to shrink the Oracle ASM disks and the grid disks, you would expect to see the following free space on the cell disks:

```
[root@exa01adm01 tmp]# dcli -g ~/cell_group -l root "cellcli -e list
celldisk \
attributes name,freespace"

exa01celadm01: CD_00_exa01celadm01 276.0625G
exa01celadm01: CD_01_exa01celadm01 276.0625G
exa01celadm01: CD_02_exa01celadm01 276.0625G
exa01celadm01: CD_03_exa01celadm01 276.0625G
exa01celadm01: CD_04_exa01celadm01 276.0625G
exa01celadm01: CD_05_exa01celadm01 276.0625G
exa01celadm01: CD_06_exa01celadm01 276.0625G
exa01celadm01: CD_07_exa01celadm01 276.0625G
exa01celadm01: CD_08_exa01celadm01 276.0625G
exa01celadm01: CD_09_exa01celadm01 276.0625G
exa01celadm01: CD_10_exa01celadm01 276.0625G
exa01celadm01: CD_11_exa01celadm01 276.0625G
...
```

2. For each storage cell, increase the size of the DATA grid disks to the desired new size.

   Use the size calculated in Determine the Amount of Available Space.

```
dcli -c exa01celadm01 -l root "cellcli -e alter griddisk
DATAC1_CD_00_exa01celadm01 \
,DATAC1_CD_01_exa01celadm01 \
,DATAC1_CD_02_exa01celadm01 \
,DATAC1_CD_03_exa01celadm01 \
,DATAC1_CD_04_exa01celadm01 \
,DATAC1_CD_05_exa01celadm01 \
,DATAC1_CD_06_exa01celadm01 \
,DATAC1_CD_07_exa01celadm01 \
,DATAC1_CD_08_exa01celadm01 \
,DATAC1_CD_09_exa01celadm01 \
,DATAC1_CD_10_exa01celadm01 \
,DATAC1_CD_11_exa01celadm01 \
size=692288M "
...
dcli -c exa01celadm14 -l root "cellcli -e alter griddisk
DATAC1_CD_00_exa01celadm14 \
,DATAC1_CD_01_exa01celadm14 \
,DATAC1_CD_02_exa01celadm14 \
,DATAC1_CD_03_exa01celadm14 \
,DATAC1_CD_04_exa01celadm14 \
,DATAC1_CD_05_exa01celadm14 \
,DATAC1_CD_06_exa01celadm14 \
,DATAC1_CD_07_exa01celadm14 \
,DATAC1_CD_08_exa01celadm14 \
,DATAC1_CD_09_exa01celadm14 \
,DATAC1_CD_10_exa01celadm14 \
,DATAC1_CD_11_exa01celadm14 \
size=692288M "
```

3. Verify the new size of the grid disks associated with the DATAC1 disk group using the following command:

```
dcli -g cell_group -l root "cellcli -e list griddisk attributes
name,size \
where name like \'DATAC1.*\' "

exa01celadm01: DATAC1_CD_00_exa01celadm01 676.0625G
exa01celadm01: DATAC1_CD_01_exa01celadm01 676.0625G
exa01celadm01: DATAC1_CD_02_exa01celadm01 676.0625G
exa01celadm01: DATAC1_CD_03_exa01celadm01 676.0625G
exa01celadm01: DATAC1_CD_04_exa01celadm01 676.0625G
exa01celadm01: DATAC1_CD_05_exa01celadm01 676.0625G
exa01celadm01: DATAC1_CD_06_exa01celadm01 676.0625G
exa01celadm01: DATAC1_CD_07_exa01celadm01 676.0625G
exa01celadm01: DATAC1_CD_08_exa01celadm01 676.0625G
exa01celadm01: DATAC1_CD_09_exa01celadm01 676.0625G
exa01celadm01: DATAC1_CD_10_exa01celadm01 676.0625G
exa01celadm01: DATAC1_CD_11_exa01celadm01 676.0625G
```

Instead of increasing the size of the DATA disk group, you could instead create new disk groups with the new free space or keep it free for future use. In general, Oracle recommends using the smallest number of disk groups needed (typically DATA, RECO, and DBFS_DG) to give the greatest flexibility and ease of administration. However, there may be cases, perhaps when using virtual machines or consolidating many databases, where additional disk groups or available free space for future use may be desired.

If you decide to leave free space on the grid disks in reserve for future use, please see the My Oracle Support Note 1684112.1 for the steps on how to allocate free space to an existing disk group at a later time.

Related Topics

- How to resize ASM disks in Exadata (My Oracle Support Doc ID 1684112.1)
- Determine the Amount of Available Space
  To increase the size of the disks in a disk group you must either have unallocated disk space available, or you have to reallocate space currently used by a different disk group.

3.3.3.5 Increase the Size of the Oracle ASM Disks

You can increase the size used by the Oracle ASM disks after increasing the space allocated to the associated grid disks.

This task is a continuation of a example where space in the RECOC1 disk group is being reallocated to the DATAC1 disk group.

You must have completed the task of resizing the grid disks before you can resize the corresponding Oracle ASM disk group.

1. Increase the Oracle ASM disks for DATAC1 disk group to the new size of the grid disks on the storage cells.

```
SQL> ALTER DISKGROUP datac1 RESIZE ALL;
```

This command resizes the Oracle ASM disks to match the size of the grid disks.

> **Note:**
>
> If the specified disk group has quorum disks configured within the disk group, then the `ALTER DISKGROUP ... RESIZE ALL` command could fail with error `ORA-15277`. Quorum disks are configured if the requirements specified in *Oracle Exadata Database Machine Maintenance Guide* are met.
>
> As a workaround, you can specify the storage server failure group names (for the ones of FAILURE_TYPE "REGULAR", not "QUORUM") explicitly in the SQL command, for example:
>
> ```
> SQL> ALTER DISKGROUP datac1 RESIZE DISKS IN FAILGROUP
> exacell01, exacell02, exacell03;
> ```

2. Wait for the rebalance operation to finish.

```
SQL> set lines 250 pages 1000
SQL> col error_code form a10
SQL> SELECT dg.name, o.* FROM gv$asm_operation o, v$asm_diskgroup dg
     WHERE o.group_number = dg.group_number;
```

Do not continue to the next step until the query returns zero rows for the disk group that was altered.

3. Verify that the new sizes for the Oracle ASM disks and disk group is at the desired sizes.

```
SQL> SELECT name, total_mb, free_mb, total_mb - free_mb used_mb,
     ROUND(100*free_mb/total_mb,2) pct_free
     FROM v$asm_diskgroup
     ORDER BY 1;

NAME                             TOTAL_MB    FREE_MB    USED_MB
PCT_FREE
------------------------------ ---------- ---------- ----------
----------
DATAC1                          116304384   57439796   58864588
49.39
RECOC1                           47488896   34542516   12946380
72.74

SQL>  SELECT dg.name, d.total_mb, d.os_mb, COUNT(1) num_disks
      FROM  v$asm_diskgroup dg, v$asm_disk d
      WHERE dg.group_number = d.group_number
      GROUP BY dg.name, d.total_mb, d.os_mb;

NAME                             TOTAL_MB      OS_MB  NUM_DISKS
------------------------------ ---------- ---------- ----------
DATAC1                             692288     692288        168
```

```
RECOC1                               282672      282672         168
```

The results of the queries show that the RECOC1 and DATAC1 disk groups and disk have been resized.

Related Topics

- Determine the Amount of Available Space
  To increase the size of the disks in a disk group you must either have unallocated disk space available, or you have to reallocate space currently used by a different disk group.

3.3.4 Determining Which Oracle ASM Disk Group Contains an Oracle Exadata Storage Server Grid Disk

If a grid disk name matches the Oracle ASM disk name, and the name contains the Oracle ASM disk group name, then you can determine the Oracle ASM disk group to which the grid disk belongs.

You can also use SQL commands on the Oracle ASM instance to find the Oracle ASM disk group that matches part of the specific grid disk name. This can help you to determine which Oracle ASM disk group contains a specific grid disk.

Example 3-7 Determining Grid Disks in an Oracle ASM Disk Group

This example shows how to find the Oracle ASM disk group that contains grid disks that begin with DATA0, for example DATA0_CD_03_CELL04.

```
SQL> SELECT d.label AS asmdisk, dg.name AS diskgroup
     FROM V$ASM_DISK d, V$ASM_DISKGROUP dg
     WHERE dg.name LIKE 'DATA0%'
           AND d.group_number = dg.group_number;

ASMDISK                DISKGROUP
---------------------- -------------
DATA0_CD_00_CELL04      DATA0
DATA0_CD_01_CELL04      DATA0
DATA0_CD_02_CELL04      DATA0
DATA0_CD_03_CELL04      DATA0
```

3.3.5 Determining Which Oracle Exadata Storage Server Grid Disks Belong to an Oracle ASM Disk Group

If a grid disk name contains the Oracle ASM disk group name, then you can use SQL commands on the Oracle ASM instance to list the Oracle ASM disk group names.

You can use the CellCLI utility to search for specific grid disk names.

Example 3-8 Displaying Oracle ASM Disk Group Names

This example shows how to use a SQL command to display the Oracle ASM disk group names on the Oracle ASM instance.

```
SQL> SELECT name FROM V$ASM_DISKGROUP;

NAME
------------------------------
CONTROL
DATA0
DATA1
DATA2
LOG
STANDBY
```

Example 3-9 Searching for Grid Disks by Name

This example shows how to display similar grid disk group names on the cell using the `dcli` utility.

```
$ ./dcli "cellcli -e list griddisk where -c cell04"

data0_CD_01_cell04
data0_CD_02_cell04
data0_CD_03_cell04
...
```

Related Topics

- Naming Conventions for Oracle Exadata Storage Server Grid Disks
  Using a consistent naming convention helps to identify Exadata components.

3.3.6 Handling Disk Replacement

If a disk has a problem, the physical disk status changes.

When a physical disk is removed, its status becomes `not present`. Oracle ASM may take a grid disk offline when getting I/O errors while trying to access a grid disk on the physical disk. When the physical disk is replaced, Oracle Exadata System Software automatically puts the grid disks on the physical disk online in their respective Oracle ASM disk groups. If a grid disk remains offline longer than the time specified by the `disk_repair_time` attribute, then Oracle ASM force drops that grid disk and starts a rebalance to restore data redundancy. Oracle ASM monitors the rebalance operation, and Oracle Exadata System Software sends an e-mail message when the operation is complete.

The following table summarizes the physical disk statuses, and how Oracle ASM handles grid disks when the physical disk has a problem.

Table 3-1 Physical Disk Status

Physical Disk Status	Oracle Exadata System Software Action
`normal` Disk is functioning normally	No action.
`not present` Disk has been removed	Oracle Exadata System Software offlines disk, then uses the `DROP ... FORCE` command after `disk_repair_time` limit exceeded. The rebalance operation begins.
`predictive failure` Disk is having problems, and may fail	Oracle Exadata System Software drops the grid disks on the affected physical disk without the `FORCE` option from Oracle ASM, and the rebalance operation copies the data on the affected physical disk to other disks. After all grid disks have been successfully removed from their respective Oracle ASM disk groups, administrators can proceed with disk replacement.
`critical` Disk has failed	Oracle Exadata System Software drops the grid disks using the `DROP ... FORCE` command on the affected physical disk from Oracle ASM, and the rebalance operation restores data redundancy. Administrators can proceed with disk replacement immediately. This status is only available for releases 11.2.3.1.1 and earlier.
`poor performance` Disk is performing poorly	Oracle Exadata System Software attempts to drop the grid disks using the `FORCE` option on the affected physical disk from Oracle ASM. If the `DROP ... FORCE` command is successful, then the rebalance operation begins to restore data redundancy and administrators can proceed with disk replacement immediately. If the `DROP ... FORCE` command fails due to offline partners, Oracle Exadata System Software drops the grid disks on the affected physical disk without the `FORCE` option from Oracle ASM, and the rebalance operation copies the data on the affected physical disk to other disks. After all grid disks have been successfully removed from their respective Oracle ASM disk groups, administrators can proceed with disk replacement.

After a physical disk is replaced, Oracle Exadata System Software automatically creates the grid disks on the replacement disk, and adds them to the respective Oracle ASM disk groups. An Oracle ASM rebalance operation relocates data to the newly-added grid disks. Oracle ASM monitors the rebalance operation, and Oracle Exadata System Software sends an e-mail message when the operation is complete.

Related Topics

- Maintaining the Hard Disks of Exadata Storage Servers

4
Maintaining Oracle Exadata System Software

This section explains how to maintain Oracle Exadata System Software.

> **Caution:**
>
> All operations in this chapter must be performed with extreme caution and only after you have ensured you have complete backups of the data. If not, then you may experience irrecoverable data loss.

- Recommendations for Changing the Exadata Storage Server Network Address
  Review the following advice before changing the fundamental configuration of a storage server, such as changing the IP address, host name, or RDMA Network Fabric address.
- Using the ipconf Utility
  The ipconf utility is used to set and change the following parameters on Oracle Exadata Storage Servers.
- Oracle Exadata System Software Validation Tests and Utilities
  You can use a variety of commands and utilities to validate the Oracle Exadata System Software and hardware configurations.
- Locating Serial Numbers for System Components
  You may need to provide the serial numbers for the system components when contacting Oracle Support Services.
- Diagnostic and Repair Utilities
  Oracle Exadata System Software includes utilities for diagnostics and repair of Oracle Exadata Storage Server.
- System Diagnostics Data Gathering with sosreports and Oracle ExaWatcher
  You can use the sosreport utility and Oracle ExaWatcher to diagnose problems with your system.
- Serial Console Support
  The storage servers and database servers of Oracle Exadata Database Machine are configured to provide serial console access.
- Oracle Linux Kernel Crash Core Files
  The storage servers and database servers of Oracle Exadata Database Machine are configured to generate Oracle Linux kernel crash core files in the `/var/crash` directory, when the Oracle Linux operating system malfunctions or crashes.

4.1 Recommendations for Changing the Exadata Storage Server Network Address

Review the following advice before changing the fundamental configuration of a storage server, such as changing the IP address, host name, or RDMA Network Fabric address.

- Before changing the storage server configuration, ensure that all Oracle Automatic Storage Management (Oracle ASM), Oracle Real Application Clusters (Oracle RAC) and database instances that use the storage servers do not access the storage server while you are changing the IP address.
- After changing the storage server configuration, ensure that consumers of storage server services are correctly reconfigured to use the new connect information of the storage server. If Oracle Auto Service Request (ASR) is being used, then deactivate the asset from Oracle ASR Manager, and activate the asset with the new IP address.
- When changing a storage server configuration, change only one storage server at a time to ensure that Oracle ASM and Oracle RAC work properly during the changes.

Related Topics

- About Oracle Auto Service Request

4.2 Using the ipconf Utility

The ipconf utility is used to set and change the following parameters on Oracle Exadata Storage Servers.

During initial configuration of Oracle Exadata Database Machine, the utility also configures the database servers.

- IP address
- Host name
- NTP server
- Time zone
- DNS name servers
- InfiniBand addresses

The ipconf utility makes a back up copy of the files it modifies. When the utility is rerun, it overwrites the existing backup file. The log file maintains the complete history of every ipconf operation performed.

Table 4-1 ipconf Options

Option	Description
no option	Utility starts in main editing mode.

Table 4-1 (Cont.) ipconf Options

Option	Description
`-ignoremismatch`	Starts utility in main editing mode when there is a mismatch between the stored cell configuration and the running configuration.
`-ilom print`	Prints basic ILOM settings.
`-ilom set`	Sets basic ILOM settings.
`-verify [-verbose]`	Verifies the consistency between the stored cell configuration and the running configuration. The `-verbose` option shows all details. If `-verbose` is not used, then only errors are displayed. Success returns zero errors.
`-verify -semantic [-verbose]`	Verifies consistency and checks for access to all DNS and NTP servers. The `-verbose` option shows all details. If `-verbose` is not used, then only errors are displayed. Success returns zero errors.
`-verify -semantic-min [-verbose]`	Verifies the consistency and checks for access to at least one NTP and one DNS server. The `-verbose` option shows all details. If `-verbose` is not used, then only errors are displayed. Success returns zero errors.

The following example shows the display for the ipconf utility when setting the Sun ILOM interface.

Example 4-1 Using the ipconf Utility to Set the Sun ILOM Interface

```
# ipconf
Logging started to /var/log/cellos/ipconf.log
Interface ib0 is Linked. hca: mxx4_0
Interface ib1 is Linked. mxx4_0
Interface eth0 is Linked. driver/mac: igb/00:00:00:01:cd:01
Interface eth1 is ... Unlinked. driver/mac: igb/00:00:00:01:cd:02
Interface eth2 is ... Unlinked. driver/mac: igb/00:00:00:01:cd:03
Interface eth3 is ... Unlinked. driver/mac: igb/00:00:00:01:cd:04

Network interfaces
Name  State      IP address      Netmask         Gateway
Hostname
ib0
Linked
ib1
Linked
eth0
Linked
eth1
Unlinked
eth2
Unlinked
eth3
Unlinked
```

```
Warning. Some network interface(s) are disconnected. Check cables and
switches
         and retry
Do you want to retry (y/n) [y]: n

The current nameserver(s): 192.0.2.10 192.0.2.12 192.0.2.13
Do you want to change it (y/n) [n]:
The current timezone: America/Los_Angeles
Do you want to change it (y/n) [n]:
The current NTP server(s): 192.0.2.06 192.0.2.12 1192.0.2.13
Do you want to change it (y/n) [n]:

Network interfaces
Name  State   IP address      Netmask       Gateway        Hostname
eth0  Linked  192.0.2.151  255.255.252.0 192.0.2.15  Managment
myg.example.com
eth1  Unlinked
eth2  Unlinked
eth3  Unlinked
bond0 ib0,ib1    192.168.13.101  255.255.252.0        Private
myg.example.com

Select interface name to configure or press Enter to continue:

Select canonical hostname from the list below
1: myg.example.com
2: myg-private.example.com
Canonical fully qualified domain name [1]:

Select default gateway interface from the list below
1: eth01
Default gateway interface [1]:

Canonical hostname: myg.example.com
Nameservers: 192.0.2.10 192.0.2.12 192.0.2.13
Timezone: America/Los_Angeles
NTP servers: 192.0.2.06 192.0.2.12 192.0.2.13
Network interfaces
Name  State  IP address      Netmask        Gateway        Hostname
eth0  Linked 192.0.2.151 255.255.252.0  192.0.2.15  myg.example.com
eth1  Unlinked
eth2  Unlinked
eth3  Unlinked
bond0 ib0,ib1  192.168.13.101 255.255.252.0 Private  myg-priv.example.com
Is this correct (y/n) [y]:

Do you want to configure basic ILOM settings (y/n) [y]: y
Loading configuration settings from ILOM ...
ILOM Fully qualified hostname [myg_ilom.example.com]:
ILOM IP discovery (static/dhcp) [static]:
ILOM IP address [192.0.2.201]:
ILOM Netmask [255.255.252.0]:
ILOM Gateway or none [192.0.2.15]:
ILOM Nameserver or none: [192.0.2.10]:
ILOM Use NTP Servers (enabled/disabled) [enabled]:
```

```
ILOM First NTP server. Fully qualified hostname or ip address or none
[192.0.2.06]:
ILOM Second NTP server. Fully qualified hostname or ip address or none
[none]:

Basic ILOM configuration settings:
Hostname             : myg.example.com
IP Discovery         : static
IP Address           : 192.0.2.10
Netmask              : 255.255.252.0
Gateway              : 192.0.2.15
DNS servers          : 192.0.2.10
Use NTP servers      : enabled
First NTP server     : 192.0.2.06
Second NTP server    : none
Timezone (read-only) : America/Los Angeles

Is this correct (y/n) [y]:
```

4.3 Oracle Exadata System Software Validation Tests and Utilities

You can use a variety of commands and utilities to validate the Oracle Exadata System Software and hardware configurations.

- Summary of Software and Firmware Components on Oracle Exadata Storage Servers
  The `imageinfo` command located in the `/usr/local/bin/` directory provides a summary of release and status of the software and firmware components on Oracle Exadata Storage Servers.
- Oracle Exadata Storage Server Image History
  The `imagehistory` command lists the version history for Oracle Exadata Storage Server.
- Validation of the State and Health of the System
  The validation framework runs different tests under certain conditions, such as on first boot after recovery of an Oracle Exadata Storage Server using the rescue and recovery functionality of the CELLBOOT USB flash drive, or when patching an Oracle Exadata Storage Server.

4.3.1 Summary of Software and Firmware Components on Oracle Exadata Storage Servers

The `imageinfo` command located in the `/usr/local/bin/` directory provides a summary of release and status of the software and firmware components on Oracle Exadata Storage Servers.

The software and firmware components comprise the storage server image. The release and status information is required when working with My Oracle Support.

The following table lists the output fields from the `imageinfo` command.

Table 4-2 Description of `imageinfo` Command Output

Field	Description
Active image activated	Date stamp in UTC format when the image on the cell was considered completed, either successfully or unsuccessfully. A cell patch updates the time stamp to indicate the time the cell was patched.
Active image status	Status of the cell image based on the success or failure of a set of self-tests and configuration actions, collectively known as *validations*. When this status is undefined, empty or failure, then examine the different validation logs in the `/var/log/cellos` directory to determine the cause for the status.
Active image version	Main release version of the overall cell image indicating a specific combination of releases of operating system, core Oracle Exadata System Software (the cell rpm), and the firmware levels for most key components of the cell. A cell patch usually updates this information. The first five separated fields of the version match the standard way Oracle product releases are identified. The last field is the exact build number of the release. It corresponds to `YYMMDD` format of the build date.
Active system partition on device	Cell operating system root (`/`) partition device. A typical successful cell patch switches the cell from its active partitions to inactive partitions. Each successful cell patch keeps the cell switching between the active and inactive partitions. There are few occasions when the cell patch does not switch partitions. These are rare, and are known as *in-partition patches*.
Boot area has rollback archive for version	For a patched cell using non in-partition cell patch, this indicates whether there is a suitable back up archive that can be used to roll the cell back to the inactive image version. Existence of this archive is necessary but not sufficient for rolling back to inactive version of the cell image.
Cell boot usb partition	Oracle Exadata Storage Server boot and rescue USB partition.
Cell boot usb version	Version of the software on the boot USB. On a healthy cell this release must be identical to the value of the Active image version line.
Cell rpm version	Cell software version or cell rpm version as reported by the CellCLI utility.
Cell version	Release version as reported by the CellCLI utility.

Table 4-2 (Cont.) Description of imageinfo Command Output

Field	Description
In partition rollback	Some cell patches do not switch the partitions. These are in-partition patches. This field indicates whether there is enough information to roll back such patch.
Inactive image activated	Time stamp for activation of the inactive image. This field is similar to active image activated field.
Inactive image status	Status of the inactive image. This field is similar to the status of the active image.
Inactive image version	Version of the cell before the most-recent patch was applied.
Inactive software partition on device	Oracle Exadata System Software file system partition, /opt/oracle, for the inactive image.
Inactive system partition on device	The root (/) file system partition for the inactive image.
Kernel version	Operating system kernel version of the cell.
Rollback to inactive partition	Summary indicator for a non-in-partition patched cell indicating whether rollback can be run on the cell to take it back to inactive version of the cell image. On a new cell, this field is empty or has the value undefined.

The following is an example of the output from the imageinfo command:

```
Kernel version: 2.6.18-194.3.1.0.3.el5 #1 SMP Tue Aug 31 22:41:13 EDT 2010 x86_64
Cell version: OSS_MAIN_LINUX.X64_101105
Cell rpm version: cell-11.2.2.1.1_LINUX.X64_101105-1

Active image version: 11.2.2.1.1.101105
Active image activated: 2010-11-06 21:52:08 -0700
Active image status: success
Active system partition on device: /dev/md5
Active software partition on device: /dev/md7

In partition rollback: Impossible

Cell boot usb partition: /dev/sdm1
Cell boot usb version: 11.2.2.1.1.101105

Inactive image version: 11.2.1.3.1
Inactive image activated: 2010-08-28 20:01:30 -0700
Inactive image status: success
Inactive system partition on device: /dev/md6
Inactive software partition on device: /dev/md8

Boot area has rollback archive for the version: 11.2.1.3.1
Rollback to the inactive partitions: Possible
```

Related Topics

- Validation of the State and Health of the System
  The validation framework runs different tests under certain conditions, such as on first boot after recovery of an Oracle Exadata Storage Server using the rescue and recovery functionality of the CELLBOOT USB flash drive, or when patching an Oracle Exadata Storage Server.

4.3.2 Oracle Exadata Storage Server Image History

The `imagehistory` command lists the version history for Oracle Exadata Storage Server.

For example, if a storage server was updated from release 11.2.1.2.6 to release 11.2.1.3.1, and then updated to release 11.2.1.2.3, the `imagehistory` command displays this history. The following is an example of the output:

```
# imagehistory
Version                    : 11.2.1.2.3
Image activation date      : 2012-12-03 06:06:46 -0700
Imaging mode               : fresh
Imaging status             : success

Version                    : 11.2.3.2.0.120713
Image activation date      : 2012-12-12 17:56:31 -0700
Imaging mode               : out of partition upgrade
Imaging status             : success
```

4.3.3 Validation of the State and Health of the System

The validation framework runs different tests under certain conditions, such as on first boot after recovery of an Oracle Exadata Storage Server using the rescue and recovery functionality of the CELLBOOT USB flash drive, or when patching an Oracle Exadata Storage Server.

Validation framework is a set of validation tests that run at boot time at the `rc.local` level. The logs for the tests are available in the `/var/log/cellos/validations` directory.

Health check validations are a set of quick health checks on the system on each boot, such as basic health of the disks, and report the status. If a validation fails, then you should examine the log file for the cause as it may indicate potential problem requiring attention.

Automatic patch rollback occurs if one or more validation checks fail after patch application. Refer to the documentation for the specific patch.

Check for any failures reported in the `/var/log/cellos/vldrun.first_boot.log` file after the first boot configuration. For all subsequent boots, the `/var/log/cellos/validations.log` file contains information about failed validations. For each failed validation, perform the following procedure:

1. Look for `/var/log/cellos/validations/`*`failed_validation_name`*`.SuggestedRemedy` file. The file exists only if the validation process has identified some corrective action. Follow the suggestions in the file to correct the cause of the failure.

2. If the `SuggestedRemedy` file does not exist, then examine the log file for the failed validation in `/var/log/cellos/validations` to track down the cause, and correct it as needed.

4.4 Locating Serial Numbers for System Components

You may need to provide the serial numbers for the system components when contacting Oracle Support Services.

Serial numbers for system components can be determined by using the following procedure:

1. Log in as the `root` user.
2. Use the `CheckHWnFWProfile` command to view the serial numbers.

```
# /opt/oracle.SupportTools/CheckHWnFWProfile -action list -mode
serial_numbers
```

Each time the system is booted, the serial numbers are written to the `/var/log/cellos/validations/SerialNumbers` file. This file can be used as a historic record of the serial numbers. The file also contains configuration information for some components.

4.5 Diagnostic and Repair Utilities

Oracle Exadata System Software includes utilities for diagnostics and repair of Oracle Exadata Storage Server.

The utilities help diagnose and repair problems that may occur during the normal life cycle of Oracle Exadata Storage Servers. The utilities are in the `/opt/oracle.SupportTools` directory.

> **Note:**
>
> All utilities must be run as the `root` user from the `/opt/oracle.SupportTools` directory.

- The CheckHWnFWProfile Utility
  The CheckHWnFWProfile utility checks that the system meets the required hardware and firmware specifications, and reports any mismatches.
- The diagnostics.iso Utility
  The `diagnostics.iso` utility can be used to boot a server to diagnose serious problems when no other way exists to analyze the system due to damage to the system and its CELLBOOT USB flash drive.
- The ibdiagtools Utilities
  The most useful utilities of the ibdiagtools utilities are the `verify-topology`, `checkbadlinks.pl`, and `infinicheck` utilities.

- The make_cellboot_usb Utility
  The `make_cellboot_usb` utility allows you to rebuild a damaged CELLBOOT USB flash drive.

4.5.1 The CheckHWnFWProfile Utility

The CheckHWnFWProfile utility checks that the system meets the required hardware and firmware specifications, and reports any mismatches.

Table 4-3 CheckHWnFWProfile Utility Commands

Command	Description
`./CheckHWnFWProfile`	When run without options, the utility checks the existing hardware and firmware components against the expected values.
`./CheckHWnFWProfile -action` *`list`*	View the existing hardware and firmware versions on the system.
`./CheckHWnFWProfile -action alter_config -property HWFW_Checker_Updater_Status -value Disabled`	Disable the `CheckHWnFWProfile` utility.
`./CheckHWnFWProfile -action alter_config -property HWFW_Checker_Updater_Status -value Enabled`	Enable the `CheckHWnFWProfile` utility.
`./CheckHWnFWProfile -action check -component` *`list_of_components`*	Check specified components against the expected values.
`./CheckHWnFWProfile -action list -component` *`list_of_components`*	View the hardware and firmware versions of specified components on the system.
`./CheckHWnFWProfile -action list -mode serial_numbers`	List serial numbers. The list includes the following serial numbers: • System • Disk controller • Each disk • RDMA Network Fabric host channel adapter (HCA) Depending on the system, serial numbers for all the memory (RAM) modules may be included.
`./CheckHWnFWProfile -action list -mode supported_info`	View the expected hardware and firmware.

`./CheckHWnFWProfile -h`
`./CheckHWnFWProfile --help`
	View help and utility usage.

4.5.2 The diagnostics.iso Utility

The `diagnostics.iso` utility can be used to boot a server to diagnose serious problems when no other way exists to analyze the system due to damage to the system and its CELLBOOT USB flash drive.

Use this utility only under the guidance of Oracle Support Services. The `root` password should be available to Oracle Support Services, if needed.

4.5.3 The ibdiagtools Utilities

The most useful utilities of the ibdiagtools utilities are the `verify-topology`, `checkbadlinks.pl`, and `infinicheck` utilities.

> **Note:**
>
> These utilities only work on systems that have RDMA over InfiniBand Network Fabric.

The `verify-topology` utility checks the correctness and health of RDMA over InfiniBand Network Fabric connections. For example, it can determine if both cables from the server go to the same switch in the Oracle Exadata Rack. When both cables go to the same switch, the system loses the ability to fail over to another switch if the first switch fails.

The `checkbadlinks.pl` utility reports the links that are operating at 5 Gbps. This is usually an indication that the cables are loose, and need to be reseated.

The `infinicheck` utility reports the base RDMA over InfiniBand Network Fabric performance between servers in Oracle Exadata Database Machine, such as expected minimum throughput between the database server and storage server, between storage servers, and between a database server and another database server. This utility can help identify potential issues in the RDMA over InfiniBand Network Fabric fabric.

> **See Also:**
>
> - For detailed information about the ibdiagtools utilities, refer to the `README.txt` file in the `/opt/oracle.SupportTools/ibdiagtools/` directory.
> - Sample outputs from each utility are included in the `/opt/oracle.SupportTools/ibdiagtools/SampleOutputs.txt` file.

4.5.4 The make_cellboot_usb Utility

The `make_cellboot_usb` utility allows you to rebuild a damaged CELLBOOT USB flash drive.

Do not connect more than one USB flash drive to the system when running this utility. The utility builds on the first discovered USB flash drive on the system.

> **Note:**
>
> This utility can only be used on Oracle Exadata Storage Server.

- To see what is done before rebuilding the USB flash drive:

  ```
  cd /opt/oracle.SupportTools
  ./make_cellboot_usb -verbose
  ```

- To rebuild the USB flash drive, run the command with one of the following options: `-execute`, `-force`, or `-rebuild`.

  ```
  ./make_cellboot_usb -execute
  ```

  Or:

  ```
  ./make_cellboot_usb -force
  ```

  Or:

  ```
  ./make_cellboot_usb -rebuild
  ```

4.6 System Diagnostics Data Gathering with sosreports and Oracle ExaWatcher

You can use the sosreport utility and Oracle ExaWatcher to diagnose problems with your system.

Every time a server is started, system-wide configuration information is collected by the sosreport utility, and stored in the `/var/log/cellos/sosreports` directory. You can generate a new sosreport by running the following command as the `root` user. The script starts collecting the information 30 minutes after entering the command.

```
/opt/oracle.cellos/vldrun -script sosreport
```

In addition, the `/opt/oracle.ExaWatcher` directory contains the Oracle ExaWatcher system data gathering and reporting utilities. Gathered data is stored in archive subdirectories. The following table describes the data gathered at different intervals by the utility:

Table 4-4 Oracle ExaWatcher Collector Names and Descriptions

Collector Name	Description
CellSrvStat	Cell server status.
Diskinfo	I/O statistics of the disk, such as successfully completed reads, merged reads, time spent reading, and so on.
FlashSpace	RAW value of the flash card space. Minimum interval limit is 300 seconds.
IBCardInfo (Currently not available for X8M systems)	RDMA Network Fabric card information, and status of RDMA over InfiniBand Network Fabric ports. Minimum interval is 300 seconds.

Table 4-4 (Cont.) Oracle ExaWatcher Collector Names and Descriptions

Collector Name	Description
IBprocs	Commands that check the RDMA Network Fabric card status. Minimum interval is 600 seconds.
Iostat	CPU statistics, and I/O statistics for devices and partitions.
Lsof	Files opened by current processes. Minimum interval limit is 120 seconds.
MegaRaidFW	MegaRaid firmware information, such as battery information. Minimum interval is 86400 seconds.
Meminfo	Memory management by the kernel.
Mpstat	Microprocessor statistics.
Netstat	Current network connection statistics.
Ps	Active processes statistics.
RDSinfo	Availability of cell servers. Interval limit is 30 seconds.
Slabinfo	Caches for frequently-used objects in the kernel.
Top	Dynamic, real-time view of the system.
Vmstat	Virtual memory status.

To use Oracle ExaWatcher, do the following:

1. As the `root` user, start the Oracle ExaWatcher processes and service.

   ```
   # systemctl start ExaWatcher
   ```

2. Run the Oracle ExaWatcher utility at the `root` user.

   ```
   /opt/oracle.ExaWatcher/ExaWatcher.sh [options]
   ```

The following options are available for use with the Oracle ExaWatcher utility:

Option	Description
No options specified	The utility runs using the default options.

Option	Description
-c|--command '*collector_name* ;; "*default_command*; ... " '	To change the core command to be run on the current group. Only the following core commands can be changed: CellSrvStat Iostat Mpstat Netstat Ps Top Vmstat Example: --command 'Vmstat;; "vmstat -a"'
--createconf "*config_file_to_create*"|null	The utility parses all command line inputs, validates them, and creates a configuration file. If the file path and name is not specified, then the utility overwrites the default configuration file.
-d|--disable "*collector_name*"	The name of the collector to be disabled on the utility. Example: --disable "Vmstat"
-e |--end "*end_time*"	The ending tine for the current group. The default value is 10 years from current time. Example: --end "11/06/2013 12:01:00"
--fromconf "*configuration_file*"|null	The configuration file to use with the Oracle ExaWatcher utility. The default configuration files are as follows: /opt/oracle.ExaWatcher/ExaWatcher.conf for Oracle Linux
-g|--group	Starts a new group for gathering data. Other options can be specified with the group option.
-h|--help	Displays help information.
-i|--interval "*interval_length*"	The sampling interval for the current group, in seconds. The default value is 5 seconds. Certain collection modules cannot be run every second because the modules consume resources. Example: --interval 10
-l|--spacelimit	Sets a limit for the amount of space used by the utility. The limit is specified in MB. The default value is 300 GB. Example: --spacelimit 600
--lastconf	The most-recent configuration file used with the utility. Data is not collected when using this option.

Option	Description
--listcmd "Full"|"Nameonly"|"Core"|"CMD"|"Enabled"|null	The information about the command inputs. The following are the options: Full displays all the information about the commands and samplers. Nameonly displays all names and if it is enabled. Core displays only the core sampler information. CMD displays the name, if it is enabled, and the default commands.
-m| --commandmode {"ALL" | "CORE" | "SELECTED"}	The type of collection modules to run for the current group. The following are the options: ALL runs all collection modules. CORE runs only the core collection modules. SELECTED runs only the specified collection modules. The default value is ALL. Example: --commandmode "CORE"
-o| --count "*archiving_count*"	The archive count of the current group. The default value is 720. Example: --count 500
-r|--resultdir "*result_directory*"	The directory path to store the results of the data collection. Example: --r "/opt/oracle.ExaWatcher/archive"
--stop	To stop the utility and all its processes, and then to zip the data files.
-t|--start "*start_time*"	The starting time for the current group. The default is 20 seconds from the current time. Example: --start "11/05/2013 12:00:00"
-u|--customcmd '*sample_name* ;; "*custom_command*;... " '	To include a custom collection module in the current group. Example: --customcmd 'Lsl; "/bin/ls -l"'
-z|--zip *"bzip2" "gzip"*	The compression program to use on the collected data. The default program is bzip2. Example: --zip "gzip"

4.7 Serial Console Support

The storage servers and database servers of Oracle Exadata Database Machine are configured to provide serial console access.

The serial console is useful when collecting Oracle Linux kernel traces or creating crash dump files to help diagnose severe malfunctions.

To access the serial console, perform the following procedure:

1. Connect to the Integrated Lights Out Manager (ILOM) using SSH and log in as an ILOM administrator.
2. Run the `start /SP/console` command

   To stop using the console, use the `stop /SP/console` command.

4.8 Oracle Linux Kernel Crash Core Files

The storage servers and database servers of Oracle Exadata Database Machine are configured to generate Oracle Linux kernel crash core files in the `/var/crash` directory, when the Oracle Linux operating system malfunctions or crashes.

The `crash` utility can be used to analyze the crash files. The crash files are automatically removed by the ExaWatcher utility so that the files do not occupy more than 10 percent of the free disk space on the file system. Older crash files are removed first.

5 Managing I/O Resources

I/O Resource Management (IORM) is a tool for managing how multiple databases and the workloads within the databases share the I/O resources of Oracle Exadata System Software.

To manage workloads within a database, the Oracle Database Resource Manager has been enhanced to work with IORM to provide database resource management.

- Overview of Resource Management
  You can use I/O Resource Management (IORM) to manage storage needs for multiple databases.
- Understanding I/O Resource Management (IORM)
  IORM manages the storage server I/O resources on a per-cell basis. Whenever the I/O requests start to saturate a cell's capacity, IORM schedules incoming I/O requests according to the configured resource plans.
- About Consumer Groups and Resource Plans
  Oracle Exadata Database Machine provides out-of-the-box consumer groups and resource plans specifically designed for data warehouses that use Oracle Exadata System Software.
- About CDB Plans and Pluggable Databases
  The container database (CDB) can have multiple workloads within multiple PDBs competing for resources.
- Administering IORM
  You can perform various administrative tasks related to I/O Resource Management (IORM).

Related Topics

- Managing Resources with Oracle Database Resource Manager
- Master Note for Oracle Database Resource Manager (My Oracle Support Doc ID 1339769.1)
- Tool for Gathering I/O Resource Manager Metrics: metric_iorm.pl (My Oracle Support Doc ID 1337265.1)

5.1 Overview of Resource Management

You can use I/O Resource Management (IORM) to manage storage needs for multiple databases.

Storage is often shared by multiple types of workloads and databases. Shared storage has several benefits over dedicated storage. One benefit is that sharing lowers administration costs by reducing the number of storage systems that need to be maintained. Another benefit is that sharing leads to much more efficient use of storage, both from a space and a bandwidth standpoint. When a storage system is dedicated to a single database, the administrator must size the storage system based on the database's peak anticipated load and size. This practice leads to unused I/O bandwidth and space for some databases and, if the estimate was not large enough, then there is insufficient bandwidth and space for other databases. The correct balance of storage resources across the multiple databases is seldom achieved because real world workloads are very dynamic.

On the other hand, running multiple types of workloads and databases on shared storage often leads to performance problems. For example, large parallel queries on one production data warehouse can impact the performance of critical queries on another production data warehouse. Also, a data load on a data warehouse can impact the performance of critical queries also running on it. You can mitigate these problems by over-provisioning the storage system, but this diminishes the cost-savings benefit of shared storage. You can also schedule non-critical tasks at off-peak hours, but this manual process is laborious. It becomes impossible when databases have different administrators who do not coordinate their activities.

IORM allows workloads and databases to share Oracle Exadata Storage Servers according to user-defined policies. To manage workloads within a database, you can define database resource plans, using Oracle Database Resource Manager which has been enhanced to manage Oracle Exadata Storage Server I/O resources. Define a container database (CDB) resource plan that allows management for the various pluggable databases (PDBs). To manage multiple databases, you can define interdatabase plans.

Flash IORM protects the latency of critical OLTP I/O requests in flash cache. When table scans are running on flash concurrently with OLTP I/O requests, the OLTP latency is impacted significantly. Flash IORM queues and throttles the table scan, and other low priority I/O requests. The critical OLTP I/O requests are never queued. When the flash disks are not busy serving critical OLTP I/O requests, the queued I/O requests are issued based on the resource allocations in the interdatabase plan.

5.2 Understanding I/O Resource Management (IORM)

IORM manages the storage server I/O resources on a per-cell basis. Whenever the I/O requests start to saturate a cell's capacity, IORM schedules incoming I/O requests according to the configured resource plans.

- About I/O Resource Management (IORM) in Exadata Database Machine
  IORM provides many features for managing resource allocations. Each feature can be used independently or in conjunction with other features.
- About Database Resource Management
  A database often has many types of workloads. These workloads may differ in their performance requirements and the amount of I/O that they issue.
- About Interdatabase Resource Management
  Inter-database resource management is managed with an interdatabase plan.
- About I/O Resource Management Profiles
  I/O Resource Management (IORM) interdatabase plans support profiles to ease management, and configuration of interdatabase plans for hundreds of databases.
- About Category Resource Management
  Categories represent collections of consumer groups across all databases.

5.2.1 About I/O Resource Management (IORM) in Exadata Database Machine

IORM provides many features for managing resource allocations. Each feature can be used independently or in conjunction with other features.

IORM schedules I/O by immediately issuing some I/O requests and queuing others. The I/O requests that are immediately issued are from workloads that have not exceeded their resource allocation, according to the resource plans. The I/O requests that are queued are from workloads that have exceeded their resource allocation. They are issued when their workload no longer exceeds its resource allocation or when the cell is operating below capacity. When the cell is operating below capacity, IORM does not queue I/O requests, and lets a workload exceed its resource allocation as the system is not maxed out.

For example, if a production database and test database are sharing Oracle Exadata Storage Server resources, you can configure resource plans that give priority to the production database. In this case, whenever the test database load would affect the production database performance, IORM schedules the I/O requests such that the production database I/O performance is not impacted. This means that the test database I/O requests are queued until they can be issued without disturbing the production database I/O performance.

Database resource management enables you to manage workloads within a database. Database resource management is configured at the database level, using Oracle Database Resource Manager to create database resource plans. You should use this feature if you have multiple types of workloads within a database. You can define a policy for specifying how these workloads share the database resource allocations. If only one database is using the Oracle Exadata Storage Server resources, then this is the only IORM feature that you need.

Interdatabase resource management enables you to manage resources across multiple databases. Interdatabase resource management is configured by using the CellCLI utility to create interdatabase plans. The interdatabase plans specify a resource allocation for each database. You should use this feature if you have multiple databases using Oracle Exadata Storage Server resources.

If an interdatabase plan has been configured, then each database can have a database plan and a resource allocation. A database resource plan specifies how the database resource allocation should be divided across its workloads, or consumer groups. If a database does not have a database resource plan enabled, then the database resource allocation is not divided, and all I/O requests from the database are treated as a single workload. If you have databases with the same `DB_UNIQUE_NAME` but associated with different Oracle Automatic Storage Management (Oracle ASM) clusters, then, starting with Oracle Exadata System Software release 19.1.0, you can use the `asmcluster` attribute to uniquely identify each database in the interdatabase plan.

Category resource management is an advanced feature. It is useful when Oracle Exadata Storage Server is supporting multiple databases, and you want to allocate resources primarily by the category of the work being done. For example, suppose all databases have three categories of workloads: OLTP, reports, and maintenance. To allocate the I/O resources based on these workload categories you would use category resource management.

The concept of maximum utilization limit (`limit`) is supported by IORM. In addition to specifying the resource allocation values, you can also provide a maximum utilization limit for a given database. This directive ensures that the database never utilizes I/O resources beyond the specified limits. For example, if a production and test database are sharing Oracle Exadata Storage Server resources, then set a maximum utilization limit for the test database to limit the I/O utilization for that database.

If a maximum utilization limit is specified, then excess capacity is never used by the databases. It is possible that the storage devices are running below full capacity when maximum utilization limits are specified.

I/Os issued to hard disks are not subjected to the maximum utilization limit, but I/Os on flash devices are. This is because almost all of the I/Os are served by Exadata Smart Flash Cache. By enforcing the maximum utilization limit only on the flash devices, the system avoids high latencies and outliers due to low utilization limits on the I/Os that miss the flash cache, such as cache misses or population I/Os.

IORM supports flash cache and flash log management. The `ALTER IORMPLAN flashcache` attribute can be set to `off` to prevent databases from using the flash cache. Similarly, the `ALTER IORMPLAN flashlog` attribute can be set to `off` to prevent databases from using the flash log. If you have databases with the same `DB_UNIQUE_NAME` but associated with different Oracle ASM clusters, then, starting with Oracle Exadata System Software release 19.1.0, you can use the `asmcluster` attribute to uniquely identify each database. These attributes allow flash cache and flash log to be reserved for mission-critical databases, especially in consolidated environments.

In addition to simply turning off flash cache for certain databases, IORM supports specifying a quota for each flash cache group. A flash cache group can either be a database or a pluggable database (PDB). These quotas support reservation of space in flash cache for certain critical groups and also prevent less important or rogue databases or PDBs from consuming the entire flash cache. The minimum and maximum quotas can be specified using either the interdatabase plan or container database (CDB) resource plan.

You can specify a *soft* maximum that enables a group to grow beyond its allocation if the flash cache is not full. You also have the option of partitioning the flash cache for Oracle Cloud and *pay for performance* deployments where the database or PDB cannot exceed its allocation even when the flash cache is not full.

IORM interdatabase plans support profiles to ease management and configuration of interdatabase plans for hundreds of databases. Currently, the storage administrator has to specify resources for every database in the interdatabase plan. The plan also needs to be updated each time a new database is created. IORM profiles help alleviate this problem. The first step is to create profile directives that let you define different profile types based on performance requirements. The second step is to map new and existing databases to one of the profiles defined in the interdatabase plan, and the database inherits all its attributes from the profile directive automatically.

5.2.2 About Database Resource Management

A database often has many types of workloads. These workloads may differ in their performance requirements and the amount of I/O that they issue.

For each database, you can use the Oracle Database Resource Manager for the following tasks:

- **Create resource consumer groups**

Resource consumer groups provide a way to group sessions that comprise a particular workload. For example, if your database is running four different applications, then you can create four consumer groups, one for each application. If your data warehouse has three types of workloads, such as critical queries, normal queries, and ETL (extraction, transformation, and loading), then you can create a consumer group for each type of workload.

- **Map user sessions to consumer groups**

  Once you have created the consumer groups, you must create rules that specify how sessions are mapped to consumer groups. Oracle Database Resource Manager allows you to create mapping rules based on session attributes such as the Oracle user name, the service that the session used to connect to the database, client machine, client program name, client user name, and so on. If you are creating consumer groups for each application and each application has a dedicated service, then you should create mapping rules based on service names. If you want to dedicate a consumer group to a particular set of users, then you should create mapping rules based on their user names. Sessions that are not explicitly assigned to a consumer group are placed in the `OTHER_GROUPS` consumer group.

- **Create CDB resource plans**

  A container database (CDB) resource plan specifies how CPU and I/O resources are allocated among the different pluggable databases (PDBs) that are part of the same container. The CDB plan is created using Oracle Database Resource Manager. The CDB plan contains a directive for each PDB. The directive defines the number of shares that are allocated to that PDB. The shares define the relative priority of that PDB as compared to other PDBs in the plan.

  A maximum utilization limit can be specified for a PDB.

  A CDB resource plan also lets you specify `memory_min` and `memory_limit` for each PDB. These parameters specify the flash cache minimum and maximum quotas for each PDB.

- **Create resource plans**

  The database resource plan, also known as an **intradatabase resource plan**, specifies how CPU and I/O resources are allocated among consumer groups in its database. The resource plan is created using Oracle Database Resource Manager. It contains a resource allocation directive for each consumer group, which consists of a percentage and a level. You can specify up to eight levels.

  – Consumer groups at level 2 get resources that were not allocated at level 1 or were not consumed by a consumer group at level 1.

  – Consumer groups at level 3 are allocated resources only when some allocation remains from levels 1 and 2.

  – The same rules apply to levels 4 through 8.

  Multiple levels not only provide a way of prioritizing, they also provide a way of explicitly specifying how all primary and leftover resources are to be used. You can construct resource plans that allocate resources across consumer groups using percentages, priorities, or a combination of the two.

  You can also specify a maximum utilization limit for a consumer group. This works in the same way as a maximum utilization limit for a database, and limits the I/O utilization for the consumer group to the specified value.

In addition to a CDB plan, each PDB can also create a resource plan to manage the workloads running within the PDB. PDBs only support single level plans with a maximum of 8 consumer groups.

- **Enable a resource plan**

  A database resource plan can be manually enabled with the `RESOURCE_MANAGER_PLAN` initialization parameter or automatically enabled with the job scheduler window.

When you set a database resource plan on the database, a description of the plan is automatically sent to each cell. For Oracle Real Application Clusters (Oracle RAC) database running on Oracle Exadata Database Machine, all instances in the Oracle RAC cluster must be set to the same resource plan. When a new cell is added or an existing cell is restarted, the current plan of the database is automatically sent to the cell. The resource plan is used to manage resources on both the database server and storage servers (cells).

Background I/Os are scheduled based on their priority relative to the user I/Os. For example, redo writes, and control file reads and writes are critical to performance and are always prioritized above all user I/Os. The database writer process (DBWR) writes are scheduled at the same priority level as the user I/Os. If a resource plan is not enabled for a database, then all user I/Os are treated equally, and background I/Os are treated as described in this paragraph.

Oracle provides several predefined plans. The most commonly used are `mixed_workload_plan`, `dss_plan`, and `default_maintenance_plan`.

> **See Also:**
>
> Managing Resources with Oracle Database Resource Manager in *Oracle Database Administrator's Guide* for additional information about PDBs, CDB plans, consumer groups, assigning user sessions to consumer groups, and managing resource plans

5.2.3 About Interdatabase Resource Management

Inter-database resource management is managed with an interdatabase plan.

An interdatabase plan specifies how resources are allocated by percentage or share among multiple databases for each cell. The directives in an interdatabase plan specify allocations to databases, rather than consumer groups. The interdatabase plan is configured and enabled with the CellCLI utility at each cell.

- About Interdatabase IORM Plan Directives
  Interdatabase plan directives are specified using the `db_unique_name` of the database as the identifier.
- Using Interdatabase Plans for Consolidation and Isolation
  Interdatabase resource management plans help manage resources when consolidating or isolating databases.
- About Flash Cache Management in Interdatabase Plans
  I/O Resource Management can provide predictable performance by guaranteeing space in flash cache.

- About PMEM Cache Management in Interdatabase Plans
  I/O Resource Management can provide predictable performance by guaranteeing space in PMEM cache.
- Tips for Managing Resource Plans
  Note the following information when creating and managing resource plans.

Related Topics

- Administering Interdatabase Resource Management
  You can configure an interdatabase or category plan for Oracle Exadata Database Machine using the CellCLI ALTER IORMPLAN command.
- ALTER IORMPLAN

5.2.3.1 About Interdatabase IORM Plan Directives

Interdatabase plan directives are specified using the db_unique_name of the database as the identifier.

Note:

For Oracle Data Guard deployments only, I/O Resource Management (IORM) does not support specifying directives using db_name unless the role attribute is also specified in the directive.

For example, you might specify that 70 percent of the I/O resources are allocated to database SALES, 30 percent to HR, and any of their unused allocations are reallocated to the TEST_SALES database. The interdatabase plan is similar to a database resource plan, in that each directive consists of an allocation amount and a level from 1 to 8. For a given plan, the total allocations at any level must be less than or equal to 100 percent. An interdatabase plan differs from a database resource plan in that it cannot contain subplans and only contains I/O resource directives. Only one interdatabase plan can be active on a cell at any given time.

Share-based plans use a relative share instead of percentage allocations and levels. These plans are simpler to implement, but as effective as percentage allocations. Each database is given a share value which is an integer between 1 and 32. The sum of the shares can be greater than 100. Share-based plans support up to 1024 directives within the interdatabase plan. For example, if a critical database, FINANCE, has 4 shares, and a low-priority database, REPORTING, has 1 share, then the FINANCE database is four times more likely to issue I/Os compared to the REPORTING database.

Oracle Exadata System Software uses the IORM and database resource plans together to allocate I/O resources.

- First, the interdatabase plan allocates the I/O resources to individual databases. Any unused resources are reallocated to other databases, as specified by the plan. This is similar to database resource plans.
- Next, the database resource plan for each database allocates the I/O resources to consumer groups. If a database does not have an active database resource plan, all user I/Os are treated the same. Background I/Os are automatically prioritized relative to the user I/Os based on their importance.

As a best practice, you should create a directive for each database that is using the same cell storage. This is done automatically for shared-based plans, but not for percentage allocation plans. Each database that is not explicitly mapped in a share-based plan gets the default share of 1. To ensure that any database without an explicit directive can be managed with percentage allocation plans, create an allocation named `OTHER`. Databases without explicit directives are managed using the allocation of the `OTHER` group directive.

Share-based plans use the `DEFAULT` directive to specify the default values for the different IORM attributes for databases not explicitly named in the plan. The default share is 1, and the limit is set to 100%. Exadata Smart Flash Cache and Exadata Smart Flash Logging are enabled by default and there are no minimum or maximum flash cache quotas setup by default.

5.2.3.2 Using Interdatabase Plans for Consolidation and Isolation

Interdatabase resource management plans help manage resources when consolidating or isolating databases.

Consider the consolidation of four different applications on the same Oracle Exadata Storage Servers. If all the applications have similar priority requirements, you can use IORM to equally allocate 25% of the I/O resources. If one application is more important then the others, then you can use IORM to give it a larger percentage of the I/O resources.

If you have databases with the same `DB_UNIQUE_NAME` but associated with different Oracle ASM clusters, then, starting with Oracle Exadata System Software release 19.1.0, you can use the `asmcluster` attribute to uniquely identify each database in the interdatabase plan.

Maximum utilization limit is also useful in such consolidation scenarios.You can isolate each application in case one application sees a sudden increase in its workload. Use the maximum utilization limit to isolate the applications, such as specifying a maximum utilization limit of 40% for each application. In such a scenario, each application can utilize a maximum of 40% of the I/O resources, and does not completely monopolize the system.

Another situation where maximum utilization limits make sense is in ensuring a pay-for-performance model. Service providers want to guarantee performance to their customers corresponding to the level of service being purchased. A customer that is not purchasing the gold tier service should not receive gold level performance. You can use IORM resource plans to distribute the I/O resources among customer applications based on their performance model. Flash cache minimum and maximum quotas are also useful in this model.

5.2.3.3 About Flash Cache Management in Interdatabase Plans

I/O Resource Management can provide predictable performance by guaranteeing space in flash cache.

With multiple databases, and pluggable databases (PDBs) sharing the storage, capacity in flash cache becomes a critical resource that needs to be managed. IORM can prevent one database or PDB from taking over the entire flash cache. This is achieved using the `flashcachemin`, `flashcachelimit` and `flashcachesize` attributes.

- `flashcachemin` — specifies the minimum size in flash cache that is guaranteed for a database even if the blocks are cold

- `flashcachelimit` — specifies a *soft* maximum size in flash cache; if the flash cache is not full, a database can exceed the `flashcachelimit` value.
- `flashcachesize` — specifies a **hard** maximum size in flash cache; a database cannot exceed the `flashcachesize` value at any time. Starting with Oracle Exadata System Software release 19.2, `flashcachesize` is not a guaranteed reservation if the sum of the `flashcachesize` across all directives is more than the size of the flash cache. Otherwise, the flash cache is partitioned.

A flash cache group with `flashcachesize` specified cannot exceed its quota even if flash cache is not full. However, if the flash cache usage for the database was already over the `flashcachesize` value prior to setting the resource plan, Oracle Exadata System Software does not proactively evict the data from flash cache. The data is lazily evicted when there is space pressure in flash cache.

Because `flashcachemin` is a guaranteed reservation, the sum of `flashcachemin` across all directives should be less than the size of the flash cache to ensure that each group gets its respective quota.

5.2.3.4 About PMEM Cache Management in Interdatabase Plans

I/O Resource Management can provide predictable performance by guaranteeing space in PMEM cache.

With multiple databases, and pluggable databases (PDBs) sharing the storage, capacity in PMEM cache becomes a critical resource that needs to be managed. IORM can prevent one database or PDB from taking over the entire PMEM cache. This is achieved using the `pmemcachemin`, `pmemcachelimit` and `pmemcachesize` attributes, available in Oracle Exadata Storage Server X8M and later.

A PMEM cache group with `pmemcachesize` specified cannot exceed its quota even if PMEM cache is not full. However, if the PMEM cache usage for the database was already over the `pmemcachesize` value prior to setting the resource plan, Oracle Exadata System Software does not proactively evict the data from PMEM cache. The data is lazily evicted when there is space pressure in PMEM cache.

- `pmemcachemin` — specifies the minimum size in PMEM cache that is guaranteed for a database even if the blocks are cold.
- `pmemcachelimit` — specifies a soft maximum size in PMEM cache; if the PMEM cache is not full, a database can exceed the `pmemcachelimit` value.
- `pmemcachesize` — specifies a hard maximum size in PMEM cache; a database cannot exceed the `pmemcachesize` value at any time. `pmemcachesize` is not a guaranteed reservation if the sum of the `pmemcachesize` across all directives is more than the size of the PMEM cache. Otherwise, the PMEM cache is partitioned.

Because `pmemcachemin` is a guaranteed reservation, the sum of `pmemcachemin` across all directives should be less than the size of the PMEM cache to ensure that each group gets its respective quota.

5.2.3.5 Tips for Managing Resource Plans

Note the following information when creating and managing resource plans.

- If Oracle Exadata Database Machine is only hosting one database, then an interdatabase plan is not needed.

- If an interdatabase plan is not specified, then all databases receive an equal allocation.
- If only one database is mapped to the OTHER directive and all other databases have an explicit directive, then Oracle Exadata System Software uses the database resource plan of that database to determine how the allocation of the OTHER database is redistributed among the consumer groups in that database.
- If multiple databases are mapped to the OTHER directive, then Oracle Exadata System Software does not use Oracle Database Resource Manager for these databases. All of the I/O requests are treated the same.
- For share-based plans, each database gets its own directive even when it is not explicitly named in the plan. Oracle Exadata System Software uses the database resource plan of the database to determine how the allocation is to be distributed between the consumer groups in the database.
- If you have databases with the same DB_UNIQUE_NAME but associated with different Oracle ASM clusters, then, starting with Oracle Exadata System Software release 19.1.0, you can use the asmcluster attribute to uniquely identify each database when specifying directives.
- If the container database (CDB) plan specifies memory_min or memory_limit, and the interdatabase plan specifies **only** flashcachesize, then memory_min in the CDB plan is ignored.
- If the container database (CDB) plan specifies memory_min or memory_limit, and the interdatabase plan specifies **only** pmemcachesize, then memory_min in the CDB plan is ignored.

5.2.4 About I/O Resource Management Profiles

I/O Resource Management (IORM) interdatabase plans support profiles to ease management, and configuration of interdatabase plans for hundreds of databases.

Profiles introduce a way to allocate I/O resources for a database. This is done using the database initialization parameter db_performance_profile. Database administrators can classify different databases as, GOLD, SILVER, BRONZE, by setting the db_performance_profile parameter. As with Oracle Database Resource Manager plans, the db_performance_profile information is automatically pushed to all the storage servers (cells). The following SQL command displays how the profile parameter can be set for a database:

```
SQL> ALTER SYSTEM SET db_performance_profile=gold SCOPE=spfile;
```

Profiles are specified as directives for the interdatabase plan, and are configured using the CellCLI utility. A profile directive consists of an identifier (name), and a set of attributes. To differentiate between a database directive and a profile directive, a qualifier attribute called type is used. The type attribute can be set to either database or profile. The following is an example of the type attribute syntax:

```
CellCLI> ALTER IORMPLAN DBPLAN=((name=gold, share=10, limit=100,
type=profile),  \
(name=silver, share=5, limit=60, type=profile), (name=bronze, share=1,
limit=20, \
 type=profile))
```

The preceding example contains three directives for profiles, GOLD, SILVER and BRONZE. All databases with db_performance_profile set to GOLD would automatically get 10 shares, and 100% limit on the cells. Likewise, in the example above, databases with the SILVER profile would get 5 shares and 60% limit, and databases with the BRONZE profile would get 1 share and 20% limit.

When adding a new database, you can set the db_performance_profile parameter and restart the database. The database automatically inherits the profile attributes without having to modify the interdatabase plan. It is also supported to create interdatabase plans with a mix of profile directives, and database directives.

Note the following when managing interdatabase profile plans:

- The db_performance_profile parameter is not a dynamic parameter, so profile updates require a database restart.
- If the type attribute is not specified, then the directive defaults to the database directive.
- An interdatabase plan can specify only 8 profile directives, and 1024 database directives.
- Level, allocation, and role cannot be specified with a profile directive.
- The words OTHER, and DEFAULT are reserved words. A profile name cannot be OTHER or DEFAULT.
- The type attribute cannot be specified with category plans.
- Profiles cannot be specified in conjunction with category plans.
- If multiple databases are mapped to the OTHER directive, then Oracle Exadata Storage Server does not use Oracle Database Resource Manager for these databases. All of the I/O requests are treated the same.

Related Topics

- ALTER IORMPLAN

5.2.5 About Category Resource Management

Categories represent collections of consumer groups across all databases.

Oracle Database Resource Manager enables you to specify a category for every consumer group. You can manage I/O resources based on categories by creating a category plan. For example, you can specify precedence to consumer groups in the interactive category over consumer groups in the batch category for all databases sharing Oracle Exadata Storage Server.

You can add any number of categories, or modify the predefined categories. You should map consumer groups to the appropriate category for all databases that use the same cell storage. Any consumer group without an explicitly specified category defaults to the OTHER category.

When a category plan is enabled, the category plan is first used to allocate resources among the categories. For each category that is selected, the interdatabase plan is used to select those databases that have consumer groups with the selected category. Finally, the database resource plan of the selected database is used to select one of its consumer groups.

Category plans are configured and enabled using the CellCLI utility on the cell. Only one category plan can be enabled at a time. The predefined categories provided in Oracle Database are described in the following table, along with sample percentages.

Table 5-1 Sample Category Resource Management Plan

Category Name	Category Description	Level 1 (%)	Level 2 (%)	Level 3 (%)
ADMINISTRATIVE	For extremely high-priority work, such as urgent administrative tasks. This category is required.	80	not set	not set
INTERACTIVE	For high-priority, performance-sensitive work, such as OLTP transactions.	not set	70	not set
BATCH	For low-priority work, such as noncritical reports and backup.	not set	not set	70
MAINTENANCE	For low-priority work, such as automated tasks.	not set	not set	10
OTHER	For all consumer groups that do not have a category label or reference a category that is not in the current category plan. This category is required.	not set	not set	20

The sample plan shown in the above table prioritizes administrative activity across all databases. It also prioritizes interactive activity over batch, maintenance, and other activities. In the sample plan, the following are the resource allocations:

- Level 1 is given 80 percent of the I/O resources. The ADMINISTRATIVE category is the only category in level 1.
- Level 2 is given all resources that were unallocated or unused by level 1. In this example, level 2 is given 20 percent of the I/O resources and any resources unused by the ADMINISTRATIVE category. The INTERACTIVE category gets 70 percent of the level 2 amount.
- Level 3 categories are given the remaining resources, including those not used by the INTERACTIVE category. Of the remaining resources, the BATCH category gets 70 percent, the OTHER category gets 20 percent, and the MAINTENANCE category gets 10 percent.

All administrative consumer groups in all databases should be mapped to the ADMINISTRATIVE category. All high-priority user activity, such as consumer groups for important OLTP transactions and time-critical reports, should be mapped to the INTERACTIVE category. All low-priority user activity, such as reports, maintenance, and low-priority OLTP transactions, should be mapped to the BATCH, MAINTENANCE, and OTHER categories.

Related Topics

- Administering Database Resource Management
  To set up database resource management, you must use Oracle Database Resource Manager to configure the consumer groups, assign sessions to consumer groups, create a database resource plan, and enable the plan.

5.3 About Consumer Groups and Resource Plans

Oracle Exadata Database Machine provides out-of-the-box consumer groups and resource plans specifically designed for data warehouses that use Oracle Exadata System Software.

These resource plans can be modified to suit the needs of your environment.

The following consumer groups are for data warehouses:

- ETL_GROUP: Consumer group for ETL (extract, transform, and load) jobs.
- DSS_GROUP: Consumer group for non-critical decision support system (DSS) queries.
- DSS_CRITICAL_GROUP: Consumer group for critical DSS queries.

The following resource plans are for data warehouses:

- DSS_PLAN Resource Plan
  The DSS_PLAN resource plan is designed for data warehouses that prioritize critical DSS queries over non-critical DSS queries and ETL jobs.
- ETL_CRITCAL_PLAN Resource Plan
  The ETL_CRITICAL_PLAN prioritizes ETL over DSS queries.

5.3.1 DSS_PLAN Resource Plan

The DSS_PLAN resource plan is designed for data warehouses that prioritize critical DSS queries over non-critical DSS queries and ETL jobs.

In this plan, SYS_GROUP has the highest priority, followed by DSS_CRITICAL_GROUP, DSS_GROUP, and then a combination of ETL_GROUP and BATCH_GROUP. No consumer group is allowed to consume all the bandwidth.

Table 5-2 DSS_PLAN Resource Plan for Data Warehouses

Consumer Group	Level 1 (%)	Level 2 (%)	Level 3 (%)	Level 4 (%)
SYS_GROUP	75	not set	not set	not set
DSS_CRITICAL_GROUP	not set	75	not set	not set
DSS_GROUP	not set	not set	75	not set
ETL_GROUP	not set	not set	not set	45
BATCH_GROUP	not set	not set	not set	45
ORA$DIAGNOSTICS	not set	5	not set	not set
ORA$AUTOTASK_SUB_PLAN	not set	5	not set	not set
OTHER_GROUPS	not set	not set	not set	10

As shown in the previous table, the DSS_CRITICAL_GROUP group is only allocated 75 percent at level 2. Any unused allocation goes to the next level, not to other consumer groups at the same level. That means that if the DSS_CRITICAL_GROUP group does not completely consume its allocation, then the allocation is not given to the ORA$DIAGNOSTICS or ORA$AUTOTASK_SUBPLAN groups at the same level, but instead the allocation is given to the DSS_GROUP group at level 3.

5.3.2 ETL_CRITCAL_PLAN Resource Plan

The ETL_CRITICAL_PLAN prioritizes ETL over DSS queries.

In this plan, the SYS_GROUP group is given 75 percent of the bandwidth. The remaining bandwidth is divided between the other consumer groups in the ratios specified by the level 2 allocations. The ETL_GROUP and DSS_CRITICAL_GROUP groups have a higher allocation (35 percent) than the DSS_GROUP and BATCH_GROUP groups (10 percent).

Table 5-3 ETL_CRITICAL_PLAN Resource Plan for Data Warehouses

Consumer Group	Level 1 (%)	Level 2 (%)	Level 3 (%)	Level 4 (%)
SYS_GROUP	75	not set	not set	not set
DSS_CRITICAL_GROUP	not set	35	not set	not set
DSS_GROUP	not set	10	not set	not set
ETL_GROUP	not set	35	not set	not set
BATCH_GROUP	not set	10	not set	not set
ORA$DIAGNOSTICS	not set	3	not set	not set
ORA$AUTOTASK_SUB_PLAN	not set	3	not set	not set
OTHER_GROUPS	not set	3	not set	not set

5.4 About CDB Plans and Pluggable Databases

The container database (CDB) can have multiple workloads within multiple PDBs competing for resources.

The Oracle Multitenant container database (CDB) supports many user-defined pluggable databases (PDBs). In a CDB, resources are managed at the following levels:

- **CDB level**:Oracle Database Resource Manager manages the workloads for multiple PDBs that are contending for system and CDB resources. The administrator can specify how resources are allocated to PDBs, and can limit the resource utilization of specific PDBs.
- **PDB level**: Oracle Database Resource Manager manages the workloads within each PDB.

The following CDB plan contains three PDBs named SALES, SERVICES and HR. The PDBs have different shares and max utilization limits in the CDB plan.

PDB Name	Directive for Shares	Directive for Utilization Limit	Memory_min	Memory_limit
SALES	3	Unlimited	20	not set
SERVICES	3	Unlimited	20	not set
HR	1	70	not set	50

5.5 Administering IORM

You can perform various administrative tasks related to I/O Resource Management (IORM).

To perform the tasks, use the `DBMS_RESOURCE_MANAGER` package to define database resource plans on the database servers, and the CellCLI utility to specify the IORM and category plans for each cell.

- Enabling IORM to Prioritize Latency
  I/O Resource Management (IORM) is enabled by default and in its default state IORM manages flash cache and flash log.
- Administering Database Resource Management
  To set up database resource management, you must use Oracle Database Resource Manager to configure the consumer groups, assign sessions to consumer groups, create a database resource plan, and enable the plan.
- Administering Interdatabase Resource Management
  You can configure an interdatabase or category plan for Oracle Exadata Database Machine using the CellCLI `ALTER IORMPLAN` command.
- Flash Cache Attributes for IORM Plans
  The `flashcachemin` and `flashcachesize` attributes can be used to specify the guaranteed minimum and maximum quotas for flash cache groups.
- Managing Flash Cache Quotas for Databases and PDBs
  IORM enables you to control how you want the flash cache to be shared among different databases and pluggable databases (PDBs).
- PMEM Cache Attributes for IORM Plans
  The `pmemcachemin` and `pmemcachesize` attributes can be used to specify the guaranteed minimum and maximum quotas for PMEM cache groups.
- Managing PMEM Cache Quotas for Databases and PDBs
  I/O Resource Management (IORM) enables you to control how you want the PMEM cache to be shared among different databases and pluggable databases (PDBs).
- Managing Flash Cache and Flash Log
  You can use IORM to manage flash cache and flash log instead of I/O resources for the databases.
- Listing an I/O Resource Management Plan
  You can view the current interdatabase plan for a storage server using the CellCLI `LIST IORMPLAN` command on the storage server.
- Verifying the Configuration of I/O Resource Management
  Use this checklist to verify that I/O Resource Management (IORM) is configured correctly.
- Resetting Default Values in an Interdatabase Plan
  To reset an interdatabase plan attribute to the default value, set the attribute to an empty string.

Related Topics

- Using the CellCLI Utility
  You use the Cell Control Command-Line Interface (CellCLI) utility to manage Oracle Exadata System Software.

5.5.1 Enabling IORM to Prioritize Latency

I/O Resource Management (IORM) is enabled by default and in its default state IORM manages flash cache and flash log.

IORM also guards against extremely high latencies for log writes, buffer cache reads, and other critical I/Os. Running very heavy I/O loads on hard disks results in high disk I/O latencies. Without IORM, the only way to lower the disk I/O latency is to reduce the overall I/O throughput by reducing smart scans, and other disk intensive I/O operations. To manage these I/O intensive operations, set the `objective` option to specify the optimization mode for IORM. The default `objective` option is `basic`.

When the `objective` option is set to `basic`, the database resource plan maximum utilization limits are not enforced. Database resource plan allocations are only used to guard against extremely high latencies, and there is no consideration for plan conformance.

Smart scan and low priority I/Os to flash devices are queued by IORM in this mode. The critical I/Os usually do not keep the disks busy all the time. During these idle intervals, IORM uses the interdatabase plan allocations to issue the queued smart scan, and low priority I/Os to flash devices. This is done independently of how I/Os to hard disks are processed.

IORM also prioritizes different scan workloads among themselves when no critical OLTP I/Os are being issued. Flash IORM is enabled on hybrid systems with flash cache and Exadata Extreme Flash storage servers. The flash IORM device queue depth varies based on the hardware and flash device type.

For stricter plan conformance, and enforcement of maximum utilization limits, the `objective` option must be set to something other than `basic`. The supported IORM objectives are `auto`, `low_latency`, `balanced`, and `high_throughput`. The recommended `objective` option is `auto` which allows IORM to continuously monitor the workloads, and select the best mode based on the active workloads currently on the cells. If the `objective` option is set to something other than `basic`, then IORM prioritizes the I/O resources as follows:

1. IORM manages disk I/O if any database has a database resource plan.
2. IORM manages disk I/O if there is a configured interdatabase or category plan.

To deactivate disk I/O prioritization and throttling, set the IORM objective to `basic`.

Flash IORM behavior cannot be controlled by changing the IORM objective except on Exadata Extreme Flash storage servers. Flash IORM is designed to protect the OLTP latency and does not take the objective into account while protecting the critical I/Os.

On Exadata Extreme Flash storage servers, the IORM objective provides some control over the behavior of flash IORM. The default objective is `basic`. The objective values of `auto` and `balanced` have the same behavior. If the degradation in scan throughput is considered to be too high, the objective can be changed to `high_throughput` which will increase the scan throughput at the cost of critical I/O latency. The objective may also be changed to `low_latency` in which case the latency for critical I/Os is very good but there is a significant degradation in scan throughput when both workloads are running concurrently.

5.5.2 Administering Database Resource Management

To set up database resource management, you must use Oracle Database Resource Manager to configure the consumer groups, assign sessions to consumer groups, create a database resource plan, and enable the plan.

- Setting Up Consumer Groups and Categories
  Consumer groups and categories are set up with the procedures in the PL/SQL `DBMS_RESOURCE_MANAGER` package.
- Assigning Sessions to Consumer Groups
  You can assign a session to a resource consumer group manually or automatically using consumer group mapping rules.
- Creating a CDB Plan
  The CDB plan manages CPU resources on the database servers, and flash cache space and I/O bandwidth on the Exadata storage servers.
- Creating a Database Plan
  Database resource plans, also known as **intradatabase plans**, are created using the PL/SQL procedures `DBMS_RESOURCE_MANAGER.CREATE_PLAN()` and `CREATE_PLAN_DIRECTIVE()`.
- Enabling a Database Resource Plan
  You can manually enable database resource plans by setting the `RESOURCE_MANAGER_PLAN` parameter. You can automatically enable resource plans by defining an Oracle Scheduler window with a resource plan.
- Managing Fast File Creation
  Oracle Exadata System Software features fast file creation, allowing accelerated initialization of data files.
- Managing Data Import
  You can control the priority of ETL as well as the amount of I/O resources that ETL consumes using I/O Resource Management (IORM).
- Managing Oracle Recovery Manager Backups and Copies
  You can use I/O Resource Management (IORM) to control the resource consumption and priority of RMAN I/Os.

5.5.2.1 Setting Up Consumer Groups and Categories

Consumer groups and categories are set up with the procedures in the PL/SQL `DBMS_RESOURCE_MANAGER` package.

You can create new consumer groups and categories, or use one of the predefined consumer groups or categories. You do not need to set up categories if you are not planning on using a category plan.

> **Note:**
>
> Consumer groups and categories are created in the database and cannot be created explicitly on a cell.

Before running the DBMS_RESOURCE_MANAGER procedures for administering consumer groups and categories, you must first create a pending area. You must have the system privilege ADMINISTER_RESOURCE_MANAGER to run the procedures in the DBMS_RESOURCE_MANAGER PL/SQL package.

The following PL/SQL commands are used with consumer groups and categories:

- To manage categories: CREATE_CATEGORY(), DELETE_CATEGORY(), and UPDATE_CATEGORY()
- To manage consumers groups: CREATE_CONSUMER_GROUP() and UPDATE_CONSUMER_GROUP()
- To assign consumer groups to categories: CREATE_CONSUMER_GROUP() or UPDATE_CONSUMER_GROUP()

In addition to the consumer groups that you set up, the database contains predefined consumer groups. The DBA_RSRC_CONSUMER_GROUPS view displays information about consumer groups, and the DBA_RSRC_CATEGORIES view displays information about categories in the database.

Example 5-1 Setting Up Consumer Groups and Categories with PL/SQL in the Database

This example shows how to set up consumer groups and categories in a database. The MAINTENANCE category is predefined, and is not created in this example.

```
BEGIN
  DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();

  DBMS_RESOURCE_MANAGER.CREATE_CATEGORY(
     CATEGORY => 'dss',
     COMMENT => 'DSS consumer groups');

  DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP(
     CONSUMER_GROUP => 'critical_dss',
     CATEGORY => 'dss',
     COMMENT => 'performance-critical DSS queries');

  DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP(
     CONSUMER_GROUP => 'normal_dss',
     CATEGORY => 'dss',
     COMMENT => 'non performance-critical DSS queries');

  DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP(
     CONSUMER_GROUP => 'etl',
     CATEGORY => 'maintenance',
     COMMENT => 'data import operations');

  DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
END;
/
```

Example 5-2   Consumer Groups and Categories in an Oracle Database

This example shows a query on the DBA_RSRC_CONSUMER_GROUPS view.

```
SQL> SELECT consumer_group, category FROM DBA_RSRC_CONSUMER_GROUPS where
     consumer_group not like 'ORA%' ORDER BY category;

CONSUMER_GROUP                 CATEGORY
------------------------------ ------------------------------
SYS_GROUP                      ADMINISTRATIVE
ETL_GROUP                      BATCH
BATCH_GROUP                    BATCH
DSS_GROUP                      BATCH
CRITICAL_DSS                   DSS
NORMAL_DSS                     DSS
DSS_CRITICAL_GROUP             INTERACTIVE
INTERACTIVE_GROUP              INTERACTIVE
ETL                            MAINTENANCE
LOW_GROUP                      OTHER
OTHER_GROUPS                   OTHER
AUTO_TASK_CONSUMER_GROUP       OTHER
DEFAULT_CONSUMER_GROUP         OTHER

13 rows selected
```

Related Topics

- *Oracle Database Administrator's Guide*
- *Oracle Database Reference*

5.5.2.2 Assigning Sessions to Consumer Groups

You can assign a session to a resource consumer group manually or automatically using consumer group mapping rules.

For both approaches, you must give explicit permission for a user to switch to a consumer group. In order to control which consumer groups a user can switch to, use the PL/SQL procedure DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUMER_GROUP().

The consumer group mapping rules are based on session attributes such as the user name, the name of the service that the session used to connect to the database, and the name of the client program. To create a consumer group mapping rule, use the SET_CONSUMER_GROUP_MAPPING procedure, as shown in Example 5-3. Before running the SET_CONSUMER_GROUP_MAPPING procedure, you must first create a pending area.

You can also manually switch a session to a particular consumer group, using the PL/SQL DBMS_RESOURCE_MANAGER.SWITCH_CONSUMER_GROUP_FOR_USER() or SWITCH_CONSUMER_GROUP_FOR_SESS() procedures.

Example 5-3   Creating Consumer Group Mapping Rules, Based on Service and User Name

```
BEGIN
DBMS_SERVICE.CREATE_SERVICE('SALES', 'SALES');
DBMS_SERVICE.CREATE_SERVICE('AD_HOC', 'AD_HOC');

DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING
     (DBMS_RESOURCE_MANAGER.ORACLE_USER, 'SYS', 'CRITICAL_DSS');
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING
     (DBMS_RESOURCE_MANAGER.SERVICE_NAME, 'SALES', 'CRITICAL_DSS');
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING
     (DBMS_RESOURCE_MANAGER.SERVICE_NAME, 'AD_HOC', 'NORMAL_DSS');

DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();

DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUMER_GROUP (
   GRANTEE_NAME   => 'PUBLIC',
   CONSUMER_GROUP => 'CRITICAL_DSS',
   GRANT_OPTION   =>  FALSE);
DBMS_RESOURCE_MANAGER_PRIVS.GRANT_SWITCH_CONSUMER_GROUP (
   GRANTEE_NAME   => 'PUBLIC',
   CONSUMER_GROUP => 'NORMAL_DSS',
   GRANT_OPTION   =>  FALSE);
END;
/
```

See Also:

Oracle Database Administrator’s Guide for additional information about the following:

- Assigning Sessions to Resource Consumer Groups
- Creating Consumer Group Mapping Rules

5.5.2.3 Creating a CDB Plan

The CDB plan manages CPU resources on the database servers, and flash cache space and I/O bandwidth on the Exadata storage servers.

CDB plans are created using the PL/SQL procedures `DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN()` and `CREATE_CDB_PLAN_DIRECTIVE()`. The CDB plan can only be configured from the root PDB.

Example 5-4   Using a CDB Plan to Distribute Resources Between PDBs

This example shows how to distribute resources between three PDBs named SALES, SERVICES and HR. SALES and SERVICES have higher priority and get three shares each compared to one share for HR. The limit on the HR PDB is set to 70% maximum utilization limit.

```
BEGIN
DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();

DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN(
                plan    => ''NEWCDB_PLAN ',
                comment => 'CDB resource plan for newcdb');

              DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE(
                plan                  => 'NEWCDB_PLAN',
                pluggable_database    => 'SALESPDB',
                shares                => 3,
                memory_min            => 20,
                utilization_limit     => 100);
              DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE(
                plan                  => ' NEWCDB_PLAN ',
                pluggable_database    => 'SERVICESPDB',
                shares                => 3,
                memory_min            => 20,
                memory_limit          => 75);
              DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE(
                plan                  => ' NEWCDB_PLAN ',
                pluggable_database    => 'HRPDB',
                shares                => 1,
                memory_limit          => 50,
                utilization_limit     => 70);

            DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
            DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
            END;
            /
```

5.5.2.4 Creating a Database Plan

Database resource plans, also known as **intradatabase plans**, are created using the PL/SQL procedures DBMS_RESOURCE_MANAGER.CREATE_PLAN() and CREATE_PLAN_DIRECTIVE().

You must always begin resource plan creations or updates with the PL/SQL procedure CREATE_PENDING_AREA() and complete them with the PL/SQL procedure SUBMIT_PENDING_AREA(). You must also include a directive for OTHER_GROUPS, which includes all sessions that are not explicitly mapped to a consumer group.

You must have the system privilege ADMINISTER_RESOURCE_MANAGER to run the procedures in the DBMS_RESOURCE_MANAGER PL/SQL package. This resource plan manages both CPU resources on database instances and I/O resources on the cells.

Example 5-5   Sharing Resources Across Applications

In this example, assume you have multiple applications sharing a database where the I/O resources should be divided across the applications using a particular ratio. For example, there are three applications named SALES, FINANCE, and MARKETING. You would like the I/O resources to be allocated as 60 percent, 25 percent, and 10 percent, respectively, with the remaining 5 percent allocated to any sessions that do not map into these consumer groups. In this scenario, you would create a consumer group for each application, and then create a single-level resource plan and specify the percentage of I/O resources for each consumer group. This allocation is actually the minimum I/O resources that the consumer group can use. If a consumer group does not use its allocation, then it is redistributed to the other consumer groups in the ratio specified by the plan. You can specify the allocations using the MGMT_P1 parameter.

```
BEGIN
DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
DBMS_RESOURCE_MANAGER.CREATE_PLAN('DAYTIME_PLAN', 'Resource plan for 
managing all
 applications between 9 am and 5 pm');
DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP('SALES', 'Sales App');
DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP('FINANCE', 'Finance App');
DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP('MARKETING', 'Marketing App');
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE('DAYTIME_PLAN', 'SALES', 
'Allocation
for SALES', MGMT_P1 => 60);
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE('DAYTIME_PLAN', 'FINANCE', 
'Allocation
for FINANCE', MGMT_P1 => 25);
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE('DAYTIME_PLAN', 'MARKETING', 
'Allocation for MARKETING', MGMT_P1 => 10);
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE('DAYTIME_PLAN', 'OTHER_GROUPS', 
'Allocation for default group', MGMT_P1 => 5);
DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
END;
/
```

Example 5-6 Sharing Resources Across Workloads

In this example, assume you want to prioritize one workload over another. For example, suppose that you load data into your data warehouse while also servicing queries, and you want to always prioritize the queries over the data load. For this scenario, you would create two consumer groups for queries (reporting and ad-hoc) and one consumer group for data load. You would like to share the I/O resources between the two query consumer groups using a 75/25 ratio. In addition, you would like to issue I/Os for data load only if the query consumer groups do not use all of their allocation. You can use resource plan levels to specify the allocation priorities.

```
BEGIN
DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
DBMS_RESOURCE_MANAGER.CREATE_PLAN('DAYTIME_PLAN', 'Resource plan for 
prioritizing
queries between 9 am and 5 pm');
DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP('REPORT_QUERIES', 'Report 
Queries');
DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP('AD-HOC_QUERIES', 'Ad-Hoc 
Queries');
DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP('DATA_LOAD', 'Data Load');

DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE('DAYTIME_PLAN', 
'REPORT_QUERIES',
'Allocation for REPORT_QUERIES', MGMT_P1 => 75);
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE('DAYTIME_PLAN', 'AD-
HOC_QUERIES',
'Allocation for AD-HOC_QUERIES', MGMT_P1 => 25);
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE('DAYTIME_PLAN', 'DATA_LOAD', 
            'Allocation for DATA_LOAD', MGMT_P2 => 100);
            DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE('DAYTIME_PLAN', 'OTHER_GROUPS',
            'Allocation for default group', MGMT_P3 => 100);
            DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
            END;
            /
```

Related Topics

- *Oracle Database Administrator's Guide*

5.5.2.5 Enabling a Database Resource Plan

You can manually enable database resource plans by setting the RESOURCE_MANAGER_PLAN parameter. You can automatically enable resource plans by defining an Oracle Scheduler window with a resource plan.

When the Oracle Scheduler window opens, the resource plan is enabled. When the Oracle Scheduler window closes, the resource plan is disabled.

When a resource plan is enabled, the database alerts all cells about this event and provides the resource plan. The database also alerts all cells when a resource plan is disabled. Because only one resource plan can be active for any database, you are required to enable the same resource plan on all instances of a database. If no database resource plan is enabled for a database, then all I/O requests are treated equally.

Related Topics

- *Oracle Database Administrator's Guide*

5.5.2.6 Managing Fast File Creation

Oracle Exadata System Software features fast file creation, allowing accelerated initialization of data files.

This feature automatically runs whenever you create a new tablespace, add a data file to an existing tablespace, or autoextend an existing tablespace. Oracle Exadata System Software can initialize files very quickly because it issues many concurrent I/O requests. However, these concurrent I/O requests create a heavy load that can interfere with performance-critical queries.

Using I/O Resource Management (IORM), you can control the priority of fast file creations for creating a new tablespace or adding a data file to an existing tablespace. These operations are run under the FASTFILECRE function. By default, the FASTFILECRE function is mapped to a hidden consumer group that has lower priority than all consumer group and background I/Os. If you choose to increase the priority, and thereby performance, of file creations, add a mapping rule based on the mapping attribute DBMS_RESOUCRE_MANAGER.ORACLE_FUNCTION, and mapping value FASTFILECRE.

Because autoextending an existing tablespace is a brief and often time-critical operation, you cannot modify its priority using IORM.

Example 5-7 Managing Fast File Creation

```
BEGIN
DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP('MAINTENANCE_GROUP',
'Maintenance
activity');
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(DBMS_RESOURCE_MANAGER.ORAC
LE_
FUNCTION, 'FASTFILECRE', 'MAINTENANCE_GROUP');
DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
END;
/
```

5.5.2.7 Managing Data Import

You can control the priority of ETL as well as the amount of I/O resources that ETL consumes using I/O Resource Management (IORM).

Data import, or extract, transform, load (ETL), is an important part of maintaining a data warehouse. In some cases, ETL is extremely critical to performance because reports or queries cannot be run until the data has been loaded. In these cases, ETL should be prioritized above all other queries. In other cases, ETL is a low-priority background activity that only needs to be prioritized in the rare event that it does not complete by a certain time.

To manage ETL, do the following:

- Map the ETL sessions to the ETL_GROUP consumer group.

  The mapping rules for ETL are typically based on user name or client program name. Data pump is run under the DATALOAD function. By default, the DATALOAD function is mapped to the ETL_GROUP consumer group.
- Include the ETL_GROUP group in your resource plans.

To import non-compressed data as compressed data,

Example 5-8   Mapping a Program to the ETL_GROUP Consumer Group

This example shows how to map a program to the ETL_GROUP consumer group.

```
BEGIN
DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING
  (DBMS_RESOURCE_MANAGER.CLIENT_PROGRAM, 'SQLLDR', 'ETL_GROUP');
DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
END;
/
```

Importing Non-compressed Data as Compressed Data

Non-compressed data can be imported as compressed data when using the TRANSFORM:SEGMENT_ATTRIBUTES=n option, and the target tablespace has been configured to create new tables as Exadata Hybrid Columnar Compression tables by default.

Related Topics

- TRANSFORM

5.5.2.8 Managing Oracle Recovery Manager Backups and Copies

You can use I/O Resource Management (IORM) to control the resource consumption and priority of RMAN I/Os.

Backups are an I/O intensive operation. You can control the rate of Oracle Recovery Manager (RMAN) I/Os by configuring the number of channels. You can use IORM for a greater degree of control over the resource consumption and priority of RMAN I/Os. For example, you can map RMAN to a low priority consumer group. If the Oracle Exadata Storage Server is busy, then the RMAN operations run very slowly and not interfere with the other database operations. However, whenever the Oracle Exadata Storage Server is not fully utilized, then IORM schedules the RMAN I/Os, allowing it to consume the unutilized bandwidth.

RMAN backups run under the `BACKUP` function. RMAN copies run under the `COPY` function. By default, both the `BACKUP` and `COPY` functions are mapped to the `BATCH_GROUP` consumer group. You can remap these functions to any other consumer group, as shown in the following example.

Example 5-9   Using Consumer Groups to Manage Resources

This example shows how to map the `BACKUP` function to the `BATCH_GROUP` consumer group and the `COPY` function to the `MAINTENANCE_GROUP` consumer group.

```
BEGIN
DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(DBMS_RESOURCE_MANAGER.ORAC
LE_
FUNCTION, 'BACKUP', 'BATCH_GROUP');
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING(DBMS_RESOURCE_MANAGER.ORAC
LE_
FUNCTION, 'COPY', 'MAINTENANCE_GROUP');
DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
END;
/
```

5.5.3 Administering Interdatabase Resource Management

You can configure an interdatabase or category plan for Oracle Exadata Database Machine using the CellCLI `ALTER IORMPLAN` command.

- How to Create an Interdatabase or Category Plan
  You can run the `ALTER IORMPLAN` commands in CellCLI.
- Required Settings for an Interdatabase or Category Plan
  Each category or interdatabase plan contain the required settings.
- Optional Settings for an Interdatabase or Category Plan
  After specifying the required directives, you can include additional directives to customize your plan.

5.5.3.1 How to Create an Interdatabase or Category Plan

You can run the `ALTER IORMPLAN` commands in CellCLI.

You can also put the `ALTER IORMPLAN` commands in a text file, such as `alter_iorm`, then run the commands using the text file with the CellCLI `START alter_iorm` command. The plan name is automatically set to *`cellname`*`_IORMPLAN`.

The `catPlan` parameter specifies the category plan. The `dbPlan` parameter specifies the interdatabase plan. To manage the I/O resources using a category plan, interdatabase plan or database resource plan, set the objective.

The default for the `objective` option is `basic`, and I/O Resource Management (IORM) manages I/O resources based on the configured resource plans. The `objective` option can be set to `basic`, `auto`, `low_latency`, `balanced`, or `high_throughput`.

If a container database (CDB) plan specifies the `memory_min` and `memory_limit` attributes for pluggable database (PDB), then it is computed as a percentage of the `flashcachemin` and `flashcachelimit` or `flashcachesize` values for the database or the total flash cache size when nothing is specified. The `memory_min` and `memory_limit` are also applied to PMEM cache and it is also computed as a percentage of the `pmemcachemin` and `pmemcachelimit` or `pmemcachesize` values for the database or the total PMEM cache size when nothing is specified.

Related Topics

- Enabling IORM to Prioritize Latency
  I/O Resource Management (IORM) is enabled by default and in its default state IORM manages flash cache and flash log.
- ALTER IORMPLAN
- START and @

5.5.3.2 Required Settings for an Interdatabase or Category Plan

Each category or interdatabase plan contain the required settings.

For interdatabase plan directives to be valid, you must provide either a level and an allocation, share, maximum utilization limit, or a flash cache quota directive. A directive without an allocation, share or maximum utilization limit is invalid. A directive that specifies both allocation and share is also invalid.

When specifying `catPlan` or `dbPlan`, a directive with `name=other` is required. For category plans, the `other` directive provides an allocation for all active consumer groups whose category is not specified in the category plan. For interdatabase plans, the `other` directive provides an allocation for all databases that are using Oracle Exadata System Software, but are not explicitly specified in the interdatabase plan. If the `other` directive is not specified, then the CellCLI utility returns an error.

- Using Allocations to Specify Resources
  You can prioritize the I/O requests from the four example databases by specifying an interdatabase plan.
- Using Shares to Specify Resources
  You can prioritize the I/O requests from the four example databases by assigning a share to each database in an interdatabase plan.

5.5.3.2.1 Using Allocations to Specify Resources

You can prioritize the I/O requests from the four example databases by specifying an interdatabase plan.

Consider the example of four databases sharing the same storage. The four databases are:

- An OLTP production database, named PROD
- A test database, named PROD_TEST
- A development database, named PROD_DEV
- A data warehouse database, named DW

An OLTP production database typically issues small I/O requests, and low latency for these requests is the critical requirement. A data warehouse issues large numbers of large I/O requests and is more sensitive to the I/O throughput than the latency of each individual I/O request. Without any I/O resource management, the large number of I/O requests issued by the DW database could overwhelm the storage subsystem and increase the latency of the I/O requests issued by the PROD database. Additionally, the I/O requests issued by the test and development databases, PROD_TEST and PROD_DEV, could adversely affect the performance of the PROD and the DW databases.

The interdatabase plan has the following characteristics:

- The OLTP database PROD gets 80 percent of the I/O resources with the highest priority level.
- The DW database gets 20 percent of the remaining I/O resources and 80 percent of PROD's unused allocation.
- The PROD_TEST and PROD_DEV databases and the OTHER consumer group get any unused I/O in the amount of 50 percent, 40 percent, and 10 percent respectively.

An example of this interdatabase plan is shown in the following table.

Table 5-4 Interdatabase Plan Using Allocations

Database Name	Level 1 (%)	Level 2 (%)	Level 3 (%)
PROD	**80**	not set	not set
DW	not set	**80**	not set
PROD_TEST	not set	not set	**50**
PROD_DEV	not set	not set	**40**
OTHER	not set	not set	**10**

If the PROD_TEST and PROD_DEV databases put out an inordinately large I/O load, then the performance of databases PROD and DW are not affected. Also, if database DW issues large amounts of I/O, then the performance of the PROD database is not adversely affected.

Example 5-10 ALTER IORMPLAN Syntax for Allocations

This example shows the command to create the interdatabase plan described previously in this topic.

```
CellCLI> ALTER IORMPLAN                                       -
         dbPlan=(                                             -
                 (name=prod, level=1,allocation=80),          -
                 (name=dw, level=2, allocation=80),           -
                 (name=prod_test,  level=3, allocation=50),   -
                 (name=prod_dev, level=3, allocation=40),     -
                 (name=other, level=3, allocation=10))
```

Example 5-11 Configuring an Interdatabase Plan Using Allocations

This example shows how to configure an interdatabase plan using allocation percentages for a cell.

```
CellCLI> ALTER IORMPLAN                                                -
 catPlan=((name=administrative, level=1, allocation=80),               -
          (name=interactive,    level=2, allocation=90),               -
          (name=batch,          level=3, allocation=80),               -
          (name=maintenance,    level=4, allocation=50),               -
          (name=other,          level=4, allocation=50)                -
          ),                                                           -
 dbplan=((name=sales_prod, share=8, role=primary),                     -
         (name=sales_prod, share=1, limit=50, role=standby),           -
         (name=sales_test, share=1, limit=25),                         -
         (name=default, share=2))
```

5.5.3.2.2 Using Shares to Specify Resources

You can prioritize the I/O requests from the four example databases by assigning a share to each database in an interdatabase plan.

Consider the example of four databases sharing the same storage. The four databases are:

- An OLTP production database, named PROD
- A test database, named PROD_TEST
- A development database, named PROD_DEV
- A data warehouse database, named DW

An OLTP production database typically issues small I/O requests, and low latency for these requests is the critical requirement. A data warehouse issues large numbers of large I/O requests and is more sensitive to the I/O throughput than the latency of each individual I/O request. Without any I/O resource management, the large number of I/O requests issued by the DW database could overwhelm the storage subsystem and increase the latency of the I/O requests issued by the PROD database. Additionally, the I/O requests issued by the test and development databases, PROD_TEST and PROD_DEV, could adversely affect the performance of the PROD and the DW databases.

Because the OLTP database PROD is most critical, you can specify 16 shares for the PROD database. The DW database gets 4 shares, PROD_TEST gets 2 shares, and

PROD_DEV gets 1 share. This plan ensures that the PROD database is 4 times more likely than the DW database to issue an I/O.

If PROD does not have any I/Os to issue, then the other databases are picked based on their shares.

Example 5-12 ALTER IORMPLAN Syntax for Shares

This example shows the command to create the interdatabase plan by assigning shares to each database.

```
CellCLI> ALTER IORMPLAN                                    -
         dbPlan=(                                          -
                (name=prod, share=16),                     -
                (name=dw, share=4),                        -
                (name=prod_test, share=2),                 -
                (name=prod_dev, share=1))
```

Example 5-13 Configuring an Interdatabase Plan Using Shares

This example shows how to configure an interdatabase plan using shares for a cell.

```
ALTER IORMPLAN                                                  -
     dbPlan=(                                                   -
        (name=dev01, share=1, limit=50, flashLog=off),          -
        (name=dev02, share=1, limit=25, flashCache=off)         -
        (name=other, share=4))
```

5.5.3.3 Optional Settings for an Interdatabase or Category Plan

After specifying the required directives, you can include additional directives to customize your plan.

When configuring an interdatabase plan, catPlan and dbPlan are optional. If catPlan is not specified, then IORM between categories is not enabled. Similarly, if dbPlan is not specified, then IORM between databases is not enabled.

The optional directives include:

- role Attribute for IORM Plans
  The role attribute allows a different allocation to be specified, based on whether the database has the Oracle Data Guard primary or standby role.
- asmcluster Attribute for IORM Plans
  You can use the asmcluster attribute to uniquely identify databases with the same DB_UNIQUE_NAME.

5.5.3.3.1 role Attribute for IORM Plans

The role attribute allows a different allocation to be specified, based on whether the database has the Oracle Data Guard primary or standby role.

By default, all interdatabase plan allocations apply when the database is in either role. If you want the allocation to apply only when the database is in the primary role, then include `role=primary` in the plan. Similarly, if you want the allocation to apply only when the database is in the `standby` role, then include `role=standby` in the plan.

```
ALTER IORMPLAN                                            -
dbPlan=((name=prod, share=8, role=primary),               -
        (name=prod, share=1, limit=25, role=standby)      -
        (name=default, share=2))
```

Related Topics

- Configuring Resource Manager for Oracle Active Data Guard (My Oracle Support Doc ID 1930540.1)

5.5.3.3.2 asmcluster Attribute for IORM Plans

You can use the `asmcluster` attribute to uniquely identify databases with the same `DB_UNIQUE_NAME`.

If you have databases with the same `DB_UNIQUE_NAME` but associated with different Oracle ASM clusters, then, starting with Oracle Exadata System Software release 19.1.0, you can use the `asmcluster` attribute to uniquely identify the databases.

The databases must be clients of different Oracle ASM clusters, and you must have already configured ASM-scoped security.

```
ALTER IORMPLAN                                            -
dbplan=((name=pdb1, share=4, flashcachemin=5G, asmcluster=asm1),  -
        (name=pdb1, share=2, limit=80, asmcluster=asm2),  -
        (name=pdb2, share=2, flashcachelimit=2G, asmcluster=asm1),  -
        name=default, share=1, flashcachelimit=1G))
```

Related Topics

- Setting Up Oracle ASM-Scoped Security on Oracle Exadata Storage Servers

5.5.4 Flash Cache Attributes for IORM Plans

The `flashcachemin` and `flashcachesize` attributes can be used to specify the guaranteed minimum and maximum quotas for flash cache groups.

These attributes allows you to control how the flash cache space is distributed across the databases. These attributes can only be specified with an interdatabase plan, and are configured using the CellCLI utility. The `flashcachemin` attribute specifies the minimum space in flash cache that is guaranteed for a database.

The `flashcachelimit` attribute specifies the maximum amount of space in the flash cache that a database can use. The `flashcachelimit` is a *soft* maximum which means that the flash cache group can exceed its quota if the flash cache is not full. This model does not work for Oracle Cloud and pay-for-performance environments. A different attribute called `flashcachesize` helps alleviate this limitation. `flashcachesize` is a **hard** maximum that limits the flash cache usage for a group. Because this attribute is intended for use in Oracle Cloud environments, `flashcachesize` is not exceeded even if the flash cache has free space.

Example 5-14 Configuring an Interdatabase Plan for Flash Cache Groups

This example shows how to create a plan that guarantees minimum and maximum quotas for flash cache groups for the `sales` database using `flashCacheSize` parameter. For the `test` database, it guarantees the maximum quotas only. The `sales` and `test` databases are guaranteed a certain amount of space in the flash cache, but the space available in the flash cache for these databases will not exceed the specified size, even if the flash cache has free space.

The `finance` and `dev` databases use `flashCacheMin` for guaranteed minimum quotas, but the maximum quotas are guaranteed only if there is free space in the flash cache. These databases can also exceed the specified size when there is free space in the flash cache.

```
ALTER IORMPLAN                                                            -
 dbplan=((name=sales, share=8, flashCacheSize=10G),                       -
         (name=finance, share=8, flashCacheLimit=10G, flashCacheMin=2G), -
         (name=dev, share=2, flashCacheLimit=4G, flashCacheMin=1G),       -
         (name=test, share=1, limit=10, flashCacheSize=1G))
```

Example 5-15 Configuring an Interdatabase Plan for Flash Cache Groups

Starting with Oracle Exadata System Software release 19.3.0, you can specify both `flashCacheMin` and `flashCacheSize` for the same target.

```
ALTER IORMPLAN                                                   -
 dbplan=((name=sales, share=8, flashCacheMin=3G, flashCacheSize=10G),    -
         (name=finance, share=8, flashCacheLimit=10G, flashCacheMin=2G), -
         (name=dev, share=2, flashCacheLimit=4G, flashCacheMin=1G),       -
         (name=test, share=1, limit=10, flashCacheSize=1G))
```

5.5.5 Managing Flash Cache Quotas for Databases and PDBs

IORM enables you to control how you want the flash cache to be shared among different databases and pluggable databases (PDBs).

This can be done using just the CDB resource plan or both the CDB plan and the I/O Resource Management (IORM) interdatabase plan.

Consider a CDB resource plan that specifies `memory_min` and `memory_limit` for the 3 PDBs mentioned in the plan. Note the following:

- These values are specified in percentages and range from 0 and 100. Because over-provisioning is supported, the sum of the percentages is not restricted to 100%. If the sum of these values is greater than 100%, then the values are normalized down to 100%.
- If `memory_min` is not specified, then it defaults to 0.
- If `memory_limit` is not specified, then it defaults to 100.
- For `CDB$ROOT`, there is a 5% `memory_limit` value.

The following example shows how to create an interdatabase plan for 3 PDBs. The sum of the `memory_min` values is 40%, and the sum of the `memory_limit` values is 175%, which must be normalized. If an interdatabase plan is not specified, then these percentages apply to the entire size of the flash cache. If an interdatabase plan is specified, then the quotas for the PDBs are computed as a percentage of the memory_min and memory_limit values for each database as specified in the interdatabase plan directive.

```
BEGIN
DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();

DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN(
    plan    => ''NEWCDB_PLAN ',
    comment => 'CDB resource plan for newcdb');

  DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE(
    plan                  => 'NEWCDB_PLAN',
    pluggable_database    => 'SALESPDB',
    memory_min            => 20);
  DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE(
    plan                  => ' NEWCDB_PLAN ',
    pluggable_database    => 'SERVICESPDB',
    memory_min            => 20,
    memory_limit          => 50);
  DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE(
    plan                  => ' NEWCDB_PLAN ',
    pluggable_database    => 'HRPDB',
    memory_limit          => 25);

DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
END;
/
```

In the example above, if no interdatabase plan is specified and the size of the flash cache is 10 GB, then the following table shows the breakdown of the quotas after normalization of the limit (because the sum of the memory_limit values is greater than 100%). If we end up with a minimum value that is greater than the limit, we reduce the minimum value to make it equal to the limit.

Table 5-5   Case 1: PDB Flash Cache Limits with No Interdatabase Plan

PDB	Flash Cache Min	FC Soft Limit	Normalized Soft Limit	FC Hard Limit
SALESPDB	20% = 2 GB	100 (default)	100 / 175 = 5.7 GB	n/a
SERVICESPDB	20% = 2 GB	50	50 / 175 = 2.85 GB	n/a
HRPDB	0	25	25 / 175 = 1.4 GB	n/a

The following example shows an interdatabase plan running on a system where the flash cache size is 50 GB and PMEM cache size is 10 GB.

```
ALTER IORMPLAN dbplan=                                            -
((name=newcdb,  share=8,  pmemCacheSize= 2G, flashCacheSize=10G), -
(name=finance, share=8,  pmemCacheMin= 1G, pmemCacheLimit= 2G,
flashCacheLimit=10G, flashCacheMin=2G), -
            (name=dev,     share=2,  pmemCacheMin= 100M, pmemCacheLimit= 1G,
            flashCacheLimit=4G, flashCacheMin=1G),  -
            (name=test,    share=1,  limit=10))
```

In addition to the NEWCDB CDB, three other databases (finance, dev, and test) share the same storage servers. Flash cache quotas are only enforced if the directives specify the flashcachesize, flashcachelimit, or flashcachemin attribute. The flashcachesize attribute specifies a hard limit and cannot be specified with flashcachelimit in the same directive. The database test does not specify any flash cache directive; so that database and its PDBs (if any exist) are not managed for any flash cache quotas.

If the CDB has flashcachesize specified, then the memory_min value from the CDB resource plan is ignored, and the memory_limit value is normalized and used to compute the flash cache sizes for the different PDBs. Because the newcdb CDB has flashcachesize specified in the example above, the memory_min values are ignored. The flashcachesize is computed after normalizing the memory_limit values, as was shown previously. The only difference is that this will be a hard limit since the CDB has specified the flashcachesize directive.

Table 5-6 Case 2: PDB Flash Cache Limits with an InterDatabase Plan

PDB	Flash Cache Min	FC Hard Limit	Normalized Hard Limit	FC Soft Limit
SALESPDB	0	100 (default)	100 / 175 = 5.71 GB	n/a
SERVICESPDB	0	50	50 / 175 = 2.86 GB	n/a
HRPDB	0	25	25 / 175 = 1.43 GB	n/a

For non-CDB databases, the flashcachesize, flashcachemin, and flashcachelimit values are specified in absolute terms and no additional normalization is required. Because flashcachemin is a guaranteed reservation, the sum of flashcachemin across all the directives should be less than the total size of the flash cache.

5.5.6 PMEM Cache Attributes for IORM Plans

The pmemcachemin and pmemcachesize attributes can be used to specify the guaranteed minimum and maximum quotas for PMEM cache groups.

These attributes allows you to control how the PMEM cache space is distributed across the databases. These attributes can only be specified with an interdatabase plan, and are configured using the CellCLI utility.

- The pmemcachemin attribute specifies the minimum space in PMEM cache that is guaranteed for a database.
- The pmemcachelimit attribute specifies the maximum amount of space in the PMEM cache that a database can use. The pmemcachelimit is a soft maximum which means that the PMEM cache group can exceed its quota if the PMEM cache is not full.

- For Oracle Cloud and pay-for-performance environments, use a different attribute called pmemcachesize. pmemcachesize is a hard maximum that limits the PMEM cache usage for a group. Because this attribute is intended for use in Oracle Cloud, pmemcachesize is not exceeded even if the PMEM cache has free space.

Example 5-16 Configuring an Interdatabase Plan for PMEM Cache Groups

This example shows how to create a plan that guarantees minimum and maximum quotas for PMEM cache groups for the sales database using pmemCacheSize parameter. For the test database, it guarantees the maximum quotas only. The sales and test databases are guaranteed a certain amount of space in the PMEM cache, but the space available in the PMEM cache for these databases will not exceed the specified size, even if the PMEM cache has free space.

The finc and dev databases use pmemCacheMin for guaranteed minimum quotas, but the maximum quotas are guaranteed only if there is free space in the PMEM cache. These databases can also exceed the specified size when there is free space in the PMEM cache.

```
ALTER IORMPLAN dbplan=                                            -
((name=sales, share=8, pmemCacheSize= 2G, flashCacheSize=10G), -
(name=finc, share=8, pmemCacheMin= 1G, pmemCacheLimit= 2G,
flashCacheLimit=10G, flashCacheMin=2G), -
(name=dev, share=2, pmemCacheMin= 500M, pmemCacheLimit= 1G,
flashCacheLimit=4G, flashCacheMin=1G), -
(name=test, share=1, limit=10, pmemCacheSize= 200M))
```

5.5.7 Managing PMEM Cache Quotas for Databases and PDBs

I/O Resource Management (IORM) enables you to control how you want the PMEM cache to be shared among different databases and pluggable databases (PDBs).

Consider a CDB resource plan that specifies memory_min and memory_limit for the 3 PDBs mentioned in the plan. Note the following:

- These values are specified in percentages and range from 0 and 100. Because over-provisioning is supported, the sum of the percentages is not restricted to 100%. If the sum of these values is greater than 100%, then the values are normalized down to 100%.
- If memory_min is not specified, then it defaults to 0.
- If memory_limit is not specified, then it defaults to 100.
- For CDB$ROOT, there is a 5% memory_limit value.

The following example shows how to create an interdatabase plan for 3 PDBs. The sum of the memory_min values is 40%, and the sum of the memory_limit values is 175%, which must be normalized. If an interdatabase plan is not specified, then these percentages apply to the entire size of the PMEM cache. If an interdatabase plan is specified, then the quotas for the PDBs are computed as a percentage of the memory_min and memory_limit values for the database as specified in the interdatabase plan directive.

```
BEGIN
DBMS_RESOURCE_MANAGER.CREATE_PENDING_AREA();
            DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN(
                plan    => ''NEWCDB_PLAN ',
                comment => 'CDB resource plan for newcdb');

              DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE(
                plan                  => 'NEWCDB_PLAN',
                pluggable_database    => 'SALESPDB',
                memory_min            => 20);
              DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE(
                plan                  => ' NEWCDB_PLAN ',
                pluggable_database    => 'SERVICESPDB',
                memory_min            => 20,
                memory_limit          => 50);
              DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE(
                plan                  => ' NEWCDB_PLAN ',
                pluggable_database    => 'HRPDB',
                memory_limit          => 25);

            DBMS_RESOURCE_MANAGER.VALIDATE_PENDING_AREA();
            DBMS_RESOURCE_MANAGER.SUBMIT_PENDING_AREA();
            END;
            /
```

In the example above, if no interdatabase plan is specified and the size of the PMEM cache is 10 GB, then the following table shows the breakdown of the quotas after normalization of the limit where the sum of the `memory_limit` values is greater than 100%. If we end up with a minimum value that is greater than the limit, we reduce the minimum value to make it equal to the limit.

Table 5-7   Case 3: PDB PMEM Cache Limits with No Interdatabase Plan

PDB	PMEM Cache Min	PMEM Soft Limit	Normalized Soft Limit	PMEM Hard Limit
SALESPDB	20% = 2 GB	100 (default)	100 / 175 = 5.7 GB	n/a
SERVICESPDB	20% = 2 GB	50	50 / 175 = 2.85 GB	n/a
HRPDB	0	25	25 / 175 = 1.4 GB	n/a

Next, consider an interdatabase plan, as shown below, running on a system where the flash cache size is 50 GB and PMEM cache size is 10 GB.

```
ALTER IORMPLAN dbplan=                                            -
((name=newcdb,  share=8,  pmemCacheSize= 2G, flashCacheSize=10G), -
(name=finance, share=8,  pmemCacheMin= 1G, pmemCacheLimit= 2G,
flashCacheLimit=10G, flashCacheMin=2G), -
(name=dev,     share=2,  pmemCacheMin= 100M, pmemCacheLimit= 1G,
flashCacheLimit=4G, flashCacheMin=1G),  -
(name=test,    share=1,  limit=10))
```

In addition to the `newcdb` CDB, three other databases (`finance`, `dev`, and `test`) share the same storage servers. PMEM cache quotas are only enforced if the directives specify the pmemcachemin, pmemcachelimit and pmemcachesize attributes. pmemcachesize is a hard limit and it cannot be specified with pmemcachelimit in the same directive. Database test does not specific any PMEM cache directive, so that database and its PDBs (if any exist) are not managed for any PMEM cache quotas.

If the CDB has pmemcachesize specified, then the memory_min value from the CDB resource plan is ignored, and the memory_limit value is normalized and used to compute the PMEM cache sizes for the different PDBs. Because the newcdb CDB has pmemcachesize specified, the memory_min values are ignored. The pmemcachesize is computed after normalizing the memory_limit values, as was shown previously. The only difference is that this will be a hard limit since the CDB has specified the pmemcachesize directive.

Table 5-8   Case 4: PDB PMEM Cache Limits with an InterDatabase Plan

PDB	PMEM Cache Min	PMEM Hard Limit	Normalized Hard Limit	PMEM Soft Limit
SALESPDB	0	100 (default)	100 / 175 = 1.14 GB	n/a
SERVICESPDB	0	50	50 / 175 = 0.57 GB	n/a
HRPDB	0	25	25 / 175 = 0.28 GB	n/a

For non-CDB databases, the pmemcachesize, pmemcachemin, and pmemcachelimit values are specified in absolute terms and no additional normalization is required. Because pmemcachemin is a guaranteed reservation, the sum of pmemcachemin across all the directives should be less than the total size of the PMEM cache.

5.5.8 Managing Flash Cache and Flash Log

You can use IORM to manage flash cache and flash log instead of I/O resources for the databases.

1. Disable I/O prioritization using the ALTER IORMPLAN command.

   ```
   CellCLI> ALTER IORMPLAN objective='basic';
   ```

2. Disable flash cache and flash log for the PROD_TEST and PROD_DEV databases

   ```
   CellCLI> ALTER IORMPLAN                                    -
            dbPlan=(                                          -
                   (name=prod, flashcache=on, flashLog=on),          -
                   (name=dw, flashcache=on, flashLog=on),            -
                   (name=prod_test, flashcache=off, flashLog=off),   -
                   (name=prod_dev, flashcache=off, flashLog=off))
   ```

   Explicitly setting flash cache and flash log to on for the PROD and DW databases is not required because flash cache and flash log are set to on by default. The settings are included in this example to contrast with the settings for the PROD_TEST and PROD_DEV databases,

5.5.9 Listing an I/O Resource Management Plan

You can view the current interdatabase plan for a storage server using the CellCLI `LIST IORMPLAN` command on the storage server.

Example 5-17 Displaying Interdatabase Plan Details

This example shows how to get a detailed list of the interdatabase plan attributes.

```
CellCLI> LIST IORMPLAN DETAIL
   name:                   cell01_IORMPLAN
   status:                 active
   catPlan:                name=administrative,level=1,allocation=80
                           name=interactive,level=2,allocation=90
                           name=batch,level=3,allocation=80
                           name=maintenance,level=4,allocation=50
                           name=other,level=4,allocation=50
   dbplan:                 name=sales_prod, share=8, role=primary
                           name=sales_prod, share=1, limit=50, role=standby
                           name=sales_test, share=1, limit=25
                           name=default, share=2
   objective:              balanced
```

Related Topics

- LIST IORMPLAN

5.5.10 Verifying the Configuration of I/O Resource Management

Use this checklist to verify that I/O Resource Management (IORM) is configured correctly.

- Verify that the following criteria are met when using IORM to manage I/O resources within a database:
  - A resource plan has been enabled.
  - The same resource plan has been enabled on all database instances.
    - If Oracle Database Resource Manager is enabled using Scheduler Window, then the same plan is always enabled on all database instances.
    - If Oracle Database Resource Manager is enabled using the `RESOURCE_MANAGER_PLAN` parameter, then use `sid='*'` to set the parameter for all database instances.
  - The resource plan includes `MGMT_P[1-8]` directives for each consumer group in the resource plan.

  The following query can be used to verify the preceding criteria have been met:

  ```
  SELECT DECODE(count(*), 0, 'Intra-Instance IORM Plan Enabled',
  'No Intra-Instance IORM Plan Enabled') status
  FROM gv$instance
  WHERE inst_id NOT IN
    (SELECT inst_id FROM gv$rsrc_plan WHERE cpu_managed = 'ON');
  ```

- Verify that the interdatabase plan has been configured properly when IORM is used to manage I/O resources from multiple databases

  ```
  CellCLI> LIST IORMPLAN DETAIL
  ```

  If no interdatabase plan has been configured, then use the CellCLI `ALTER IORMPLAN` command to configure a plan. Each active database should have its own directive in the `dbPlan` parameter.

- Verify that sessions are mapped to the correct consumer group.

  Run the following query while a workload is running:

  ```
  SELECT r.sid,
         c.consumer_group current_consumer_group
    FROM v$rsrc_session_info r, dba_rsrc_consumer_groups c
    WHERE r.current_consumer_group_id = c.consumer_group_id
  UNION
  SELECT sid, 'OTHER_GROUPS' from v$rsrc_session_info
    WHERE current_consumer_group_id = 0;
  ```

  A session may not be in the expected consumer group due to the following configuration errors:

  – **Missing privilege**: In order for a session to switch into a consumer group, its user or role must have permission to switch into that consumer group. The following query shows the permissions for all consumer groups.

    ```
    SELECT grantee, granted_group
    FROM DBA_RSRC_CONSUMER_GROUP_PRIVS
    ORDER BY granted_group;
    ```

    Use the following command to grant permission for any session to switch into the consumer group. This example shows to how grant the permission for `BATCH_GROUP`.

    ```
    EXEC dbms_resource_manager_privs.grant_switch_consumer_group -
      ('public', 'BATCH_GROUP', FALSE);
    ```

  – **Inactive consumer group**: If a session maps to or is manually switched to a consumer group that is not part of the current resource plan, then the session is switched into the default consumer group, `OTHER_GROUPS`.

    If sessions are being assigned to consumer groups using mapping rules, then the following query can be used to determine the consumer group that the mapping rules selected, the mapping attribute that was used, and the consumer group that the session started in originally.

    ```
    SELECT r.sid,
           r.mapped_consumer_group,
           r.mapping_attribute,
           c.consumer_group original_consumer_group
    FROM v$rsrc_session_info r, dba_rsrc_consumer_groups c
    WHERE r.orig_consumer_group_id = c.consumer_group_id;
    ```

If the mapped consumer group differs from the original consumer group, then the mapped consumer group was not part of the resource plan.

- Use CellCLI to list the number of small and large I/O requests that were issued for each consumer group across all databases.

  ```
  CellCLI> LIST METRICCURRENT CG_IO_RQ_LG, CG_IO_RQ_SM   ATTRIBUTES name,
  -
           metricObjectName, metricValue, collectionTime;
  ```

  Each consumer group that has an active I/O workload should generate small or large I/O requests according to these metrics.

- While a workload is running, verify that I/O loads are being managed in the correct consumer groups.

  The following CellCLI command lists the number of small and large I/O requests that were issued for each consumer group across all databases:

  ```
  CellCLI> LIST METRICCURRENT CG_IO_RQ_LG, CG_IO_RQ_SM   ATTRIBUTES name,
  -
           metricObjectName, metricValue, collectionTime;
  ```

- While the workload is running, query the actual I/O utilization for each category, database and consumer group.

  The following CellCLI command lists the small and large I/O utilization for each database running on Oracle Exadata Storage Server:

  ```
  CellCLI> LIST METRICCURRENT DB_IO_UTIL_LG, DB_IO_UTIL_SM ATTRIBUTES
  name, -
           metricObjectName, metricValue, collectionTime;
  ```

  The output shows the percentage of disk resources utilized by small and large requests from the databases.

Related Topics

- Monitoring IORM Utilization
  You use metrics to monitor IORM utilization.

5.5.11 Resetting Default Values in an Interdatabase Plan

To reset an interdatabase plan attribute to the default value, set the attribute to an empty string.

You can reset the entire plan, or separately reset the `catPlan` or `dbPlan`.

```
CellCLI> ALTER IORMPLAN dbPlan="", catPlan=""
CellCLI> ALTER IORMPLAN dbPlan=""
CellCLI> ALTER IORMPLAN catPlan=""
```

6 Monitoring and Tuning Oracle Exadata System Software

Oracle provides utilities for monitoring tasks in the Oracle Exadata Storage Server environment.

- Understanding Metrics and Alerts
  Metrics and alerts help you monitor Oracle Exadata System Software.
- Monitoring Oracle Exadata Storage Server with Metrics
  You can monitor a cell by viewing the Oracle Exadata Storage Server metrics.
- Monitoring IORM with Metrics
  I/O Resource Management (IORM) can be monitored using Oracle Exadata System Software metrics.
- Monitoring Requests and Alerts for Oracle Exadata Storage Server
  You can monitor the active requests and alerts for a storage server and you can receive e-mail notifications for alerts.
- Monitoring Oracle Exadata Storage Server using Views
  Dynamic `V$` data dictionary views can help monitor the Oracle Exadata Storage Server environment.
- Understanding Oracle Exadata System Software Wait Events
  Oracle Exadata System Software uses a different set of wait events for disk I/O to Oracle Exadata Storage Server than disk I/O to conventional storage.
- Optimizing Performance
  You can use various settings to optimize the performance of Oracle Exadata System Software.
- Using the SQL EXPLAIN PLAN Command with Oracle Exadata System Software
  The SQL `EXPLAIN PLAN` command was enhanced to display the predicates that are evaluated on Oracle Exadata Storage Server.
- Enabling or Disabling In-Memory Columnar Caching on Storage Servers
  In-Memory Columnar Caching on cells is enabled by default when the `INMEMORY_SIZE` is configured. You do not need to do anything to get this enhancement.

Related Topics

- Monitoring and Tuning the Database
- Monitoring the Database
- Performance Tuning Overview
- Introduction to SQL Performance Analyzer
- *Oracle Enterprise Manager Exadata Management Getting Started Guide*

6.1 Understanding Metrics and Alerts

Metrics and alerts help you monitor Oracle Exadata System Software.

Metrics are associated with objects such as cells and cell disks, and can be cumulative, rate, or instantaneous. By reviewing metrics and setting thresholds for alerts, you can tune and optimize resource management.

- About Metrics
  Metrics are recorded observations of important run-time properties or internal instrumentation values of the storage server and its components, such as cell disks or grid disks.
- About Alerts
  Alerts represent events of importance occurring within the storage cell, typically indicating that storage cell functionality is either compromised or in danger of failure.

6.1.1 About Metrics

Metrics are recorded observations of important run-time properties or internal instrumentation values of the storage server and its components, such as cell disks or grid disks.

Metrics are a series of measurements that are computed and retained in memory for an interval of time, and stored on a disk for a more permanent history. Metric values typically record either the absolute value of a cumulative counter or a rate of change in a cumulative counter over an observed time period. Some metrics are used to record the time of state transitions as well.

Metric values can cause alerts to signal by comparing those values against threshold boundaries. Metrics can be associated with warning and critical thresholds when extreme values in the metric might indicate a problem or other event of interest to an administrator.

Related Topics

- Monitoring Oracle Exadata Storage Server with Metrics
  You can monitor a cell by viewing the Oracle Exadata Storage Server metrics.

6.1.2 About Alerts

Alerts represent events of importance occurring within the storage cell, typically indicating that storage cell functionality is either compromised or in danger of failure.

An administrator should investigate these alerts, because they might require corrective or preventive action. There are three types of alerts, informational, warning or critical. Alerts are typically propagated to a monitoring infrastructure, such as Oracle Enterprise Manager Cloud Control, for notification to storage administrators. Examples of possible alerts that can be used are physical disk failure, disk read/write errors, cell temperature exceeding recommended value, Oracle Exadata System Software failure, and excessive I/O latency. Metrics can be used to signal alerts using warning or critical threshold values. When the metric value exceeds the threshold value, an alert can be signaled.

Alerts are either stateful or stateless. Stateful alerts represent observable cell states that can be subsequently retested to detect whether the state has changed, so that a previously observed alert condition is no longer a problem. Stateless alerts represent point-in-time events that do not represent a persistent condition; they simply show that something has occurred.

> **Note:**
>
> There are built-in thresholds for ambient temperature. If the temperature is too low or too high, an alert is automatically sent.

Related Topics

- Monitoring Requests and Alerts for Oracle Exadata Storage Server
  You can monitor the active requests and alerts for a storage server and you can receive e-mail notifications for alerts.

6.2 Monitoring Oracle Exadata Storage Server with Metrics

You can monitor a cell by viewing the Oracle Exadata Storage Server metrics.

Metrics are of the following type:

- Cumulative: Statistics since the metric was created.
- Instantaneous: Values at the time the metric is collected.
- Rate: Computed value for metric by averaging statistics over a period of time.

In the table descriptions in the following topics, small means less than or equal to 128 KB, and large means more than 128 KB.

- Displaying Metrics
  You can use the CellCLI `LIST` command to display and monitor metrics for cell objects.
- Monitoring Cell Metrics
  Cell metrics provide information about the cell, such as CPU utilization, memory or network interconnections.
- Monitoring Cell Disk Metrics
  Cell disk metrics provide information about the I/O load for cell disks, such as the number of large blocks read from a cell disk.
- Monitoring Flash Cache Metrics
  Flash cache metrics provide information about the utilization of flash cache, such as the number of megabytes read per second from flash cache.
- Monitoring Exadata Smart Flash Log with Metrics
  Exadata Smart Flash Log metrics provide information about flash log utilization, such as the number of megabytes written per second.
- Monitoring PMEM Cache Metrics
  PMEM cache metrics provide information about the utilization of the PMEM cache.
- Monitoring Grid Disk Metrics
  Grid disk metrics provide information about the I/O load for grid disks, such as the number of large blocks read from a grid disk.

- Monitoring Host Interconnection Metrics
  Host interconnection metrics provide information about the I/O transmission for hosts that access cell storage.
- Monitoring Smart I/O Metrics
  Smart I/O metrics provide information about smart I/O performance. Smart I/O is the I/O processing, such as smart scan, and smart incremental backup, performed by the Oracle Exadata Storage Server offload functionality.

6.2.1 Displaying Metrics

You can use the CellCLI `LIST` command to display and monitor metrics for cell objects.

You can display metric definitions, current metrics, and metric history using the `LIST` command.

- Displaying Metric Definitions
  Use the `LIST METRICDEFINITION` command to display the metric definitions for a storage server.
- Displaying Current Metrics
  Use the `LIST METRICCURRENT` command to display the current metric values for a storage server.
- Displaying Metric History
  Use the `LIST METRICHISTORY` command to display the metric history for a storage server.

6.2.1.1 Displaying Metric Definitions

Use the `LIST METRICDEFINITION` command to display the metric definitions for a storage server.

A metric definition listing shows the configuration of a metric.

Example 6-1 Displaying Metric Definitions

This example shows how to display attributes for the `METRICDEFINITION` object.

```
CellCLI> LIST METRICDEFINITION CL_CPUT DETAIL

CellCLI> LIST METRICDEFINITION WHERE objectType = 'GRIDDISK'

CellCLI> LIST METRICDEFINITION WHERE name LIKE 'CD_IO_RQ.*' -
         ATTRIBUTES name, metricType, description
```

Related Topics

- LIST METRICDEFINITION

6.2.1.2 Displaying Current Metrics

Use the `LIST METRICCURRENT` command to display the current metric values for a storage server.

A current metric listing shows a set of observations on the current value of an individual metric.

Example 6-2   Displaying Current Metric Values

This example shows how to display attributes for the `METRICCURRENT` object.

```
CellCLI> LIST METRICCURRENT CL_TEMP DETAIL

         name:                   CL_TEMP
         alertState:             normal
         collectionTime:         2009-12-17T15:32:25-08:00
         metricObjectName:       abcd2x3
         metricType:             Instantaneous
         metricValue:            48.0 C
         objectType:             CELL

CellCLI> LIST METRICCURRENT WHERE objectType = 'CELLDISK'
AND                 -
         metricValue != 0 ATTRIBUTES name,
metricObjectName,                  -
         metricValue, collectionTime

         CD_IO_BY_R_LG   CD_00_abcd2x3    1.9 MB  2009-12-17T15:46:52-08:00
         CD_IO_BY_R_LG   CD_01_abcd2x3    1.0 MB  2009-12-17T15:46:52-08:00
         CD_IO_BY_R_LG   CD_02_abcd2x3    4.1 MB  2009-12-17T15:46:52-08:00
         CD_IO_BY_R_LG   CD_03_abcd2x3    9.5 MB  2009-12-17T15:46:52-08:00
         CD_IO_BY_R_LG   CD_04_abcd2x3    0.1 MB  2009-12-17T15:46:52-08:00
         CD_IO_BY_R_LG   CD_05_abcd2x3    0.4 MB  2009-12-17T15:46:52-08:00
         ...
```

Related Topics

- DESCRIBE METRICCURRENT
- LIST METRICCURRENT

6.2.1.3 Displaying Metric History

Use the `LIST METRICHISTORY` command to display the metric history for a storage server.

A metric history listing shows a collection of past individual observations of all metric values.

The retention period for metric history files is specified by the `metricHistoryDays` cell attribute. You can modify this setting with the CellCLI `ALTER CELL` command.

Example 6-3   Displaying Metric History Values

This example shows how to display attributes for the `METRICHISTORY` object.

```
CellCLI> LIST METRICHISTORY CD_IO_RQ_R_LG WHERE alertState='critical'
DETAIL

CellCLI> LIST METRICHISTORY WHERE objectType = 'CELLDISK' AND metricValue !
= 0   -
         AND collectionTime > '2009-08-12T09:10:51-07:00' -ATTRIBUTES
```

```
name,      -
        metricObjectName, metricValue, collectionTime
```

Related Topics

- DESCRIBE METRICHISTORY
- LIST METRICHISTORY
- ALTER CELL
- DESCRIBE CELL

6.2.2 Monitoring Cell Metrics

Cell metrics provide information about the cell, such as CPU utilization, memory or network interconnections.

To display cell metrics, use an object type attribute equal to `CELL` with the `METRICCURRENT`, `METRICDEFINITION`, and `METRICHISTORY` objects.

Oracle Exadata System Software may utilize most of the memory available on a storage server. Oracle Exadata System Software has internal controls to avoid over-utilization of the memory. In the event of low swap space, Restart Server (RS) generates an alert, and restarts the CELLSRV process on the Oracle Exadata Storage Server. These alerts should be reported to Oracle Support Services. Oracle does not recommend the configuration of other alerts for memory utilization monitoring for Oracle Exadata Storage Servers by Oracle Enterprise Manager.

The following table lists the metric names and descriptions for cell metrics. The storage server associated with the metric is specified by the `metricObjectName` attribute of the `METRICCURRENT` and `METRICHISTORY` objects.

Table 6-1   Cell Metrics and Descriptions

Metric	Description
`CL_BBU_CHARGE`	Disk controller battery charge.
`CL_BBU_TEMP`	Temperature of disk controller battery.
`CL_CPUT`	The storage server CPU utilization which is the instantaneous percentage of time over the previous minute that the system CPUs were not idle (from `/proc/stat`).
`CL_CPUT_CS`	The percentage of CPU time used by CELLSRV.
`CL_CPUT_MS`	The percentage of CPU time used by MS.
`CL_FANS`	The instantaneous number of working fans on the storage server.
`CL_FSUT`	The percentage of total space utilized on the file system that is currently in use. This metric shows the space utilization in the various files systems on the storage server.
`CL_IO_RQ_NODATA`	The number of I/O requests that did not return data.

Table 6-1 (Cont.) Cell Metrics and Descriptions

Metric	Description
CL_IO_RQ_NODATA_SEC	The number of I/O requests that did not return data per second.
CL_MEMUT	The percentage of total physical memory used on the storage server.
CL_MEMUT_CS	The percentage of physical memory used by CELLSRV.
CL_MEMUT_MS	The percentage of physical memory used by MS.
CL_RUNQ	The instantaneous average number (over the preceding minute) of processes in the Oracle Linux run queue marked running or uninterruptible (from /proc/loadavg).
CL_SWAP_IN_BY_SEC	The number of swap pages read in KB per second.
CL_SWAP_OUT_BY_SEC	The number of swap pages written in KB per second.
CL_SWAP_USAGE	The percentage of swap space used.
CL_TEMP	The instantaneous temperature (Celsius) of the server, provided by the Baseboard Management Controller (BMC).
CL_VIRTMEM_CS	The amount of virtual memory used by CELLSRV in megabytes.
CL_VIRTMEM_MS	Total memory (resident and swap) used by MS in megabytes.
IORM_MODE	The I/O Resource Management (IORM) objective for the storage server.
N_HCA_MB_RCV_SEC	The number of megabytes received by the InfiniBand interfaces per second.
N_HCA_MB_TRANS_SEC	The number of megabytes transmitted by the InfiniBand interfaces per second.
N_NIC_KB_RCV_SEC	The number of kilobytes received by the Ethernet interfaces per second.
N_NIC_KB_TRANS_SEC	The number of kilobytes transmitted by the Ethernet interfaces per second.
N_NIC_RCV_SEC	The rate which is the total number of I/O packets received by interconnections per second.
N_NIC_TRANS_SEC	The rate which is the total number of I/O packets transmitted by interconnections per second.

Example 6-4 Displaying Cell Metrics

This example shows how to display the storage server metric CL_CPUT.

```
CellCLI> LIST METRICCURRENT CL_CPUT DETAIL
         name:                   CL_CPUT
         alertState:             normal
         collectionTime:         2020-01-03T17:26:45-08:00
         metricObjectName:       exa01celadm13
         metricType:             Instantaneous
         metricValue:            4.3 %
         objectType:             CELL
```

6.2.3 Monitoring Cell Disk Metrics

Cell disk metrics provide information about the I/O load for cell disks, such as the number of large blocks read from a cell disk.

To display cell disk metrics, use an object type attribute equal to CELLDISK with the METRICCURRENT, METRICDEFINITION, and METRICHISTORY objects.

The following table lists the metric names and descriptions for cell disk metrics. The cell disk for the metric is specified by the metricObjectName attribute of the METRICCURRENT and METRICHISTORY objects. For cumulative metrics, the metric value for a specific time period can be determined by subtracting values from different collectionTime periods. For rate metrics, the time period for the metric value is over the previous minute.

Table 6-2 Cell Disk Metrics and Descriptions

Metric	Description
CD_BY_FC_DIRTY	The number of megabytes in flash cache that are not synchronized to the cell disk.
CD_IO_BY_R_LG	The cumulative number of megabytes read in large blocks from a cell disk.
CD_IO_BY_R_LG_SEC	The rate which is the number of megabytes read in large blocks per second from a cell disk.
CD_IO_BY_R_SCRUB	The number of megabytes read from a cell disk by the scrubbing job.
CD_IO_BY_R_SCRUB_SEC	The number of megabytes read per second from a cell disk by the scrubbing job.
CD_IO_BY_R_SM	The cumulative number of megabytes read in small blocks from a cell disk.
CD_IO_BY_R_SM_SEC	The rate which is the number of megabytes read in small blocks per second from a cell disk.
CD_IO_BY_W_LG	The cumulative number of megabytes written in large blocks on a cell disk.
CD_IO_BY_W_LG_SEC	The rate which is the number of megabytes written in large blocks per second on a cell disk.
CD_IO_BY_W_SM	The cumulative number of megabytes written in small blocks on a cell disk.

Table 6-2 (Cont.) Cell Disk Metrics and Descriptions

Metric	Description
`CD_IO_BY_W_SM_SEC`	The rate which is the number of megabytes written in small blocks per second on a cell disk.
`CD_IO_ERRS`	The cumulative number of I/O errors on a cell disk.
`CD_IO_ERRS_MIN`	The rate of I/O errors on a cell disk per minute.
`CD_IO_ERRS_SCRUB`	The number of bad 1 MB blocks detected during a scrubbing job on a cell disk.
`CD_IO_LOAD`	The average I/O load from cell disks. I/O load specifies the length of the disk queue. It is similar to iostat's `avgqu-sz`, but I/O load is a weighted value depending on the type of disk: • For hard disks, a large I/O has three times the weight of a small I/O. • For flash disks, large and small I/Os have the same weight. Because this metric is computed by I/O Resource Management (IORM), this metric is also available per database or pluggable database (PDB).
`CD_IO_RQ_R_LG`	The cumulative number of requests to read large blocks from a cell disk.
`CD_IO_RQ_R_LG_SEC`	The rate which is the number of requests to read large blocks per second from a cell disk.
`CD_IO_RQ_R_SCRUB`	The number of requests to read from a cell disk by the scrubbing job.
`CD_IO_RQ_R_SCRUB_SEC`	The number of requests to read per second from a cell disk by the scrubbing job.
`CD_IO_RQ_R_SM`	The cumulative number of requests to read small blocks from a cell disk.
`CD_IO_RQ_R_SM_SEC`	The rate which is the number of requests to read small blocks per second from a cell disk.
`CD_IO_RQ_W_LG`	The cumulative number of requests to write large blocks to a cell disk.
`CD_IO_RQ_W_LG_SEC`	The rate which is the number of requests to write large blocks per second to a cell disk.
`CD_IO_RQ_W_SM`	The cumulative number of requests to write small blocks to a cell disk.
`CD_IO_RQ_W_SM_SEC`	The rate which is the number of requests to write small blocks per second to a cell disk.
`CD_IO_ST_RQ`	Average service time per request for small I/O requests to a cell disk. This metric cannot be used for flash devices as explained in Using iostat to monitor I/O service time statistics.
`CD_IO_TM_R_LG`	The cumulative latency of reading large blocks from a cell disk. Units are microseconds per request.
`CD_IO_TM_R_LG_RQ`	The rate which is the average latency of reading large blocks per request to a cell disk. Units are microseconds per request.
`CD_IO_TM_R_SM`	The cumulative latency of reading small blocks from a cell disk.

Table 6-2 (Cont.) Cell Disk Metrics and Descriptions

Metric	Description
CD_IO_TM_R_SM_RQ	The rate which is the average latency of reading small blocks per request from a cell disk. Units are microseconds per request.
CD_IO_TM_W_LG	The cumulative latency of writing large blocks to a cell disk. Units are microseconds per request.
CD_IO_TM_W_LG_RQ	The rate which is the average latency of writing large blocks per request to a cell disk. Units are microseconds per request.
CD_IO_TM_W_SM	The cumulative latency of writing small blocks to a cell disk. Units are microseconds per request.
CD_IO_TM_W_SM_RQ	The rate which is the average latency of writing small blocks per request to a cell disk. Units are microseconds per request.
CD_IO_UTIL	The percentage of device utilization for the cell disk. This metric is similar to %util of iostat for hard disks, but not for flash devices as mentioned in Using iostat to monitor I/O service time statistics. For flash devices, it is a percentage of maximum I/O bandwidth specified in the Exadata data sheet for your system. Because this metric is computed by IORM, it is also available per database, PDB, and consumer group.
CD_IO_UTIL_LG	The percentage of disk resources utilized by large requests for the cell disk.
CD_IO_UTIL_SM	The percentage of disk resources utilized by small requests for the cell disk.

6.2.4 Monitoring Flash Cache Metrics

Flash cache metrics provide information about the utilization of flash cache, such as the number of megabytes read per second from flash cache.

To display flash cache metrics, use an object type attribute equal to FLASHCACHE with the METRICCURRENT, METRICDEFINITION, and METRICHISTORY objects.

The following table lists the metric names and descriptions for flash cache metrics. The flash cache for the metric is specified by the metricObjectName attribute of the METRICCURRENT and METRICHISTORY objects.

Table 6-3 Flash Cache Metrics and Descriptions

Metric	Description
FC_BY_ALLOCATED	The number of megabytes allocated in flash cache. Tracks how many cachelines are used in the flash cache. If the value is close to the flash cache size, then the flash cache is fully populated.
FC_BY_ALLOCATED_DIRTY	The number of megabytes allocated for unflushed data in flash cache.
FC_BY_ALLOCATED_OLTP	The number of megabytes allocated for OLTP data in flash cache.

Table 6-3 (Cont.) Flash Cache Metrics and Descriptions

Metric	Description
FC_BY_DIRTY	The number of megabytes in flash cache that are not synchronized to the grid disks.
FC_BY_STALE_DIRTY	The number of megabytes in flash cache which cannot be synchronized because the cached disks are not accessible.
FC_BY_USED	The number of megabytes used in flash cache. Tracks the valid bytes in flash cache. For some workloads, like OLTP, the FC_BY_USED can be much less than the value of FC_BY_ALLOCATED because an OLTP write might only use 8K out of a 64K cacheline.
FC_BYKEEP_DIRTY	The number of megabytes allocated for unflushed keep objects in flash cache.
FC_BYKEEP_OLTP	The number of megabytes for OLTP keep objects in flash cache.
FC_BYKEEP_OVERWR	The number of megabytes pushed out of flash cache because of the space limit for keep objects.
FC_BYKEEP_OVERWR_SEC	The number of megabytes per second pushed out of flash cache because of space limit for keep objects.
FC_BYKEEP_USED	The number of megabytes used for keep objects on flash cache.
FC_COL_BY_USED	The number of megabytes used in columnar flash cache.
FC_COL_BYKEEP_USED	The number of megabytes used for keep objects in columnar flash cache.
FC_COL_IO_BY_R	The number of megabytes that were read from columnar flash cache.
FC_COL_IO_BY_R_ELIGIBLE	The number of megabytes eligible to read from columnar flash cache.
FC_COL_IO_BY_R_ELIGIBLE_SEC	The number of megabytes per second eligible to read from columnar flash cache.
FC_COL_IO_BY_R_SEC	The number of megabytes per second that were read from columnar flash cache.
FC_COL_IO_BY_SAVED	The number of megabytes saved by reads from columnar flash cache.
FC_COL_IO_BY_SAVED_SEC	The number of megabytes saved per second by reads from columnar flash cache.
FC_COL_IO_BY_W_POPULATE	The number of megabytes that are population writes into columnar flash cache due to read misses.
FC_COL_IO_BY_W_POPULATE_SEC	The number of megabytes per second that are population writes into columnar flash cache due to read misses.

Table 6-3 (Cont.) Flash Cache Metrics and Descriptions

Metric	Description
FC_COL_IO_BYKEEP_R	The number of megabytes read from columnar flash cache for keep objects.
FC_COL_IO_BYKEEP_R_SEC	The number of megabytes read per second from columnar flash cache for keep objects.
FC_COL_IO_RQ_R	The number of requests that were read from columnar flash cache.
FC_COL_IO_RQ_R_ELIGIBLE	The number of read requests eligible for columnar flash cache.
FC_COL_IO_RQ_R_ELIGIBLE_SEC	The number of read requests per second eligible for columnar flash cache.
FC_COL_IO_RQ_R_SEC	The number of requests per second that were read from columnar flash cache.
FC_COL_IO_RQ_W_POPULATE	The number of requests that are population writers into columnar flash cache due to read misses.
FC_COL_IO_RQ_W_POPULATE_SEC	The number of requests per second that are population writes into columnar flash cache due to read misses.
FC_COL_IO_RQKEEP_R	The number of requests read for keep objects from columnar flash cache.
FC_COL_IO_RQKEEP_R_SEC	The number of requests read per second for keep objects from columnar flash cache.
FC_IO_BY_ALLOCATED_OLTP	The number of megabytes allocated for OLTP data in flash cache.
FC_IO_BY_DISK_WRITE	The number of megabytes written from flash cache to hard disks.
FC_IO_BY_DISK_WRITE_SEC	The number of megabytes per second written from flash cache to hard disks.
FC_IO_BY_R	The number of megabytes of data read from flash cache for small data requests (OLTP). Combine this value with FC_IO_BY_R_DW to get the total flash cache IO reads.
FC_IO_BY_R_ACTIVE_SECONDARY	The number of megabytes for active secondary reads satisfied from flash cache.
FC_IO_BY_R_ACTIVE_SECONDARY_MISS	The number of megabytes for active secondary reads not satisfied from flash cache.
FC_IO_BY_R_ACTIVE_SECONDARY_MISS_SEC	The number of megabytes per second for active secondary reads not satisfied from flash cache.
FC_IO_BY_R_ACTIVE_SECONDARY_SEC	The number of megabytes per second for active secondary reads satisfied from flash cache.
FC_IO_BY_R_DISK_WRITER	The number of megabytes read from flash cache by disk writer.

Table 6-3 (Cont.) Flash Cache Metrics and Descriptions

Metric	Description
FC_IO_BY_R_DISK_WRITER_SEC	The number of megabytes per second read from flash cache by disk writer.
FC_IO_BY_R_DW	The number of megabytes of data read from flash cache for large data requests (DW). Combine this value with FC_IO_BY_R to get the total flash cache reads hits.
FC_IO_BY_R_MISS	The number of megabytes of data read from disks for small data requests (OLTP) because some of the requested data was not in flash cache. Combine this value with FC_IO_BY_R_MISS_DW to get the total flash cache read misses.
FC_IO_BY_R_MISS_DW	The number of megabytes of data read from disks for large data requests (DW) because some of the requested data was not in flash cache. Combine this value with FC_IO_BY_R_MISS to get the total flash cache read misses.
FC_IO_BY_R_MISS_SEC	The number of megabytes read from disks per second because not all requested data was in flash cache.
FC_IO_BY_R_SEC	The number of megabytes read per second from flash cache.
FC_IO_BY_R_SKIP	The number of megabytes read from disks for I/O requests that bypass flash cache. Read I/O requests that bypass flash cache go directly to disks. These requests do not populate flash cache after reading the requested data. For example: reads for database objects with CELL_FLASH_CACHE=NONE, or Smart Scan I/Os for database objects with CELL_FLASH_CACHE=DEFAULT.
FC_IO_BY_R_SKIP_FC_THROTTLE	The number of megabytes read from disk for I/O requests that bypass flash cache due to heavy load on flash cache.
FC_IO_BY_R_SKIP_FC_THROTTLE_SEC	The number of megabytes read per second from disk for I/O requests that bypass flash cache due to heavy load on flash cache.
FC_IO_BY_R_SKIP_LG	The number of megabytes read from disk for I/O requests that bypass flash cache due to the large I/O size.
FC_IO_BY_R_SKIP_LG_SEC	The number of megabytes read per second from disk for I/O requests that bypass flash cache due to the large I/O size.
FC_IO_BY_R_SKIP_NCMIRROR	The number of megabytes read from disk for I/O requests that bypass flash cache as the I/O is on non-primary, non-active secondary mirror.

Table 6-3 (Cont.) Flash Cache Metrics and Descriptions

Metric	Description
FC_IO_BY_R_SKIP_SEC	The number of megabytes read from disks per second for I/O requests that bypass flash cache. Read I/O requests that bypass flash cache go directly to disks. These requests do not populate flash cache after reading the requested data. For example: reads for database objects with CELL_FLASH_CACHE=NONE, or Smart Scan I/Os for database objects with CELL_FLASH_CACHE=DEFAULT.
FC_IO_BY_W	The number of megabytes written to flash cache.
FC_IO_BY_W_DISK_WRITER	The number of megabytes written to hard disks by disk writer.
FC_IO_BY_W_DISK_WRITER_SEC	The number of megabytes per second written to hard disks by disk writer.
FC_IO_BY_W_FIRST	The number of megabytes that are first writes into flash cache.
FC_IO_BY_W_FIRST_SEC	The number of megabytes per second for first writes into flash cache.
FC_IO_BY_W_METADATA	The number of megabytes that are flash cache metadata writes
FC_IO_BY_W_METADATA_SEC	The number of megabytes per second that are flash cache metadata writes
FC_IO_BY_W_OVERWRITE	The number of megabytes that are overwrites into flash cache.
FC_IO_BY_W_OVERWRITE_SEC	The number of megabytes per second that are overwrites into flash cache.
FC_IO_BY_W_POPULATE	The number of megabytes for population writes into flash cache due to read misses.
FC_IO_BY_W_POPULATE_SEC	The number of megabytes per second that are population writes into flash cache due to read misses into flash cache.
FC_IO_BY_W_SEC	The number of megabytes per second written to flash cache.
FC_IO_BY_W_SKIP	The number of megabytes written to disk for I/O requests that bypass flash cache.
FC_IO_BY_W_SKIP_FC_THROTTLE	The number of megabytes written to disk for I/O requests that bypass flash cache due to heavy load on flash cache.
FC_IO_BY_W_SKIP_FC_THROTTLE_SEC	The number of megabytes written per second to disk for I/O requests that bypass flash cache due to heavy load on flash cache.
FC_IO_BY_W_SKIP_LG	The number of megabytes written to disk for I/O requests that bypass flash cache due to the large I/O size.

Table 6-3 (Cont.) Flash Cache Metrics and Descriptions

Metric	Description
FC_IO_BY_W_SKIP_LG_SEC	The number of megabytes written per second to disk for I/O requests that bypass flash cache due to the large I/O size.
FC_IO_BY_W_SKIP_NCMIRROR	The number of megabytes written to disk for I/O requests that bypass flash cache as the I/O is on non-primary, non-active secondary mirror.
FC_IO_BY_W_SKIP_SEC	The number of megabytes written to disk per second for I/O requests that bypass flash cache.
FC_IO_BYKEEP_R	The number of megabytes read from flash cache for keep objects.
FC_IO_BYKEEP_R_SEC	The number of megabytes read per second from flash cache for keep objects
FC_IO_BYKEEP_W	The number of megabytes written to flash cache for keep objects.
FC_IO_BYKEEP_W_SEC	The number of megabytes per second written to flash cache for keep objects.
FC_IO_ERRS	The number of I/O errors on flash cache.
FC_IO_RQ_DISK_WRITE	The number of requests written from flash cache to hard disks.
FC_IO_RQ_DISK_WRITE_SEC	The number of requests per second for flash cache writing data to hard disks.
FC_IO_RQ_R	The number of read I/O requests satisfied from flash cache for small data requests (OLTP). Combine this value with FC_IO_RQ_R_DW to get the total read hit requests for the flash cache.
FC_IO_RQ_R_ACTIVE_SECONDARY	The number of requests for active secondary reads satisfied from flash cache.
FC_IO_RQ_R_ACTIVE_SECONDARY_MISS	The number of requests for active secondary reads not satisfied from flash cache.
FC_IO_RQ_R_ACTIVE_SECONDARY_MISS_SEC	The number of requests per second for active secondary reads not satisfied from flash cache.
FC_IO_RQ_R_ACTIVE_SECONDARY_SEC	The number of requests per second for active secondary reads satisfied from flash cache.
FC_IO_RQ_R_DISK_WRITER	The number of requests read from flash cache by disk writer.
FC_IO_RQ_R_DISK_WRITER_SEC	The number of requests per second read from flash cache by disk writer.
FC_IO_RQ_R_DW	The number of read I/O requests satisfied from the flash cache for large data requests (DW). Combine this value with FC_IO_RQ_R to get the total read hit requests for the flash cache.

Table 6-3 (Cont.) Flash Cache Metrics and Descriptions

Metric	Description
`FC_IO_RQ_R_MISS`	The number of read I/O requests that did not find all data in flash cache for small data requests (OLTP). Combine this value with `FC_IO_RQ_R_MISS_DW` to get the total read cache request misses for the flash cache.
`FC_IO_RQ_R_MISS_DW`	The number of read I/O requests that did not find all data in flash cache for large data requests (DW). Combine this value with `FC_IO_RQ_R_MISS` to get the total read cache request misses for the flash cache.
`FC_IO_RQ_R_MISS_SEC`	The number of read I/O requests per second which did not find all data in flash cache.
`FC_IO_RQ_R_SEC`	The number of read I/O requests satisfied per second from flash cache.
`FC_IO_RQ_R_SKIP`	The number of read I/O requests that bypass flash cache. Read I/O requests that bypass flash cache go directly to disks. These requests do not populate flash cache after reading the requested data. For example: reads for database objects with `CELL_FLASH_CACHE=NONE`, or Smart Scan I/Os for database objects with `CELL_FLASH_CACHE=DEFAULT`.
`FC_IO_RQ_R_SKIP_FC_THROTTLE`	The number of requests read from disk that bypass flash cache due to heavy load on flash cache.
`FC_IO_RQ_R_SKIP_FC_THROTTLE_SEC`	The number of requests per second read from disk that bypassed flash cache due to heavy load on flash cache.
`FC_IO_RQ_R_SKIP_LG`	The number of read I/O requests that bypass flash cache due to the large I/O size.
`FC_IO_RQ_R_SKIP_LG_SEC`	The number of read I/O requests per second that bypass flash cache due to the large I/O size.
`FC_IO_RQ_R_SKIP_NCMIRROR`	The number of requests read from disk that bypass flash cache as the I/O is on non-primary non-active secondary mirror.
`FC_IO_RQ_R_SKIP_SEC`	The number of read I/O requests per second that bypass flash cache. Read I/O requests that bypass flash cache go directly to disks. These requests do not populate flash cache after reading the requested data. For example: reads for database objects with `CELL_FLASH_CACHE=NONE`, or Smart Scan I/Os for database objects with `CELL_FLASH_CACHE=DEFAULT`.

Table 6-3 (Cont.) Flash Cache Metrics and Descriptions

Metric	Description
FC_IO_RQ_REPLACEMENT_ATTEMPTED	The number of requests attempted to find space in the flash cache.
FC_IO_RQ_REPLACEMENT_DW_FAILED	Number of times that client DW IOs failed to find a replacement buffer
FC_IO_RQ_REPLACEMENT_DW_SUCCEEDED	Number of times that client DW IOs succeeded in finding a replacement buffer.
FC_IO_RQ_REPLACEMENT_OLTP_FAILED	Number of times that client OLTP IOs failed to find a replacement buffer.
FC_IO_RQ_REPLACEMENT_OLTP_SUCCEEDED	Number of times that client OLTP IOs succeeded in finding a replacement buffer.
FC_IO_RQ_REPLACEMENT_FAILED	The number of requests that failed to find space in the flash cache.
FC_IO_RQ_W	The number of I/O requests which resulted in flash cache being populated with data.
FC_IO_RQ_W_DISK_WRITER	The number of requests written to hard disks by disk writer.
FC_IO_RQ_W_DISK_WRITER_SEC	The number of requests per second written to hard disks by disk writer.
FC_IO_RQ_W_FIRST	The number of requests that are first writes into flash cache.
FC_IO_RQ_W_FIRST_SEC	The number of requests per second that are first writes into flash cache.
FC_IO_RQ_W_METADATA	The number of requests that are flash cache metadata writes
FC_IO_RQ_W_METADATA_SEC	The number of requests per second that are flash cache metadata writes
FC_IO_RQ_W_OVERWRITE	The number of requests that are overwrites into flash cache.
FC_IO_RQ_W_OVERWRITE_SEC	The number of requests per second that are overwrites into flash cache.
FC_IO_RQ_W_POPULATE	The number of requests that are population writes into flash cache due to read misses.
FC_IO_RQ_W_POPULATE_SEC	The number of requests per second that are population writes into flash cache due to read misses.
FC_IO_RQ_W_SEC	The number of I/O requests per second which resulted in flash cache being populated with data.
FC_IO_RQ_W_SKIP	The number of write I/O requests that bypass flash cache.
FC_IO_RQ_W_SKIP_FC_THROTTLE	The number of requests written to disk that bypass flash cache due to heavy load on flash cache.

Table 6-3 (Cont.) Flash Cache Metrics and Descriptions

Metric	Description
FC_IO_RQ_W_SKIP_FC_THROTTLE_SEC	The number of requests written to disk per second that bypass flash cache due to heavy load on flash cache.
FC_IO_RQ_W_SKIP_LG	The number of requests written to disk that bypass flash cache due to the large I/O size.
FC_IO_RQ_W_SKIP_LG_SEC	The number of requests written to disk per second that bypass flash cache due to the large I/O size.
FC_IO_RQ_W_SKIP_NCMIRROR	The number of requests written to disk that bypass flash cache as the I/O is on non-primary, non-active secondary mirror.
FC_IO_RQ_W_SKIP_SEC	The number of write I/O requests per second that bypass flash cache.
FC_IO_RQKEEP_R	The number of read I/O requests for keep objects from flash cache.
FC_IO_RQKEEP_R_MISS	The number of read I/O requests for keep objects which did not find all data in flash cache.
FC_IO_RQKEEP_R_MISS_SEC	The number of read I/O requests per second for keep objects which did not find all data in flash cache.
FC_IO_RQKEEP_R_SEC	The number of read I/O requests per second for keep objects satisfied from flash cache.
FC_IO_RQKEEP_R_SKIP	The number of read I/O requests for keep objects that bypass flash cache. Read I/O requests that bypass flash cache go directly to disks. These requests do not populate flash cache after reading the requested data. For example: reads for database objects with CELL_FLASH_CACHE=NONE, or Smart Scan I/Os for database objects with CELL_FLASH_CACHE=DEFAULT.
FC_IO_RQKEEP_R_SKIP_SEC	The rate which is the number of read I/O requests per second for keep objects that bypass flash cache. Read I/O requests that bypass flash cache go directly to disks. These requests do not populate flash cache after reading the requested data. For example: reads for database objects with CELL_FLASH_CACHE=NONE, or Smart Scan I/Os for database objects with CELL_FLASH_CACHE=DEFAULT.
FC_IO_RQKEEP_W	The number of I/O requests for keep objects which resulted in flash cache being populated with data.

Table 6-3 (Cont.) Flash Cache Metrics and Descriptions

Metric	Description
FC_IO_RQKEEP_W_SEC	The number of I/O requests per second for keep objects which resulted in flash cache being populated with data.
FC_PREFETCH_HIT	The number of prefetch hits in flash cache. A hit means a page is prefetched and later used by a client for read or write. Once a prefetched page is hit, it is no longer marked as prefetched.
FC_PREFETCH_MISS	The number of prefetch misses in flash cache. A miss means a page is prefetched but never used by a client before the page is freed or reused.
FC_PREFETCH_RQ	The number of prefetch requests in flash cache.
FC_PREFETCH_SKIP	The number of prefetch requests skipped in flash cache. A prefetch request can be skipped if a prefetch disk IO is not issued due to error, throttling, or no buffer.
FC_PREFETCH_USED	The number of megabytes in flash cache holding prefetched pages that have not been used by client yet.

The following tables and views also have metrics for flash cache:

- Table 6-2
- Table 6-5
- Table 6-8
- Table 6-10
- Table 6-14
- V$SEGMENT_STATISTICS
- V$SQL

Related Topics

- Using V$SEGMENT_STATISTICS with Oracle Exadata System Software
  The V$SEGMENT_STATISTICS view lists statistics on a per segment basis.
- Using V$SQL with Oracle Exadata System Software
  The V$SQL view lists statistics on shared SQL areas and contains one row for each child of the original SQL text entered.

6.2.5 Monitoring Exadata Smart Flash Log with Metrics

Exadata Smart Flash Log metrics provide information about flash log utilization, such as the number of megabytes written per second.

To display Exadata Smart Flash Log metrics, use an object type attribute equal to FLASHLOG with the METRICCURRENT, METRICDEFINITION, and METRICHISTORY objects.

You can set the size of Exadata Smart Flash Log. If the size is set too small, then the FL_IO_W_SKIP_LARGE and FL_IO_W_SKIP_BUSY_MIN metrics can have high values.

The following table lists the metric names and descriptions for Exadata Smart Flash Log metrics. The flash log for the metric is specified by the metricObjectName attribute of the METRICCURRENT and METRICHISTORY objects.

Exadata Smart Flash Log Metrics and Descriptions

- FL_ACTUAL_OUTLIERS — The number of redo writes written to flash and disk that exceeded the outlier threshold
- FL_BY_KEEP — The number of redo data bytes saved on flash due to disk I/O errors
- FL_DISK_FIRST — The number of redo writes first written to disk
- FL_DISK_IO_ERRS — The number of disk I/O errors encountered by Exadata Smart Flash Log
- FL_EFFICIENCY_PERCENTAGE — The efficiency of Exadata Smart Flash Log expressed as a percentage
- FL_EFFICIENCY_PERCENTAGE_HOUR — The efficiency of Exadata Smart Flash Log over the past hour expressed as a percentage
- FL_FLASH_FIRST — The number of redo writes first written to flash
- FL_FLASH_IO_ERRS — The number of flash I/O errors encountered by Exadata Smart Flash Log
- FL_FLASH_ONLY_OUTLIERS — The number of redo writes written to flash that exceeded the outlier threshold.
- FL_IO_DB_BY_W — The number of megabytes written to hard disk by Exadata Smart Flash Log
- FL_IO_DB_BY_W_SEC — The number of megabytes written per second were written to hard disk by Exadata Smart Flash Log
- FL_IO_FL_BY_W — The number of megabytes written to flash by Exadata Smart Flash Log
- FL_IO_FL_BY_W_SEC — The number of megabytes written per second were written to flash by Exadata Smart Flash Log
- FL_IO_TM_W —

  Cumulative redo log write latency. It includes latency for requests not handled by Exadata Smart Flash Log.
- FL_IO_TM_W_RQ — Average redo log write latency. It includes write I/O latency only
- FL_IO_W — The number of writes serviced by Exadata Smart Flash Log
- FL_IO_W_SKIP_BUSY — The number of redo writes that bypassed Exadata Smart Flash Log because the flash log disks contained too much active data, which is data that has not yet been written to disk, or has been checkpointed by flash log.

  There are several factors that can cause redo writes to bypass flash log:

  - Flash disks are not performing as fast as they should, that is, write latencies are high. You can verify this by checking relevant flash disk performance metrics and also by checking FL_FLASH_ONLY_OUTLIERS. If this metric value is high, then it would indicate a flash disk performance issue.

  - Hard disks containing log file data are not performing as fast as they should. You can verify this by checking relevant hard disk performance metrics and also by checking `FL_PREVENTED_OUTLIERS`. If this metric value is high, then it would indicate a hard disk performance issue.
  - Redo log data is being generated at a high volume and rate, and Exadata Smart Flash Logging cannot keep up. You can verify this by checking `FL_IO_DB_BY_W_SEC`. If this metric value is several megabytes per second, then the redo log generation rate may be higher than what flash log can handle. To address this issue, you can make the flash log size larger, but this would require re-configuring the size of the Exadata Smart Flash Cache to make it smaller, so that the corresponding space can be allocated to the flash log.

  Generally, `FL_IO_W_SKIP_BUSY` is not as important as other factors, such as "log file parallel write" wait times, as indicated in Automatic Workload Repository (AWR), and `FL_ACTUAL_OUTLIERS`.

  If `log file parallel write` wait times are high (more than 0.5 seconds) or `FL_ACTUAL_OUTLIERS` is substantial, then this means that the flash log feature is not performing efficiently.

- `FL_IO_W_SKIP_BUSY_MIN` — The number of redo writes during the last minute that could not be serviced by Exadata Smart Flash Log because too much data had not yet been written to disk.
- `FL_IO_W_SKIP_DISABLED_GD` — The number of times Exadata Smart Flash Log was disabled for the underlying grid disk, usually caused by recent write errors on that disk
- `FL_IO_W_SKIP_IORM_LIMIT` — The number of times IORM limit was reached for the disk containing the redo log
- `FL_IO_W_SKIP_IORM_PLAN` — The number of times the flash log was disabled in the IORM plan
- `FL_IO_W_SKIP_LARGE` — The number of large redo writes that could not be serviced by Exadata Smart Flash Log because the size of the data was larger than the amount of available space on any flash disk.
- `FL_IO_W_SKIP_LOG_ON_FLASH` — The number of times the redo log resided on flash (for systems that are not Extreme Flash systems)
- `FL_IO_W_SKIP_NO_BUFFER` — The metric shows the number of redo writes that could not be serviced by Exadata Smart Flash Log due to a lack of available buffers.
- `FL_IO_W_SKIP_NO_FL_DISKS` — The number of times there were no active flash log disks because the disks were inactive or unhealthy
- `FL_PREVENTED_OUTLIERS` — The number of redo writes written to disk that exceeded the outlier threshold. These writes would have been outliers if not for Exadata Smart Flash Log.
- `FL_RQ_TM_W` — Cumulative redo log write request latency. It includes networking and other overhead.

  To get the latency overhead due to factors such as network and processing, you can use (`FL_RQ_TM_W - FL_IO_TM_W`).
- `FL_RQ_TM_W_RQ` — Average redo log write request latency.

- `FL_RQ_W` — Total number of redo log write requests. It includes requests not handled by Exadata Smart Flash Log.

  To get the number of redo log write requests not handled by Exadata Smart Flash Log, you can use (`FL_RQ_W - FL_IO_W`).
- `FL_SKIP_OUTLIERS` — The number of outliers when redo log write skips use of the flash log. The use of the flash log can be skipped because of `FL_IO_W_SKIP_DISABLED_GD`, `FL_IO_W_SKIP_IORM_LIMIT`, `FL_IO_W_SKIP_IORM_PLAN`, `FL_IO_W_SKIP_LOG_ON_FLASH`, or `FL_IO_W_SKIP_NO_FL_DISKS`.

> **Note:**
>
> - An outlier is a redo log write that exceeds 0.5 seconds.
> - Often, there are more redo log write operations that complete to disk first before flash. The value of `FL_DISK_FIRST` is greater than the value of `FL_FLASH_FIRST`. This does not mean that the Exadata Smart Flash Logging feature is ineffective or unnecessary.

Related Topics

- Smart Flash Technology
  The Exadata Smart Flash Cache feature of the Oracle Exadata System Software intelligently caches database objects in flash memory, replacing slow, mechanical I/O operations to disk with very rapid flash memory operations.

6.2.6 Monitoring PMEM Cache Metrics

PMEM cache metrics provide information about the utilization of the PMEM cache.

To display PMEM cache metrics, use an object type attribute equal to `PMEMCACHE` with the `METRICCURRENT`, `METRICDEFINITION`, and `METRICHISTORY` objects.

The following table lists the metric names and descriptions for PMEM cache metrics. The PMEM cache for the metric is specified by the `metricObjectName` attribute of the `METRICCURRENT` and `METRICHISTORY` objects.

Table 6-4 PMEM Cache Metrics and Descriptions

Metric	Description
`DB_PC_BY_ALLOCATED`	Number of megabytes allocated in PMEM cache for this database.
`PC_BY_ALLOCATED`	The number of megabytes allocated in PMEM cache. Tracks how many cachelines are used in the PMEM cache. If the value is close to the PMEM cache size, then the PMEM cache is fully populated.
`PDB_PC_BY_ALLOCATED`	Number of megabytes allocated in PMEM cache for this pluggable database (PDB).

The following tables and views also have metrics for PMEM cache:

- Table 6-2
- Table 6-5
- Table 6-8
- Table 6-10
- Table 6-14
- V$SEGMENT_STATISTICS
- V$SQL

Related Topics

- Using V$SEGMENT_STATISTICS with Oracle Exadata System Software
  The V$SEGMENT_STATISTICS view lists statistics on a per segment basis.
- Using V$SQL with Oracle Exadata System Software
  The V$SQL view lists statistics on shared SQL areas and contains one row for each child of the original SQL text entered.

6.2.7 Monitoring Grid Disk Metrics

Grid disk metrics provide information about the I/O load for grid disks, such as the number of large blocks read from a grid disk.

To display grid disk metrics, use an object type attribute equal to GRIDDISK with the METRICCURRENT, METRICDEFINITION, and METRICHISTORY objects.

The following tale lists the metric names and descriptions for grid disk metrics. The grid disk for the metric is specified by the metricObjectName attribute of the METRICCURRENT and METRICHISTORY objects. For cumulative metrics, the metric value for a specific time period can be determined by subtracting values from different collectionTime periods. For rate metrics, the time period for the metric value is over the previous minute.

Table 6-5 Grid Disk Metrics and Descriptions

Metric	Description
GD_BY_FC_DIRTY	The number of megabytes cached in flash cache that are not synchronized to the grid disk.
GD_IO_BY_R_LG	The cumulative number of megabytes read in large blocks from a grid disk.
GD_IO_BY_R_LG_SEC	The rate which is the number of megabytes read in large blocks per second from a grid disk.
GD_IO_BY_R_SCRUB	The number of megabytes read from a grid disk by the scrubbing job.
GD_IO_BY_R_SCRUB_SEC	The number of megabytes read per second from a grid disk by the scrubbing job.
GD_IO_BY_R_SM	The cumulative number of megabytes read in small blocks from a grid disk.
GD_IO_BY_R_SM_SEC	The rate which is the number of megabytes read in small blocks per second from a grid disk.

Table 6-5 (Cont.) Grid Disk Metrics and Descriptions

Metric	Description
GD_IO_BY_W_LG	The cumulative number of megabytes written in large blocks on a grid disk.
GD_IO_BY_W_LG_SEC	The rate which is the number of megabytes written in large blocks per second on a grid disk.
GD_IO_BY_W_SM	The cumulative number of megabytes written in small blocks on a grid disk.
GD_IO_BY_W_SM_SEC	The rate which is the number of megabytes written in small blocks per second on a grid disk.
GD_IO_ERRS	The cumulative number of I/O errors on a grid disk.
GD_IO_ERRS_MIN	The rate of I/O errors on a grid disk per minute.
GD_IO_ERRS_SCRUB	The number of bad 1 MB blocks detected during a scrubbing job on a grid disk.
GD_IO_RQ_R_LG	The cumulative number of requests to read large blocks from a grid disk.
GD_IO_RQ_R_LG_SEC	The rate which is the number of requests to read large blocks per second from a grid disk.
GD_IO_RQ_R_SCRUB	The number of requests to read from a grid disk by the scrubbing job.
GD_IO_RQ_R_SCRUB_SEC	The number of requests to read per second from a grid disk by the scrubbing job.
GD_IO_RQ_R_SM	The cumulative number of requests to read small blocks from a grid disk.
GD_IO_RQ_R_SM_SEC	The rate which is the number of requests to read small blocks per second from a grid disk.
GD_IO_RQ_W_LG	The cumulative number of requests to write large blocks to a grid disk.
GD_IO_RQ_W_LG_SEC	The rate which is the number of requests to write large blocks per second to a grid disk.
GD_IO_RQ_W_SM	The cumulative number of requests to write small blocks to a grid disk.
GD_IO_RQ_W_SM_SEC	The rate which is the number of requests to write small blocks per second to a grid disk.
GD_SP_BY_ALLOCATED	The allocated physical space for grid disk in bytes.
GD_SP_IO_BY_PARTIAL	The bytes returned by partial I/Os.
GD_SP_IO_BY_PARTIAL_SEC	The bytes returned by partial I/Os per second.
GD_SP_IO_BY_REDIRECTED	The sparse bytes redirected to original data block.
GD_SP_IO_BY_REDIRECTED_SEC	The sparse bytes redirected to original data block per second.
GD_SP_IO_RQ_PARTIAL	The number of I/O requests that returned partial data.
GD_SP_IO_RQ_PARTIAL_SEC	The number of I/O requests that returned partial data per second.

Table 6-5 (Cont.) Grid Disk Metrics and Descriptions

Metric	Description
GD_SP_PRCT_ALLOCATED	The allocated physical space for grid disk by percentage. A warning alert is generated when space usage is more than 95%. A critical alert is generated when space usage reaches 99%. If an alert occurs, then either increase the size of the sparse grid disk, or drop some snapshot databases to free space.

6.2.8 Monitoring Host Interconnection Metrics

Host interconnection metrics provide information about the I/O transmission for hosts that access cell storage.

To display host interconnection metrics, use an object type attribute equal to HOST_INTERCONNECT with the METRICCURRENT, METRICDEFINITION, and METRICHISTORY objects.

The following table lists the metric names and descriptions for host interconnection metrics. The host interconnection for the metric is specified by the metricObjectName attribute of the METRICCURRENT and METRICHISTORY objects. For cumulative metrics, the metric value for a specific time period can be determined by subtracting values from different collectionTime periods. For rate metrics, the time period for the metric value is over the previous minute.

Table 6-6 Host Interconnection Metrics and Descriptions

Metric	Description
N_MB_DROP	The cumulative number of megabytes dropped during transmission to a particular host.
N_MB_DROP_SEC	The rate which is the number of megabytes dropped per second during transmission to a particular host.
N_MB_RDMA_DROP	The cumulative number of megabytes dropped during remote direct memory access (RDMA) transmission to a particular host.
N_MB_RDMA_DROP_SEC	The rate which is the number of megabytes dropped per second during RDMA transmission to a particular host.
N_MB_RECEIVED	The cumulative number of megabytes received from a particular host.
N_MB_RECEIVED_SEC	The rate which is the number of megabytes received per second from a particular host.
N_MB_RESENT	The cumulative number of megabytes retransmitted to a particular host.
N_MB_RESENT_SEC	The rate which is the number of megabytes retransmitted per second to a particular host.
N_MB_SENT	The cumulative number of megabytes transmitted to a particular host.
N_MB_SENT_SEC	The rate which is the number of megabytes transmitted per second to a particular host.

Table 6-6 (Cont.) Host Interconnection Metrics and Descriptions

Metric	Description
`N_RDMA_RETRY_TM`	The cumulative latency of the retry action during RDMA transmission to a particular host.

6.2.9 Monitoring Smart I/O Metrics

Smart I/O metrics provide information about smart I/O performance. Smart I/O is the I/O processing, such as smart scan, and smart incremental backup, performed by the Oracle Exadata Storage Server offload functionality.

To display smart I/O metrics, use the object type attribute equal to `SMARTIO` with the `METRICCURRENT`, `METRICDEFINITION`, and `METRICHISTORY` objects.

The following table lists the metric names and descriptions for the smart I/O metrics. The smart I/O for the metric is specified by the `metricObjectName` attribute of the `METRICCURRENT` and `METRICHISTORY` objects. For cumulative metrics, the metric value for a specific time period can be determined by subtracting values from different `collectionTime` periods.

Table 6-7 Smart I/O Metrics and Descriptions

Metric	Description
`SIO_IO_EL_OF`	The cumulative number of megabytes eligible for offload by smart I/O.
`SIO_IO_EL_OF_SEC`	The number of megabytes per second eligible for offload by smart I/O.
`SIO_IO_OF_RE`	The cumulative number of interconnect megabytes returned by smart I/O.
`SIO_IO_OF_RE_SEC`	The number of interconnect megabytes per second returned by smart I/O.
`SIO_IO_PA_TH`	The cumulative number of megabytes of pass-through I/O by smart I/O.
`SIO_IO_PA_TH_SEC`	The number of megabytes per second of pass-through I/O by smart I/O.
`SIO_IO_RD_FC`	The cumulative number of megabytes read from flash cache by smart I/O.
`SIO_IO_RD_FC_HD`	The cumulative number of megabytes read from both flash cache and hard disk by smart I/O.
`SIO_IO_RD_FC_HD_SEC`	The number of megabytes per second read from both flash cache and hard disk by smart I/O.
`SIO_IO_RD_FC_SEC`	The number of megabytes per second read from flash cache by smart I/O.
`SIO_IO_RD_HD`	The cumulative number of megabytes read from hard disk by smart I/O.
`SIO_IO_RD_HD_SEC`	The number of megabytes per second read from hard disk by smart I/O.

Table 6-7 (Cont.) Smart I/O Metrics and Descriptions

Metric	Description
`SIO_IO_RD_RQ_FC`	The cumulative number of read I/O requests from flash cache by smart I/O.
`SIO_IO_RD_RQ_FC_HD`	The cumulative number of read I/O requests from both flash cache and hard disk by smart I/O.
`SIO_IO_RD_RQ_FC_HD_SEC`	The number of read I/O requests per second from both flash cache and hard disk by smart I/O.
`SIO_IO_RD_RQ_FC_SEC`	The number of read I/O requests per second from flash cache by smart I/O.
`SIO_IO_RD_RQ_HD`	The cumulative number of megabytes read I/O requests from hard disk by smart I/O.
`SIO_IO_RD_RQ_HD_SEC`	The number of read I/O requests per second from hard disk by smart I/O.
`SIO_IO_RV_OF`	The cumulative number of megabytes sent to the database node to balance CPU by smart I/O.
`SIO_IO_RV_OF_SEC`	The number of megabytes per second sent to the database node to balance CPU by smart I/O.
`SIO_IO_SI_SV`	The cumulative number of megabytes saved by the storage index.
`SIO_IO_SI_SV_SEC`	The number of megabytes per second saved by the storage index.
`SIO_IO_WR_FC`	The cumulative number of megabytes of flash cache population writes by smart I/O.
`SIO_IO_WR_FC_SEC`	The number of megabytes per second of flash cache population writes by smart I/O.
`SIO_IO_WR_HD`	The cumulative number of megabytes written to hard disk by smart I/O.
`SIO_IO_WR_HD_SEC`	The number of megabytes per second written to hard disk by smart I/O.
`SIO_IO_WR_RQ_FC`	The cumulative number of I/O requests for flash cache population writes by smart I/O.
`SIO_IO_WR_RQ_FC_SEC`	The number of I/O requests per second for flash cache population writes by smart I/O.
`SIO_IO_WR_RQ_HD`	The cumulative number of write I/O requests to hard disk by smart I/O.
`SIO_IO_WR_RQ_HD_SEC`	The number of write I/O requests per second to hard disk by smart I/O.

6.3 Monitoring IORM with Metrics

I/O Resource Management (IORM) can be monitored using Oracle Exadata System Software metrics.

IORM uses the database unique name, not the database identifier, to collect statistics and display output. Starting with Oracle Exadata System Software release 19.1.0, if you configured ASM-scoped security for the Oracle Automatic Storage Management

(Oracle ASM) cluster used by the database, then the database name is prefixed with the Oracle ASM cluster name.

- Monitoring IORM with Category Metrics
  Category metrics provide information about the size of the I/O load from each category specified in the current IORM category plan.
- Monitoring IORM with Pluggable Database (PDB) Metrics
  PDB metrics provide information about the size of the I/O load from each PDB specified in the container database (CDB) plan.
- Monitoring IORM with Database Metrics
  Database metrics provide information about the size of the I/O load from each database specified in the inter-database plan.
- Monitoring IORM with Consumer Group Metrics
  Consumer group metrics provide information about the size of the I/O load from each consumer group specified in a database resource plan.
- Monitoring IORM Utilization
  You use metrics to monitor IORM utilization.
- Tuning Interdatabase Plans with Metrics
  The Oracle Exadata System Software metrics relevant to I/O Resource Management (IORM) can be used in multiple ways.

6.3.1 Monitoring IORM with Category Metrics

Category metrics provide information about the size of the I/O load from each category specified in the current IORM category plan.

The `objectType` attribute of the `METRICCURRENT`, `METRICDEFINITION`, and `METRICHISTORY` objects is equal to `IORM_CATEGORY` for category metrics.

The following table lists the metric name and description for category metrics. The category for the metric is specified by the `metricObjectName` attribute of the `METRICCURRENT` and `METRICHISTORY` objects. For cumulative metrics, the metric value for a specific time period can be determined by subtracting values from different `collectionTime` periods. For rate metrics, the time period for the metric value is over the previous minute. In the table descriptions, small means less than or equal to 128 KB, and large means more than 128 KB.

Table 6-8 Category Metrics and Descriptions

Metric	Description
`CT_FC_IO_BY_SEC`	The number of megabytes of I/O per second for this category to flash cache.
`CT_FC_IO_RQ`	The number of I/O requests issued by an IORM category to flash cache.
`CT_FC_IO_RQ_LG`	The number of large I/O requests issued by an IORM category to flash cache.
`CT_FC_IO_RQ_LG_SEC`	The number of large I/O requests issued by an IORM category to flash cache per second.
`CT_FC_IO_RQ_SEC`	The number of I/O requests issued by an IORM category to flash cache per second.

Table 6-8 (Cont.) Category Metrics and Descriptions

Metric	Description
`CT_FC_IO_RQ_SM`	The number of small I/O requests issued by an IORM category to flash cache.
`CT_FC_IO_RQ_SM_SEC`	The number of small I/O requests issued by an IORM category to flash cache per second.
`CT_FD_IO_BY_SEC`	The number of megabytes of I/O per second for this category to flash disks.
`CT_FD_IO_LOAD`	The average I/O load from this category for flash disks. For a description of I/O load, see "CD_IO_LOAD".
`CT_FD_IO_RQ_LG`	The number of large I/O requests issued by an IORM category to flash disks.
`CT_FD_IO_RQ_LG_SEC`	The number of large I/O requests issued by an IORM category to flash disks per second.
`CT_FD_IO_RQ_SM`	The number of small I/O requests issued by an IORM category to flash disks.
`CT_FD_IO_RQ_SM_SEC`	The number of small I/O requests issued by an IORM category to flash disks per second.
`CT_FD_IO_TM`	The cumulative latency of reading or writing blocks for this category from flash disks.
`CT_FD_IO_TM_LG`	The cumulative latency of reading or writing large blocks for this category from flash disks.
`CT_FD_IO_TM_LG_RQ`	The rate which is the average latency of reading or writing large blocks per request for this category from flash disks.
`CT_FD_IO_TM_RQ`	The rate which is the average latency of reading or writing blocks per request for this category from flash disks.
`CT_FD_IO_TM_SM`	The cumulative latency of reading or writing small blocks for this category from flash disks.
`CT_FD_IO_TM_SM_RQ`	The rate which is the average latency of reading or writing small blocks per request for this category from flash disks.
`CT_FD_IO_UTIL`	The percentage of flash resources utilized from this category.
`CT_FD_IO_UTIL_LG`	The percentage of flash resources utilized by large requests from this category.
`CT_FD_IO_UTIL_SM`	The percentage of flash resources utilized by small requests from this category.
`CT_FD_IO_WT_LG`	The IORM wait time for large I/O requests issued to flash disks by an IORM category.
`CT_FD_IO_WT_LG_RQ`	The average IORM wait time per request for large I/O requests issued to flash disks by an IORM category.
`CT_FD_IO_WT_SM`	The IORM wait time for small I/O requests issued to flash disks by an IORM category.
`CT_FD_IO_WT_SM_RQ`	The average IORM wait time per request for small I/O requests issued to flash disks by an IORM category.
`CT_IO_BY_SEC`	The number of megabytes of I/O per second for this category to hard disks.

Table 6-8 (Cont.) Category Metrics and Descriptions

Metric	Description
`CT_IO_LOAD`	The average I/O load from this category for hard disks. For a description of I/O load, see "CD_IO_LOAD".
`CT_IO_RQ_LG`	The cumulative number of large I/O requests issued by the category for hard disks. A large value indicates a heavy I/O workload from this category.
`CT_IO_RQ_LG_SEC`	This metric is derived from `CT_IO_RQ_LG`. It specifies the rate of large I/O requests issued by the category for hard disks. Its units are number of I/O requests per second. A large value indicates a heavy I/O workload from this category in the past minute.
`CT_IO_RQ_SM`	The cumulative number of small I/O requests issued by the category for hard disks. A large value indicates a heavy I/O workload from this category.
`CT_IO_RQ_SM_SEC`	This metric is derived from `CT_IO_RQ_SM`. It specifies the rate of small I/O requests issued by the category for hard disks. Its units are number of I/O requests per second. A large value indicates a heavy I/O workload from this category in the past minute.
`CT_IO_TM_LG`	The cumulative latency of reading or writing large blocks for this category from hard disks.
`CT_IO_TM_LG_RQ`	The rate which is the average latency of reading or writing large blocks per request for this category from hard disks.
`CT_IO_TM_SM`	The cumulative latency of reading or writing small blocks for this category from hard disks.
`CT_IO_TM_SM_RQ`	The rate which is the average latency of reading or writing small blocks per request for this category from hard disks.
`CT_IO_UTIL_LG`	The percentage of disk resources utilized by large requests from this category.
`CT_IO_UTIL_SM`	The percentage of disk resources utilized by small requests from this category.
`CT_IO_WT_LG`	The cumulative number of milliseconds that large I/O requests issued to hard disks by the category have waited to be scheduled by IORM. A large value indicates that the I/O workload from this category is exceeding the allocation specified for it in the category plan.
`CT_IO_WT_LG_RQ`	The average IORM wait time per request for large I/O requests issued to hard disks by an IORM category.
`CT_IO_WT_SM`	The cumulative number of milliseconds that small I/O requests issued to hard disks by the category have waited to be scheduled by IORM. A large value indicates that the I/O workload from this category is exceeding the allocation specified for it in the category plan.
`CT_IO_WT_SM_RQ`	The average IORM wait time per request for small I/O requests issued to hard disks by an IORM category.

All category cumulative metrics are reset to zero whenever a category, IORM, or any database resource plan is modified. To list the category metric history for an interdatabase plan, use the following CellCLI command:

```
CellCLI> LIST METRICHISTORY WHERE objectType = 'IORM_CATEGORY'
AND             -
         metricValue != 0 ATTRIBUTES name, metricObjectName,
metricValue,      -
         collectionTime
```

In addition, category metrics are provided for the following internally-generated and automatically-managed categories:

- _ASM_: Oracle ASM-related I/Os
- _ORACLE_BG_CATEGORY_: High-priority I/Os issued by Oracle Database background processes
- _ORACLE_MEDPRIBG_CATEGORY_: Medium-priority I/Os issued by Oracle Database background processes
- _ORACLE_LOWPRIBG_CATEGORY_: Low-priority I/Os issued by Oracle Database background processes

6.3.2 Monitoring IORM with Pluggable Database (PDB) Metrics

PDB metrics provide information about the size of the I/O load from each PDB specified in the container database (CDB) plan.

The `objectType` attribute of the `METRICCURRENT`, `METRICDEFINITION`, and `METRICHISTORY` objects is equal to `IORM_PLUGGABLE_DATABASE` for database metrics.

The following table lists the metric name and description for database metrics. Note the following:

- The PDB for the metric is specified by the `metricObjectName` attribute of the `METRICCURRENT` and `METRICHISTORY` objects. The PDB name is a concatenation of the CDB name with the PDB name.
- Starting with Oracle Exadata System Software release 19.1.0, if you configured ASM-scoped security for the Oracle Automatic Storage Management (Oracle ASM) cluster used by the database, then the database name is prefixed with the Oracle ASM cluster name.
- For cumulative metrics, the metric value for a specific time period can be determined by subtracting values from different `collectionTime` periods.
- For rate metrics, the time period for the metric value is over the previous minute.
- In the table descriptions, small means less than or equal to 128 KB, and large means more than 128 KB.

Table 6-9 Pluggable Database Metrics and Descriptions

Metric	Description
`PDB_FC_BY_ALLOCATED`	The number of megabytes allocated in flash cache for this PDB.

Table 6-9 (Cont.) Pluggable Database Metrics and Descriptions

Metric	Description
PDB_FC_IO_BY_SEC	The number of megabytes of I/O per second for the PDB to flash cache.
PDB_FC_IO_RQ	The number of I/O requests issued by this PDB to flash cache.
PDB_FC_IO_RQ_LG	The number of large I/O requests issued by this PDB to flash cache.
PDB_FC_IO_RQ_LG_SEC	The number of large I/O requests issued by this PDB to flash cache per second.
PDB_FC_IO_RQ_SEC	The number of I/O requests issued by this PDB to flash cache per second.
PDB_FC_IO_RQ_SM	The number of small I/O requests issued by this PDB to flash cache.
PDB_FC_IO_RQ_SM_SEC	The number of small IO requests issued by this PDB to flash cache per second.
PDB_FD_IO_BY_SEC	The number of megabytes of I/O per second for the PDB to flash disks.
PDB_FD_IO_LOAD	The average I/O load from this PDB for flash disks. For a description of I/O load, see CD_IO_LOAD.
PDB_FD_IO_RQ_LG	The number of large I/O requests issued by a PDB to flash disks.
PDB_FD_IO_RQ_LG_SEC	The number of large I/O requests issued by a PDB to flash disks per second.
PDB_FD_IO_RQ_SM	The number of small I/O requests issued by a PDB to flash disks.
PDB_FD_IO_RQ_SM_SEC	The number of small I/O requests issued by a PDB to flash disks per second.
PDB_FD_IO_TM	The cumulative latency of reading or writing blocks by this PDB from flash disks.
PDB_FD_IO_TM_LG	The cumulative latency of reading or writing large blocks by this PDB from flash disks.
PDB_FD_IO_TM_LG_RQ	The rate which is the average latency of reading or writing large blocks per request by this PDB from flash disks.
PDB_FD_IO_TM_RQ	The rate which is the average latency of reading or writing blocks per request by this PDB from flash disks.
PDB_FD_IO_TM_SM	The cumulative latency of reading or writing small blocks by this PDB from flash disks.
PDB_FD_IO_TM_SM_RQ	The rate which is the average latency of reading or writing small blocks per request by this PDB from flash disks.
PDB_FD_IO_UTIL	The percentage of flash resources utilized from this PDB.
PDB_FD_IO_UTIL_LG	The percentage of flash resources utilized by large requests from this PDB.
PDB_FD_IO_UTIL_SM	The percentage of flash resources utilized by small requests from this PDB.
PDB_FD_IO_WT_LG	The IORM wait time for large I/O requests issued to flash disks by this PDB.

Table 6-9 (Cont.) Pluggable Database Metrics and Descriptions

Metric	Description
PDB_FD_IO_WT_LG_RQ	The average IORM wait time per request for large I/O requests issued to flash disks by this PDB.
PDB_FD_IO_WT_SM	The IORM wait time for small I/O requests issued to flash disks by this PDB.
PDB_FD_IO_WT_SM_RQ	The average IORM wait time per request for small I/O requests issued to flash disks by this PDB.
PDB_IO_BY_SEC	The number of megabytes of I/O per second for the PDB to hard disks.
PDB_IO_LOAD	The average I/O load from this PDB for hard disks. For a description of I/O load, see "CD_IO_LOAD".
PDB_IO_RQ_LG	The cumulative number of large I/O requests issued by the PDB to hard disks. A large value indicates a heavy large I/O workload from this PDB.
PDB_IO_RQ_LG_SEC	The rate of large I/O requests issued by a PDB per second over the past minute to hard disks. A large value indicates a heavy, large I/O workload from this PDB in the past minute.
PDB_IO_RQ_SM	The cumulative number of small I/O requests issued by the PDB to hard disks. A large value indicates a heavy small I/O workload from this PDB.
PDB_IO_RQ_SM_SEC	The rate of small I/O requests issued by a PDB per second over the past minute to hard disks. A large value indicates a heavy, small I/O workload issued by this PDB in the past minute.
PDB_IO_TM_LG	The cumulative latency of reading or writing large blocks by this PDB from hard disks.
PDB_IO_TM_LG_RQ	The rate which is the average latency of reading or writing large blocks per request by this PDB from hard disks.
PDB_IO_TM_SM	The cumulative latency of reading or writing small blocks by this PDB from hard disks.
PDB_IO_TM_SM_RQ	The rate which is the average latency of reading or writing small blocks per request by this PDB from hard disks.
PDB_IO_UTIL_LG	The percentage of disk resources utilized by large requests from this PDB.
PDB_IO_UTIL_SM	The percentage of disk resources utilized by small requests from this PDB.
PDB_IO_WT_LG	The IORM wait time for large I/O requests issued to hard disks by this PDB.
PDB_IO_WT_LG_RQ	The average IORM wait time per request for large I/O requests issued to hard disks by this PDB.
PDB_IO_WT_SM	The IORM wait time for small I/O requests issued to hard disks by this PDB.
PDB_IO_WT_SM_RQ	The average IORM wait time per request for small I/O requests issued to hard disks by this PDB.
PDB_PC_BY_ALLOCATED	Number of megabytes allocated in PMEM cache for this PDB.

6.3.3 Monitoring IORM with Database Metrics

Database metrics provide information about the size of the I/O load from each database specified in the inter-database plan.

The `objectType` attribute of the `METRICCURRENT`, `METRICDEFINITION`, and `METRICHISTORY` objects is equal to `IORM_DATABASE` for database metrics.

The following table lists the metric name and description for database metrics. Note the following:

- The database for the metric is specified by the `metricObjectName` attribute of the `METRICCURRENT` and `METRICHISTORY` objects. Starting with Oracle Exadata System Software release 19.1.0, if you configured ASM-scoped security for the Oracle Automatic Storage Management (Oracle ASM) cluster used by the database, then the database name is prefixed with the Oracle ASM cluster name.
- For cumulative metrics, the metric value for a specific time period can be determined by subtracting values from different `collectionTime` periods.
- For rate metrics, the time period for the metric value is over the previous minute.
- In the metric descriptions, small means less than or equal to 128 KB, and large means more than 128 KB.

Table 6-10 Database Metrics and Descriptions

Metric	Description
`DB_FC_BY_ALLOCATED`	The number of megabytes allocated in flash cache for this database.
`DB_FC_IO_BY_SEC`	The number of megabytes of I/O per second for this database to flash cache.
`DB_FC_IO_RQ`	The number of I/O requests issued by a database to flash cache.
`DB_FC_IO_RQ_LG`	The number of large I/O requests issued by a database to flash cache.
`DB_FC_IO_RQ_LG_SEC`	The number of large I/O requests issued by a database to flash cache per second.
`DB_FC_IO_RQ_SEC`	The number of I/O requests issued by a database to flash cache per second.
`DB_FC_IO_RQ_SM`	The number of small I/O requests issued by a database to flash cache.
`DB_FC_IO_RQ_SM_SEC`	The number of small I/O requests issued by a database to flash cache per second.
`DB_FD_IO_BY_SEC`	The number of megabytes of I/O per second for this database to flash disks.
`DB_FD_IO_LOAD`	The average I/O load from this database for flash disks. For a description of I/O load, see CD_IO_LOAD.
`DB_FD_IO_RQ_LG`	The number of large I/O requests issued by a database to flash disks.
`DB_FD_IO_RQ_LG_SEC`	The number of large I/O requests issued by a database to flash disks per second.

Table 6-10 (Cont.) Database Metrics and Descriptions

Metric	Description
DB_FD_IO_RQ_SM	The number of small I/O requests issued by a database to flash disks.
DB_FD_IO_RQ_SM_SEC	The number of small I/O requests issued by a database to flash disks per second.
DB_FD_IO_TM	The cumulative latency of reading or writing blocks by a database from flash disks.
DB_FD_IO_TM_LG	The cumulative latency of reading or writing large blocks by a database from flash disks.
DB_FD_IO_TM_LG_RQ	The rate which is the average latency of reading or writing large blocks per request by a database from flash disks.
DB_FD_IO_TM_RQ	The rate which is the average latency of reading or writing blocks per request by a database from flash disks.
DB_FD_IO_TM_SM	The cumulative latency of reading or writing small blocks by a database from flash disks.
DB_FD_IO_TM_SM_RQ	The rate which is the average latency of reading or writing small blocks per request by a database from flash disks.
DB_FD_IO_UTIL	The percentage of flash resources utilized from this database.
DB_FD_IO_UTIL_LG	The percentage of flash resources utilized by large requests from this database.
DB_FD_IO_UTIL_SM	The percentage of flash resources utilized by small requests from this database.
DB_FD_IO_WT_LG	The IORM wait time for large I/O requests issued to flash disks by a database.
DB_FD_IO_WT_LG_RQ	The average IORM wait time per request for large I/O requests issued to flash disks by a database.
DB_FD_IO_WT_SM	The IORM wait time for small I/O requests issued to flash disks by a database.
DB_FD_IO_WT_SM_RQ	The average IORM wait time per request for small I/O requests issued to flash disks by a database.
DB_FL_IO_BY	The number of megabytes written to Exadata Smart Flash Log.
DB_FL_IO_BY_SEC	The number of megabytes written per second to Exadata Smart Flash Log.
DB_FL_IO_RQ	The number of I/O requests issued to Exadata Smart Flash Log.
DB_FL_IO_RQ_SEC	The number of I/O requests per second issued to Exadata Smart Flash Log.
DB_IO_BY_SEC	The number of megabytes of I/O per second for this database to hard disks.
DB_IO_LOAD	The average I/O load from this database for hard disks. For a description of I/O load, see CD_IO_LOAD.
DB_IO_RQ_LG	The cumulative number of large I/O requests issued to hard disks by the database. A large value indicates a heavy large I/O workload from this database.

Table 6-10 (Cont.) Database Metrics and Descriptions

Metric	Description
`DB_IO_RQ_LG_SEC`	The rate of large I/O requests issued to hard disks by a database per second over the past minute. A large value indicates a heavy large I/O workload from this database in the past minute.
`DB_IO_RQ_SM`	The cumulative number of small I/O requests issued to hard disks by the database. A large value indicates a heavy small I/O workload from this database.
`DB_IO_RQ_SM_SEC`	The rate of small I/O requests issued to hard disks by a database per second over the past minute. A large value indicates a heavy small I/O workload issued by this database in the past minute.
`DB_IO_TM_LG`	The cumulative latency of reading or writing large blocks by a database from hard disks.
`DB_IO_TM_LG_RQ`	The rate which is the average latency of reading or writing large blocks per request by a database from hard disks.
`DB_IO_TM_SM`	The cumulative latency of reading or writing small blocks by a database from hard disks.
`DB_IO_TM_SM_RQ`	The rate which is the average latency of reading or writing small blocks per request by a database from hard disks.
`DB_IO_UTIL_LG`	The percentage of hard disk resources utilized by large requests from this database.
`DB_IO_UTIL_SM`	The percentage of hard disk resources utilized by small requests from this database.
`DB_IO_WT_LG`	The IORM wait time for large I/O requests issued to hard disks by a database.
`DB_IO_WT_LG_RQ`	The average IORM wait time per request for large I/O requests issued to hard disks by a database.
`DB_IO_WT_SM`	The IORM wait time for small I/O requests issued to hard disks by a database.
`DB_IO_WT_SM_RQ`	The average IORM wait time per request for small I/O requests issued to hard disks by a database.
`DB_PC_BY_ALLOCATED`	Number of megabytes allocated in PMEM cache for this database.

All database cumulative metrics are reset to zero whenever a category, IORM, or any database resource plan is modified.

To list the database metric history for an inter-database plan, use the following CellCLI command:

```
CellCLI> LIST METRICHISTORY WHERE objectType = 'IORM_DATABASE' AND -
metricValue != 0 ATTRIBUTES name, metricObjectName, metricValue,
collectionTime
```

In addition, database metrics are provided for Oracle ASM and all other databases in a metric named `_OTHER_DATABASE_`.

6.3.4 Monitoring IORM with Consumer Group Metrics

Consumer group metrics provide information about the size of the I/O load from each consumer group specified in a database resource plan.

Each database in the interdatabase plan has metrics for each of its consumer groups. The `objectType` attribute of the `METRICCURRENT`, `METRICDEFINITION`, and `METRICHISTORY` objects is equal to `IORM_CONSUMER_GROUP` for consumer group metrics.

The following table lists the metric name and description for consumer group metrics. Note the following:

- The consumer group and database for the metric are specified by the `metricObjectName` attribute of the `METRICCURRENT` and `METRICHISTORY` objects. The name is formed by the database name followed by a period (`.`) and the consumer group name. For example, for a database named `PRODUCTIONDB` and a consumer group named `OLTP`, the `metricObjectName` would be `PRODUCTIONDB.OLTP`.
- Starting with Oracle Exadata System Software release 19.1.0, if you configured ASM-scoped security for the Oracle Automatic Storage Management (Oracle ASM) cluster used by the database, then the database name is prefixed with the Oracle ASM cluster name.
- For cumulative metrics, the metric value for a specific time period can be determined by subtracting values from different `collectionTime` periods.
- For rate metrics, the time period for the metric value is over the previous minute.
- In the metric descriptions, small means less than or equal to 128 KB, and large means more than 128 KB.

Table 6-11 Consumer Group Metrics and Descriptions

Metric	Description
`CG_FC_IO_BY_SEC`	The number of megabytes of I/O per second for this consumer group to flash cache.
`CG_FC_IO_RQ`	The number of I/O requests issued by a consumer group to flash cache.
`CG_FC_IO_RQ_LG`	The number of large I/O requests issued by a consumer group to flash cache.
`CG_FC_IO_RQ_LG_SEC`	The number of large I/O requests issued by a consumer group to flash cache per second.
`CG_FC_IO_RQ_SEC`	The number of I/O requests issued by a consumer group to flash cache per second.
`CG_FC_IO_RQ_SM`	The number of small I/O requests issued by a consumer group to flash cache.
`CG_FC_IO_RQ_SM_SEC`	The number of small I/O requests issued by a consumer group to flash cache per second.
`CG_FD_IO_BY_SEC`	The number of megabytes of I/O per second for this consumer group to flash disks.
`CG_FD_IO_LOAD`	The metric shows the average I/O load from this consumer group for flash disks. For a description of I/O load, see CD_IO_LOAD.

Table 6-11 (Cont.) Consumer Group Metrics and Descriptions

Metric	Description
`CG_FD_IO_RQ_LG`	The number of large I/O requests issued by a consumer group to flash disks.
`CG_FD_IO_RQ_LG_SEC`	The number of large I/O requests issued by a consumer group to flash disks per second.
`CG_FD_IO_RQ_SM`	The number of small I/O requests issued by a consumer group to flash disks.
`CG_FD_IO_RQ_SM_SEC`	The number of small I/O requests issued by a consumer group to flash disks per second.
`CG_FD_IO_TM`	The cumulative latency of reading or writing blocks by a consumer group from flash disks.
`CG_FD_IO_TM_LG`	The cumulative latency of reading or writing large blocks by a consumer group from flash disks.
`CG_FD_IO_TM_LG_RQ`	The rate which is the average latency of reading or writing large blocks per request by a consumer group from flash disks.
`CG_FD_IO_TM_RQ`	The rate which is the average latency of reading or writing blocks per request by a consumer group from flash disks.
`CG_FD_IO_TM_SM`	The cumulative latency of reading or writing small blocks by a consumer group from flash disks.
`CG_FD_IO_TM_SM_RQ`	The rate which is the average latency of reading or writing small blocks per request by a consumer group from flash disks.
`CG_FD_IO_UTIL`	The percentage of flash resources utilized from this consumer group.
`CG_FD_IO_UTIL_LG`	The percentage of flash resources utilized by large requests from this consumer group.
`CG_FD_IO_UTIL_SM`	The percentage of flash resources utilized by small requests from this consumer group.
`CG_FD_IO_WT_LG`	The IORM wait time for large I/O requests issued to flash disks by a consumer group.
`CG_FD_IO_WT_LG_RQ`	The average IORM wait time per request for large I/O requests issued to flash disks by a consumer group.
`CG_FD_IO_WT_SM`	The IORM wait time for small I/O requests issued to flash disks by a consumer group.
`CG_FD_IO_WT_SM_RQ`	The average IORM wait time per request for small I/O requests issued to flash disks by a consumer group.
`CG_IO_BY_SEC`	The number of megabytes of I/O per second for this consumer group to hard disks.
`CG_IO_LOAD`	The metric shows the average I/O load from this consumer group for hard disks. For a description of I/O load, see CD_IO_LOAD.
`CG_IO_RQ_LG`	The cumulative number of large I/O requests issued by the consumer group. A large value indicates a heavy I/O workload from this consumer group.

Table 6-11 (Cont.) Consumer Group Metrics and Descriptions

Metric	Description
`CG_IO_RQ_LG_SEC`	This metric is derived from `CG_IO_RQ_LG`. It specifies the rate of large I/O requests issued by a consumer group per second over the past minute. A large value indicates a heavy I/O workload from this consumer group in the past minute.
`CG_IO_RQ_SM`	The cumulative number of small I/O requests issued by the consumer group. A large value indicates a heavy I/O workload from this consumer group.
`CG_IO_RQ_SM_SEC`	This metric is derived from `CG_IO_RQ_SM`. It specifies the rate of small I/O requests issued by a consumer group per second over the past minute. A large value indicates a heavy I/O workload from this consumer group in the past minute.
`CG_IO_TM_LG`	The cumulative latency of reading or writing large blocks by a consumer group from hard disks.
`CG_IO_TM_LG_RQ`	The rate which is the average latency of reading or writing large blocks per request by a consumer group from hard disks.
`CG_IO_TM_SM`	The cumulative latency of reading or writing small blocks by a consumer group from hard disks
`CG_IO_TM_SM_RQ`	The rate which is the average latency of reading or writing small blocks per request by a consumer group from hard disks.
`CG_IO_UTIL_LG`	The percentage of disk resources utilized by large requests from this consumer group.
`CG_IO_UTIL_SM`	The percentage of disk resources utilized by small requests from this consumer group.
`CG_IO_WT_LG`	The IORM wait time for large I/O requests issued to hard disks by a consumer group.
`CG_IO_WT_LG_RQ`	The average IORM wait time per request for large I/O requests issued to hard disks by a consumer group.
`CG_IO_WT_SM`	The IORM wait time for small I/O requests issued to hard disks by a consumer group.
`CG_IO_WT_SM_RQ`	The average IORM wait time per request for small I/O requests issued to hard disks by a consumer group.

All consumer group cumulative metrics are reset to zero whenever a category, IORM, or any database resource plan is modified.

To list the current metrics for consumer groups, use the following CellCLI command:

```
CellCLI> LIST METRICCURRENT WHERE objectType = 'IORM_CONSUMER_GROUP'
AND        -
         metricValue != 0 ATTRIBUTES name, metricObjectName,
metricValue,        -
         collectionTime
```

For Oracle ASM and all other databases, metrics are only provided for the BACKGROUND and OTHER consumer groups. The BACKGROUND consumer groups are:

- _ORACLE_BACKGROUND_GROUP_: High-priority I/O requests from Oracle Database background processes
- _ORACLE_MEDPRIBG_GROUP_: Medium-priority I/O requests from Oracle Database background processes
- _ORACLE_LOWPRIBG_GROUP_: Low-priority I/O requests from Oracle Database background processes

Related Topics

- LIST METRICHISTORY
- LIST METRICCURRENT

6.3.5 Monitoring IORM Utilization

You use metrics to monitor IORM utilization.

When OLTP and DSS workloads share Oracle Exadata Storage Servers, IORM determines whether to optimize for low latency or high throughput. To optimize for low latency, large I/O requests should be distributed so the disk is not fully utilized. To optimize for high throughput, each Oracle Exadata Storage Server must handle many concurrent large I/O requests, allowing the cell to be fully utilized while applying optimization algorithms. However, when a cell has many concurrent large I/O requests, I/O latency is high because each I/O is queued behind many other I/Os.

The utilization metrics for I/O requests from database, pluggable database (PDB), and consumer groups correspond to the amount of time a database, PDB, or consumer group utilized a storage server. Large I/O requests utilize more of a storage server than small I/O requests. The following are the utilization metrics for determining IORM optimization:

- `CG_IO_UTIL_LG`
- `CG_IO_UTIL_SM`
- `PDB_IO_UTIL_LG`
- `PDB_IO_UTIL_SM`
- `CT_IO_UTIL_LG`
- `CT_IO_UTIL_SM`
- `DB_IO_UTIL_LG`
- `DB_IO_UTIL_SM`

By comparing the amount of I/O resources consumed with the I/O resource allocations, the database administrator can determine if IORM should be tuned for latency or throughput. The IORM metric, `IORM_MODE`, shows the mode for IORM. The metric value ranges between 1 and 3. The following are the definitions for the values:

- 1 means the cell IORM objective was set to `low_latency`.
- 2 means the cell IORM objective was set to `balanced`.
- 3 means the cell IORM objective was set to `high_throughput`.

A value in between 1-2 or 2-3 indicates the IORM objective was not the same throughout the metric period, and the value indicates proximity to a given objective. It is also indicative of a constantly-changing mix of workloads.

Using iostat to monitor I/O service time statistics

Device utilization and I/O service time statistics monitored using `iostat` or any other operating system tool are erroneous for a flash device because the flash device services I/Os in parallel. The following is stated in the Linux `man` page for `iostat`:

```
svctm - The average service time (in milliseconds) for I/O requests that
were
issued to the device. Warning! Do not trust this field any more. This
field will be removed
in a future sysstat version.
```

The utilization computation depends upon the I/O service time which makes it inaccurate. To monitor the actual I/O utilization for a cell disk, database, pluggable database or consumer group, use the corresponding IORM metrics.

6.3.6 Tuning Interdatabase Plans with Metrics

The Oracle Exadata System Software metrics relevant to I/O Resource Management (IORM) can be used in multiple ways.

Metrics can be used to understand the size of the I/O workload as a whole and which portion of it corresponds to each category, database, or consumer group. For example, the metrics might show that a particular database is issuing a higher I/O workload than expected.

In addition, metrics can be used to understand which category, database, pluggable database (PDB), or consumer group is not using its resource allocation and which one is exceeding its resource allocation. For example:

- If the wait times (`DB_IO_WT_SM_RQ`, `DB_IO_WT_LG_RQ`, `PDB_IO_WT_SM_RQ`, `PDB_IO_WT_LG_RQ` and `CG_IO_WT_SM_RQ`) are typically small or zero, then the plan allocation is probably sufficient.
- If the wait times are large, then the plan allocation is insufficient.
- If the wait times cause an unacceptable drop in performance, then the plans should be adjusted to give a larger allocation, or more cells or disks should be added to increase the total I/O resources.

6.4 Monitoring Requests and Alerts for Oracle Exadata Storage Server

You can monitor the active requests and alerts for a storage server and you can receive e-mail notifications for alerts.

An active request provides a client-centric or application-centric view of client I/O requests that are currently being processed by a cell. Alerts represent events of importance occurring within the cell. Typically, alerts indicate that Oracle Exadata Storage Server functionality is compromised or in danger of failure.

To receive notifications, use the `ALTER CELL` command.

- Displaying Active Requests
  Use the LIST ACTIVEREQUEST command to display the active requests for the storage server.
- Displaying Alert Definitions
  Use the LIST ALERTDEFINITION command to display the alert definitions for the storage server.
- Receiving Alert Notifications
  Administrators for Oracle Exadata System Software can receive alert notifications by e-mail or by Simple Network Management Protocol (SNMP) trap alerts.
- Monitoring syslog Messages Remotely
  By default, storage server syslog messages are written to local log files.
- Displaying Alert History
  Use the LIST ALERTHISTORY command to display the alert history that has occurred on a cell.
- Modifying Alert History
  Use the ALTER ALERTHISTORY command to update the alert history for the cell.

Related Topics

- ALTER CELL
- About Alerts
  Alerts represent events of importance occurring within the storage cell, typically indicating that storage cell functionality is either compromised or in danger of failure.

6.4.1 Displaying Active Requests

Use the LIST ACTIVEREQUEST command to display the active requests for the storage server.

To view the ACTIVEREQUEST attributes, use the DESCRIBE ACTIVEREQUEST command.

Example 6-5 Listing Active Request Attributes

This example shows how to display a detailed list of attributes for a specified request I/O type.

```
CellCLI> LIST ACTIVEREQUEST WHERE IoType = 'predicate pushing' DETAIL
```

Related Topics

- LIST ACTIVEREQUEST
- DESCRIBE ACTIVEREQUEST

6.4.2 Displaying Alert Definitions

Use the LIST ALERTDEFINITION command to display the alert definitions for the storage server.

An alert definition provides a definition for every alert that can be produced on a storage server.

Example 6-6 Listing Alert Definition Attributes

This example shows how to display a detailed list of attributes for the alert `ADRAlert`.

```
CellCLI> LIST ALERTDEFINITION ADRAlert DETAIL
         name:                   ADRAlert
         alertShortName:         ADR
         alertSource:            "Automatic Diagnostic Repository"
         description:            "Incident Alert"
```

Example 6-7 Listing Alert Definition Name and Description Attributes

You can display a list of specified attributes for an alert definition. This example shows how to display the alert name, metric name, and description. The metric name identifies the metric on which the alert is based. `ADRAlert`, `HardwareAlert`, `Stateful_HardwareAlert`, and `Stateful_SoftwareAlert` are not based on a metric, and therefore do not have metric names.

```
CellCLI> LIST ALERTDEFINITION ATTRIBUTES name, metricName, description
    ADRAlert                                        "Incident Alert"
    HardwareAlert                                   "Hardware Alert"
    MetricAlert                                     "Threshold Alert"
    SoftwareAlert                                   "Software Alert"
    StatefulAlert_CD_IO_ERRS_MIN    CD_IO_ERRS_MIN  "Threshold Alert"
    StatefulAlert_CG_IO_RQ_LG       CG_IO_RQ_LG     "Threshold Alert"
    StatefulAlert_CG_IO_RQ_LG_SEC   CG_IO_RQ_LG_SEC "Threshold Alert"
    StatefulAlert_CG_IO_RQ_SM       CG_IO_RQ_SM     "Threshold Alert"
    StatefulAlert_CG_IO_RQ_SM_SEC   CG_IO_RQ_SM_SEC "Threshold Alert"
    StatefulAlert_CG_IO_WT_LG       CG_IO_WT_LG     "Threshold Alert"
    StatefulAlert_CG_IO_WT_LG_RQ    CG_IO_WT_LG_RQ  "Threshold Alert"
    StatefulAlert_CG_IO_WT_SM       CG_IO_WT_SM     "Threshold Alert"
    StatefulAlert_CG_IO_WT_SM_RQ    CG_IO_WT_SM_RQ  "Threshold Alert"
    StatefulAlert_CL_FSUT           CL_FSUT         "Threshold Alert"
    StatefulAlert_CL_MEMUT          CL_MEMUT        "Threshold Alert"
    StatefulAlert_CT_IO_RQ_LG       CT_IO_RQ_LG     "Threshold Alert"
    StatefulAlert_CT_IO_RQ_LG_SEC   CT_IO_RQ_LG_SEC "Threshold Alert"
    StatefulAlert_CT_IO_RQ_SM       CT_IO_RQ_SM     "Threshold Alert"
    StatefulAlert_CT_IO_RQ_SM_SEC   CT_IO_RQ_SM_SEC "Threshold Alert"
    StatefulAlert_CT_IO_WT_LG       CT_IO_WT_LG     "Threshold Alert"
    StatefulAlert_CT_IO_WT_LG_RQ    CT_IO_WT_LG_RQ  "Threshold Alert"
    StatefulAlert_CT_IO_WT_SM       CT_IO_WT_SM     "Threshold Alert"
    StatefulAlert_CT_IO_WT_SM_RQ    CT_IO_WT_SM_RQ  "Threshold Alert"
    StatefulAlert_DB_IO_RQ_LG       DB_IO_RQ_LG     "Threshold Alert"
    StatefulAlert_DB_IO_RQ_LG_SEC   DB_IO_RQ_LG_SEC "Threshold Alert"
    StatefulAlert_DB_IO_RQ_SM       DB_IO_RQ_SM     "Threshold Alert"
    StatefulAlert_DB_IO_RQ_SM_SEC   DB_IO_RQ_SM_SEC "Threshold Alert"
    StatefulAlert_DB_IO_WT_LG       DB_IO_WT_LG     "Threshold Alert"
    StatefulAlert_DB_IO_WT_LG_RQ    DB_IO_WT_LG_RQ  "Threshold Alert"
    StatefulAlert_DB_IO_WT_SM       DB_IO_WT_SM     "Threshold Alert"
    StatefulAlert_DB_IO_WT_SM_RQ    DB_IO_WT_SM_RQ  "Threshold Alert"
    StatefulAlert_GD_IO_ERRS_MIN    GD_IO_ERRS_MIN  "Threshold Alert"
    Stateful_HardwareAlert                          "Hardware Stateful
Alert"
```

```
    Stateful_SoftwareAlert                               "Software Stateful 
Alert"
```

Related Topics

- Displaying Metric Definitions
  Use the `LIST METRICDEFINITION` command to display the metric definitions for a storage server.
- DESCRIBE ALERTDEFINITION
- LIST ALERTDEFINITION

6.4.3 Receiving Alert Notifications

Administrators for Oracle Exadata System Software can receive alert notifications by e-mail or by Simple Network Management Protocol (SNMP) trap alerts.

Use of SNMP alerts allows Oracle Exadata Storage Servers to be monitored by a management application, such as Oracle Enterprise Manager Cloud Control. To configure Oracle Exadata Storage Server to send e-mail messages or SNMP trap alerts, use the `ALTER CELL` command.

> **Note:**
>
> The SNMP alerts conform to a MIB (management information base) which is included with each installation of Oracle Exadata System Software. The MIB file on Oracle Exadata Storage Server is available at `/opt/oracle/cell/cellsrv/deploy/config/cell_alert.mib`. The SNMP alerts and MIB conform to SNMP version 1 (SNMPv1).

Related Topics

- Format of Alert Messages for SNMP Notification

6.4.4 Monitoring syslog Messages Remotely

By default, storage server syslog messages are written to local log files.

A separate management server, known as a *loghost server*, can receive syslog messages from Oracle Exadata Storage Servers.

- To monitor the syslog messages remotely, configure the syslog service on the loghost server to listen for incoming syslog messages by setting `SYSLOGD_OPTIONS -r` in the loghost server `/etc/sysconfig/syslog` file.
- Configure each Oracle Exadata Storage Server to forward specified syslog messages to the loghost server using the `ALTER CELL` command.

  The configuration on Oracle Exadata Storage Servers is maintained across restarts and updates.
- The `ALTER CELL VALIDATE SYSLOGCONF` command can be used to test that messages are sent by the cells and received by the loghost server.

Related Topics

- ALTER CELL

6.4.5 Displaying Alert History

Use the `LIST ALERTHISTORY` command to display the alert history that has occurred on a cell.

Alert history entries are retained for a maximum of 100 days. If the number of alert history entries exceeds 500, then the alert history entries are only retained for 7 days. When stateful alerts are cleared, meaning that the underlying metric, hardware or software condition has returned to normal, then the stateful alert is retained either 100 or 7 days, depending on the number of alert history entries. Stateful alerts that are not cleared are retained, regardless of their age.

Example 6-8 Listing Alert History Attributes

This example shows how to display a detailed list of attributes for alert history entries where the `severity` attribute is set to `critical` and the `examinedBy` attribute has not been set.

```
CellCLI> LIST ALERTHISTORY WHERE severity = 'critical' AND examinedBy = ''
DETAIL
```

Related Topics

- DESCRIBE ALERTHISTORY
- LIST ALERTHISTORY
- ALTER CELL
- DESCRIBE CELL

6.4.6 Modifying Alert History

Use the `ALTER ALERTHISTORY` command to update the alert history for the cell.

Example 6-9 Altering Alert History Attributes

This example shows how to set the `examinedBy` attribute to the user ID of the administrator that examined the alert.

```
CellCLI> ALTER ALERTHISTORY 1671443714 examinedBy="jdoe"
```

The `examinedBy` attribute is the only `ALERTHISTORY` attribute that can be modified.

Related Topics

- ALTER ALERTHISTORY
- DESCRIBE ALERTHISTORY

6.5 Monitoring Oracle Exadata Storage Server using Views

Dynamic V$ data dictionary views can help monitor the Oracle Exadata Storage Server environment.

- Using the V$CELL and GV$CELL Views to Display Oracle Exadata Storage Server Identification
  The V$CELL view provides identifying information about cells.
- Using V$BACKUP_DATAFILE with Oracle Exadata Storage Server
  The V$BACKUP_DATAFILE view contains columns relevant to Oracle Exadata Storage Server during Oracle Recovery Manager (RMAN) incremental backups.
- Using V$SYSSTAT with Oracle Exadata System Software
  You can query the V$SYSSTAT view for statistics that can be used to compute Oracle Exadata System Software effectiveness.
- Using V$SEGMENT_STATISTICS with Oracle Exadata System Software
  The V$SEGMENT_STATISTICS view lists statistics on a per segment basis.
- Using V$SQL with Oracle Exadata System Software
  The V$SQL view lists statistics on shared SQL areas and contains one row for each child of the original SQL text entered.
- Using V$CELL_STATE to Display Oracle Exadata System Software Performance Statistics
  The V$CELL_STATE view describes the state of all the cells accessible from the database client.
- Using V$CELL_THREAD_HISTORY to Display Oracle Exadata System Software Threads
  The V$CELL_THREAD_HISTORY view contains the samples that Cell Server (CELLSRV) takes of the threads in the cells visible to the database client.
- Using V$CELL_REQUEST_TOTALS with Oracle Exadata System Software
  The V$CELL_REQUEST_TOTALS view contains a historic view of the types and frequencies of the requests being run by a cell.
- Using V$ASM_DISK_SPARSE and V$ASM_DISKGROUP_SPARSE to Monitor Sparse Disks
  The V$ASM_DISK_SPARSE and V$ASM_DISKGROUP_SPARSE views contain information about sparse disks.

6.5.1 Using the V$CELL and GV$CELL Views to Display Oracle Exadata Storage Server Identification

The V$CELL view provides identifying information about cells.

Table 6-12 V$CELL View Columns and Descriptions

Column	Description
CELL_HASHVAL	A numeric hash value for the cell. For example: 138889696 **Note:** This value is useful because the P1 argument is often the cell hash name for cell-related wait events in the V$SESSION_WAIT and V$ACTIVE_SESSION_HISTORY views. Using this value, you can use a join with the V$CELL view on the CELL_HASHVAL column to determine the corresponding cell name.
CELL_PATH	A character string (maximum 400) that specifies the IP addresses of the cell. These are the IP addresses specified in the cellip.ora file.
CELL_TYPE	The type of storage cell.

The GV$CELL view contains the same columns as the V$CELL view, and includes the INST_ID column. The INST_ID column displays the instance number from which the associated V$ view information was obtained. Querying a GV$ view retrieves the V$ view information from all qualified instances.

6.5.2 Using V$BACKUP_DATAFILE with Oracle Exadata Storage Server

The V$BACKUP_DATAFILE view contains columns relevant to Oracle Exadata Storage Server during Oracle Recovery Manager (RMAN) incremental backups.

Table 6-13 V$BACKUP_DATAFILE Columns and Descriptions

Column	Description
BLOCKS	Size of the backup data file in blocks.
BLOCKS_READ	The number of blocks that were scanned while taking this backup. If this is an incremental backup, and block change tracking was used to optimize the backup, then the value of this column is smaller than DATAFILE_BLOCKS. Otherwise, the value of this column is the same as DATAFILE_BLOCKS.
BLOCKS_SKIPPED_IN_CELL	The number of blocks that were read and filtered at the Oracle Exadata Storage Server to optimize the RMAN incremental backup.
DATAFILE_BLOCKS	Size of the data file in blocks at backup time. This value is also the number of blocks taken by the data file restarted from this backup.

The percentage of blocks skipped by Oracle Exadata System Software is calculated as follows:

```
(BLOCKS_SKIPPED_IN_CELL / BLOCKS_READ) * 100
```

This number changes significantly based on block change tracking.

If block change tracking for fast incremental backups is used, then most of the filtering is done at the database using the change tracking file, and the blocks are skipped before making an I/O request to the cell. If block change tracking is not used, then all of the blocks are filtered at the cell.

Related Topics

- *Oracle Database Reference*
- *Oracle Database Backup and Recovery User's Guide*

6.5.3 Using V$SYSSTAT with Oracle Exadata System Software

You can query the `V$SYSSTAT` view for statistics that can be used to compute Oracle Exadata System Software effectiveness.

Table 6-14 V$SYSSTAT View Key Statistics

Statistic	Description
`cell flash cache read hits`	The number of read requests that were a cache hit on Exadata flash cache.
`cell IO uncompressed bytes`	The total size of uncompressed data that is processed on the cell. For scan on Exadata Hybrid Columnar Compression-compressed tables, this statistic is the size of data after decompression.
`cell overwrites in flash`	Total number of mirror write requests that overwrote an existing mirror in Exadata Smart Flash Cache that had not been written to disk. This statistic is incremented once per mirror write.
`cell partial writes in flash`	Total number of mirror write requests written to both Exadata Smart Flash Cache and disk. Part of the data was written to flash, and the rest was written to disk. This statistic is incremented once per mirror write.
`cell physical IO bytes eligible for predicate offload`	The total number of I/O bytes processed with physical disks when processing was offloaded to the cell.
`cell physical IO bytes saved by storage index`	The number of bytes saved by storage index.
`cell physical IO bytes saved during optimized file creation`	The number of I/O bytes saved by the database host by offloading the file creation operation to cells. This statistic shows the Oracle Exadata System Software benefit due to optimized file creation operations.
`cell physical IO bytes saved during optimized RMAN file restore`	The number of I/O bytes saved by the database host by offloading the RMAN file restore operation to cells. This statistic shows the Oracle Exadata System Software benefit due to optimized RMAN file restore operations.

Table 6-14 (Cont.) V$SYSSTAT View Key Statistics

Statistic	Description
`cell physical IO bytes sent directly to DB node to balance CPU usage`	The number of I/O bytes sent back to the database server for processing due to CPU usage on Oracle Exadata Storage Server.
`cell physical IO interconnect bytes`	The number of I/O bytes exchanged over the interconnection between the database host and cells.
`cell physical IO interconnect bytes returned by smart scan`	The number of bytes that are returned by the cell for Smart Scan only, and does not include bytes for other database I/O.
`cell writes to flash`	Total number of mirror write requests written entirely to Exadata Smart Flash Cache. This statistic is incremented once per mirror write.
`physical read partial requests`	Total number of read requests satisfied from both Exadata Smart Flash Cache and hard disks. Part of the data was dirty on flash and rest was retrieved from disk.
`physical read requests optimized`	Total number of read requests satisfied either by using Exadata Smart Flash Cache or storage index.
`physical read snap IO requests no data`	The number of physical read I/O requests for which no physical I/O is done on the child file level.
`physical read snap IO requests base`	The number of physical I/Os on the base level.
`physical read snap IO requests copy`	The number of physical I/Os on any snapshot hierarchy.
`physical read snap bytes base`	The number of bytes read from the base.
`physical read snap bytes copy`	The number of bytes read from the snapshot.
`physical write snap IO requests new allocation`	The total number of new allocations on the snapshot.
`physical read total bytes`	Total amount of I/O bytes for reads processed with physical disks. This includes when processing was offloaded to the cell and when processing was not offloaded.
`physical read total bytes optimized`	Total number of bytes read from Exadata Smart Flash Cache or storage index.
`physical read total IO requests`	The number of read requests that read one or more database blocks for all instance activity including application, backup, recovery, and other utilities.
`physical write requests optimized`	Total number of write requests in which all mirrors were written to the Exadata Smart Flash Cache.
`physical write total bytes`	Total amount of I/O bytes for writes processed with physical disks. This includes when processing was offloaded to the cell and when processing was not offloaded.

Table 6-14 (Cont.) V$SYSSTAT View Key Statistics

Statistic	Description
`physical write total bytes optimized`	Total number of bytes written to Exadata Smart Flash Cache first. These bytes are synchronized to the disk in a lazy manner to create free space in flash cache.
`physical write total IO requests`	The number of write requests written to one or more database blocks from all instance activity including application activity, backup, recovery, and other utilities.

- Monitoring Write Statistics
  The write statistics for the Oracle RAC interconnect can be monitored using the `V$SYSSTAT` view.
- Querying Storage Server Statistics in the V$SYSSTAT View
  This example shows the how to query the `V$SYSSTAT` statistics.

Related Topics

- V$SYSSTAT

6.5.3.1 Monitoring Write Statistics

The write statistics for the Oracle RAC interconnect can be monitored using the `V$SYSSTAT` view.

Exafusion direct-to-wire protocol allows processes that read and serve cache blocks across an Oracle Real Application Clusters (Oracle RAC) interconnect to directly send and receive messages over the RDMA Network Fabric. The following table lists the write statistics for the storage server:

Table 6-15 Write Statistics for Cell Monitoring

Statistic	Description
`cell physical write IO bytes eligible for offload`	This write event is the total number of write I/O bytes that became eligible for high throughput write offload.
`cell physical write IO host network bytes written during offload`	This write event is the total number of write I/O bytes that became eligible for high throughput write offload, and were written directly from the database server process which created the logical write.
`cell logical write IO requests`	This write event is the total number of logical write requests issued by the process.
`cell logical write IO requests eligible for offload`	This write event is the total number of logical write requests that became eligible for high throughput write offload mode.

6.5.3.2 Querying Storage Server Statistics in the V$SYSSTAT View

This example shows the how to query the `V$SYSSTAT` statistics.

Example 6-10 Querying Cell Statistics in the V$SYSSTAT View

```
SQL> SELECT name, value/1024/1024 MB FROM v$sysstat a WHERE
        a.name = 'physical read total bytes' OR
        a.name = 'physical write total bytes' OR
        a.name = 'cell physical IO interconnect bytes' OR
        a.name = 'cell physical IO bytes eligible for predicate offload' OR
        a.name = 'cell physical IO bytes saved during optimized file
creation' OR
        a.name = 'cell physical IO bytes saved during optimized RMAN file
restore'
        OR
        a.name = 'cell IO uncompressed bytes' OR
        a.name = 'cell physical IO interconnect bytes returned by smart
scan' OR
        a.name = 'cell physical IO bytes saved by storage index';


NAME
   MB
---------------------------------------------------------------------------
-------
physical read total bytes
5425.11
physical write total
bytes                                                   26.24
cell physical IO interconnect
bytes                                         550.94
cell physical IO bytes saved during optimized file
creation                      0
cell physical IO bytes saved during optimized RMAN file
restore                  0
cell physical IO bytes eligible for predicate offload
5372.53
cell physical IO bytes saved by storage
index                                    0
cell physical IO interconnect bytes returned by smart
scan                  472.12
cell IO uncompressed bytes
5372.53

9 rows selected.
```

You can determine the amount of disk I/O performed by calculating the following:

```
physical read total bytes  + physical write total bytes - cell physical IO
bytes
saved by storage index
```

Using the output in this example, the disk I/O performed is 5425.11 + 26.24 - 0 = 5451.35 MB.

6.5.4 Using V$SEGMENT_STATISTICS with Oracle Exadata System Software

The V$SEGMENT_STATISTICS view lists statistics on a per segment basis.

The segment-level statistics can be used to detect specific objects, such as tables or indexes, that are performing optimized reads from the cell. The optimized physical read segment statistic provides the number of read requests for an objects that were read from Exadata Smart Flash Cache or from the storage index. The optimized physical writes statistic provides the number of write requests for an object which went to Exadata Smart Flash Cache first. These write requests can later be synchronized to the disk in a lazy manner to create free space on the flash.

Example 6-11   Using the V$SEGMENT_STATISTICS View

This example shows a query that returns all objects that have performed more than 1000 optimized reads from a cell. A similar query can be used to determine which objects have performed very few optimized reads.

```
SELECT object_name, value FROM V$segment_statistics WHERE \
statistic_name='optimized physical reads' AND value>1000 ORDER BY value;
```

6.5.5 Using V$SQL with Oracle Exadata System Software

The V$SQL view lists statistics on shared SQL areas and contains one row for each child of the original SQL text entered.

You can query the following columns in the V$SQL view for physical I/O read disk bytes, physical I/O write disk bytes, cell physical I/O interconnect bytes, cell physical I/O bytes eligible for predicate offload, cell uncompressed bytes, cell interconnect bytes returned for Smart Scan, and the number of physical read requests returned by Exadata Smart Flash Cache or storage index:

- PHYSICAL_READ_BYTES
- PHYSICAL_WRITE_BYTES
- IO_INTERCONNECT_BYTES
- IO_CELL_OFFLOAD_ELIGIBLE_BYTES
- IO_CELL_UNCOMPRESSED_BYTES
- IO_CELL_OFFLOAD_RETURNED_BYTES
- OPTIMIZED_PHY_READ_REQUESTS

The values returned in the view are the number of bytes read by the parallel queries for the particular instance.

Note:

The preceding columns are also available in the following views:

- V$SQLAREA
- V$SQLAREA_PLAN_HASH
- V$SQLSTATS
- V$SQLSTATS_PLAN_HASH

Example 6-12 Querying the V$SQL View

This example shows a query returning Oracle Exadata System Software offload processing performance data. A WHERE predicate was used to select only the V$SQL command of interest, which was the query that scanned the SALES table. The performance data stored in V$SQL for the query shows that of the approximately 5 GB of data in the SALES table, all of it was eligible for offload processing. Because of the effectiveness of Oracle Exadata System Software offload processing, only 417 MB of data was delivered over the network to the database host.

```
SELECT sql_text,
       io_cell_offload_eligible_bytes/1024/1024 cell_offload_eligible_mb,
       io_cell_uncompressed_bytes/1024/1024 io_uncompressed_mb,
       io_interconnect_bytes/1024/1024 io_interconnect_mb,
       io_cell_offload_returned_bytes/1024/1024 cell_return_bytes_mb,
       (physical_read_bytes + physical_write_bytes)/1024/1024 io_disk_mb
       FROM v$sql WHERE
       sql_text LIKE '%from sales%';


SQL_TEXT CELL_OFFLOAD_ELIGIBLE_MB IO_UNCOMPRESSED_MB IO_INTERCONNECT_MB 
CELL_RETURN_BYTES_MB
IO_DISK_MB
-----------------------------------------------------------------------------------------
--
select count(*) from sales
5283.06   5283.06  520.34  417.65  5385.75
```

Related Topics

- Using V$SYSSTAT with Oracle Exadata System Software
  You can query the V$SYSSTAT view for statistics that can be used to compute Oracle Exadata System Software effectiveness.
- *Oracle Database Reference*

6.5.6 Using V$CELL_STATE to Display Oracle Exadata System Software Performance Statistics

The V$CELL_STATE view describes the state of all the cells accessible from the database client.

The state includes key performance statistics, such as Cell Server (CELLSRV) network layer information, scheduling layer information, and Cell Server configuration information. This view is used by Oracle Support Services.

Table 6-16 V$CELL_STATE Columns and Descriptions

Column	Description
CELL_NAME	A character string (maximum 400) that specifies the IP addresses of the cell. These are the IP addresses specified in the cellip.ora file. For example: 172.16.50.28
STATISTICS_TYPE	The statistics type, such as thread_stats.
OBJECT_NAME	Key for a specific statistics type, such as the thread ID if STATISTICS_TYPE is thread_stats.
STATISTICS_VALUE	The actual statistic values in an XML document with attribute and value pairs.

Example 6-13 Displaying Statistics from the V$CELL_STATE View

This example shows how to retrieve the information in the STATISTICS_VALUE column in a readable format.

```
SQL> SPOOL /tmp/cell_state_080619.log
SQL> SET PAGESIZE 10000
SQL> SET LONG 500000
SQL> SELECT  statistics_type, XMLTYPE(statistics_value) FROM v$cell_state;
SQL> SPOOL OFF
```

6.5.7 Using V$CELL_THREAD_HISTORY to Display Oracle Exadata System Software Threads

The V$CELL_THREAD_HISTORY view contains the samples that Cell Server (CELLSRV) takes of the threads in the cells visible to the database client.

This view is used by Oracle Support Services.

Table 6-17 V$CELL_THREAD_HISTORY Columns and Descriptions

Column	Description
CELL_NAME	A character string (maximum 400) that specifies the IP addresses of the cell. These are the IP addresses specified in the cellip.ora file. For example: 172.16.50.28
SNAPSHOT_ID	The ID of the snapshot (NUMBER).
SNAPSHOT_TIME	The date and time of the snapshot.
THREAD_ID	The thread ID (NUMBER).
JOB_TYPE	The job that the thread was running when the snapshot was taken.
WAIT_STATE	A unique state that identifies the location of the wait, if any exists.
WAIT_OBJECT_NAME	Object being waited on, if any exists. For example, the mutex name.
SQL_ID	The identifier of the SQL command that the client was processing for the job that is running.
DATABASE_ID	The ID of the database (NUMBER).
INSTANCE_ID	The ID of the instance (NUMBER).
SESSION_ID	The ID of the session (NUMBER).
SESSION_SERIAL_NUM	The session serial number (NUMBER).

6.5.8 Using V$CELL_REQUEST_TOTALS with Oracle Exadata System Software

The V$CELL_REQUEST_TOTALS view contains a historic view of the types and frequencies of the requests being run by a cell.

CELLSRV samples these requests periodically and stores them for use in this view. This view is used by Oracle Support Services.

Table 6-18 V$CELL_REQUEST_TOTALS Columns and Descriptions

Column	Description
CELL_NAME	A character string (maximum 400) that specifies the IP addresses of the cell. These are the IP addresses specified in the cellip.ora file. For example: 172.16.51.28

Table 6-18 (Cont.) V$CELL_REQUEST_TOTALS Columns and Descriptions

Column	Description
SNAPSHOT_ID	The ID of the snapshot (NUMBER).
SNAPSHOT_TIME	The date and time of the snapshot.
STATISTICS_NAME	The name of the statistic.
STATISTICS_VALUE	The value of the statistic.

6.5.9 Using V$ASM_DISK_SPARSE and V$ASM_DISKGROUP_SPARSE to Monitor Sparse Disks

The V$ASM_DISK_SPARSE and V$ASM_DISKGROUP_SPARSE views contain information about sparse disks.

Table 6-19 V$ASM_DISK_SPARSE Columns and Descriptions

Column	Description
GROUP_NUMBER	The number of the disk group containing the disk.
DISK_NUMBER	The number assigned to the disk within this disk group.
INCARNATION	The incarnation number for the disk.
ALLOCATED_MAT_MB	The total used physical and materialized capacity on the disk.
TOTAL_MAT_MB	The total physical capacity on the disk.
SPARSE_READS	The total number of I/O read requests on non-materialized regions of the disk.
SPARSE_BYTES_READ	The total number of bytes read from non-materialized regions of the disk.
SPARSE_READ_TIME	The time taken by sparse read I/O operations.

Table 6-20 V$ASM_DISKGROUP_SPARSE Columns and Descriptions

Column	Description
GROUP_NUMBER	The cluster-wide number assigned to the disk group.
ALLOCATED_MAT_MB	The total used physical and materialized capacity of the disk group.
TOTAL_MAT_MB	The total physical capacity of the disk group.

6.6 Understanding Oracle Exadata System Software Wait Events

Oracle Exadata System Software uses a different set of wait events for disk I/O to Oracle Exadata Storage Server than disk I/O to conventional storage.

The wait events that are designed for Oracle Exadata Storage Server directly show the cell and grid disk being accessed. This information is more useful for performance and diagnostics purposes than the database file# and block# information provided by wait events for conventional storage. Information about wait events is displayed in V$ dynamic performance views.

You can use the V$SESSION_ACTIVE_HISTORY view to analyze wait events. This view shows what has happened, when a query was run, and how it ran. It also shows what events the query had to wait on.

- Monitoring Wait Events for Oracle Exadata Storage Server
  This topic describes the wait events useful for monitoring a storage server.
- Using V$SESSION_WAIT to Monitor Sessions
  The V$SESSION_WAIT view displays the current or last wait for each session.
- Using V$SYSTEM_EVENT to Monitor Wait Events
  The V$SYSTEM_EVENT view displays information about the number of total waits for an event.
- Using V$SESSION_EVENT to Monitor Events by Sessions
  The V$SESSION_EVENT view displays information about waits for an event by a session.

Related Topics

- V$SESSION_WAIT
- V$SYSTEM_EVENT
- V$SESSION_EVENT

6.6.1 Monitoring Wait Events for Oracle Exadata Storage Server

This topic describes the wait events useful for monitoring a storage server.

If a cell hash number or disk hash number is associated with one of the wait events in the following table, then the value can be joined with the CELL_HASHVAL column of V$CELL and the HASH_VALUE column of V$ASM_DISK to help identify slow cells or disks.

Table 6-21 Wait Events Useful for Storage Server Monitoring

Wait Event	Description
cell interconnect retransmit during physical read	This wait event appears during retransmission for an I/O of a single-block or multiblock read. The cell hash number in the P1 column in the V$SESSION_WAIT view is the same cell identified for cell single block physical read and cell multiblock physical read. The P2 column contains the subnet number to the cell, and the P3 column contains the number of bytes processed during the I/O read operation.
cell list of blocks physical read	This wait event is equivalent to database file parallel read for a cell. The P1, P2, and P3 columns in V$SESSION_WAIT view for this event identify the cell hash number, disk hash number, and the number of blocks processed during the I/O read operation.

Table 6-21 (Cont.) Wait Events Useful for Storage Server Monitoring

Wait Event	Description
`cell multiblock physical read`	This wait event is equivalent to `db file scattered read` for a cell. The `P1`, `P2`, and `P3` columns in the `V$SESSION_WAIT` view for this event identify the cell hash number, disk hash number, and the total number of bytes processed during the I/O read operation.
`cell single block physical read`	This wait event is equivalent to `db file sequential read` for a cell. The `P1`, `P2`, and `P3` columns in the `V$SESSION_WAIT` view for this event identify the cell hash number, disk hash number, and the number of bytes processed during the I/O read operation.
`cell smart file creation`	This wait event appears when the database is waiting for the completion of a file creation on a cell. The cell hash number in the `P1` column in the `V$SESSION_WAIT` view for this event should help identify a slow cell compared to the rest of the cells.
`cell smart incremental backup`	This wait event appears when the database is waiting for the completion of an incremental backup on a cell. The cell hash number in the `P1` column in the `V$SESSION_WAIT` view for this event should help identify a slow cell when compared to the rest of the cells.
`cell smart index scan`	This wait event appears when the database is waiting for index or index-organized table (IOT) fast full scans. The cell hash number in the `P1` column in the `V$SESSION_WAIT` view for this event should help identify a slow cell when compared to the rest of the cells.
`cell smart restore from backup`	This wait event appears when the database is waiting for the completion of a file initialization for restore from backup on a cell. The cell hash number in the `P1` column in the `V$SESSION_WAIT` view for this event should help identify a slow cell when compared to the rest of the cells.
`cell smart table scan`	This wait event appears when the database is waiting for table scans to complete on a cell. The cell hash number in the `P1` column in the `V$SESSION_WAIT` view for this event should help identify a slow cell when compared to the rest of the cells.
`cell sparse block physical read`	This wait event appears when an I/O did not return any data.
`cell statistics gather`	This wait event appears when a select is done on the `V$CELL_STATE`, `V$CELL_THREAD_HISTORY`, or `V$CELL_REQUEST_TOTALS` tables. During the select, data from the cells and any wait events are shown in this wait event.

6.6.2 Using V$SESSION_WAIT to Monitor Sessions

The `V$SESSION_WAIT` view displays the current or last wait for each session.

Example 6-14 Using the V$SESSION_WAIT View

This example shows how to query the V$SESSION_WAIT view. The second SELECT query displays the cell path and disk name.

```
SELECT w.event, w.p1, w.p2, w.p3 FROM V$SESSION_WAIT w, V$EVENT_NAME e
       WHERE e.name LIKE 'cell%' AND e.wait_class_id = w.wait_class_id;

SELECT w.event, c.cell_path, d.name, w.p3 FROM V$SESSION_WAIT w,
       V$EVENT_NAME e, V$ASM_DISK d, V$CELL c
       WHERE e.name LIKE 'cell%' AND e.wait_class_id = w.wait_class_id
       AND w.p1 = c.cell_hashval AND w.p2 = d.hash_value;
```

6.6.3 Using V$SYSTEM_EVENT to Monitor Wait Events

The V$SYSTEM_EVENT view displays information about the number of total waits for an event.

Example 6-15 Using the V$SYSTEM_EVENT View

This example shows how to query the V$SYSTEM_EVENT view.

```
SELECT s.event FROM V$SYSTEM_EVENT s, V$EVENT_NAME e WHERE e.name LIKE
'cell%'
       AND e.event_id = s.event_id;
```

6.6.4 Using V$SESSION_EVENT to Monitor Events by Sessions

The V$SESSION_EVENT view displays information about waits for an event by a session.

Example 6-16 Using the V$SESSION_EVENT View

This example shows how to query the V$SESSION_EVENT view.

```
SELECT s.event FROM V$SESSION_EVENT s, V$EVENT_NAME e WHERE e.name LIKE
'cell%'
       AND e.event_id = s.event_id;
```

6.7 Optimizing Performance

You can use various settings to optimize the performance of Oracle Exadata System Software.

- About Exadata Hybrid Columnar Compression
  With Exadata Hybrid Columnar Compression, the database stores the same column for a group of rows together.
- About SQL Processing Offload
- About SQL Tuning Advisor
  SQL Tuning Advisor takes one or more SQL statements as input and uses the Automatic Tuning Optimizer to perform SQL tuning on the statements.

- About SQL Monitor
  Oracle Database has SQL monitor which allows you to monitor SQL statements as they are run.
- About Quality of Service Management
- About Fault Isolation
  Oracle Exadata System Software has the ability to learn from the past events to avoid a potential fatal error.
- About Fast Disk Scan Rates
- About Indexes and Scan Rates
  In the past, storage required indexes for good performance. Oracle Exadata System Software can have superior scan rates without using indexes.
- About Automatic Extent Management
- About Minimum Extent Size
- About SQL*Loader in Parallel Mode
- About Fragmentation
- About Disk Group Balance
- About Oracle ASM Disk Repair Timer

Related Topics

- Oracle Exadata Best Practices (My Oracle Support Doc ID 757552.1)

6.7.1 About Exadata Hybrid Columnar Compression

With Exadata Hybrid Columnar Compression, the database stores the same column for a group of rows together.

Storing the database column values together improves the effectiveness of compression algorithms. The data block does not store the rows in the row-major format. The database compresses data manipulated by any SQL operation, although compression levels are higher for direct path loads.

Exadata Hybrid Columnar Compression provides a very high degree of size reduction. Database operations work transparently against compressed objects, so no application changes are required.

You can specify the following types of Exadata Hybrid Columnar Compression, depending on your requirements:

- Warehouse compression: This type of compression is optimized for query performance, and is intended for data warehouse applications.
- Archive compression: This type of compression is optimized for maximum compression levels, and is intended for historic data and data that does not change.

Assume that you apply Exadata Hybrid Columnar Compression to a `daily_sales` table. At the end of every day, the table is populated with items and the number sold, with the item ID and date forming a composite primary key. A row subset is shown in the following table.

Table 6-22 Sample Table daily_sales

Item_ID	Date	Num_Sold	Shipped_From	Restock
1000	01-JUN-07	2	WAREHOUSE1	Y
1001	01-JUN-07	0	WAREHOUSE3	N
1002	01-JUN-07	1	WAREHOUSE3	N
1003	01-JUN-07	0	WAREHOUSE2	N
1004	01-JUN-07	2	WAREHOUSE1	N
1005	01-JUN-07	1	WAREHOUSE2	N

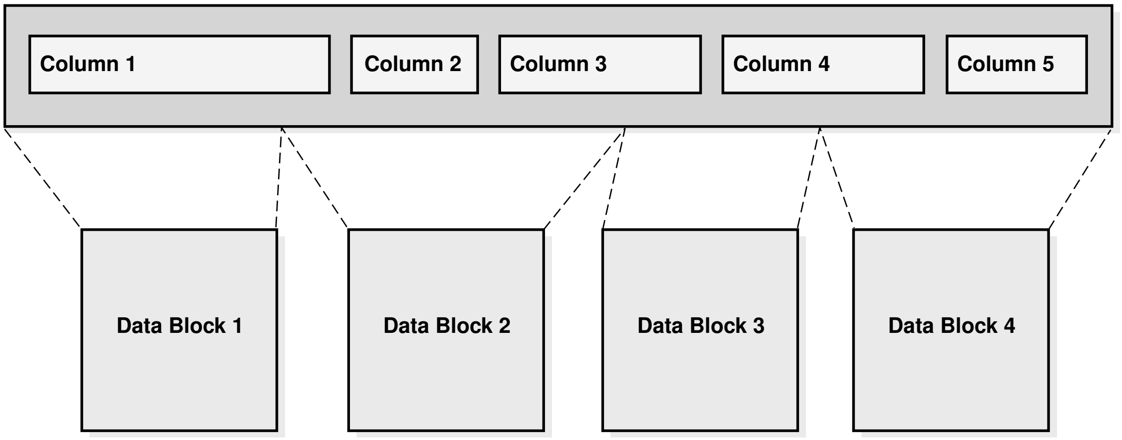
The database stores a set of rows in an internal structure called a compression unit. For example, assume that the rows in the previous table are stored in one unit. Exadata Hybrid Columnar Compression stores each unique value from column 4 with metadata that maps the values to the rows. Conceptually, the compressed value can be represented as:

```
WAREHOUSE1WAREHOUSE3WAREHOUSE2
```

The database then compresses the repeated word `WAREHOUSE` in this value by storing it once and replacing each occurrence with a reference. If the reference is smaller than the original word, then the database achieves compression. The compression benefit is particularly evident for the `Date` column, which contains only one unique value.

As shown in the following illustration, each compression unit can span multiple data blocks. The values for a particular column may or may not span multiple blocks.

Figure 6-1 Compression Unit



Exadata Hybrid Columnar Compression has implications for row locking. When an update occurs for a row in an uncompressed data block, only the updated row is locked. In contrast, the database must lock all rows in the compression unit if an update is made to any row in the unit. Updates to rows using Exadata Hybrid Columnar Compression cause rowids to change.

Note:

When tables use Exadata Hybrid Columnar Compression, Oracle DML locks larger blocks of data (compression units) which may reduce concurrency.

Oracle Database supports four methods of table compression.

Table 6-23 Table Compression Methods

Table Compression Method	Compression Level	CPU Overhead	Applications
Basic compression	High	Minimal	DSS
OLTP compression	High	Minimal	OLTP, DSS
Warehouse compression	Higher (compression level depends on compression level specified (LOW or HIGH))	Higher (CPU overhead depends on compression level specified (LOW or HIGH))	DSS
Archive compression	Highest (compression level depends on compression level specified (LOW or HIGH))	Highest (CPU overhead depends on compression level specified (LOW or HIGH))	Archiving

Warehouse compression and archive compression achieve the highest compression levels because they use Exadata Hybrid Columnar Compression technology. Exadata Hybrid Columnar Compression technology uses a modified form of columnar storage instead of row-major storage. This enables the database to store similar data together, which improves the effectiveness of compression algorithms. Because Exadata Hybrid Columnar Compression requires high CPU overhead for DML, use it only for data that is updated infrequently.

The higher compression levels of Exadata Hybrid Columnar Compression are achieved only with data that is direct-path inserted. Conventional inserts and updates are supported, but result in a less compressed format, and reduced compression level.

The following table lists characteristics of each table compression method.

Table 6-24 Table Compression Characteristics

Table Compression Method	CREATE/ALTER TABLE Syntax	Direct-Path Insert	DML
Basic compression	`COMPRESS [BASIC]` `COMPRESS` and `COMPRESS BASIC` are equivalent	Yes	Yes **Note:**Inserted and updated rows are uncompressed.
OLTP compression	`COMPRESS FOR OLTP`	Yes	Yes

Table 6-24 (Cont.) Table Compression Characteristics

Table Compression Method	CREATE/ALTER TABLE Syntax	Direct-Path Insert	DML
Warehouse compression	`COMPRESS FOR QUERY [LOW|HIGH]`	Yes	Yes High CPU overhead. **Note:**Inserted and updated rows go to a block with a less compressed format and have lower compression level.
Archive compression	`COMPRESS FOR ARCHIVE [LOW|HIGH]`	Yes	Yes **Note:** Inserted and updated rows are uncompressed. Inserted and updated rows go to a block with a less compressed format and have lower compression level.

The `COMPRESS FOR QUERY HIGH` option is the default data warehouse compression mode. It provides good compression and performance. The `COMPRESS FOR QUERY LOW` option should be used in environments where load performance is critical. It loads faster than data compressed with the `COMPRESS FOR QUERY HIGH` option.

The `COMPRESS FOR ARCHIVE LOW` option is the default archive compression mode. It provides a high compression level and good query performance. It is ideal for infrequently-accessed data. The `COMPRESS FOR ARCHIVE HIGH` option should be used for data that is rarely accessed.

A compression advisor, provided by the `DBMS_COMPRESSION` package, helps you determine the expected compression level for a particular table with a particular compression method.

You specify table compression with the `COMPRESS` clause of the `CREATE TABLE` command. You can enable compression for an existing table by using these clauses in an `ALTER TABLE` statement. In this case, only data that is inserted or updated is compressed after compression is enabled. Similarly, you can disable table compression for an existing compressed table with the `ALTER TABLE...NOCOMPRESS` command. In this case, all data that was already compressed remains compressed, and new data is inserted uncompressed.

- Determining If a Table Is Compressed
  Query the `*_TABLES` or `*_TAB_PARTITIONS` data dictionary views to determine whether a table or partitioned is compressed.
- Determining Which Rows are Compressed
  When Exadata Hybrid Columnar Compression tables are updated, the rows change to a lower level of compression.
- Changing Compression Level
  The compression level can be changed for a partition, table, or tablespace.

- Importing and Exporting Exadata Hybrid Columnar Compression Tables
  You can use the impdp and expdp commands to import and export Exadata Hybrid Columnar Compression tables.
- Restoring an Exadata Hybrid Columnar Compression Table
  The compressed table backup can be restored to a system that supports Exadata Hybrid Columnar Compression, or to a system that does not support Exadata Hybrid Columnar Compression.

6.7.1.1 Determining If a Table Is Compressed

Query the *_TABLES or *_TAB_PARTITIONS data dictionary views to determine whether a table or partitioned is compressed.

- Query the *_TABLES data dictionary views to determine table compression.

  In the *_TABLES data dictionary views, compressed tables have ENABLED in the COMPRESSION column.

  ```
  SQL> SELECT table_name, compression, compress_for FROM user_tables;

  TABLE_NAME       COMPRESSION   COMPRESS_FOR
  ---------------- ------------  -----------------
  T1               DISABLED
  T2               ENABLED       BASIC
  T3               ENABLED       OLTP
  T4               ENABLED       QUERY HIGH
  T5               ENABLED       ARCHIVE LOW
  ```

  For partitioned tables, the COMPRESSION column is NULL in the *_TABLES data dictionary views.

- For partitioned tables, query the *_TAB_PARTITIONS data dictionary views.

  The COMPRESSION column of the *_TAB_PARTITIONS views indicates the table partitions that are compressed. The COMPRESS_FOR column indicates the compression method in use for the table or partition.

  ```
  SQL> SELECT table_name, partition_name, compression, compress_for
    FROM user_tab_partitions;

  TABLE_NAME  PARTITION_NAME   COMPRESSION   COMPRESS_FOR
  ----------- ---------------- -----------
  ------------------------------
  SALES       Q4_2004          ENABLED       ARCHIVE HIGH
    ...
  SALES       Q3_2008          ENABLED       QUERY HIGH
  SALES       Q4_2008          ENABLED       QUERY HIGH
  SALES       Q1_2009          ENABLED       OLTP
  SALES       Q2_2009          ENABLED       OLTP
  ```

6.7.1.2 Determining Which Rows are Compressed

When Exadata Hybrid Columnar Compression tables are updated, the rows change to a lower level of compression.

For example, the compression level might change from COMP_FOR_QUERY_HIGH to COMP_FOR_OLTP or COMP_NOCOMPRESS.

By sampling the table rows, you can determine the percentage of rows that are no longer at the higher compression level.

- Use the following query to determine the compression level of a row:

```
DBMS_COMPRESSION.GET_COMPRESSION_TYPE (
   ownname    IN    VARCHAR2,
   tabname    IN    VARCHAR2,
   row_id     IN    ROWID)
  RETURN NUMBER;
```

You can use ALTER TABLE or MOVE PARTITION to set the rows to a higher compression level. For example, if 10 percent of the rows are no longer at the highest compression level, then you might alter or move the rows to a higher compression level.

Related Topics

- GET_COMPRESSION_TYPE Function

6.7.1.3 Changing Compression Level

The compression level can be changed for a partition, table, or tablespace.

An example of when you might want to change the compression level is: A company uses warehouse compression for its sales data, but sales data older than six months is rarely accessed. If the sales data is stored in a table that is partitioned based on the age of the data, then the compression level for the older data can be changed to archive compression to free up disk space.

- To change the compression level of a partitioned table use the DBMS_REDEFINITION package.

  This package performs online redefinition of a table by creating a temporary copy of the table which holds the table data while it is being redefined. The table being redefined remains available for queries and DML statements during the redefinition. The amount of free space for online table redefinition depends on the relative compression level for the existing table, and the new table. Ensure you have enough hard disk space on your system before using the DBMS_REDEFINITION package.

- To change the compression level for a single partition of a partitioned table, use the ALTER TABLE...MODIFY PARTITION command.

- To change the compression level of a non-partitioned table use the ALTER TABLE...MOVE...COMPRESS FOR... command.

  The ALTER TABLE...MOVE command does not permit DML statements against the table while the command is running.

- To change the compression level for a tablespace, use the `ALTER TABLESPACE` command.

Related Topics

- *Oracle Database Administrator’s Guide*
- *Oracle Database PL/SQL Packages and Types Reference*

6.7.1.4 Importing and Exporting Exadata Hybrid Columnar Compression Tables

You can use the `impdp` and `expdp` commands to import and export Exadata Hybrid Columnar Compression tables.

Exadata Hybrid Columnar Compression tables can be imported using the `impdp` command of the Data Pump Import utility. By default, the `impdp` command preserves the table properties and the imported table is Exadata Hybrid Columnar Compression table. The tables can also be exported using the `expdp` command.

On tablespaces not supporting Exadata Hybrid Columnar Compression, the `impdp` command fails with the following error:

```
ORA-6430: hybrid columnar compression is only supported in tablespaces residing on
Exadata storage
```

You can import the Exadata Hybrid Columnar Compression table as an uncompressed table using the `TRANSFORM:SEGMENT_ATTRIBUTES=n` option clause of the `impdp` command.

An uncompressed or OLTP-compressed table can be converted to Exadata Hybrid Columnar Compression format during import. To convert a non-Exadata Hybrid Columnar Compression table to an Exadata Hybrid Columnar Compression table, do the following:

1. Specify default compression for the tablespace using the `ALTER TABLESPACE ... SET DEFAULT COMPRESS` command.
2. Override the `SEGMENT_ATTRIBUTES` option of the imported table during import.

Related Topics

- Oracle Data Pump Import
- ALTER TABLESPACE

6.7.1.5 Restoring an Exadata Hybrid Columnar Compression Table

The compressed table backup can be restored to a system that supports Exadata Hybrid Columnar Compression, or to a system that does not support Exadata Hybrid Columnar Compression.

- When restoring a table with Exadata Hybrid Columnar Compression to a system that supports Exadata Hybrid Columnar Compression, restore the file using Oracle Recovery Manager (RMAN).
- When an Exadata Hybrid Columnar Compression table is restored to a system that does not support Exadata Hybrid Columnar Compression, you must convert the table from Exadata Hybrid Columnar Compression to OLTP compression or a uncompressed format.

Use the following steps to convert an Exadata Hybrid Columnar Compression table to OLTP compression, or to an uncompressed format.

1. Ensure there is sufficient storage in the non-Oracle Exadata Storage Server environment to hold the data in uncompressed or OLTP compression format.
2. Use RMAN to restore the Exadata Hybrid Columnar Compression tablespace.
3. Alter the table compression from Exadata Hybrid Columnar Compression to `NOCOMPRESS`:

   ```
   ALTER TABLE table_name MOVE NOCOMPRESS
   ```

4. If the table partitioned, then alter the compression method for each partition:

   ```
   ALTER TABLE table_name MOVE PARTITION partition_name NOCOMPRESS
   ```

   Change each partition separately.
5. Use the following command to move the data in parallel:

   ```
   ALTER TABLE table_name MOVE NOCOMPRESS PARALLEL
   ```

6. Change the data compression from Exadata Hybrid Columnar Compression to `OLTP COMPRESS`:

   ```
   ALTER TABLE table_name MOVE COMPRESS for OLTP
   ```

Related Topics

- Recovering Tables and Table Partitions from RMAN Backups
- ALTER TABLE

6.7.2 About SQL Processing Offload

To optimize the performance of queries that do table and index scans, the database can offload data search and retrieval processing to the storage cell. This feature is managed by the CELL_OFFLOAD_PROCESSING and CELL_OFFLOAD_PLAN_DISPLAY initialization parameters.

- CELL_OFFLOAD_PROCESSING
- CELL_OFFLOAD_PLAN_DISPLAY

- CELL_OFFLOAD_PROCESSING
  The CELL_OFFLOAD_PROCESSING initialization parameter enables SQL processing offload to Oracle Exadata Storage Server.
- CELL_OFFLOAD_PLAN_DISPLAY
  The database parameter `CELL_OFFLOAD_PLAN_DISPLAY` determines whether the SQL `EXPLAIN PLAN` command displays the predicates that can be evaluated by Oracle Exadata System Software as `STORAGE` predicates for a given SQL command.
- CELL_OFFLOAD_DECRYPTION

6.7.2.1 CELL_OFFLOAD_PROCESSING

The CELL_OFFLOAD_PROCESSING initialization parameter enables SQL processing offload to Oracle Exadata Storage Server.

When the value of the parameter is set to `TRUE`, predicate evaluation can be offloaded to cells. The default value of the parameter is `TRUE`. If the parameter is set to `FALSE` at the session or system level, then the database performs all the predicate evaluation with cells serving blocks. You can set CELL_OFFLOAD_PROCESSING dynamically with the SQL `ALTER SYSTEM` or `ALTER SESSION` commands, for example:

```
SQL> ALTER SESSION SET CELL_OFFLOAD_PROCESSING = TRUE;
```

The CELL_OFFLOAD_PROCESSING parameter can also be set with the `OPT_PARAM` optimizer hint to enable or disable predicate filtering for a specific SQL command.

- To disable CELL_OFFLOAD_PROCESSING for a SQL command:

  ```
  SELECT /*+ OPT_PARAM('cell_offload_processing' 'false') */ COUNT(*)
  FROM EMPLOYEES;
  ```

- To enable CELL_OFFLOAD_PROCESSING for a SQL command:

  ```
  SELECT /*+ OPT_PARAM('cell_offload_processing' 'true') */ COUNT(*) FROM
  EMPLOYEES;
  ```

> **Note:**
>
> The CELL_OFFLOAD_PROCESSING initialization parameter cannot be used to compare the performance of Oracle Exadata Storage Server with conventional storage. Even when CELL_OFFLOAD_PROCESSING is set to `FALSE`, Oracle Exadata Storage Server has many advantages over conventional storage. Oracle Exadata Storage Server is highly optimized for fast processing of large queries. It has no bottlenecks at the controller or other levels inside the cell. Oracle Exadata System Software uses a modern scale-out architecture and a state-of-the-art InfiniBand network that has much higher throughput than conventional storage networks. Oracle Exadata System Software is tightly integrated with the Oracle Database, and has unique capabilities for setup, execution, monitoring, diagnostics, resource management, and corruption prevention.

Related Topics

- *Oracle Database SQL Language Reference*

6.7.2.2 CELL_OFFLOAD_PLAN_DISPLAY

The database parameter CELL_OFFLOAD_PLAN_DISPLAY determines whether the SQL EXPLAIN PLAN command displays the predicates that can be evaluated by Oracle Exadata System Software as STORAGE predicates for a given SQL command.

The values for the CELL_OFFLOAD_PLAN_DISPLAY parameter are AUTO, ALWAYS, or NEVER. The default value is AUTO.

- AUTO instructs the SQL EXPLAIN PLAN command to display the predicates that can be evaluated as STORAGE only if a cell is present and if a table is on the cell.
- ALWAYS produces changes to the SQL EXPLAIN PLAN command based on Oracle Exadata System Software, whether or not Oracle Exadata System Software is present or the table is on the cell. You can use this setting to see what can be offloaded to Oracle Exadata Storage Server before migrating to Oracle Exadata Storage Server.
- NEVER produces no changes to the SQL EXPLAIN PLAN command for Oracle Exadata System Software.

You can set the CELL_OFFLOAD_PLAN_DISPLAY parameter dynamically with the SQL ALTER SYSTEM or ALTER SESSION commands. For example:

```
SQL> ALTER SESSION SET cell_offload_plan_display = ALWAYS;
```

Related Topics

- Using the SQL EXPLAIN PLAN Command with Oracle Exadata System Software
  The SQL EXPLAIN PLAN command was enhanced to display the predicates that are evaluated on Oracle Exadata Storage Server.

6.7.2.3 CELL_OFFLOAD_DECRYPTION

The CELL_OFFLOAD_DECRYPTION initialization parameter enables decryption offload to Oracle Exadata Storage Servers. This decryption applies to both encrypted tablespaces and encrypted columns. When the value of the parameter is set to TRUE, decryption can be offloaded to cells. The default value of the parameter is TRUE. If the parameter is set to FALSE at the system level, then the database performs all decryption with cells serving blocks. You can set CELL_OFFLOAD_DECRYPTION dynamically with the SQL ALTER SYSTEM command. For example:

```
SQL> ALTER SYSTEM SET CELL_OFFLOAD_DECRYPTION = FALSE;
```

6.7.3 About SQL Tuning Advisor

SQL Tuning Advisor takes one or more SQL statements as input and uses the Automatic Tuning Optimizer to perform SQL tuning on the statements.

The output of SQL Tuning Advisor is in the form of advice or recommendations, along with a rationale for each recommendation and its expected benefit. SQL Tuning Advisor provides information about the following:

- Missing and stale statistics
- Better execution plans

- Better access paths and objects
- Better SQL statements

System statistics measure the performance of CPU and storage so that the optimizer can use these inputs when evaluating plans. The database automatically gathers default system statistics, called *noworkload statistics*, at the first instance startup. System statistics ensure the optimizer is aware of Oracle Exadata Database Machine performance.

Use the following SQL command to see if Oracle Exadata-specific statistics have been gathered.

```
SELECT pname, PVAL1  FROM aux_stats$ WHERE pname='MBRC';
```

If the PVAL1 is null or not set, then the system statistics have not been gathered manually and the default system statistics are being used.

It is not recommended to gather system statistics in Exadata mode for *all* Exadata environments. Oracle recommends using the defaults for system statistics in most cases. If you have acceptable performance with your current execution plans, then do not gather system statistics in Exadata mode. If the cardinality estimates in your execution plans are accurate, but suffer from the optimizer over-estimating the cost of a full table scan where the full scan performs better, then you should gather system statistics in Exadata mode. For new applications or deployments where the impact can be assessed from the beginning, and dealt with easily if there is a problem, you can gather system statistics in Exadata mode.

The following SQL command gathers system statistics in Exadata mode:

```
exec dbms_stats.gather_system_stats('EXADATA');
```

If your new application or new deployment testing reveals that the default system statistics performed as well or better than using manually collected system statistics, you can reset system statistics to their default values using the DBMS_STATS.DELETE_SYSTEM_STATS procedure. After executing this procedure, shut down and then reopen the database.

> **Note:**
>
> Oracle Exadata System Software must be using the latest patch bundle.

Related Topics

- *Oracle Database 2 Day DBA*
- *Oracle Database SQL Tuning Guide*

6.7.4 About SQL Monitor

Oracle Database has SQL monitor which allows you to monitor SQL statements as they are run.

The SQL monitor provides extensive statistics for queries, and works with serial and parallel queries. By default, SQL monitoring is automatically started when a SQL statement runs parallel queries, or when the statement has consumed at least 5 seconds of CPU or I/O time in a single execution. To monitor shorter queries, use the `MONITOR` hint.

The `REPORT_SQL_MONITOR` function builds a report in text, HTML or XML with the monitoring information collected about the execution of the SQL statement.

Related Topics

- Real-Time SQL Monitoring and Real-Time Database Operations
- REPORT_SQL_MONITOR Function

6.7.5 About Quality of Service Management

Oracle Exadata Quality of Service Management (QoS Management) allows system administrators to manage application service levels hosted on Oracle Exadata Database Machines. Using a policy-based architecture, QoS Management correlates accurate run-time performance and resource metrics, analyzes the data with its expert system to identify bottlenecks, and produces recommended resource adjustments to meet and maintain performance objectives under dynamic load conditions. When sufficient resources are not available, QoS Management preserves the most business critical objectives at the expense of the less critical ones.

In conjunction with Oracle Clusterware Cluster Health Monitor, QoS Management Memory Guard detects servers that are at risk of failure due to memory over-commitment. It responds to such conditions by automatically preventing new connections, and preserving existing workloads. QoS Management restores connectivity when sufficient memory is available.

Related Topics

- Introduction to Oracle Database QoS Management

6.7.6 About Fault Isolation

Oracle Exadata System Software has the ability to learn from the past events to avoid a potential fatal error.

When a faulty SQL statement caused a crash of the server in the past, Oracle Exadata System Software quarantines the SQL statement so that when the faulty SQL statement occurs again, Oracle Exadata System Software does not allow the SQL statement to perform Smart Scan. This reduces the chance of server software crashes, and improves storage availability. The following types of quarantine are available:

- SQL Plan: Created when Oracle Exadata System Software crashes while performing Smart Scan for a SQL statement. As a result, the SQL Plan for the SQL statement is quarantined, and Smart Scan is disabled for the SQL plan.
- Disk Region: Created when Oracle Exadata System Software crashes while performing Smart Scan of a disk region. As a result, the 1 MB disk region being scanned is quarantined and Smart Scan is disabled for the disk region.
- Database: Created when Oracle Exadata System Software detects that a particular database causes instability to a cell. Instability detection is based on the

number of SQL Plan Quarantines for a database. Smart Scan is disabled for the database.

- Cell Offload: Created when Oracle Exadata System Software detects some offload feature has caused instability to a cell. Instability detection is based on the number of Database Quarantines for a cell. Smart Scan is disabled for all databases.
- Cell-to-Cell Offload: See "Quarantine Manager Support for Cell-to-Cell Offload Operations".

When a quarantine is created, alerts notify administrators of what was quarantined, why the quarantine was created, when and how the quarantine can be dropped manually, and when the quarantine is dropped automatically. All quarantines are automatically removed when a cell is patched or upgraded.

CellCLI commands are used to manually manipulate quarantines. For instance, the administrator can manually create a quarantine, drop a quarantine, change attributes of a quarantine, and list quarantines.

- Quarantine Manager Support for Cell-to-Cell Offload Operations

Related Topics

- Using the CellCLI Utility
  You use the Cell Control Command-Line Interface (CellCLI) utility to manage Oracle Exadata System Software.
- ADR Alert Messages
- Quarantine Manager Support for Cell-to-Cell Offload Operations
- *Oracle Database Administrator's Guide*

6.7.6.1 Quarantine Manager Support for Cell-to-Cell Offload Operations

Minimum Exadata software required: 12.2.1.1.0

Quarantine manager support is enabled for rebalance and high throughput writes in cell-to-cell offload operations. If Exadata detects a crash during these operations, the offending operation will be quarantined, and Exadata will fall back to using non-offloaded operations.

These types of quarantines are most likely caused by incompatible versions of CELLSRV. If such quarantines occur on your system, contact Oracle Support Services.

For rebalance operations, the quarantine is based on the ASM cluster ID. Rebalance will continue using the fallback path, which is slower.

For high throughput writes that originated from a database, the quarantine is based on a combination of ASM cluster ID and database ID.

For high throughput writes that originated from a CDB or PDB, the quarantine is based on a combination of ASM cluster ID and container database ID.

To identify these types of quarantine, run the `LIST QUARANTINE DETAIL` command and check the value of the `quarantineType` attribute. Values for this attribute for these quarantines are `ASM_OFFLOAD_REBALANCE` and `HIGH_THROUGHPUT_WRITE`. For the `HIGH_THROUGHPUT_WRITE` type there is a database case and a CDB case.

The `LIST QUARANTINE` statement produces output that looks like the following:

For rebalance:

```
CellCLI> list quarantine detail
 name:                   2
 asmClusterId:           b6063030c0ffef8dffcc99bd18b91a62
 cellsrvChecksum:        9f98483ef351a1352d567ebb1ca8aeab
 clientPID:              10308
 comment:                None
 crashReason:
ORA-600[CacheGet::process:C2C_OFFLOAD_CACHEGET_CRASH]
 creationTime:           2016-06-23T22:33:30-07:00
 dbUniqueID:             0
 dbUniqueName:           UnknownDBName
 incidentID:             1
 quarantineMode:         "FULL Quarantine"
 quarantinePlan:         SYSTEM
 quarantineReason:       Crash
 quarantineType:         ASM_OFFLOAD_REBALANCE
 remoteHostName:         slc10vwt
 rpmVersion:             OSS_MAIN_LINUX.X64_160623
```

For high throughput writes that originated from database:

```
CellCLI> list quarantine detail
 name:                   10
 asmClusterId:           b6063030c0ffef8dffcc99bd18b91a62
 cellsrvChecksum:        9f98483ef351a1352d567ebb1ca8aeab
 clientPID:              8377
 comment:                None
 crashReason:
ORA-600[CacheGet::process:C2C_OFFLOAD_CACHEGET_CRASH]
 creationTime:           2016-06-23T23:47:01-07:00
 conDbUniqueID:          0
 conDbUniqueName:        UnknownDBName
 dbUniqueID:             4263312973
 dbUniqueName:           WRITES
 incidentID:             25
 quarantineMode:         "FULL Quarantine"
 quarantinePlan:         SYSTEM
 quarantineReason:       Crash
 quarantineType:         HIGH_THROUGHPUT_WRITE
 remoteHostName:         slc10vwt
 rpmVersion:             OSS_MAIN_LINUX.X64_160623
```

For high throughput writes that originated from the CDB (differences noted in bold):

```
CellCLI> list quarantine detail
 name:                   10
 asmClusterId:           eff096e82317ff87bfb2ee163731f7f7
 cellsrvChecksum:        9f98483ef351a1352d567ebb1ca8aeab
 clientPID:              17206
 comment:                None
 crashReason:
ORA-600[CacheGet::process:C2C_OFFLOAD_CACHEGET_CRASH]
```

```
creationTime:             2016-06-24T12:59:06-07:00
conDbUniqueID:            4263312973
conDbUniqueName:          WRITES
dbUniqueID:               0
dbUniqueName:             UnknownDBName
incidentID:               25
quarantineMode:           "FULL Quarantine"
quarantinePlan:           SYSTEM
quarantineReason:         Crash
quarantineType:           HIGH_THROUGHPUT_WRITE
remoteHostName:           slc10vwt
rpmVersion:               OSS_MAIN_LINUX.X64_160623
```

6.7.7 About Fast Disk Scan Rates

To achieve fast disk scan rates, it is important to lay out segments with at least 4 MB of contiguous space. This allows disk scans to read 4 MB of data before performing another seek at a different location on disk. To ensure segments are laid out with 4 MB of contiguous space, set the Oracle ASM allocation unit size to 4 MB, and ensure data file extents are also at least 4 MB. The allocation unit can be set with the disk group attribute AU_SIZE when creating the disk group.

The following SQL command creates a disk group with the allocation unit set to 4 MB. The compatible.rdbms attribute is set to 11.2.0.2 in order to support both release 11.2.0.2 and release 11.2.0.3 databases in a consolidated environment.

```
SQL> CREATE DISKGROUP data NORMAL REDUNDANCY
     DISK 'o/*/data_CD*'
     ATTRIBUTE 'compatible.rdbms' = '11.2.0.2',
               'compatible.asm' = '11.2.0.3',
               'content.type' = 'data',
               'cell.smart_scan_capable' = 'TRUE',
               'au_size' = '4M';
```

Related Topics

- *Oracle Database SQL Language Reference*

6.7.8 About Indexes and Scan Rates

In the past, storage required indexes for good performance. Oracle Exadata System Software can have superior scan rates without using indexes.

Review the application execution plans that use indexes to determine if they would run faster with Oracle Exadata System Software scans. To determine if a scan would be faster when there is no index, make the index invisible to the optimizer. An invisible index is maintained by DML operations, but it is not used by the optimizer.

To make the index invisible, use the following command, where *index_name* is the name of the index:

```
ALTER INDEX index_name INVISIBLE;
```

6.7.9 About Automatic Extent Management

Table extent size is managed automatically in locally-managed tablespaces using the AUTOALLOCATE option. This option automatically increases the size of the extent depending on segment size, available free space in the tablespace, and other factors. The extent size starts at 64 KB and increases to 1 MB for the segment. The extent can increase to 8 MB if the segment size exceeds 64 MB. For a large table, the extent size automatically increases to accommodate the table.

Related Topics

- *Oracle Database SQL Language Reference*

6.7.10 About Minimum Extent Size

Extent size is based on the size of the segment, not the table. A large table that has many partitions can have many small segments. For example, a 100 GB table that is partitioned into 1000 partitions has an average segment size of 100 MB. Therefore, it is good practice to ensure that large tables with many partitions use large extents. To do this, you can provide a hint to the database when the table is created or use an instance level initialization parameter. The following option can be used to hint to the database to allocate large extents for a table:

- The INITIAL storage parameter sets the starting extent size for tables in locally-managed tablespace. The database makes best effort to size every extent in the segment to be at least the extent size defined by the INITIAL storage parameter. For tables that are large, set INITIAL to 8 MB. This storage parameter affects newly-created segments. It is not possible to alter the INITIAL parameter for objects that have already been created.

Related Topics

- *Oracle Database SQL Language Reference*

6.7.11 About SQL*Loader in Parallel Mode

When SQL*Loader is used to load data in parallel, the server uses temporary segments to load data, and then merge the segments into the base segment during transaction commit. For heavily partitioned tables, do not use SQL*Loader to load data. Instead, put the data in a flat file, and use parallel INSERT direct load to load from the flat file using the external tables feature of the database.

Related Topics

- *Oracle Database SQL Language Reference*

6.7.12 About Fragmentation

A table is considered to be fragmented when the number of small extents in the table is high relative to the size of the table. When the number of small extents is high, then the metadata used to map the segment space is bloated, and the user data is not optimally laid out on disk. This impacts the performance of most of the operations including subsequent load, scan, DDL and DML operations. If the extent size are significantly less than 8 MB, then the segment must be reorganized. Large objects must have at least 8 MB extent sizes. There are several methods to reorganize objects. During reorganization, the storage parameter INITIAL may be set to at least 8 MB for the target object. If the table is not partitioned and very large, then consider partitioning it.

> **See Also:**
>
> *Oracle Database Administrator's Guide* for additional information about the following:
>
> - Reclaiming space
> - Redefinition of tablespaces

6.7.13 About Disk Group Balance

Files should be equally balanced across all disks. The following queries and script can be used to check disk group balance:

- To check I/O balance, query the `V$ASM_DISK_IOSTAT` view before and after running a large SQL statement. For example, if a large query has a lot of reads, then the read column and the read_bytes column should be approximately the same for all disks in the disk group.
- To check all mounted disk groups, run the script available in My Oracle Support Note 367445.1. My Oracle Support is available at

  `http://support.oracle.com`

6.7.14 About Oracle ASM Disk Repair Timer

The Oracle ASM disk repair timer represents the amount of time a disk can remain offline before it is dropped by Oracle ASM. While the disk is offline, Oracle ASM tracks the changed extents so the disk can be resynchronized when it comes back online. The default disk repair time is 3.6 hours. If the default is inadequate, then the attribute value can be changed to the maximum amount of time it might take to detect and repair a temporary disk failure. The following command is an example of changing the disk repair timer value to 8.5 hours for the `DATA` disk group:

```
ALTER DISKGROUP data SET ATTRIBUTE 'disk_repair_time' = '8.5h'
```

The `disk_repair_time` attribute does not change the repair timer for disks currently offline. The repair timer for those offline disks is either the default repair timer or the repair timer specified on the command line when the disks were manually set to offline. To change the repair timer for currently offline disks, use the `OFFLINE` command and specify a repair timer value. The following command is an example of changing the disk repair timer value for disks that are offline:

```
ALTER DISKGROUP data OFFLINE DISK data_CD_06_cell11 DROP AFTER 20h;
```

> **Note:**
>
> When the disk repair time value is increased, the vulnerability of a double failure is increased.

Related Topics

- *Oracle Automatic Storage Management Administrator's Guide*

6.8 Using the SQL EXPLAIN PLAN Command with Oracle Exadata System Software

The SQL `EXPLAIN PLAN` command was enhanced to display the predicates that are evaluated on Oracle Exadata Storage Server.

- Example of using EXPLAIN PLAN with Exadata Storage Servers
  You can use EXPLAIN PLAN with Oracle Exadata Database Machine to verify SQL offloading occurs.
- About Predicate Evaluation Offloading and EXPLAIN PLAN
  The database parameter `CELL_OFFLOAD_PLAN_DISPLAY` must be set to `AUTO` or `ALWAYS` for `EXPLAIN PLAN` to display predicates pushed to Oracle Exadata Storage Server.
- SQL Storage Clause CELL_FLASH_CACHE for Exadata Smart Flash Cache
  The `CELL_FLASH_CACHE` storage clause attribute allows the override of the automatic caching policy when the `COMPATIBLE` parameter is greater than or equal to 11.2.0.2.
- Determining which SQL Operators and Functions are Supported by Oracle Exadata System Software
  One of the primary factors in determining whether Oracle Exadata System Software performs predicate evaluation is the SQL operator or function referenced in the predicate.

Related Topics

- Generating and Displaying Execution Plans
- EXPLAIN PLAN

6.8.1 Example of using EXPLAIN PLAN with Exadata Storage Servers

You can use EXPLAIN PLAN with Oracle Exadata Database Machine to verify SQL offloading occurs.

Example 6-17 Using the EXPLAIN PLAN Command

In this example, the `storage` predicate in the plan output indicates the predicates evaluated by Oracle Exadata System Software. The storage predicate is shown in bold.

```
SQL> EXPLAIN PLAN FOR
  SELECT t.prod_id, v.exp1, t2_prod_id, t2_amount_sold
  FROM   sales t, v1 v
  WHERE  t.prod_id = v.prod_id AND t.cust_id = v.cust_id
    AND  t.prod_id != 45
    AND  v.amount_sold * v.quantity_sold > 10000;

Explained.

SQL>
SQL> SELECT PLAN_TABLE_OUTPUT FROM TABLE(DBMS_XPLAN.DISPLAY());

PLAN_TABLE_OUTPUT
---------------------------------------------------------------------------
------------------------
Plan hash value: 2267424675

--------------------------------------------------
| Id  | Operation                   | Name       |
--------------------------------------------------
|   0 | SELECT STATEMENT            |            |
|*  1 |  HASH JOIN                  |            |
|*  2 |   HASH JOIN                 |            |
|*  3 |    TABLE ACCESS STORAGE FULL| SALES      |
|*  4 |    TABLE ACCESS STORAGE FULL| SALES      |
|*  5 |   TABLE ACCESS STORAGE FULL | SALES      |
--------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - access("T"."CUST_ID"="T2"."CUST_ID" AND
              "T1"."PROD_ID"="T2"."PROD_ID" AND
"T1"."CUST_ID"="T2"."CUST_ID")
   2 - access("T"."PROD_ID"="T1"."PROD_ID")
   3 - storage("T1"."PROD_ID"<200 AND
              "T1"."AMOUNT_SOLD"*"T1"."QUANTITY_SOLD">10000 AND
"T1"."PROD_ID"<>45)
       filter("T1"."PROD_ID"<200 AND
              "T1"."AMOUNT_SOLD"*"T1"."QUANTITY_SOLD">10000 AND
"T1"."PROD_ID"<>45)
   4 - storage("T"."PROD_ID"<200 AND "T"."PROD_ID"<>45)
       filter("T"."PROD_ID"<200 AND "T"."PROD_ID"<>45)
   5 - storage("T2"."PROD_ID"<200 AND "T2"."PROD_ID"<>45)
       filter("T2"."PROD_ID"<200 AND "T2"."PROD_ID"<>45)
```

6.8.2 About Predicate Evaluation Offloading and EXPLAIN PLAN

The database parameter `CELL_OFFLOAD_PLAN_DISPLAY` must be set to `AUTO` or `ALWAYS` for `EXPLAIN PLAN` to display predicates pushed to Oracle Exadata Storage Server.

Predicate evaluation is not offloaded to Oracle Exadata Storage Server in the following cases:

- The `CELL_OFFLOAD_PROCESSING` parameter is set to `FALSE`.
- The table or partition being scanned is small.

- The optimizer does not use direct path read.
- A scan is performed on a clustered table.
- A scan is performed on an index-organized table.
- A fast full scan is performed on compressed indexes.
- A fast full scan is performed on reverse key indexes.
- The table has row dependencies enabled or the `rowscn` is being fetched.
- The optimizer wants the scan to return rows in `ROWID` order.
- The command is `CREATE INDEX` using `nosort`.
- A `LOB` or `LONG` column is being selected or queried.
- A `SELECT ... VERSIONS` query is done on a table.
- A query that has more than 255 columns referenced and heap table is uncompressed, or Basic or OLTP compressed. However such queries on Exadata Hybrid Columnar Compression-compressed tables are offloaded.
- The tablespace is encrypted, and the `CELL_OFFLOAD_DECRYPTION` parameter is set to `FALSE`. In order for Oracle Exadata System Software to perform decryption, Oracle Database needs to send the decryption key to Oracle Exadata Storage Server. If there are security concerns about keys being shipped across the network to Oracle Exadata Storage Server, then disable the decryption feature.
- The tablespace is not completely stored on Oracle Exadata Storage Server.
- The predicate evaluation is on a virtual column.

Example 6-18 Using UTLXPLP to Display the EXPLAIN PLAN Command

This example shows the use of `UTLXPLP` to display the `EXPLAIN PLAN` command output when `CELL_OFFLOAD_PLAN_DISPLAY` is enabled. The `storage` predicates (shown in bold) in the plan output indicates the predicates evaluated by Oracle Exadata System Software.

```
SQL> ALTER SESSION SET CELL_OFFLOAD_PLAN_DISPLAY = ALWAYS;
Session altered.

SQL> EXPLAIN PLAN FOR SELECT COUNT(*),SUM(creditlim) FROM s_customer1,
s_region1
     WHERE s_customer1.zip = s_region1.zip AND region_code = 'NE' ;
Explained.

SQL> @?/rdbms/admin/utlxplp

...
Predicate Information (identified by operation id):
---------------------------------------------------

PLAN_TABLE_OUTPUT
---------------------------------------------------------------------------
-----
   5 - access("S_CUSTOMER1"."ZIP"="S_REGION1"."ZIP")
  10 - storage("REGION_CODE"='NE')
       filter("REGION_CODE"='NE')
```

```
15 - storage(SYS_OP_BLOOM_FILTER(:BF0000,"S_CUSTOMER1"."ZIP"))
     filter(SYS_OP_BLOOM_FILTER(:BF0000,"S_CUSTOMER1"."ZIP"))
```

6.8.3 SQL Storage Clause CELL_FLASH_CACHE for Exadata Smart Flash Cache

The CELL_FLASH_CACHE storage clause attribute allows the override of the automatic caching policy when the COMPATIBLE parameter is greater than or equal to 11.2.0.2.

In addition, this storage clause attribute can be used to specify how specific database objects are cached in Exadata Smart Flash Cache. The CELL_FLASH_CACHE *storage_clause* option supports the following values:

- NONE: The value ensures that the database object buffers are never cached in Exadata Smart Flash Cache. This allows the user to reserve the flash cache space for useful and frequently-accessed objects.
- DEFAULT: This value specifies that database objects are cached with the default LRU (least recently used) of Exadata Smart Flash Cache. This is the default value for CELL_FLASH_CACHE when the storage clause is omitted from the command.
- KEEP: This value specifies the database object buffers remain cached in the flash cache as long as the flash cache is large enough.

Example 6-19 Using CREATE TABLE with CELL_FLASH_CACHE

The storage clause can be provided during the CREATE and ALTER commands for a table or other objects. This example shows the CREATE TABLE command for CELL_FLASH_CACHE use with the storage clause.

```
CREATE TABLE ptable (c1 number, c2 clob) TABLESPACE TBS_1
          PARTITION BY RANGE(c1) ( PARTITION p1 VALUES LESS THAN (100)
             TABLESPACE TBS_2 STORAGE (CELL_FLASH_CACHE DEFAULT),
          PARTITION p2 VALUES LESS THAN (200) TABLESPACE TBS_3
             STORAGE (CELL_FLASH_CACHE KEEP));

CREATE TABLE tkbcsrbc (c1 number, l1 clob)
           lob (l1) STORE AS securefile
              (cache nologging STORAGE (CELL_FLASH_CACHE NONE))
           PCTFREE 0 TABLESPACE tbs_93 STORAGE
              (initial 128K next 128K pctincrease 0);
```

Example 6-20 Using ALTER TABLE with CELL_FLASH_CACHE

For objects where altering the storage clause is allowed, the ALTER command can be used with CELL_FLASH_CACHE, as shown in this example.

```
ALTER TABLE tkbcsrbc STORAGE( CELL_FLASH_CACHE DEFAULT);

ALTER TABLE tkbcsrbc MODIFY LOB (l1) (STORAGE (CELL_FLASH_CACHE KEEP));
```

Example 6-21 Using Views to Query CELL_FLASH_CACHE Storage Clause

The CELL_FLASH_CACHE storage clause attribute can be queried using database views based on the object involved.

```
SELECT TABLESPACE_NAME, TABLE_NAME, CELL_FLASH_CACHE FROM user_tables
WHERE table_name='TKBCSRBC';

SELECT CELL_FLASH_CACHE FROM ALL_INDEXES WHERE index_name='TKBCIDX';
```

6.8.4 Determining which SQL Operators and Functions are Supported by Oracle Exadata System Software

One of the primary factors in determining whether Oracle Exadata System Software performs predicate evaluation is the SQL operator or function referenced in the predicate.

Oracle Exadata System Software supports many SQL operators and functions. However, there are also some operators which are not supported. The dynamic performance view V$SQLFN_METADATA has been enhanced to indicate whether a given SQL operator or function is supported by Oracle Exadata System Software. The OFFLOADABLE column in the view has been added to provide this information. A YES value indicates that the operator is supported, while a NO value indicates the operator is not supported.

6.9 Enabling or Disabling In-Memory Columnar Caching on Storage Servers

In-Memory Columnar Caching on cells is enabled by default when the INMEMORY_SIZE is configured. You do not need to do anything to get this enhancement.

The Database In-Memory format cache offers a significant boost to the amount of data held in Database In-Memory format formats and to Smart Scan performance over and above that offered by the pure columnar Hybrid Columnar Compression (HCC) format.

This feature is available if you have licensed the Oracle Database In-Memory option.

You do not need to use any DDL to enable this behavior. However, if you need to override the default behavior, you can use a new DDL keyword CELLMEMORY with the ALTER TABLE command.

```
ALTER TABLE table_name  [ [ NO ] CELLMEMORY [ MEMCOMPRESS FOR [ QUERY |
CAPACITY ] [ LOW | HIGH ] ]
```

Options and Clauses	Usage Description
NO CELLMEMORY	Indicates a table is ineligible for the rewrite from 12.1.0.2 columnar flash cache format into the 12.2 Database In-Memory format.

Options and Clauses	Usage Description
CELLMEMORY and CELLMEMORY MEMCOMPRESS FOR CAPACITY	Allows a table to be cached in the default Oracle Database 12.2 In-Memory format. You only need to use this clause to undo a previously specified NO CELLMEMORY statement or to change the specified compression level.
CELLMEMORY MEMCOMPRESS FOR QUERY	This option indicates that In-Memory column store data should be compressed less than if MEMCOMPRESS FOR CAPACITY was specified. This option can provide an increase in query performance but requires almost twice as much flash space.
LOW and HIGH	Not implemented at this time.

Example 6-22 Using CELLMEMORY and INMEMORY Options on the Same Table

You can use both INMEMORY and CELLMEMORY on the same table. For example:

```
CREATE TABLE t (c1 NUMBER) INMEMORY CELLMEMORY MEMCOMPRESS FOR QUERY;
```

Specifying both options is useful for when you have a low priority table that is unlikely to get loaded in memory. By also specifying CELLMEMORY you still get columnar performance.

7
Using the CellCLI Utility

You use the Cell Control Command-Line Interface (CellCLI) utility to manage Oracle Exadata System Software.

CellCLI provides many of the features that are provided with SQL*Plus, including the use of script files.

- Overview of the CellCLI Utility
  The Cell Control Command-Line Interface (CellCLI) utility is the command-line administration tool for Oracle Exadata System Software.
- About CellCLI Administration Commands
  CellCLI administrative commands do not act directly on objects.
- About CellCLI Object Commands
  This topic describes the CellCLI object commands, object types, and object attributes.
- About CellCLI Object Types
  There are serveral Oracle Exadata System Software object types that can be used with CellCLI object commands.
- About Leading Zeros in Object Names
  If an object name begins with a 0, you must use quotes to avoid parsing errors.
- About CellCLI Object Attributes
  Each CellCLI object has a set of attributes that are assigned when the object is created or altered.
- CellCLI Command Reference
  CellCLI has both administrative and object commands.

Related Topics

- Using the dcli Utility
  The dcli utility facilitates centralized management across an Oracle Exadata System Software realm.

7.1 Overview of the CellCLI Utility

The Cell Control Command-Line Interface (CellCLI) utility is the command-line administration tool for Oracle Exadata System Software.

CellCLI runs on each cell to enable you to manage an individual cell. You use CellCLI to start and stop the cell, to manage cell configuration information, to enable or disable cells, and to manage objects in the cell environment.

The command-line utility is already installed when Oracle Exadata Storage Server is shipped.

7.1.1 Starting CellCLI

You can start CellCLI from the operating system command line on the cell that you want to manage or remotely from a network-attached client using Secure Shell (SSH).

Syntax

The command-line syntax is as follows:

```
CellCLI [port_number] [-n] [-m] [-xml] [-v | -vv | -vvv] [-x] [-e command]
```

The *port_number* in the preceding command specifies the HTTP port number of the Management Server (MS) for the cell. If the *port_number* argument is omitted, then the CellCLI utility uses the value assigned to the `HTTP_PORT` variable in the `cellinit.ora` file on the cell. If the `HTTP_PORT` variable in the `cellinit.ora` file is not set, then the port number defaults to `8888`.

Options

The following options can be used with the CellCLI command:

- `-n` — Runs the CellCLI utility in non-interactive mode. This option suppresses the command prompt and disables the command-line editing features.
- `-m` — Runs CellCLI monitor (read-only) mode.
- `-xml` — Causes output to be displayed in XML format for the Oracle Enterprise Manager.
- `-v`, `-vv`, and `-vvv` — Sets the log level. The `-v` option is for fine, `-vv` is for finer, and `-vvv` is for the finest level.
- `-x` — Suppresses the banner.

- `-e` *`command`* — Runs the specified CellCLI command. CellCLI exits after running the command. For example:

  ```
  $ CellCLI -e list cell detail
  $ CellCLI -e "list celldisk attributes name where name   -
    like '.*cell01'"
  ```

Authentication

CellCLI does not have a login parameter or a connect command. CellCLI uses the cell operating system authentication. The directory from which CellCLI is invoked is the default directory for unqualified file access in CellCLI `SPOOL` and `START` commands.

7.1.2 Understanding Command Syntax and Options for CellCLI

This topic describes the syntax and command options for CellCLI.

CellCLI syntax is as follows:

```
{admin-command | object-command object} [options] ;
```

In the preceding syntax, the following arguments are used:

- *admin-command* is an administrative action.
- *object-command* is an action performed on an object.
- *object* is an object or target on which a command performs an action.
- *options* extend the use of a command combination to include additional parameters for the command.

When using the CellCLI utility, the following rules apply:

- Commands, objects, and options are not case-sensitive except where explicitly stated, such as in string patterns used in filtering strings with the `LIKE` operator.
- Use single quotation marks or double quotation marks around the name of an object that includes spaces or punctuation. The use of quotation marks should match. For example, `"this is incorrect'` is incorrect because the first mark is double quotation marks, and the second is a single quotation mark.
- The current, local cell is the cell to which all CellCLI commands apply.
- A semicolon (;) is optional at the end of a CellCLI command.
- A hyphen (-) is used at the end of a line to continue a long command onto the next line. If you are using hyphens in names or to denote negative values, it must be immediately followed by an alphanumerical value.

7.1.3 Reserved Words

If any of the reserved keywords are used as values in commands, then they must be enclosed in quotation marks.

The following are CellCLI reserved words:

```
ABORT
ACTIVE
ACTIVEREQUEST
ALERTDEFINITION
ALERTHISTORY
ALL
ALTER
ASSIGN
BBU
BMC
CALIBRATE
CATPLAN
CELL
CONFIGUREBMC
CREATE
DBPLAN
DESCRIBE
DETAIL
DROP
EXPORT
FLASHCACHE
FLASHCACHECONTENT
FOR REPLACEMENT
FORCE
GRIDDISK
IGNORE REDUNDANCY
IMPORT
INACTIVE
IORMPLAN
KEY
LED
LIST
LUN
MAIL
MEMORY
METRICDEFINITION
METRICCURRENT
METRICHISTORY
MS
NULL
OFF
ON
PHYSICALDISK
PRIVILEGE
REALM
RESTART
RS
```

```
RULE
SHUTDOWN
SNMP
STARTUP
THRESHOLD
USER
VALIDATE
```

7.1.4 CellCLI Command-Line Editing

Most of the command editing features of CellCLI are similar to modern shells, such as `bash` and `tcsh`.

The CellCLI utility supports command-line history and editing, similar to Berkeley Software Distribution (BSD) `editline` and GNU `readline` functionality.

7.1.5 CellCLI Input and Output Options

Oracle Exadata System Software command-line utilities read commands from standard input and write output to standard output.

You can use the host operating system options for redirecting input and output to compose and process command scripts. For example, you can perform the following redirection:

```
$ CellCLI < command-script-in  > results-out
```

In this example, the output from CellCLI commands in the *`command-script-in`* file are written to the *`results-out`* file.

7.1.6 Comments in CellCLI Scripts

You can add single-line comments to CellCLI scripts using multiple formats.

You can begin the comment line with `REMARK`, `REM` or `--` (two hyphens).

For example, the following are valid syntax for comments:

```
REMARK This is a comment
REM This is a comment
-- This is a comment
```

7.1.7 Line Continuation in CellCLI Commands

To continue a long command on to the next line, insert a hyphen (-) at the end of the line.

After inserting the hyphen, press **Enter**, and continue typing the command.

For example:

```
CellCLI> LIST CELLDISK WHERE name LIKE 'CD_04.*' -
         ATTRIBUTES name, status, comment
```

7.2 About CellCLI Administration Commands

CellCLI administrative commands do not act directly on objects.

The following administration commands are available with CellCLI:

- EXIT
- HELP
- QUIT
- SET
- SPOOL
- START and @

Note:

The `celladmin` user should be used to run all services on the cell.The `cellmonitor` user is for monitoring purposes. The `cellmonitor` user can run the following commands:

- `DESCRIBE`
- `EXIT`
- `HELP`
- `LIST`
- `REMARK`
- `SET`
- `START`

7.3 About CellCLI Object Commands

This topic describes the CellCLI object commands, object types, and object attributes.

The following CellCLI commands operate on Oracle Exadata System Software objects:

- ALTER
- ASSIGN KEY
- CALIBRATE
- CREATE
- DESCRIBE
- DROP
- EXPORT CELLDISK
- GRANT

- IMPORT CELLDISK
- LIST
- REVOKE

7.4 About CellCLI Object Types

There are serveral Oracle Exadata System Software object types that can be used with CellCLI object commands.

- `ACTIVEREQUEST` — An active request provides a client-centric or application-centric view of client I/O requests that are currently being processed by a cell. The active request object can be used only with the `LIST` command.
- `ALERTDEFINITION` — An alert definition provides a definition for every alert that can be produced on the cell. Alerts are defined on metrics and other sources of alerts.
- `ALERTHISTORY` — An alert history provides a list of alerts that have occurred on the cell.
- `CELL` — Cell refers to the current or local cell. A cell is the server to which disks are attached and on which the CellCLI utility runs.
- `CELLDISK` — Each cell disk is associated with a logical unit number (LUN). One physical disk is associated with each cell disk.
- `DATABASE` — Database refers to an active database instance.
- `DIAGPACK` — A diagpack represents a compressed file under `$LOG_HOME` and contains log files and trace files.
- `FLASHCACHE` — The portion of flash storage allocated for use as a cache.
- `FLASHCACHECONTENT` — List of all objects currently cached in the flash cache.
- `FLASHLOG` — The portion of flash storage allocated for storing the Exadata Smart Flash Log.
- `GRIDDISK` — A grid disk is a logical partition of a cell disk. It is exposed on the Oracle Exadata Storage Server network to the database hosts, where it becomes an Oracle ASM disk that can be used to create an Oracle ASM disk group.
- `IBPORT` — The RDMA over InfiniBand Network Fabric ports for Oracle Exadata Storage Server. Does not apply to Oracle Exadata Database Machine X8M systems.
- `IORMPLAN` — An I/O Resource Management (IORM) interdatabase plan is a set of directives that determines allocation of I/O resources to database clients. There is one plan for the cell.
- `IORMPROFILE` — IORM interdatabase plans support profiles to ease management, and configuration of interdatabase plans for hundreds of databases. Profiles introduce a way to allocate I/O resources for a database.
- `KEY` — A key is a unique hexadecimal string that identifies clients for security purposes.
- `LUN` — Logical unit number (LUN) is the address for an individual physical disk device (a single-disk LUN). LUNs are automatically discovered when the cell is started. They are assigned to the corresponding cell disk when the cell disk is first created or when cell disks are discovered after the system is restarted. LUNs that are not yet assigned to a cell disk have a NULL value for the `cellDisk` attribute.

- METRICCURRENT — A current metric describes a set of observations on the current value of an individual metric.
- METRICDEFINITION — A metric definition describes the configuration of a metric.
- METRICHISTORY — A metric history describes a collection of past individual observations of all metric values.
- OFFLOADGROUP — An object that contains modifiable attributes of offload groups, and can be used to restart, start up, and shut down services.
- PHYSICALDISK — A disk is called a physical disk on the cell. Physical disks can be listed, but they are not managed directly by CellCLI. Physical disks are automatically discovered and assigned to the corresponding cell disk when the cell disk is first created or when cell disks are discovered after the system is restarted.
- PMEMCACHE—The portion of PMEM storage allocated for use as a cache.
- PMEMLOG—The portion of PMEM storage allocated for caching redo log data.
- PRIVILEGE — A right or permission assigned to a role.
- QUARANTINE — A quarantine stops faulty SQL statements from performing a Smart Scan. This reduces software crashes, and improves storage availability.
- ROLE — A named group of related privileges.
- SOFTWAREHISTORY — A list of final states for past software updates.
- SOFTWAREUPDATE — An object that contains the software location and time parameters for scheduling software updates
- THRESHOLD — A threshold describes the rules for generating stateful alerts based on a specific metric. The rules include boundary (threshold) values and how long the metric values can violate these boundaries before an alert is generated.
- USER — A person allowed access to the storage servers.

Not all possible command-object combinations are valid. For valid command-object combinations, review the syntax for the specific object command.

Related Topics

- About CellCLI Object Commands
This topic describes the CellCLI object commands, object types, and object attributes.
- Oracle Exadata System Software Components
This section provides a summary of the following Oracle Exadata System Software components.

7.5 About Leading Zeros in Object Names

If an object name begins with a 0, you must use quotes to avoid parsing errors.

The object name can be a number or string in commands that have the following format:

```
<verb> <object_type> <object_name>
```

For example, consider the following command:

```
LIST PHYSICALDISK 0929M00Q8H
```

LIST is the verb, PHYSICALDISK is the object type, and 0929M00Q8H is the object name.

When the command parser analyzes commands, it can parse the object name as a string or as a number. For object names that begin with leading zeros, you need to surround the object name with either single or double quotes to force the parser to treat the object name as a string. Otherwise, the parser will treat the object name as a number and strip the leading zeros from the object name.

For example, the following command produced an error because the object name began with a 0 but was not enclosed within quotation marks:

```
CellCLI> LIST PHYSICALDISK 0929M00Q8H DETAIL
CELL-04524: Physicaldisk object cannot be obtained for physical disk:
929M00Q8H
```

The parser removed the leading zero from 0929M00Q8H, and the command treated the object name as 929M00Q8H. This led to the error.

To avoid this transformation, surround the object name with either single or double quotes:

```
CellCLI> LIST PHYSICALDISK "0929M00Q8H" DETAIL
```

7.6 About CellCLI Object Attributes

Each CellCLI object has a set of attributes that are assigned when the object is created or altered.

Attribute filters and lists are used to specify which attributes and objects are displayed in the output of the LIST command.

All attributes can be displayed, but only some can be modified directly by the user. To display a list of attributes and determine which ones can be modified, use the DESCRIBE command.

- Restrictions on Values of Common Attributes
  Review the following restrictions for the values of attributes used by multiple CellCLI objects.
- Attribute Lists in LIST Command
  You can specify which attributes to display for the LIST command with the optional ATTRIBUTES clause.
- Attribute Filters in LIST Commands
  You can specify which objects to display for the LIST command with the optional WHERE clause.

Related Topics

- LIST
- DESCRIBE

7.6.1 Restrictions on Values of Common Attributes

Review the following restrictions for the values of attributes used by multiple CellCLI objects.

- The value of the `name` attribute must be less than 256 characters and composed only of the following ASCII characters (alphanumeric and underscore only):
  - Lowercase alphabetic characters (a to z)
  - Uppercase alphabetic characters (A to Z)
  - Numbers (0 to 9)
  - Underscore (_)
  - Hyphen (-), but may require the use of double quotes, for example "hyphenated-string"
- The value of the `comment` attribute must be less than 256 characters.

See the syntax of each CellCLI command for any additional restrictions on attribute values.

7.6.2 Attribute Lists in LIST Command

You can specify which attributes to display for the `LIST` command with the optional `ATTRIBUTES` clause.

This syntax of the `ATTRIBUTES` clause is:

```
ATTRIBUTES { ALL | attribute1 [, attribute2] ... }
```

`ALL` displays all possible object attributes for the `LIST` object.

Example 7-1   Listing METRICHISTORY for Specific Attributes

This example shows the `LIST METRICHISTORY` command with the `name` and `metrictype` attributes specified, and the output.

```
LIST METRICHISTORY ATTRIBUTES name, metrictype
         CL_CPUT          Instantaneous
         CL_FANS          Instantaneous
         CL_RUNQ          Instantaneous
         CL_TEMP          Instantaneous
         N_NIC_RCV_SEC    Rate
         N_NIC_TRANS_SEC  Rate
...
```

7.6.3 Attribute Filters in LIST Commands

You can specify which objects to display for the `LIST` command with the optional `WHERE` clause.

This syntax of the WHERE clause is:

```
WHERE attribute-filter1 [AND attribute-filter2, ...]
```

Each *attribute-filterN* has the following syntax:

```
attribute [ NOT | !] operator comparison_value
```

The *attribute* placeholder represents the name of the attribute to use for filtering. The supported types of *operator* are listed in the following table. These operators can be combined with NOT or !.

Table 7-1 Supported Operators in Attribute Filters

Operator	Description
=	Tests for equality between string, status, or numeric attributes. For example: `status NOT = normal`
>	Tests for values greater than the numeric attributes. For example: `size > 139920M`
<	Tests for values less than the numeric attributes. For example: `freeSpace !< 100M`
LIKE	Tests for a regular expression match with a string attribute using case-sensitive matching. For example: `LIKE 'GD_IO_RQ.*'`

When used with the supported operators, *comparison_value* is one of the following data types:

- Numeric
- Literal: Value such as active or normal
- Datetime: Time value supported only for ALERTHISTORY
- String: Value delimited by single quotation marks ('') or double quotation marks ("")
- NULL: Unassigned strings or empty lists

7.7 CellCLI Command Reference

CellCLI has both administrative and object commands.

The following commands are available with the CellCLI utility:

- ALTER
- ASSIGN KEY
- CALIBRATE

- CREATE
- DESCRIBE
- DROP
- EXIT
- EXPORT CELLDISK
- GRANT
- HELP
- IMPORT CELLDISK
- LIST
- QUIT
- REVOKE
- SET
- SPOOL
- START and @

7.7.1 ALTER

Purpose

The ALTER command performs an action on or changes attributes of a single cell object or multiple Oracle Exadata System Software objects. The ALTER command can be used to change an attribute or to take an action upon the object.

Syntax

```
ALTER { object_type object_name [, object_name]... operation
      | attribute_name = attribute_value
        [, attribute_name = attribute_value]...
     }
```

Usage Notes

When multiple objects are the target of an ALTER command, there is the possibility of partial success. If an error occurs, then the command is interrupted, and the remaining objects are not changed.

- ALTER ALERTHISTORY
- ALTER CELL
- ALTER CELLDISK
- ALTER FLASHCACHE
- ALTER GRIDDISK
- ALTER IBPORT
- ALTER IORMPLAN
- ALTER LUN

- ALTER OFFLOADGROUP
- ALTER PHYSICALDISK
- ALTER PMEMCACHE
- ALTER QUARANTINE
- ALTER SOFTWAREUPDATE
- ALTER THRESHOLD
- ALTER USER

Related Topics

- About CellCLI Object Types
  There are serveral Oracle Exadata System Software object types that can be used with CellCLI object commands.

7.7.1.1 ALTER ALERTHISTORY

Purpose

The `ALTER ALERTHISTORY` command changes the attributes of all or specified alert histories.

Syntax

```
ALTER ALERTHISTORY { ALL | alertid1  [,alertid2 ...]}
      examinedBy=user_name
```

Usage Notes

The following arguments can be used with the command:

- *alertidn*: The identifier of the alerts to be changed.
- *user_name*: The name of the user who acknowledged the alert.

Example 7-2   Altering ALERTHISTORY Attributes

This example shows the `ALTER` command used with the `ALERTHISTORY` object to update the `examinedBy` attribute. The `examinedBy` attribute is the only `ALERTHISTORY` attribute that can be modified.

```
CellCLI> ALTER ALERTHISTORY 1671443714 -
                            examinedBy="jdoe"

CellCLI> ALTER ALERTHISTORY ALL examinedBy="jdoe"
```

7.7.1.2 ALTER CELL

- Purpose
- Syntax
- Usage Notes for ALTER CELL Command
- SYSLOG Attributes
- BBU Commands and Attributes

- Configuring Alert Notifications
- Flash Cache Attributes
- Disk Scrubbing Attributes
- CA-Certified Security Certificate Attributes
- Other Attributes
- Examples

Purpose

The `ALTER CELL` command changes the attributes of the cell.

Syntax

```
ALTER CELL  {
  | SHUTDOWN SERVICES { RS | MS | CELLSRV | ALL } [IGNORE REDUNDANCY]
  | RESTART SERVICES  { RS | MS | CELLSRV | ALL } [IGNORE REDUNDANCY]
  | RESTART BMC
  | STARTUP SERVICES  { RS | MS | CELLSRV | ALL }
  | LED {ON | OFF}
  | DONOTSERVICELED {ON | OFF [FORCE]}
  | VALIDATE { MAIL | SNMP | CONFIGURATION }
  | VALIDATE SYSLOGCONF selector.node
  | CONFIGUREBMC
  | BBU { DROP FOR REPLACEMENT | REENABLE }
  | { snmpuser=((user_clause)[,(user_clause)[,..]]) |
snmpuser.name=(user_clause) }
  | attribute_name = attribute_value
        [, attribute_name = attribute_value]...
  }
```

Usage Notes for ALTER CELL Command

The following table lists the arguments and options for the `ALTER CELL` command:

Argument	Options	Description
`SHUTDOWN SERVICES`	`RS` (Restart Server) `MS` (Management Server) `CELLSRV [IGNORE REDUNDANCY]` (Cell Server) `ALL [IGNORE REDUNDANCY]`	All services are shut down, or the specified service is shutdown. When the `SHUTDOWN SERVICES ALL` or `SHUTDOWN SERVICES CELLSRV` option is specified, then the following occurs: • All grid disks are made inactive and stop accepting user I/O. • Oracle ASM takes disks offline. Communication between the cell, the database instance, and Oracle ASM instances is disrupted.

Argument	Options	Description

`RESTART SERVICES`	`RS`
	`MS`
	`CELLSRV [IGNORE REDUNDANCY]`
	`ALL [IGNORE REDUNDANCY]`
		Restart Server (RS) must be running before restarting individual services. If RS is not running, then the only possible commands are `STARTUP SERVICES {RS | ALL}` or `RESTART SERVICES {RS | ALL}`.

`RESTART BMC`	none	Restarts the Baseboard Management Controller (BMC).

`STARTUP SERVICES`	`RS`
	`MS`
	`CELLSRV`
	`ALL`
		All services are started or the specified service in the command is started.
		RS must be running before you can start other individual services. If RS is not running, then the only possible commands are `STARTUP SERVICES {RS | ALL}` or `RESTART SERVICES {RS | ALL}`.

`LED`	`ON` `OFF`	`LED ON` and `LED OFF` operations turn on and off the Fault-Service Required LED. You can manually set the LED to light to indicate that a cell requires maintenance. The LED also turns on automatically if a component fails.
`DONOTSERVICELED`	`ON` `OFF`	Turns the Do Not Service LED on and off. This LED is available with Oracle Exadata Database Machine X7 and later models.

Argument	Options	Description

`VALIDATE`	`MAIL`
	`SNMP`
	`CONFIGURATION`
		The `VALIDATE MAIL` operation sends a test message using the e-mail attributes configured for the cell.
		The `VALIDATE SNMP` operation sends a test message using the SNMP attributes configured for the cell. The `VALIDATE SNMP` TYPE=ASR operation validates Oracle ASR on Oracle Exadata Storage Server.
		The `VALIDATE CONFIGURATION` operation validates the configuration. When the validation is complete and correct, the system responds with `Cell` *`cell_name`* `successfully altered`. If there is a problem, then the system responds with an error message.

`VALIDATE SYSLOGCONF`	*`facility.priority`*	The `VALIDATE SYSLOGCONF` *`facility.priority`* sends a test message for the specified facility and priority.
`CONFIGUREBMC`	none	Configures the BMC for hardware alerts to the local cell so that Management Server (MS) can pick up the alerts.

`BBU`	`DROP FOR REPLACEMENT`
	`ENABLE`
		`BBU DROP FOR REPLACEMENT` drops the hard disk controller battery-backed unit (BBU).
		`BBU REENABLE` re-enables the BBU.

Argument	Options	Description

`snmpuser` or `snmpuser.name`	*user_clause*	Defines the user who receives SNMP alerts. You can either supply a list of user clauses for all SNMP users, or individual user clauses using the attribute `snmpuser.name`. The *user_clause* is of the form:
		`([name=user1,] authProtocol=auth_type,`
		`authPassword=*,`
		`\`
		`privProtocol=priv_type,`
		`privPassword=*) ,`
		`\`
		`(name=user2, authProtocol=auth_type,`
		`authPassword=*,`
		`\`
		`privProtocol=priv_type,`
		` privPassword=*, )`
		If updating a single user, then do not include the phrase `name=user1,` in the *user_clause*.

attributes	N/A	The attributes that can be changed using the `ALTER CELL` command are shown as `modifiable` in Example 7-83 or described below.

The following are additional usage notes for the `ALTER CELL` command:

- You might need to restart, shut down, or start up a cell for the following reasons:
  - Software upgrades
  - Service outages that include any condition under which a cell is not responding to service requests
- Before shutting down services, the system checks the status of the grid disks, as follows, depending on the command:
  - When an `ALTER CELL SHUTDOWN SERVICES CELLSRV` command is issued, the system checks the grid disks to determine if it is safe to shut down the CELLSRV service. If the `asmDeactivationOutcome` attribute returns `yes` for all grid disks, then the system inactivates the grid disks on the cell, and Oracle ASM takes the disks offline. After the offline process completes, CELLSRV is

shut down. If any gird disk cannot be shut down, then the error message CELL-01548 appears, and CELLSRV continues to run.

  - The ALTER CELL SHUTDOWN SERVICES ALL command performs the same checks as described in the preceding bullet for the ALTER CELL SHUTDOWN SERVICES CELLSRV command. If the grid disks return no, then an error message appears, and the current state of MS, RS, and CELLSRV are displayed.
  - The ALTER CELL RESTART SERVICES CELLSRV and ALTER CELL RESTART SERVICES ALL command run the same checks to ensure that all disks can be deactivated safely from Oracle ASM. If the asmDeactivationOutcome attribute returns yes for all grid disks, then the CELLSRV service is restarted immediately, followed by a restart of the MS and RS services, if applicable.

- The IGNORE REDUNDANCY option for the ALTER CELL command is only valid for the following:
  - ALTER CELL SHUTDOWN SERVICES CELLSRV IGNORE REDUNDANCY
  - ALTER CELL SHUTDOWN SERVICES ALL IGNORE REDUNDANCY
  - ALTER CELL RESTART SERVICES CELLSRV IGNORE REDUNDANCY
  - ALTER CELL RESTART SERVICES ALL IGNORE REDUNDANCY

> **Note:**
>
> The IGNORE REDUNDANCY option bypasses the asmDeactivationOutcome attribute checks. Using the IGNORE REDUNDANCY option results in immediate execution of the command. As a result, some Oracle ASM disk groups may be dismounted.

- If you change the cell name, then you must choose a unique cell name.
- The cell attribute values for the following attributes can be set to the default by using two sets of quotation mark characters as the value.
  - alertSummaryInterval
  - bbuLearnSchedule
  - hardDiskScrubInterval
  - notificationMethod
  - notificationPolicy
  - smtpPort
  - traceLevel

SYSLOG Attributes

The syslogconf attribute extends syslog rules for a cell. The attribute can be used to designate that syslog messages be forwarded to a specified management server. On the management server, the forwarded messages are directed to a file, console, or management application, depending on the syslog configuration on the management server. The syntax for configuring this attribute is:

```
syslogconf = ('selector @node' [, 'selector @node']... )
```

In the preceding syntax, *selector* is the message type, and *node* is the specified server. Both variables follow `syslog.conf` standard syntax rules.

- The `facility` option for the `syslogconf` attribute must be one of the following: `auth`, `authpriv`, `cron`, `daemon`, `ftp`, `kern`, `lpr`, `mail`, `mark`, `news`, `security`, `syslog`, `user`, `uucp`, `local0`, `local1`, `local2`, `local3`, `local4`, `local5`, `local6`, `local7`, `none`, and `*`.
- The `priority` option for the `syslogconf` attribute must be one of the following: `alert`, `crit`, `debug`, `emerg`, `err`, `error`, `info`, `notice`, `panic`, `warn`, `warning`, `none`, and `*` (asterisk).

The `ALTER CELL VALIDATE syslogconf` *`selector`* command sends a test log message. The test message is directed as specified by rules in the `/etc/syslog.conf` file. If the `syslogconf` assignment extends the syslog rules, then a test message is forwarded to the specified management servers.

Starting with Oracle Exadata System Software release 19.1.0, you can use the `syslogFormat` attribute to change the standard format for syslog to any format by setting the value to the desired format string. Setting the `syslogFormat` attribute to an empty string removes the format change, reverting the syslog format to the default format. If the format string contains a control character, it must be preceded by a backslash when entering the command.

See Example 7-34 for examples of the syntax.

Starting with Oracle Exadata System Software release 19.3.0, you can use the `syslogFormat` attribute to enable sending syslog in an encrypted format. For the complete configuration steps, refer to Encrypting System Log Information.

BBU Commands and Attributes

The `ALTER CELL BBU DROP FOR REPLACEMENT` command is run prior to replacement of a hard disk controller battery. The command changes the caching policy from `writeback` to `writethrough`, and turns on the locator LED. The new battery is enabled automatically.

The `ALTER CELL BBU REENABLE` command is run when a battery is removed and then the same battery is re-inserted. The command changes the caching policy from `writethrough` to `writeback`, and turns off the locator LED.

The `bbuLearnCycleTime` attribute is used to set the start time for the battery learn cycle. After the learn cycle has completed, the attribute reverts to its default quarterly cycle.

The `bbuLearnSchedule` attribute is used to set the next battery learn cycle. The following parameters are used with the `bbuLearnSchedule` attribute:

- `month`: Values are `1` through `12`. The month entered must be within the current month and the next three months. For example, if the `bbuLearnSchedule` attribute is set in February, then the months could be February, March, April or May.

- week: Values are 1 through 5. The value 1 represents the first week of the month, 2 represents the second week, and so on. The week value must be specified when specifying month and day.
- day: Values are 1 through 7. The value 1 represents Sunday, 2 represents Monday, and so on. The day value must be specified when specifying month and week.
- date: Values are 1 through 31. The values represent the days of the month. The default date is 17.
- hour: Values are 0 through 23. The value 0 represents 12:00 a.m., 1 represents 1:00 a.m., and so on.
- minute: Values are 0 to 59. The values represent the minutes in an hour.
- second: Values are 0 to 59. The values represent the seconds in a minute.

Configuring Alert Notifications

To set up the cell to send notifications about alerts, you can configure the following cell attributes:

- smtpServer: SMTP e-mail server used to send alert notifications
- smtpPort: SMTP e-mail server port used to send alert notifications
- smtpUser: SMTP e-mail user name that is displayed in alert notifications
- smtpPwd: Password of the SMTP e-mail user that sends alert notifications
- smtpUseSSL: Specification to use Secure Socket Layer (SSL) authentication
- smtpFrom: SMTP e-mail user name that appears in alert notifications
- smtpFromAddr: SMTP e-mail address from which alert notifications are sent
- smtpToAddr: Address to which e-mail is sent. It can be a comma-delimited list in quotation marks to allow multiple subscribers to alerts.
- snmpSubscriber: List of hosts that subscribe to the SNMP alert notifications
- snmpUser: Defines the user who receives SNMP alerts
- snmpEngineID: An identifier used by the SNMP managers to subscribe to alerts from the storage cells
- notificationMethod: Notification method for alerts
- notificationPolicy: Indicator for severity alerts to be sent to subscribers
- emailFormat: File format for e-mail messages
- emailSubscriber: List of names that subscribe to the alert notifications

The smtpUseSSL attribute enables Secure Socket Layer (SSL) encryption on the e-mail notifications when the attribute is set to true.

The smtpToAddr attribute can be used to set a list of comma-delimited e-mail addresses that are the recipients of the alert notification. The list must be enclosed in quotation marks.

The snmpSubscriber attribute can be set to a list of SNMP targets to which the SNMP alert notification is sent. These targets are specified as follows:

```
snmpSubscriber=(
  (host=host[,port=port][,community=community][,type=user_type]
[,fromIP="ip"][,asrmPort="ASRManager_port"])
[,(host=host[,port=port][,community=community][,type=user_type]
[,fromIP="ip"][,asrmPort="ASRManager_port"])] ...)
```

Note:

- The ALTER CELL ... smnpSubscriber command replaces the existing value set for smnpSubscriber. To add to the list of SNMP targets ensure that you include the existing values in the command. The following message is displayed after running the command:

  ```
  snmpSubscriber <old_value> has been replaced with <new_value>
  ```

  For example:

  ```
  snmpSubscriber ((host=hosta)) has been replaced with
  ((host=hostb))
  ```

The smnpSubscriber attribute uses the following values:

- The host must be specified as either a domain name or an IP address. Enclose the host name or IP address in quotation marks if it contains non-alphanumeric characters.
- The default value for port is 162. This value is optional.
- The default value for community is public. This value is optional.
- The snmpSubscriber types are ASR, v3, and v3ASR. Specifying a type value is optional.
  - The default value for type is NULL.
  - The ALTER CELL snmpSubscriber...type=asr command sets the Oracle ASR destination for Oracle Exadata Storage Server, and its ILOM. Removing all snmpSubscriber entries with type=asr from the SNMP subscriber list disables the trap mechanism for Oracle Exadata Storage Server and its ILOM.
  - The snmpSubscriber with type=asr or type=v3ASR should only be configured to point to Oracle ASR Manager.
  - For types v3 and v3ASR, a snmpUser must be defined, and the user name is provided instead of community.
  - For the v3ASR type, the user must be defined with authProtocol = SHA, and privProtocol = AES. These are the only protocols supported by Oracle ASR Manager. Setting the snmpSubscriber as type v3ASR also sets the ILOM properties and rules for traps sent by ILOM.

  - If type is not specified, then the default is version 1, cell_alert traps. There is no string to specify this type. To use this type, just omit the type field.

- The fromIP field enables you to specify an IP address from which the trap is sent. If this field is not specified, it defaults to the IP address associated with eth0. Use this field if the default IP address is not registered with Oracle ASR Manager. Oracle ASR Manager only processes SNMP traps that are sent from IP addresses that it recognizes.

  The fromIP field is allowed only for SNMP subscribers whose type is either ASR or v3ASR.

  For example:

  ```
  CellCLI> alter cell
  snmpSubscriber=((host=asrhost,port=162,community=public,fromIP="1.1.1.1"
  ,type=ASR))
  ```

  The following example returns an error because the type is not ASR or v3ASR.

  ```
  CellCLI> alter cell
  snmpSubscriber=((host=localhost,port=162,community=public,fromIP="1.1.1.
  1"))
  CELL-00068: The fromIP field is only supported for ASR SNMP subscribers.
  ```

- The asrmPort field enables you to specify the port number on an Oracle ASR Manager machine that MS uses to communicate with Oracle ASR Manager. This port must be the same as the HTTP port of Oracle ASR Manager's HTTP Receiver. You can check this by running asr show_http_receiver on the Oracle ASR Manager machine.

  The asrmPort field is allowed only for SNMP subscribers whose type is either ASR or v3ASR. The default value for this port is 16161.

After startup of the Management Server (MS), the snmpSubscriber list entries with type=asr are added to the ILOM for the cell. This ensures that when an ILOM is replaced, the entries are set for the new ILOM. If the entries are removed from the ILOM, then they must be manually added to the ILOM using the ALTER CELL ... snmpUser= command.

The snmpUser attribute defines the user who receives SNMP alerts. This command can only be run in interactive mode. There are two methods for configuring this attribute.

```
snmpuser=((user_clause)[,(user_clause)[,..]])

snmpuser.name=(user_clause)
```

- If you specify snmpuser, then you must provide a *user_clause* for every configured user. If you omit a user, then that user will no longer receive SNMP alerts. The ((*user_clause*)[,(*user_clause*)[,..]]) string that you provide overwrites the previous string used for the snmpuser attribute.
- If you specify snmpuser.*name*, then you must provide a *user_clause* for only the specified user. This allows you to add, delete, or modify each user individually, without having to supply the entire snmpuser attribute string each time.

- If you use `snmpuser=''`, then all SNMP users are removed. If you use `snmpuser.name=''`, then only the specified user is removed. You cannot remove an SNMP user while it is still referenced by a V3 SnmpSubscriber.

Each method uses a *user_clause*, which has the following format:

```
(([name=user1,] authProtocol=auth_type, authPassword=*,               \
privProtocol=priv_type, privPassword=*) ,                            \
(name=user2, authProtocol=auth_type, authPassword=*,                 \
privProtocol=priv_type, privPassword=*, ), ...)
```

If updating a single user, then do not include the phrase `name=user1,` in the *user_clause*, because you have already supplied the name as part of `snmpuser.name`.

- `name` is the user name.
- Only `*` is allowed for the password values in the command. Passwords are not stored or displayed. Secure hash keys are computed and used for trap authentication and encryption.
- `authProtocol` is the authentication protocol. Options are `MD5` or `SHA`.

  The `authProtocol` must be specified for the `snmpUser` attribute.

  The system prompts for the authentication password. The authentication password must have 8 to 12 alphanumeric characters.
- `privProtocol` is encryption protocol. Options are `none`, `AES`, or `DES`. The default is `none` when the `privProtocol` attribute is not specified.

  The system prompts for an encryption password if the encryption protocol is specified. The password is exactly 8 alphanumeric characters, and they are case sensitive.

The `ALTER CELL snmpEngineID` command is used by the SNMP managers to subscribe to alerts from the storage cells. The `snmpEngineID` parameter can be up to 20 characters. It should be unique for each target within a data center. The default is the cell name. This default is used if the `snmpEngineID` attribute is not set before the SNMP users are defined.

> **Note:**
>
> The engine identifier should not be changed after SNMP users are defined. Any change to an engine identifier causes the user keys to be re-computed, and user passwords must be re-entered.

The `notificationMethod` attribute value can be `mail`, `snmp`, `none`, or a combination of `mail` and `snmp`, such as `notificationMethod='mail,snmp'`. The default value is `mail`.

The `notificationPolicy` attribute value can be `none` or a combination of `critical`, `warning`, or `clear`, such as `notificationPolicy='warning,clear.'`

- The `critical` value refers to hardware-generated alerts or alerts generated by Automatic Diagnostic Repository (ADR) or BMC. The `critical` value also refers to a metric alert when the value exceeds the critical threshold specified in the metric definition.

- The warning value refers to a metric alert when the value exceeds the warning threshold specified in the metric definition.
- The clear value refers to a metric alert when the value is below the threshold boundary after having previously exceeded a warning or critical threshold.
- The maintenance value refers to all hardware-related errors. The hardware errors are reported as "Maintenance" in e-mail message subject lines.

To validate that e-mail messages are successfully sent for cell alerts or events, use the ALTER command with the VALIDATE MAIL option. The validation process sends a test e-mail message to the configured recipient. If that test e-mail message is not received, then an e-mail configuration setting is not valid.

The emailFormat attribute can be html or text. By default, e-mail notifications are sent in HTML format. Change the value to text to receive plain text e-mail notifications.

The ALTER CELL emailSubscriber command sets a list of comma-delimited e-mail addresses that are the recipients of alert notifications for specific alert types. The syntax for this command is:

```
ALTER CELL emailSubscriber = ((email="email_address1",               \
           alertType="alert_type")                                   \
          [, (email="email_address2",alertType="alert_type"), ...])
```

- The e-mail address must be a valid e-mail address. The email parameter is mandatory.
- The alertType parameter specifies the type of alert, and is optional. The alert types are HARDWARE, SOFTWARE, METRIC or ADR. If the alert type is not specified, then the subscription is for all alert types.
- An empty input string removes the current set of subscribers.
- The notification policy must be set before alert notifications can be received. The policy applies to all e-mail subscribers. The notification policy for these alerts are the same as for snmpSubscriber alerts.

Flash Cache Attributes

The flashCacheMode attribute is used to display and set the current value for flash cache. The values are writethrough (the default) or writeback. Note the following about the flashCacheMode attribute:

- If the attribute is modified from writeback to writethrough and there is existing flash cache, then an error is displayed. The flash cache must be flushed and dropped before changing the attribute to writethrough.
- If the attribute is to be modified from writethrough to writeback, then flash cache must be dropped before modifying the attribute.
- Write back caching can be disabled on the grid disks that do not need caching, such as the grid disks in the RECO disk group. This allows other objects to use the cache space.

> **See Also:**
>
> - ALTER GRIDDISK for information about disabling caching on grid disks
> - *Oracle Exadata Database Machine Maintenance Guide* for information about enabling and disabling flash cache

Flash cache compression is available in Oracle Exadata System Software releases 11.2.3.3.0 and later running on only Oracle Exadata Database Machine X3 and X4 storage servers.

The `ALTER CELL flashCacheCompress` command enables or disables flash cache compression. The options are `true` to enable flash cache compression, and `false` to disable flash cache compression. To enable flash cache compression on Oracle Exadata Database Machine X3 and X4 storage servers, use the following command:

```
CellCLI> ALTER CELL flashCacheCompress=true
```

> **Note:**
>
> Oracle Advanced Compression Option is required to enable flash cache compression.

> **See Also:**
>
> *Oracle Exadata Database Machine Maintenance Guide* for additional information about enabling flash cache compression

Flash cache compression must be disabled before a storage server is downgraded to an earlier release.

Disk Scrubbing Attributes

The `ALTER CELL hardDiskScrubStartTime` command sets the start time for proactive resilvering of latent bad sectors. Valid options are a *`date/time`* combination or `now`. The following table shows the approximate time needed to scrub one idle hard disk:

Disk Type	Hard Disk Capacity	Approximate Time for Scrubbing
High performance	600 GB	1 hour
High performance	1.2 TB	2 hours
High capacity	2 TB	4 hours, 40 minutes
High capacity	3 TB	6 hours, 30 minutes
High capacity	4 TB	8 hours
High capacity	8 TB	13 hours

Disk Type	Hard Disk Capacity	Approximate Time for Scrubbing
High capacity	10 TB	14 hours
High capacity	14 TB	18 hours

The `ALTER CELL hardDiskScrubInterval` command sets the interval for proactive resilvering of latent bad sectors. Valid options are `daily`, `weekly`, `biweekly` and `none`. Using the `none` option stops all disk scrubbing.

In the alert log, you may see messages such as `Begin scrubbing celldisk` and `Finished scrubbing celldisk`. These informational messages are expected, and no action is necessary.

If the system is idle, disk scrubbing can drive the disk utilization to 100%. This is expected. Disk scrubbing will throttle itself using IORM based on disk activity. When requests come in, disk scrubbing activity will decrease. Your workloads should not be affected by disk scrubbing.

CA-Certified Security Certificate Attributes

To set up CA-certified security certificates on the cell for use with ExaCLI, use the following attributes:

> **Note:**
>
> The following attributes can be used only if you are running the `ALTER CELL` command from ExaCLI.

- `securityPubKey` - Specifies the URL to the public key file.
- `securityPrivKey` - Specifies the URL to the private key file.
- `securityPrivKeyPW` - Specifies the password to use if the private key file is encrypted.

This example shows configuring the security keys for a storage server, including the prompt for a password after entering the command:

```
exacli -e 'ALTER CELL securityPubKey="http://www.example.com/security/
newkey.crt",  -
                   securityPrivKey="http://www.example.com/security/
newkey.key", -
                   securityPrivKeyPW=*'

password=****************
```

After you upload the CA-certified security certificate, you must restart MS before the new security certificate is visible.

```
CellCLI> ALTER CELL RESTART SERVICES MS
```

> **See Also:**
>
> Using a CA-Certified Security Certificate in *Oracle Exadata Database Machine Maintenance Guide*

Other Attributes

- The `ALTER CELL alertSummaryInterval` command sets the frequency of the open alerts summary e-mail message. The open alerts e-mail message is an HTML document that provides a concise summary of all open issues on a cell even without access to the cell. Valid options are `daily`, `weekly`, `biweekly` and `none`.The default value is `weekly`.
- The `ALTER CELL alertSummaryStartTime` command sets the delivery time for the open alerts summary e-mail message. The command accepts any valid time stamp.
- You can turn off the diagnostic pack attachment to emails by running the following command:

  ```
  alter cell diagPackEmailAttach=FALSE
  ```

  See CREATE DIAGPACK for information about diagnostic packs.
- The `ALTER CELL eighthRack` command enables or disables an Eighth Rack configuration on Oracle Exadata Database Machine X3-2 Quarter Racks or later. The options are `true` to enable the Eighth Rack configuration, and `false` to disable the Eighth Rack configuration. The `ALTER CELL eighthRack=true` command requires that there are no cell disks because enabling the Eighth Rack configures only half of the hard disks and flash capacity. After using this command you must restart Cell Server (CELLSRV) to make the new changes effective and prevent unexpected results.
- The `enableSmartStorage` attribute can be set to `TRUE` to enable the use of Oracle Exadata System Software capabilities such as Smart Scan and Storage Index on Oracle Exadata Storage Server X8-2 Extended (XT) after you have procured the necessary software licenses.
- Starting with Oracle Exadata System Software release 19.1.0, the `httpsAccess` attribute can be used to specify a list of IP addresses or IP subnet masks that control who can access the RESTful service via HTTPs. The value you specify for `httpsAccess` overwrites any previous value. You can use the following values for `httpsAccess`:
  - `ALL` — to allow access to all hosts (Default)
  - `NONE` — to disable the HTTPs port completely
  - `IP1, IP2,..., IP`*`n`* — to only allow access to hosts with IP addresses IP1, IP2,..., IP*n* where IP*n* is a valid IP address in IPv4, IPv4 subnet, IPv6 or IPv4-embedded IPv6 format. You can specify a maximum of 512 IP addresses for the access control list.

  Additionally, instead of a single IP address, you can use the `/` character to specify a range of IP addresses using a subnet mask. For example the range `'192.168.10.0/24'` corresponds to hosts having IP addresses from 192.168.10.1

to 192.168.10.255. If you specify an IP address range, you need to enclose the IP address string in quotes.

See Example 7-33 for an example of the syntax.

- The `ALTER CELL interconnectN=""` command removes the RDMA Network Fabric configuration information for the cell for the specified interface (*N*).

  If the IP address to an RDMA Network Fabric interface is changed, then the command `service openibd restart` must be run as the `root` user before the `service network restart` command.

  After changing an IP address, you must restart all services using the `ALTER CELL RESTART SERVICES ALL` command.

- The `ALTER CELL iotimeoutthreshold` command changes the timeout threshold. If cell I/O takes longer than the defined threshold, then the I/O is canceled, and Oracle ASM redirects the I/O to another mirror copy of the data. Any I/Os issued to the last valid mirror copy of the data are not canceled, even if the timeout threshold is exceeded.

  The default value for `iotimeoutthreshold` is 1000s. The command takes a value, such as `5`, and a unit. The valid unit is `s`, for seconds.

  > **Caution:**
  >
  > Setting the timeout threshold too low can negatively impact system performance. Oracle recommends reviewing the Automatic Workload Repository (AWR) reports of peak I/O loads, and setting the threshold value to a value higher than the peak I/O latency with sufficient safety margin.

- The `ramCacheMode` attribute can be set to `on`, `off`, or `auto`. The default value is `auto` , which means the RAM Cache feature is not enabled. When you modify this attribute, you must restart CELLSRV for the change to take effect.

Examples

Example 7-3 shows how to set the `asrmPort` field for an `snmpSubscriber`.

Example 7-4 shows how to enable and disable the auto diagpack upload feature.

Example 7-5 shows the `ALTER` command with the `CELL` object.

Example 7-6 shows how to set up e-mail notifications for the cell.

Example 7-7 shows how to set the RDMA Network Fabric interconnections.

Example 7-8 shows how to validate the e-mail setup on a cell.

Example 7-9 shows how to validate the Oracle ASR e-mail setup on a cell.

Example 7-10 shows how to change the format of e-mail messages.

Example 7-11 shows how to validate the SNMP setup on a cell.

Example 7-12 shows how to specify the type of e-mail alerts. In the example, one subscriber gets hardware and software alerts, and the other subscriber gets ADR alerts.

Example 7-13 shows how to modify the SNMP user.

Example 7-14 shows how to modify a single SNMP user.

Example 7-15 shows how to unsubscribe from e-mail alerts.

Example 7-16 shows how to validate the configuration on a cell.

Example 7-17 shows a sample error message when configuration on a cell is incorrect.

Example 7-18 shows how to start up and shut down cell services.

Example 7-19 shows how to set the LED on the cell.

Example 7-20 shows how to add a rule using the syslogconf attribute.

Example 7-21 shows how to add and validate a rule with test message.

Example 7-22 shows how to remove the `syslog.conf` rule.

Example 7-23 shows how to set the flash cache mode.

Example 7-24 shows how to set the frequency for the open alerts summary e-mail message to weekly.

Example 7-25 shows how to set the delivery time for the open alerts summary e-mail message.

Example 7-26 shows how to set the battery learn cycle. In the example, the command sets the battery learn cycle to occur January 17 3:00:59, and then the following learn cycles are April 17 3:00:59, July 17 3:00:59, and October 17 3:00:59. The default setting is "`MONTH 1 DATE 17 HOUR 2 MINUTE 0`."

Example 7-27 shows how to enable flash cache compression on a cell in Oracle Exadata Database Machine X4-2.

Example 7-28 shows how to set the disk scrubbing start time.

Example 7-29 shows how to modify the interval for proactive resilvering of latent bad sectors to weekly.

Example 7-30 shows how to reset the `notificationPolicy` attribute to its default value.

Example 7-31 shows to reset the `traveLevel` to the default value.

Example 7-32 shows how to modify the `iotimeoutthreshold` setting.

Example 7-33 shows how to configure an access control list for HTTPs access to the Exadata RESTful service.

Example 7-34 shows how to specify a customized format for syslog.

Example 7-3 Setting the asrmPort for an snmpSubscriber

```
CellCLI> ALTER CELL
snmpSubscriber=((host=host1,port=162,community=public,type=asr,asrmPort=161
61))
```

Example 7-4 Enabling/Disabling Auto Diagpack Upload

You can enable or disable this feature by setting the `diagPackUploadEnabled` attribute on the `cell` object.

Set the attribute to `false` to disable this feature, `true` to enable it. The default is `true`.

```
CellCLI> ALTER CELL diagPackUploadEnabled=FALSE
```

Example 7-5 Altering Cell Name, IP Block, and Cell Number Attributes

```
CellCLI> ALTER CELL name=cell02, cellNumber=2
```

Example 7-6 Configuring E-Mail Notifications for a Cell

```
CellCLI> ALTER CELL smtpServer='my_mail.example.com',              -
                    smtpFromAddr='john.doe@example.com',            -
                    smtpFrom='John Doe',                            -
                    smtpToAddr='jane.smith@example.com',            -
                    snmpSubscriber=((host=host1),(host=host2)),     -
                    notificationPolicy='clear',                     -
                    notificationMethod='mail,snmp'
```

Example 7-7 Setting RDMA Network Fabric Interconnections

For systems that use RDMA over InfiniBand Network Fabric, use a command such as the following:

```
CellCLI> ALTER CELL interconnect1='ib0', interconnect2='ib1'
```

For systems that use RDMA over RoCE Network Fabric, use a command such as the following:

```
CellCLI> ALTER CELL interconnect1='re0', interconnect2='re1'
```

Example 7-8 Validating E-mail on a Cell

```
CellCLI> ALTER CELL VALIDATE MAIL
```

Example 7-9 Validating Oracle ASR E-mail on a Cell

```
CellCLI> ALTER CELL VALIDATE SNMP type=asr
```

Example 7-10 Changing the Format of E-mail Messages

```
CellCLI> ALTER CELL emailFormat='text'
CellCLI> ALTER CELL emailFormat='html'
```

Example 7-11 Validating SNMP on a Cell

```
CellCLI> ALTER CELL VALIDATE SNMP
```

Example 7-12 Specifying the Type of E-mail Alert

```
ALTER CELL emailSubscriber=                                               \
           ((email="email1@example.com",alertType="HARDWARE,SOFTWARE"), \
           (email="email2@example.com",alertType="ADR"))
```

Example 7-13 Modifying the SNMP User

This example shows the initial configuration of a single SNMP user, where the administrator is prompted to enter the passwords.

```
CellCLI> ALTER CELL snmpuser = ((name=ASR, authprotocol=md5,
authpassword=*,    \
                    privprotocol=AES, privpassword=*))
snmpUser ASR authpassword: password
Confirm snmpUser ASR authpassword: password
snmpUser ASR privpassword: password
Confirm snmpUser ASR privpassword: password
```

Example 7-14 Modifying a Single SNMP User

The following code examples show adding an SNMP user, changing that user's password, and then removing that user.

```
## adding users individually
CellCLI> ALTER CELL snmpuser.user2=(authprotocol=SHA,authpassword=*)

snmpUser user2 authpassword: password
Confirm snmpUser user2 authpassword: password

snmpUser ((name=user1, authProtocol=SHA, privProtocol=AES)) has been
replaced with
((name=user1, authProtocol=SHA, privProtocol=AES),(name=user2,
authProtocol=SHA)).
Cell cel01 successfully altered

## changing a password of an existing user
CellCLI> ALTER CELL snmpuser.user2 =
(authprotocol=SHA,authpassword=welcome2)

Cell cel01 successfully altered

## delete a user individually
CellCLI> ALTER CELL snmpuser.user2=''

snmpUser ((name=user1, authProtocol=SHA, privProtocol=AES),(name=user2,
authProtocol=SHA)) has
 been replaced with ((name=user1, authProtocol=SHA, privProtocol=AES)).
Cell cel01 successfully altered
```

Example 7-15 Unsubscribing from E-mail Alerts

```
ALTER CELL emailSubscriber=""
```

Example 7-16 Validating Configuration on a Cell

```
CellCLI> ALTER CELL VALIDATE CONFIGURATION

Cell CD_01_cell01 successfully altered
```

Example 7-17 Checking an Incorrect Configuration on a Cell

```
CellCLI> ALTER CELL VALIDATE CONFIGURATION

CELL-02827: Cell configuration check for hardware and firmware encountered 
the
following issues:

ILOM check has detected the following issue(s):
    Attribute Name : ILOMVersion
    Required       : 3.0.6.10.a r49240
    Found          : 3.0.6.10.a r49385
```

Example 7-18 Starting Up and Shutting Down Cell Services

```
CellCLI> ALTER CELL STARTUP SERVICES CELLSRV
CellCLI> ALTER CELL STARTUP SERVICES ALL

CellCLI> ALTER CELL SHUTDOWN SERVICES MS
CellCLI> ALTER CELL SHUTDOWN SERVICES CELLSRV
CellCLI> ALTER CELL SHUTDOWN SERVICES CELLSRV IGNORE REDUNDANCY
CellCLI> ALTER CELL SHUTDOWN SERVICES ALL
CellCLI> ALTER CELL SHUTDOWN SERVICES ALL IGNORE REDUNDANCY

CellCLI> ALTER CELL RESTART SERVICES CELLSRV
CellCLI> ALTER CELL RESTART SERVICES ALL
CellCLI> ALTER CELL RESTART SERVICES ALL IGNORE REDUNDANCY
```

Example 7-19 Setting the Cell LED Off and On

```
CellCLI> ALTER CELL LED OFF
CellCLI> ALTER CELL LED ON
```

Example 7-20 Using the syslogconf Attribute

```
CellCLI> ALTER CELL syslogconf=('*.err;authpriv.none @loghost', -
         '*.emerg @loghost')
```

Example 7-21 Adding and Validating a Rule

```
CellCLI> ALTER CELL syslogconf=('kern.crit @loghost')
CellCLI> ALTER CELL VALIDATE syslogconf   'kern.crit'
```

Example 7-22 Removing All `syslog.conf` Rules

```
CellCLI> ALTER CELL syslogconf=''
```

Example 7-23 Setting the Mode for Flash Cache

```
CellCLI> ALTER CELL flashcachemode = writeback
```

Example 7-24 Setting the Frequency for the Open Alerts Summary E-mail Message

```
CellCLI> ALTER CELL alertSummaryInterval=weekly
```

Example 7-25 Setting the Time for Open Alerts Message Delivery

```
CellCLI> ALTER CELL alertSummaryStartTime="2013-04-23T12:57:00-06:00"
```

Example 7-26 Setting the Battery Learn Cycle

This example shows how to schedule for the battery learn cycle.

```
CellCLI> ALTER CELL bbuLearnSchedule = "MONTH 1 HOUR 3 SECOND 59"
```

Example 7-27 Enabling Flash Cache Compression

This example shows how to enable flash cache compression for a storage server.

```
CellCLI> ALTER CELL flashCacheCompress=true
```

Example 7-28 Setting the Start Time for Proactive Resilvering of Latent Bad Sectors

This example shows how to set the start time for resilvering the disks of a storage server.

```
CellCLI> ALTER CELL hardDiskScrubStartTime='2013-08-07T21:19:22-07:00'
```

Example 7-29 Setting the Proactive Resilvering of Latent Bad Sectors Interval to be Weekly

This example shows how to set the disk scrubbing interval.

```
CellCLI> ALTER CELL hardDiskScrubInterval=weekly
```

Example 7-30 Setting the Default Value for the notificationPolicy Attribute

This example shows how to set the default value for the `notificationPolicy` attribute.

```
CellCLI> alter cell notificationPolicy=""
```

Example 7-31 Setting the traceLevel Value to its Default Value

This example shows how to set the `traceLevel` value to its default value.

```
CellCLI> ALTER CELL traceLevel=""
```

Example 7-32 Setting the iotimeoutthreshold Value

This example demonstrates how to set the `iotimeoutthreshold` to 5 seconds.

```
CellCLI> ALTER CELL iotimeoutthreshold = '5s'
```

To reset the `iotimeoutthreshold` to the default value, use the following command:

```
CellCLI> ALTER CELL iotimeoutthreshold = ""
```

Example 7-33 Restricting HTTPS Access to the Exadata RESTful Service

This example shows how to configure an access control list for HTTPs access to the Exadata RESTful service. The following command allows HTTPs port access to hosts having IP addresses in the range from 192.168.10.1 to 192.168.10.255.

```
CellCLI> ALTER CELL httpsAccess="192.168.10.0/24"
```

Example 7-34 Setting the Syslog Format to a Custom String Then Reverting to the Default Format

```
CellCLI> ALTER CELL syslogformat="%TIMESTAMP:::date-rfc3339% %HOSTNAME%
%syslogtag%
%syslogseverity-text%:%msg:::sp-if-no-1st-sp%%msg:::drop-last-lf%\\n"

CellCLI> ALTER CELL syslogformat="%TIMESTAMP% %HOSTNAME% %msg%\\n"

CellCLI> ALTER CELL syslogformat=""
```

Related Topics

- CREATE CELL
- Restrictions on Values of Common Attributes
  Review the following restrictions for the values of attributes used by multiple CellCLI objects.
- Increasing the Number of Active Cores on Database Servers

7.7.1.3 ALTER CELLDISK

Purpose

The ALTER CELLDISK command changes the attributes of all cell disks or the specified cell disks.

Syntax

```
ALTER CELLDISK { ALL [FLASHDISK | HARDDISK] | cdisk_name [,
cdisk_name]... }
   {{FLUSH [NOWAIT] | CANCEL FLUSH} |
   { attribute_name = attribute_value
        [, attribute_name = attribute_value]...
   }
```

Usage Notes

The attributes that can be changed with the ALTER command are shown as modifiable in Example 7-84.

- The FLASHDISK option limits the ALTER CELLDISK command to cell disks that are flash disks.
- The HARDDISK option limits the ALTER CELLDISK command to cell disks that are hard disks.
- The FLUSH option synchronizes dirty data from the flash cache to the grid disks. Dirty data is data that has not been synchronized with the grid disk. Synchronization of dirty data can be a lengthy process, depending on the number of bytes to be synchronized. Use the following command to check the progress:

  ```
  LIST CELLDISK ATTRIBUTES name, flushstatus, flusherror
  ```

- The ALTER CELLDISK ... FLUSH command must be run before exporting a cell disk to ensure that the dirty data is flushed from flash cache to the grid disks.
- The FLUSH option stops new data from being cached on the flash cache until CELLSRV restarts, or the flush operation is canceled.
- The CANCEL FLUSH option terminates an earlier flush operation, and reinstates caching.
- When the ALTER CELLDISK ... FLUSH command is run for a flash cell disk, it synchronizes dirty data from the flash cache located on the specified FDOM to cached grid disks. When the command is run for a hard cell disk, it synchronizes dirty data from the flash cache located on all FDOMs to the grid disks located on the specified cell disk.

Example 7-35 Altering Cell Disk Attributes

This example shows how to change cell disk attributes.

```
CellCLI> ALTER CELLDISK cdiska name = CD_01_cell01, -
               comment = 'cdiska is now CD_01_cell01'

CellCLI> ALTER CELLDISK ALL -
```

```
               comment = 'This cell disk is on cell cell01'

CellCLI> ALTER CELLDISK ALL HARDDISK FLUSH NOWAIT

CellCLI > ALTER CELLDISK c9datafile1 CANCEL FLUSH
```

Related Topics

- Restrictions on Values of Common Attributes
  Review the following restrictions for the values of attributes used by multiple CellCLI objects.
- CREATE CELLDISK

7.7.1.4 ALTER FLASHCACHE

Purpose

The `ALTER FLASHCACHE` command stops new data from being cached on the flash cache and then flushes data not synchronized with the grid disks (dirty data) from flash cache to the specified disks.

Syntax

```
ALTER FLASHCACHE { ALL | CELLDISK="cdisk1 [,cdisk2] ..."}
      {SIZE} {FLUSH | CANCEL FLUSH} [NOWAIT] }
```

Usage Notes

> **Note:**
>
> The `FLUSH` option stops new data from being cached on the flash cache until CELLSRV restarts, or the flush operation is canceled with the `ALTER FLASHCACHE CANCEL FLUSH` command.

- The `ALL` option affects all available flash cell disks.
- The `CELLDISK` option allows specific cell disks to be flushed.
- When the `SIZE` option is specified with the `ALL` option, the size is validated. If the size is valid, then all cell disks are dropped, and flash cache is created with the specified size.
- When the `SIZE` option is specified with the `CELLDISK` option, the size is validated. If the size is valid, then the specified cell disks are dropped, and the flash cache is created on the specified cell disks with the specified size.
- The `FLUSH` option synchronizes dirty data from the flash cache to the grid disks. Dirty data is data that has not been synchronized with the grid disk. Synchronization of dirty data can be a lengthy process, depending on the number of bytes to be synchronized. Use the following command to check the progress:

  ```
  LIST CELLDISK ATTRIBUTES name, flushstatus, flusherror
  ```

- The `ALTER FLASHCACHE CELLDISK= ... FLUSH` command does not flush the dirty data when the data cannot be read from the flash cache or written to disk. To flush the dirty data from the flash disk to grid disks use the `ALTER GRIDDISK ... FLUSH` command.
- The `ALTER FLASHCACHE ... FLUSH` command stops new data from being written to the flash cache and then synchronizes all data in the flash cache with the hard disks. As a result, all data is removed from the flash cache. When the flash cache is re-enabled, the flash cache activity metrics are reset.
- The `CANCEL FLUSH` option terminates an earlier flush operation, and reinstates flash caching.
- The `NOWAIT` option allows the `ALTER` command to complete while the flush operation is in progress.
- By default, 5 percent of space on Extreme Flash Exadata X5-2 storage servers is used for write back flash cache. Use the `DROP FLASHCACHE` and `CREATE FLASHCACHE` commands to change the value.

Example 7-36 Flushing Dirty Blocks from Flash Cell Disks

This example shows how to flush dirty blocks from all flash cell disks.

```
CellCLI> ALTER FLASHCACHE ALL FLUSH
Flash cache on FD_00_scac01cel07 successfully altered
Flash cache on FD_01_scac01cel07 successfully altered
Flash cache on FD_02_scac01cel07 successfully altered
...
Flash cache on FD_14_scac01cel07 successfully altered
Flash cache on FD_15_scac01cel07 fsuccessfully altered
```

7.7.1.5 ALTER GRIDDISK

Purpose

The `ALTER GRIDDISK` command changes the attributes of all grid disks or specified grid disks.

> **Caution:**
>
> Before changing the name of a grid disk that belongs to an Oracle ASM disk group, ensure that the Oracle ASM disk group is offline.

Syntax

```
ALTER GRIDDISK { ALL  [FLASHDISK | HARDDISK] |gdisk_name1 [,
gdisk_name2]... }
   { ACTIVE    |INACTIVE | {FLUSH [NOWAIT] | CANCEL FLUSH} |
    attribute_name = attribute_value
        [, attribute_name = attribute_value]...
   [NOWAIT]
```

Usage Notes

The attributes that can be changed with the `ALTER GRIDDISK` command are shown as `modifiable` in Example 7-90.

- The `FLASHDISK` option limits the `ALTER GRIDDISK` command to grid disks that are flash disks.
- The `HARDDISK` option limits the `ALTER GRIDDISK` command to grid disks that are hard disks.
- The `ACTIVE` option notifies CELLSRV to accept I/O as normal for the specified grid disks. The grid disks are visible to the database clients.
- The `INACTIVE` option makes the grid disks visible to the cell administrator, but not visible to the database clients. CELLSRV treats the grid disks as if they were offline. This mode allows management operations on the grid disks. You can do upgrading and testing on the grid disks before making the grid disks visible to database users. This functionality is similar to starting up a database in `RESTRICTED` mode.

  > **Note:**
  >
  > When a grid disk that is currently in use by a database client is made `INACTIVE`, Oracle ASM takes the corresponding Oracle ASM disk offline when I/Os to the disk fail. To make the disk usable again, make the grid disk `ACTIVE` in the cell, and then bring the corresponding Oracle ASM disk back online in Oracle ASM.

- The `FLUSH` option synchronizes dirty data from the flash cache to the grid disks. Dirty data is data that has not been synchronized with the grid disk. Synchronization of dirty data can be a lengthy process, depending on the number of bytes to be synchronized. Use the following command to check the progress:

  ```
  LIST GRIDDISK ATTRIBUTES name, flushstatus
  ```

- The `FLUSH` option stops new data from being cached on the specified grid disks until CELLSRV restarts, or the flush operation is canceled.
- The `FLUSH` option is valid for write back disks, not write through disks.
- The `CANCEL FLUSH` option terminates an earlier d flush operation, and reinstates caching.
- The `NOWAIT` option allows the `ALTER` command to complete while a resize or flush operation continues.
- The length of a grid disk name is limited to 30 characters.
- The `size` attribute can be specified to expand or reduce space allocated to a grid disk. The corresponding Oracle ASM disk must be resized separately.

  The `size` attribute is specified as a number in bytes, unless the suffix `M` (megabytes) or `G` (gigabytes) is included with the number value. Grid disk space is allocated in 16 MB units, referred to as allocation units. The actual size allocated is the size of the largest multiple of allocation units less than or equal to the specified size. The minimum value is 16 MB. Values less than 16 MB are rounded up to 16 MB.

- A grid disk should not be renamed when the grid disk is being accessed. If you try to rename a grid disk when it is being accessed, then the operation fails. You can make the grid disk inactive or dismount the Oracle ASM disk group to stop access to the grid disk before renaming it.
- When an interleaved grid disk is resized, the contents of the grid disk are moved to achieve the interleaved space allocation across the cell disk. The resizing operation can take a few minutes. You can choose to have the data movement proceed as a background process by using the `NOWAIT` option. Use the `LIST GRIDDISK` command to check the status.

> **Note:**
>
> Interleaved grid disks are deprecated in Oracle Exadata System Software release 19.1.0.

- The `cachingPolicy` attribute is used to change the flash caching policy of a grid disk. Before changing the caching policy from `default` to `none`, ensure there is no cached data in flash cache for the grid disk by using the `ALTER GRIDDISK ... FLUSH` command.

  Flash cache is not used with data files placed in a disk group composed of grid disks with their `cachingPolicy` set to `none`.

  OEDA configures the RECO disk group to have its grid disk's `cachingPolicy` set to `none`; therefore flash cache is not used for any data files placed in RECO.

  Write back caching can be disabled on the grid disks that do not need caching, such as the grid disks in the RECO disk group. This allows other objects to use the cache space. Use the following commands to flush the cache, and then set the `cachingPolicy` attribute to `none`:

  ```
  ALTER GRIDDISK grid_disk_name FLUSH
  ALTER GRIDDISK grid_disk_name CACHINGPOLICY="none"
  ```

  To re-enable caching on the grid disk, do the following:

  ```
  ALTER GRIDDISK grid_disk_name CANCEL FLUSH
  ALTER GRIDDISK grid_disk_name CACHINGPOLICY="default"
  ```

  grid_disk_name is the name of the grid disk.

Examples

Example 7-37 Altering Grid Disk Attributes

This example shows the `ALTER` command with the `GRIDDISK` object.

```
CellCLI> ALTER GRIDDISK data1_CD_01_cell01, data2_CD_01_cell01
         comment = "This grid disk is on cell01"

CellCLI> ALTER GRIDDISK ALL INACTIVE
```

Example 7-38 Using the NOWAIT Option

This example shows the NOWAIT option being used to have data movement done as a background process.

```
CellCLI> ALTER GRIDDISK gd0 SIZE=40M NOWAIT
GridDisk gd0 alter in progress
```

Related Topics

- Restrictions on Values of Common Attributes
  Review the following restrictions for the values of attributes used by multiple CellCLI objects.
- Configuring Data Security for Exadata Storage Servers
- LIST CELLDISK
- CREATE GRIDDISK
- ALTER FLASHCACHE

7.7.1.6 ALTER IBPORT

Purpose

The ALTER IBPORT command performs an action on all RDMA over InfiniBand Network Fabric ports, or specified RDMA over InfiniBand Network Fabric ports.

> **Note:**
>
> This command does not apply to Oracle Exadata Database Machine X8M systems.

Syntax

```
ALTER IBPORT {ALL | ibport_name [, ibport_name] ...} RESET COUNTERS
```

Usage Notes

The RESET COUNTERS option resets all counters on the RDMA over InfiniBand Network Fabric port.

Example 7-39 Altering IBPORT Attributes

This example shows the ALTER command with the IBPORT object.

```
CellCLI> ALTER IBPORT ALL RESET COUNTERS

         InfiniBand Port HCA-1:1 successfully altered.
         InfiniBand Port HCA-1:2 successfully altered.

CellCLI> ALTER IBPORT "HCA-1:1" RESET COUNTERS

         InfiniBand Port HCA-1:1 successfully altered.
```

7.7.1.7 ALTER IORMPLAN

Purpose

The ALTER IORMPLAN command updates the interdatabase plan and category plans for the cell. The directives determine the allocation of I/O resources to database clients. The ALTER IORMPLAN command is used to control disk and flash I/Os by database clients.

Syntax

```
ALTER IORMPLAN [dbPlan=(directive1[, directive2]...),]
               [catPlan=( directive1[, directive2]...)]}
               [objective='iorm_objective']
```

- Parameters
- Directive Attributes for dbPlan
- Directive Attributes for catPlan
- Usage Notes
- Examples

Parameters

- dbplan: Specifies the interdatabase plan, allowing you to manage resource allocations among databases. You can activate the interdatabase plan by setting the dbplan parameter. If the dbPlan parameter is not set, then the default is that each database has an equal percentage or share of the resources.
- catplan: Specifies the category plan, allowing you to allocate resources primarily by the category of the work being done. If the catPlan parameter is not set, then the default is that no category plan is used. The resources are divided among databases.
- To deactivate the interdatabase plan, set the dbplan parameter to an empty string. Single quotation marks or double quotation marks can be used to set an empty string. The marks must match, for example "" is correct, but "' is incorrect.
- objective: The objective parameter specifies the optimization mode for IORM. By default, the objective is set to basic because IORM is always minimally enabled. The objective parameter can have the following values:
  - auto - Use this setting to have IORM determine the best mode based on active workloads and resource plans. IORM continuously and dynamically determines the optimization objective, based on the workloads observed, and resource plans enabled. This is the recommended value that users should start with.
  - high_throughput - Use this setting for critical DSS workloads that require high throughput. This setting improves throughput at the cost of I/O latency.
  - low_latency - Use this setting for critical OLTP workloads that require extremely good disk latency. This setting provides the lowest possible latency at the cost of throughput by significantly limiting disk utilization.

  - `balanced` - Use this setting for critical OLTP and DSS workloads. This setting balances low disk latency and high throughput. This setting limits disk utilization of large I/Os to a lesser extent than `low_latency` to achieve a balance between latency and throughput.
  - `basic`: Use this setting to disable I/O prioritization and limit the maximum small I/O latency.

  To enable user-defined plans, the `objective` attribute must be set to a value other than `basic`.

Directive Attributes for dbPlan

The `directive`*N* for a database plan (`dbPlan`) is of the following format:

```
( name=db_name, [share=number] [,level=number, allocation=number]
[,limit=number]
[, flashcache={on|off}] [, flashLog={on|off}] [, flashcachemin=number] [,
flashcachelimit=number]
[, flashcachesize=number] [, pmemcachelimit=number] [,
pmemcachemin=number] [, pmemcachesize=number]
[, asmcluster=asm_cluster_name [, type={ database | profile }] [,
role={ primary | standby }] )
```

- Up to 1024 directives are allowed for each `dbPlan`. These directives include the mandatory `other` database directive.

  > **Note:**
  >
  > Only 32 directives are allowed when using the `level` and `allocation` attributes.

- `name`: The name of the database. The database name cannot start with an underscore (_). For `dbplan`, each database `name` specified must have at least one directive. For example, `(name=sales, share=8)`, `(name=oltpdg, limit=80)`, or `(name=dwh, flashcachesize=50G)`.
  - The `other` value is a distinguished database name. It refers to all clients that are not specifically named in other directives. You must specify at least one directive with `name=other` for `dbPlan` when using the `allocation` attribute or the CellCLI utility returns an error. You can use the `other` directive only with `level` and `allocation`, or `limit` attributes. You cannot use `role`, `asmcluster`, `flashcachemin`, `flashcachelimit`, `flashcachesize`, `pmemcachemin`, `pmemcachelimit`, or `pmemcachesize`attributes with `other`.
  - The `default` value is a distinguished database name. It provides default values for all databases not explicitly named in a `dbPlan` so you do not have to add directives for each database. Any database that does not have an explicit directive uses the values that are specified for the `default` database name. You cannot specify `level`, `allocation`, `asmcluster` or `role` directives for `default`.
- A database plan can be defined using one of the following configurations:
  - `share`: Use the `share` attribute to specify the relative priority for a database. A higher `share` value implies higher priority and more guaranteed I/O resources.

    The `share` value is 1 to 32, with 1 being the lowest share, and 32 being the highest share. The `share` attribute allows IORM to manage up to 1024 databases. The recommended best practice is to use the `share` attribute with `dbplan`.

  - `allocation`: Use the `allocation` attribute to specify percentages for I/O distribution for each database. The `allocation` attribute allows IORM to manage up to 32 databases. You can allocate resources across different levels, using the `level` attribute. `level` can have a value from 1 to 8. For each `level` the total `allocation` value cannot exceed 100. You cannot specify `asmcluster` if also specifying `allocation`
  - `limit`: Use the `limit` attribute to restrict the I/O utilization for a database to a specified utilization limit. The `limit` directive can only be used with `dbPlan`. The `limit` attribute specifies the maximum disk I/O utilization limit for databases. It is a value greater than zero, and less than or equal to 100. This configuration is ideal for pay-for-performance use cases and should not be used to implement fairness.

    If a maximum utilization limit is specified, then excess capacity is never used by the databases. It is possible that the disks are running below full capacity when maximum utilization limits are specified.

    > **Note:**
    >
    > Specifying low `limit` values can have a significant performance impact and is not advisable for regular consolidation use cases.

    > **Note:**
    >
    > The `MAX_UTILIZATION_LIMIT` attribute specifies the disk I/O utilization limit for consumer groups. Refer to *Oracle Database Administrator's Guide* and *Oracle Database PL/SQL Packages and Types Reference* for additional information about this attribute.

- To enable disk I/O management, specify `share`, `limit`,or `level` and `allocation`.
- The `flashcache` attribute is used to prevent a database from using the flash cache. This ensures that flash cache space is reserved for mission-critical databases. The `flashcache` attribute can only be used for `dbPlan`. The `flashcache` directive cannot be set to `OFF` when specifying the `flashcachemin`, `flashcachelimit`, or `flashcachesize` directives.

  > **Note:**
  >
  > If the `flashcache` attribute is not specifically set to `off` for a database, then that database uses flash cache by default.

- The `flashLog` attribute is used to prevent a database from using the flash log. The `flashlog` attribute can only be used for `dbPlan`.

> **Note:**
>
> If the flashLog attribute is not specifically set to off for a database, then that database uses flash log by default.

- Starting with Oracle Exadata System Software release 12.1.2.2.0, the flashcachesize attribute specifies the guaranteed hard limit space in the flash cache for a database. This space in the flash cache is reserved but since this is a hard limit the database cannot exceed its quota even when the flash cache is not full.
- The flashcachemin attribute value specifies the minimum guaranteed space in flash cache for a database.
- The flashcachelimit attribute value specifies the maximum space in flash cache that a database can use. The flashcachelimit attribute is a soft limit that specifies the maximum space in the flash cache that a database can use. If the flash cache is not full, the database can exceed its allocated quota. The flashcachelimit attribute cannot be set to 0.
- The flashcachemin, flashcachelimit, and flashcachesize attributes are applicable to interdatabase plans only and cannot be specified for category plans. The directives specify their values in bytes, unless the suffix M (megabytes), G (gigabytes), or T (terabytes) is specified, for example, 300M, 150G, or 1T. The sum of flashcachemin and flashcachesize across all the directives should be less than the size of flash cache.
- Starting with Oracle Exadata System Software release 19.3.0, the pmemcachesize attribute specifies the guaranteed hard limit space in the PMEM cache for a database. This space in the PMEM cache is reserved, but since this is a hard limit the database cannot exceed its quota even when the PMEM cache is not full.
- The pmemcachemin attribute value specifies the minimum guaranteed space in PMEM cache for a database.
- The pmemcachelimit attribute value specifies the maximum space in PMEM cache that a database can use. The pmemcachelimit attribute is a soft limit that specifies the maximum space in the PMEM cache that a database can use. If the PMEM cache is not full, the database can exceed its allocated quota. The pmemcachelimit attribute cannot be set to 0.
- The pmemcachemin, pmemcachelimit, and pmemcachesize attributes are applicable to interdatabase plans only and cannot be specified for category plans. The directives specify their values in bytes, unless the suffix M (megabytes), G (gigabytes), or T (terabytes) is specified, for example, 300M, 150G, or 1T. The sum of pmemcachemin and pmemcachesize across all the directives should be less than the size of the PMEM cache.
- Starting with Oracle Exadata System Software release 19.1.0, you can use the asmcluster attribute to specify different directives for databases with the same unique name running in different Oracle ASM clusters. The asmcluster attribute value should match the asm field in the cellkey.ora file when setting up Oracle ASM-scoped security.
- The asmcluster attribute is not allowed in default and other dbPlan directives. The asmcluster attribute is not allowed in catPlan directives. You cannot specify allocation if also specifying asmcluster

- The `type` attribute must be set to either `database` or `profile`. If `type` in not specified, then the directive defaults to the `database` directive. The `type` attribute can only be specified for interdatabase plans. The `dbplan` can contain a combination of profile and individual database directives.

  Use `profile` to ease management and configuration of resource plans in large consolidation environments with hundreds of databases. You can specify `share`, `limit`, `flashcache`, `flashlog`, `flashcachemin`, `flashcachelimit`, `flashcachesize`, `pmemcachemin`, `pmemcachelimit`, or `pmemcachesize` attributes for a profile. Databases that map to a particular `profile` inherit the attribute values from the `profile` directive specified in the plan. Set the `db_performance_profile` database parameter to map a database to a `profile`.
- The `role` attribute enables you specify different directives based on the Oracle Data Guard database role. The `role` directive can have the value `primary` or `standby`. The directive is applied only when the databases are in that database role. In the case of a switchover or failover, the new directive values are automatically applied by IORM. If the `role` attribute is not specified, then the directive is applied regardless of the database role. For each `role`, the total `allocation` value cannot exceed 100. The `role` attribute is not allowed in `default` and `other` `dbPlan` directives. The `role` attribute must be the last attribute specified in the directive.

Directive Attributes for catPlan

The `directive`*N* for a category plan (`catPlan`) is of the following format:

```
( name=category_name, [,level=number, allocation=number] )
```

- Up to 28 directives are allowed for each `catPlan`. These directives include the mandatory `other` category directive.
- `name`: the name of the category. The category name cannot start with an underscore (_). For `catplan`, each category `name` specified must have at least one directive. For example, `(name=gold, share=8)`, or `(name=customer, level=1, allocation=80)`.

  The `other` value is a distinguished category name. It refers to all categories that are not specifically named in other directives. You must specify at least one directive with `name=other` for `catplan` or the CellCLI utility returns an error.
- A category plan can be defined using the following configuration:
  - `allocation`: Use the `allocation` attribute to specify percentages for I/O distribution for each category. The `allocation` attribute allows IORM to manage up to 32 categories. You can allocate resources across different levels, using the `level` attribute. `level` can have a value from 1 to 8. For each `level` the total `allocation` value cannot exceed 100.

Usage Notes

- The attributes that can be changed with the `ALTER` command are shown as `modifiable` in Example 7-92.
- To reset an interdatabase plan attribute to the default value, set the attribute to an empty string. Single quotation marks or double quotation marks can be used to set an empty string. The marks must match, for example `""` is correct, but `"'` is incorrect.

- Consider running `ALTER IORMPLAN` commands as scripts because of the length of the commands.
- The following items should be checked before implementing a plan:
  - A `catPlan` must have at least one directive for `other` set.
  - A `dbPlan` must have at least one directive for `other` when using the `allocation` attribute.
  - A `dbPlan` does not use the `other` directive when using the `share` attribute.
  - `pmemcachesize`, `pmemcachemin` or `pmemcachemax` attributes for the `other` directive are not permitted. `flashcachesize`, `flashcachemin` or `flashcachemax` attributes for the `other` directive are not permitted.
  - `pmemcachesize`, `pmemcachemin`, `pmemcachemax`, `flashcachesize`, `flashcachemin` and `flashcachemax` cannot be used in `catPlan`.
  - Allocations are not greater than 100 for each level.
  - There are no more than 32 directives for `dbPlan` if the `allocation` attribute is used.
  - There are no more than 1023 directives for `dbPlan` if the `share` attribute is used.
  - There are no more than 28 directives for a `catPlan`.
  - The `role` directive is only used with `dbPlan`.
  - The `limit` directive is only used with `dbPlan`.
  - Only a single assignment of an attribute is allowed for each database and each profile
  - The value for a `share` directive is not greater than 32.
  - The sum of `share` directives is not greater than 32768.
  - A profile name cannot be OTHER or DEFAULT. Those names are reserved words.
  - The `asmcluster` attribute cannot be used with `allocation`. `asmcluster` is not allowed in `catPlan`, `default`, and dbplan `other` directives.
  - The sum of `pmemcachemin` directives for a plan can't be greater than `pmemcachesize`. The sum of `flashcachemin` directives for a plan can't be greater than `flashcachesize`.
  - The sum of `pmemcachesize` and `pmemcachelimit` can't be less than `pmemcachemin`. The sum of `flashcachesize` and `flashcachelimit` can't be less than `flashcachemin`.
  - The `pmemcachesize` directive cannot be specified if `pmemcachelimit` is specified. The `flashcachesize` directive cannot be specified if `flashcachelimit` is specified.
  - `flashcache`, `flashcachemin`, `flashcachelimit`, `flashcachesize`, `pmemcachemin`, `pmemcachelimit`, and `pmemcachesize` must be at least 4 MB.
  - The `flashcache` attribute is not set to `off` when the `flashcachemin`, `flashcachelimit`, or `flashcachesize` attributes are set.

  - The `pmemcachesize`, `pmemcachemin`, `pmemcachelimit`, `flashcachesize`, `flashcachemin`, or `flashcachelimit` attributes cannot be set to `0`.

Examples

Example 7-40 shows the `ALTER` command with the `IORMPLAN` object.

Example 7-40 and Example 7-41 shows how to set `dbPlan` using the `share` directive. Both examples set the same resource allocations.

Example 7-42 uses the `default` directive to set the share allocation for databases `sales01`, `sales02`, and so on.

Example 7-43 shows how to set flash cache use with the `IORMPLAN` attributes.

Example 7-44 shows how to specify flash cache quotas using `IORMPLAN` attributes.

Example 7-45 shows how to reset the `IORMPLAN` attributes.

Example 7-46 shows how to specify the `asmcluster` attribute.

Example 7-40 Altering IORMPLAN Attributes

This example shows the `ALTER` command with the `IORMPLAN` object.

```
CellCLI> ALTER IORMPLAN                                        -
        catPlan=(                                              -
                 (name=administrative, level=1, allocation=80),-
                 (name=interactive, level=2, allocation=90),   -
                 (name=batch, level=3, allocation=80),         -
                 (name=maintenance, level=4, allocation=50),   -
                 (name=other, level=4, allocation=50)          -
                ),                                             -
        dbPlan=((name=sales_prod, level=1, allocation=80),     -
                (name=finance_prod, level=1, allocation=20),   -
                (name=sales_dev, level=2, allocation=100),     -
                (name=sales_test, level=3, allocation=50),     -
                (name=other, level=3, allocation=50))

CellCLI> ALTER
IORMPLAN                                                        -
         dbPlan=((name='sales_prod', level=1, allocation=30,
role=primary), -
                 (name=sales1, level=1, allocation=20,
role=standby),          -
                 (name=sales2, level=1, allocation=20,
role=standby),          -
                 (name=other, level=3, allocation = 50))

ALTER IORMPLAN                                             -
dbPlan=((name=salesprod, share=4, role=primary),           -
        (name=salesprod, share=1, limit=50, role=standby)  -
        (name=finance, share=4)                            -
        (name=hr, share=2))

CellCLI> ALTER IORMPLAN                                         -
         catPlan=((name=interactive, level=1, allocation=90),   -
                  (name=batch, level=2, allocation=80),         -
                  (name=maintenance, level=3, allocation=50),   -
                  (name=other, level=3, allocation=50)          -

CellCLI> ALTER IORMPLAN objective='low_latency'

CellCLI> ALTER IORMPLAN objective='auto'

CellCLI> ALTER IORMPLAN                                         -
         dbplan=((name=db1, limit=50),                          -
                 (name=db2, limit=50),                          -
                 (name=other, level=1, allocation=25))
```

Example 7-41 Setting a Database Plan Using the share Directive

This example shows how to set `dbPlan` using the `share` directive.

```
ALTER IORMPLAN                                                  -
dbplan=((name=sales01, share=4),                                -
        (name=sales02, share=4),                                -
        (name=sales03, share=4),                                -
        (name=fin01, share=4),                                  -
        (name=fin02, share=4),                                  -
        (name=fin03, share=4),                                  -
        (name=dev01, share=1, limit=50, flashLog=off),          -
        (name=dev02, share=1, limit=25, flashCache=off))
```

Example 7-42 Setting a Database Plan using the default Directive

This example shows how to use the `default` directive to set the share allocation for all databases except dev01 and dev02.

```
ALTER IORMPLAN                                                  -
        (name=dev01, share=1, limit=50, flashLog=off),          -
        (name=dev02, share=1, limit=25, flashCache=off)         -
        (name=default, share=4))
```

Example 7-43 Setting Flash Cache Use with IORMPLAN Attributes

This example shows how to enable flash cache use with the `IORMPLAN` attributes.

```
CellCLI> ALTER IORMPLAN                                         -
         dbPlan=((name=sales_prod, flashCache=on),              -
                 (name=sales_dev, flashCache=on),               -
                 (name=sales_test, flashCache=off),             -
                 (name=other, flashCache=off))

CellCLI> ALTER
IORMPLAN                                                           -
         dbPlan=((name=oltp, level=1, allocation=80,
flashCache=on),             -
                 (name=dss, level=1, allocation=20, limit=50,
flashCache=off),   -
                 (name=other, level=2, allocation=100))
```

Example 7-44 Specifying Flash Cache Quotas Using IORMPLAN Attributes

This example shows how to configure flash cache quotas using IORMPLAN attributes.

```
CellCLI> ALTER
IORMPLAN                                                             -
         dbPlan=((name=prod, share=8,
flashCacheMin=4096),                   -
                 (name=dev,  share=2, flashCacheMin=1G,
flashCacheLimit=4G), -
                 (name=test, share=1, limit=40, flashCacheLimit=2048))
```

Example 7-45 Resetting IORMPLAN Attributes

This example shows how to reset the IORMPLAN attributes for both the dbPlan and catPlan, or individually.

```
CellCLI> ALTER IORMPLAN dbPlan="", catPlan=""
CellCLI> ALTER IORMPLAN dbPlan=""
CellCLI> ALTER IORMPLAN catPlan=""
```

Example 7-46 Specifying an Oracle ASM Cluster Using IORMPLAN Attributes

For a multi-cluster environment with Oracle ASM-scoped security mode, specify different directives for databases with the same unique name using the asmcluster attribute.

```
CellCLI> ALTER
IORMPLAN                                                             -
       dbPlan=((name=pdb1, share=4, flashCacheMin=5G),
asmcluster=cluster1),-
              (name=pdb1, share=2, limit=80, asmcluster=cluster2), -
              (name=pdb2, share=2, flashcachelimit=2G,
asmcluster=cluster1),-
              (name=default, share=1, flashcachelimit=1G))
```

Related Topics

- Managing I/O Resources
  I/O Resource Management (IORM) is a tool for managing how multiple databases and the workloads within the databases share the I/O resources of Oracle Exadata System Software.
- Administering IORM
  You can perform various administrative tasks related to I/O Resource Management (IORM).
- About Security Keys

7.7.1.8 ALTER LUN

Purpose

The ALTER LUN command re-enables all LUNs or specified LUNs.

Syntax

```
ALTER LUN { ALL  | lun1 [ , lun2] ...  }
  REENABLE FORCE
```

Usage Notes

This command creates the cell disk and grid disk metadata on a replacement disk.

This command rebuilds redundancy for the system area of the system disks even when the system LUN is in a normal state.

> **Caution:**
>
> Data might be lost when using this command.

Example 7-47   Re-enabling a LUN

This example shows the `ALTER` command with the `LUN` object.

```
CellCLI> ALTER LUN 'x:7' REENABLE FORCE

CellCLI> ALTER LUN ALL REENABLE FORCE
```

7.7.1.9 ALTER OFFLOADGROUP

Purpose

The `ALTER OFFLOADGROUP` command enables you to alter modifiable attributes of offload groups, and also to restart, start up, and shut down services.

Syntax

```
ALTER OFFLOADGROUP { offloadgroup1 [,offloadgroup2, ...] }
{attribute_name = attribute_value [, attribute_name = attribute_value ...]]
| STARTUP | RESTART | SHUTDOWN }
```

Usage Notes

- The *offloadgroupN* (where *N* is a number) parameters specify the names of the offload groups whose attributes you want to modify, or that you want to start, shut down, or restart.
- The *attribute_name* and *attribute_value* parameters specify the name and value of the attribute you want to modify.
- The `STARTUP` parameter specifies that the offload group(s) is to be started.
- The `RESTART` parameter specifies that the offload group(s) is to be shut down, then started.
- The `SHUTDOWN` parameter specifies that the offload group(s) is to be shut down.

Examples

Example 7-48 Updating the "Comment" Attribute

```
ALTER OFFLOADGROUP offloadgroup1 comment='System group'
```

Example 7-49 Starting up the Offload Group Named "offloadgroup1"

```
ALTER OFFLOADGROUP offloadgroup1 startup
```

Related Topics

- LIST OFFLOADGROUP

7.7.1.10 ALTER PHYSICALDISK

Purpose

The `ALTER PHYSICALDISK` command prepares a disk for replacement.

Syntax

```
ALTER PHYSICALDISK { ALL [harddisk]  | disk_id1 [,disk_id2]  ...  }
 { DROP FOR REPLACEMENT [FORCE] | REENABLE }
```

Usage Notes

The `DROP FOR REPLACEMENT` option:

- Is supported only for hot-pluggable disks
- Checks if it is safe to proactively replace the specified disk. For example, if you attempt to drop the last good system disk, then replacing it would cause the system to crash.
- Offlines the data grid disks on the physical disk if they exist
- Flushes the flash cache store on the disk if it's part of flash cache
- Flushes the disk controller cache if the disk is a hard disk
- Prepares the device so that it can be removed online. For example, for flash devices this option powers off the associated PCIe slot.
- If you use the `FORCE` option with `DROP FOR REPLACEMENT`, then this allows the command to proceed even if the software deems it unsafe.

The `REENABLE` option re-enables a normal physical disk that was dropped for replacement.

The `SERVICELED` option is now obsolete. If you use this option, you will get the error message CELL-04591.

Examples

Example 7-50 Dropping a Normal, Functioning Physical Disk

This example shows how to drop a physical disk.

```
CellCLI> ALTER PHYSICALDISK FLASH_5_1 DROP FOR REPLACEMENT
```

Example 7-51 Re-enabling a Physical Disk

This example shows how to re-enable a physical disk.

```
CellCLI> ALTER PHYSICALDISK 12:3 REENABLE
```

Related Topics

- LIST PHYSICALDISK
- About Leading Zeros in Object Names
  If an object name begins with a 0, you must use quotes to avoid parsing errors.

7.7.1.11 ALTER PMEMCACHE

Purpose

The `ALTER PMEMCACHE` command can alter the set of cell disks used by PMEM cache, flush dirty blocks from PMEM cache, or cancel a previous flush operation on the specified cell disks to re-enable caching.

Syntax

```
ALTER PMEMCACHE { ALL | CELLDISK="cdisk1 [,cdisk2] ..." [FORCE]}}
      {FLUSH [NOWAIT] | CANCEL FLUSH}
```

Usage Notes

> **Note:**
>
> The `FLUSH` option stops new data from being cached on the PMEM cache until CELLSRV restarts, or the flush operation is canceled with the `ALTER PMEMCACHE CANCEL FLUSH` command.

- The `ALL` option affects all available PMEM cell disks.
- The `CELLDISK` option allows you to specify individual cell disks. *cdiskn* represents a cell disk name.
- The `FLUSH` option synchronizes dirty data from the PMEM cache to the cell disks. Dirty data is data that has been modified in the cache but not yet synchronized with the data on disk. Synchronization of dirty data can be a lengthy process,

depending on the number of bytes to be synchronized. Use the following command to check the progress of the flush operation:

```
LIST CELLDISK ATTRIBUTES name, flushstatus, flusherror
```

- The `ALTER PMEMCACHE ... FLUSH` command stops new data from being written to the PMEM cache and then synchronizes all data in the PMEM cache with the flash disks. As a result, all data is removed from the PMEM cache.
- The `CANCEL FLUSH` option terminates an earlier flush operation, and reinstates PMEM caching.
- The `NOWAIT` option allows the `ALTER` command to complete while the flush operation is in progress.
- The `FORCE` option can be used to forcefully change the set of cell disks used by the PMEM cache.

Example 7-52 Using the ALTER PMEMCACHE Command

The following command specifies that the PMEM cache uses all PMEM cell disks.

```
CellCLI> ALTER PMEMCACHE ALL
```

The following command specifies that the PMEM cache uses only two PMEM cell disks, ignoring any errors or warnings.

```
CellCLI> ALTER PMEMCACHE CELLDISK='PM_01_mycell, PM_03_mycell' FORCE
```

The following command specifies initiates a flush operation for all PMEM cell disks, and returns the prompt before the operation completes.

```
CellCLI> ALTER PMEMCACHE ALL FLUSH NOWAIT
```

7.7.1.12 ALTER QUARANTINE

Purpose

The `ALTER QUARANTINE` command changes the attributes for a quarantine.

Syntax

```
ALTER QUARANTINE { ALL  | quarantine1 [,quarantine2]  ...  }
   attribute_name = attribute_value
   [, attribute_name = attribute_value]...
```

Usage Notes

Only modifiable fields can be changed.

Examples

The following example shows the `ALTER` command with the `QUARANTINE` object.

Example 7-53 Altering a Quarantine

```
CELLCLI> ALTER QUARANTINE 12 comment='bugX'
```

7.7.1.13 ALTER SOFTWAREUPDATE

Purpose

You can schedule software updates by setting SOFTWAREUPDATE attributes. The ALTER SOFTWAREUPDATE command enables you to alter modifiable Software Update attributes, to validate the pre-requirements for the software update, or to start the upgrade immediately.

You can also run the ALTER SOFTWAREUPDATE command using exacli.

Syntax

```
ALTER SOFTWAREUPDATE {VALIDATE PREREQ | UPGRADE [FORCE] | attribute =
attribute value [,attribute = attribute value...]}
```

Options for the ALTER SOFTWAREUPDATE Command

- VALIDATE PREREQ

  Run software update check pre-requirement steps now. This will download the software update pre-requirement code for the update specified by the store attribute. These checks are run automatically as part of update. Use this option only if you want to run prerequisite checks explicitly. Any error found will be displayed. A stateful alert will be raised if any error is found by the VALIDATE PREREQ command.

- UPGRADE [FORCE]

  Run the software update (including the pre-requirement steps) now, using the software location specified by the Software Update store attribute. Use this command if you want to perform the update now rather than wait for the time specified by the Software Update time attribute.

  If FORCE is specified, then the upgrade continues despite any pre-requirement check errors.

- *attribute = attribute value*

  Modify the specified Software Update attributes to the values provided.

Attributes for SOFTWAREUPDATE

The following attributes for the ALTER SOFTWAREUPDATE command are modifiable:

- frequency: Storage server updates can automatically be done periodically by setting the frequency attribute to the desired frequency. You can specify one of the following values: {none | daily | weekly | biweekly }. If the value specified for frequency is '' or none, then the scheduled update is only done once. The value none can be used for the frequency in Oracle Exadata System Software release 19.1.0 or later.
- name: The name of the patch to use in the update, which includes the software version string such as 18.1.1.0.0.171018. If you use the ALTER SOFTWAREUPDATE UPGRADE or ALTER SOFTWAREUPDATE VALIDATE PREREQ command, then the software store is checked and the name attribute is set automatically (if not already set) to the latest available version in the software store. Otherwise, the name attribute has

a value of `unknown`. If there are multiple software versions at the store site, then this attribute can be used to specify which version should be used.

Patches downloaded from My Oracle Support use a different name format, for example `p26875767_181100_Linux-x86-64.zip`. If you are using Oracle Exadata System Software 18c (18.1.0) or 18c (18.1.1), then you must rename the downloaded patch file so it has a name like `18.1.1.0.0.171018`. Starting with Oracle Exadata System Software release 18.1.2, the `ALTER SOFTWAREUPDATE` command accepts patch names of the form `p26875767_181100`.

- `store`: A URL for the location of the software update file. MS finds and downloads the software update file using the protocol specified by the URL. You can use the following protocols for the store URL: `HTTP` or `HTTPS`. The update software can be staged anywhere that is accessible using the management or RDMA Network Fabric network of the storage server.
- `time`: A future date and time at which the software update should be performed. The time can be specified as a local informal date and time, for example `"1 AM, next Tuesday"`. If the date and time is valid then the output from setting this attribute shows the interpreted time in standard format with timezone offset, such as `2017-08-22T01:00:00-08:00`.

  If you set this attribute to the empty string, `""`, then it cancels the scheduled software update.
- `timeLimitInMinutes`: An update may wait for other storage servers to complete their updates in order to preserve disk group redundancy. By default, there is no limit on the amount of time which can be spent waiting to update. This attribute may be set to a positive, maximum integer which represents the number of minutes a storage server will spend waiting to update. If an update does not start within the time specified by the limit, then the update is canceled and an update alert is reported.

Usage Notes

- The `store` attribute directs storage servers to a URL from which they can automatically download update software. Each storage server downloads the software to its active partition, and then loads the software on its passive partition. The storage servers reboot to the new software version according to the desired schedule.
- Software download and the prerequisite check will begin up to a week before the scheduled update time.
- The update progress can be monitored by displaying the non-modifiable Software Update `status` attribute.
- Software updates do not occur if the upgrade software is already installed
- You can use `dcli` or `exacli` to schedule and install updates using the `ALTER SOFTWAREUPDATE` command.
- The Software Update feature supports using HTTPS transport for software downloads. When using HTTPS, TLS certificate checks are performed by default. If the remote server’s certificate cannot be validated then the following error is reported:

  ```
  CELL-00076: An error occurred during download of software update:
  source https://hostname:port is not available.
  ```

```
CELL-00092: The store's TLS certificate cannot be authenticated with
known CA certificates.
```

This can happen if the remote server uses a self-signed certificate or if the remote server uses a certificate signed by a certificate authority (CA) that is not included in the storage server's CA store. You can use the following procedure to add a CA certificate to the storage server's CA store. This is a security setup step which requires shell access as `root` on the storage server.

1. Get a CA certificate that can verify the remote server. The certificate should be stored in PEM or DER file format.
2. Copy the file to the storage server at this directory: `/etc/pki/ca-trust/source/anchors/`
3. Run following commands:

   ```
   update-ca-trust enable
   ```

   ```
   update-ca-trust extract
   ```

Use the operating system `man` utility to get more information about the `update-ca-trust` command.

Examples

Example 7-54 Modifying the Software Update time Attribute

Modify the scheduled time of the next software update to 1 a.m. on Thursday.

```
CellCLI> ALTER SOFTWAREUPDATE time = "1 AM Thursday"
CELL update is scheduled to begin at 2017-08-24T01:00:00-08:00
```

Example 7-55 Setting the Software Update store Attribute

This example shows how to set the store attribute to a location that uses HTTPS protocol.

```
ALTER SOFTWAREUPDATE store="https://my-exa-store/cell"
```

Example 7-56 Starting a Software Update Immediately

This example shows how to immediately start a software update using the attribute values already specified.

```
ALTER SOFTWAREUPDATE UPGRADE FORCE
```

7.7.1.14 ALTER THRESHOLD

Purpose

The `ALTER THRESHOLD` command updates the attribute values of all thresholds or the specified thresholds.

Syntax

```
ALTER THRESHOLD { ALL |threshold_name [, threshold_name ...] }
   attribute_name = attribute_value
   [, attribute_name = attribute_value]...
```

Usage Notes

The attributes that can be changed with the `ALTER` command are shown as `modifiable` in Describing the THRESHOLD Object.

Examples

The following example shows how to alter threshold attributes.

Example 7-57   Altering Threshold Attributes

```
CellCLI> ALTER THRESHOLD ct_io_wt_rq.interactive warning=10, critical=20, -
                comparison='=', occurrences=2, observation=10

CellCLI> ALTER THRESHOLD ALL occurrences=3
```

Related Topics

- CREATE THRESHOLD

7.7.1.15 ALTER USER

Purpose

The `ALTER USER` command changes the attributes of a user role.

Syntax

```
ALTER USER user1 attribute_name1 = attribute_value1      \
[, attribute_name2 = attribute_value2, ...]
```

Usage Notes

- The user name cannot be `root`, `celladmin` or `cellmonitor`. Those are reserved.
- The user name should be unique.
- The system prompts for a password for the new user. The password must have 12 to 40 alphanumeric characters or special characters !@#$%^&*() with at least one digit, one lowercase letter, and one uppercase letter. Starting with Oracle Exadata System Software release 18.1.0.0.0, the password can be 8 to 40 characters in length and can also utilize the special characters - and _.
- The new password cannot be the same as the current password for the user.

Example 7-58   Using the ALTER USER Command

This example shows how to change a user's password.

```
CellCLI> ALTER USER sjones password=TOPsecret2345
```

7.7.2 ASSIGN KEY

Purpose

The `ASSIGN KEY` command assigns or removes a security key to or from a client.

Syntax

```
ASSIGN KEY FOR [ASMCLUSTER] 'client_name1' = 'key-value1' [,
'client_name2' = 'key-value2'...]

ASSIGN KEY FOR CELL 'key-value'

ASSIGN KEY FOR [REMOTE | LOCAL] CELL 'client_name1' = 'key-value1' [,
'client_name2' = 'key-value2'...]
```

Options

- *client_name* is an alias that is the unique name (`DB_UNIQUE_NAME`) for a database client or Oracle ASM cluster.

  > **Note:**
  >
  > The client name or Oracle ASM cluster name not case-sensitive. For example, `ASM1` and `asm1` are treated as the same value.

- *key-value* is a hexadecimal string key that is assigned to the client as a security key. The key value is generated with the `CREATE KEY` command. The key values assigned with the `ASSIGN` command must match the key in the client `cellkey.ora` file on the database servers. The key value can be the same for multiple clients that need the same access. An empty string for the *key-value* removes a previously assigned key.
- Starting with Oracle Exadata System Software release 12.2.1.1.0, you can use the optional keyword `ASMCLUSTER` to indicate that the client is an Oracle ASM cluster. The Oracle ASM cluster alias must not be longer than 14 characters, and only alphanumeric and hyphen characters are allowed.
- Starting with Oracle Exadata System Software release 12.2.1.1.0, the use of the `CELL` keyword can be used to assign a single key to all storage servers to enable cell-to-cell direct operations. You specify only a single *key-value*; you do not specify a *client_name*. You cannot use a list of values with the `CELL` keyword.
- Starting with Oracle Exadata System Software release 12.2.1.1.0, the `FOR LOCAL CELL` clause assigns a cell key to the local (current) cell. If you specify `FOR LOCAL CELL`, there can be only one key; a list of values is not supported. The *client_name* is a unique identifier for each cell.
- Starting with Oracle Exadata System Software release 12.2.1.1.0, the `FOR REMOTE CELL` clause specifies the cell keys that the current cell will accept. The *client_name* is the unique identifier for the cell assigned the *key-value*. You can specify a single client and key, or a list of values.

Usage Notes

- For ASM-scoped security or DB-scoped security, the client aliases must be entered in the `availableTo` attribute of the `GRIDDISK` object.
- When using the `ASMCLUSTER` keyword in Oracle Exadata System Software release 12.2.1.1.0 or later, if you specify a client name and key that already exists (that is a key was already specified for an Oracle ASM client prior to Oracle Exadata System Software release 12.2.1.1.0), then the client will be changed to be an Oracle ASM cluster client. In this case, the name and key will be removed from the ASM-scoped security list, and added as an Oracle ASM cluster client. Grid disks with this Oracle ASM client in their ACL can remain online for this operation.

Examples

Example 7-59 Assigning Keys to Clients

This example shows how to use the `ASSIGN KEY` command to assign keys to one or multiple clients.

```
CellCLI> ASSIGN KEY FOR 'db0' ='b67d5587fe728118af47c57ab8da650a'

CellCLI> ASSIGN KEY FOR '+asm'='7c57ab8da650ab118587feaf467d5728'

CellCLI> ASSIGN KEY FOR '+asm'='ed63f41779c262ddd34a00c0d83590b8',  -
                         'db1' ='118af47c57ab8da650ab67d5587fe728',  -
                         'db2' ='8a65313e8de6cd8bcbab7f4bdddb0498',  -
                         'db3' ='9140c767bd92d1b45783e7fe6520e6d'


CellCLI> ASSIGN KEY FOR LOCAL CELL mykey='fa292e11b31b210c4b7a24c5f1bb4d32'

CellCLI> ASSIGN KEY FOR REMOTE CELL -
          'cellkey1'='b67d5587fe728118af47c57ab8da650a', -
          'cellkey2'='118af47c57ab8da650ab67d5587fe728'


CellCLI> ASSIGN KEY FOR CELL '4839deff903625aab394df7638e7b29a'

CellCLI> ASSIGN KEY FOR ASMCLUSTER asm1='118af47c57ab8da650ab67d5587fe728'
```

Example 7-60 Removing Keys from Clients

This example shows how to use the `ASSIGN KEY` command to remove keys from clients.

```
CellCLI> ASSIGN KEY FOR 'db1'='', 'db2'='', 'db3'='', '+asm'=''

CellCLI> ASSIGN KEY FOR ASMCLUSTER asm1=''
```

Related Topics

- Enabling Cell-to-Cell Operations
- CREATE KEY

- About Security Keys

7.7.3 CALIBRATE

Purpose

The CALIBRATE command runs raw performance tests on cell disks, enabling you to verify the disk performance before the cell is put online.

Syntax

```
CALIBRATE [FLASHDISK | HARDDISK | LUN1 [, LUN2]] [FORCE]
```

Usage Notes

You must be logged on to the cell as the root user to run CALIBRATE.

The FORCE option enables you to run the tests when Cell Server is running. If you do not use the FORCE option, then Cell Server must be shut down. Running CALIBRATE at the same time as a Cell Server process impacts performance.

Use the FLASHDISK option to specify that only flash LUNs be calibrated.

Use the HARDDISK option to specify that only hard disk LUNs be calibrated.

Use the LUN*n* option to specify a list of LUNs by name be calibrated.

Examples

Example 7-61 Output from CALIBRATE Command with FORCE Option on Oracle Exadata Storage Server

This example shows the output when using CALIBRATE with FORCE option on Oracle Exadata Storage Server.

```
CellCLI> CALIBRATE FORCE
Calibration will take a few minutes...
Aggregate random read throughput across all hard disk luns: 1604 MBPS
Aggregate random read throughput across all flash disk luns: 4242.9 MBPS
Aggregate random read IOs per second (IOPS) across all hard disk luns: 4927
Aggregate random read IOs per second (IOPS) across all flash disk luns:
148695
Controller read throughput: 1608.05 MBPS
Calibrating hard disks (read only) ...
Lun 0_0  on drive [20:0      ] random read throughput: 153.41 MBPS, and
412 IOPS
Lun 0_1  on drive [20:1      ] random read throughput: 155.38 MBPS, and
407 IOPS
Lun 0_10 on drive [20:10     ] random read throughput: 155.32 MBPS, and
423 IOPS
Lun 0_11 on drive [20:11     ] random read throughput: 151.24 MBPS, and
427 IOPS
Lun 0_2  on drive [20:2      ] random read throughput: 152.70 MBPS, and
422 IOPS
Lun 0_3  on drive [20:3      ] random read throughput: 155.42 MBPS, and
423 IOPS
```

```
Lun 0_4  on drive [20:4      ] random read throughput: 153.14 MBPS, and
428 IOPS
Lun 0_5  on drive [20:5      ] random read throughput: 154.06 MBPS, and
424 IOPS
Lun 0_6  on drive [20:6      ] random read throughput: 150.82 MBPS, and
409 IOPS
Lun 0_7  on drive [20:7      ] random read throughput: 154.61 MBPS, and
426 IOPS
Lun 0_8  on drive [20:8      ] random read throughput: 154.46 MBPS, and
424 IOPS
Lun 0_9  on drive [20:9      ] random read throughput: 154.63 MBPS, and
426 IOPS
Calibrating flash disks (read only, note that writes will be significantly
slower) ...
Lun 1_0  on drive [[10:0:0:0]] random read throughput: 269.11 MBPS, and
19635 IOPS
Lun 1_1  on drive [[10:0:1:0]] random read throughput: 268.86 MBPS, and
19648 IOPS
Lun 1_2  on drive [[10:0:2:0]] random read throughput: 268.68 MBPS, and
19645 IOPS
Lun 1_3  on drive [[10:0:3:0]] random read throughput: 268.92 MBPS, and
19640 IOPS
Lun 2_0  on drive [[12:0:0:0]] random read throughput: 269.78 MBPS, and
20436 IOPS
Lun 2_1  on drive [[12:0:1:0]] random read throughput: 269.69 MBPS, and
20394 IOPS
Lun 2_2  on drive [[12:0:2:0]] random read throughput: 269.04 MBPS, and
20439 IOPS
Lun 2_3  on drive [[12:0:3:0]] random read throughput: 269.51 MBPS, and
20420 IOPS
Lun 4_0  on drive [[9:0:0:0] ] random read throughput: 269.07 MBPS, and
19668 IOPS
Lun 4_1  on drive [[9:0:1:0] ] random read throughput: 269.24 MBPS, and
19697 IOPS
Lun 4_2  on drive [[9:0:2:0] ] random read throughput: 269.09 MBPS, and
19676 IOPS
Lun 4_3  on drive [[9:0:3:0] ] random read throughput: 269.03 MBPS, and
19681 IOPS
Lun 5_0  on drive [[11:0:0:0]] random read throughput: 268.06 MBPS, and
19714 IOPS
Lun 5_1  on drive [[11:0:1:0]] random read throughput: 268.24 MBPS, and
19696 IOPS
Lun 5_2  on drive [[11:0:2:0]] random read throughput: 268.33 MBPS, and
19717 IOPS
Lun 5_3  on drive [[11:0:3:0]] random read throughput: 268.14 MBPS, and
19693 IOPS
CALIBRATE results are within an acceptable range.

CALIBRATE stress test is now running...
Calibration has finished.
```

Example 7-62   Calibrating LUNs by Name

```
CALIBRATE '2_1', '2_3' FORCE
```

Related Topics

- ALTER CELL

7.7.4 CREATE

Purpose

The CREATE command creates a new object and assigns initial attributes to the object.

Syntax

```
CREATE object_type [name] ...
   [attribute_name = attribute_value [, attribute_name =
attribute_value]...]
```

Usage Notes

- When multiple objects are valid as the target of a CREATE command, there is the possibility of partial success. If an error occurs, then the command is interrupted, and the remaining objects are not created.

- CREATE CELL
- CREATE CELLDISK
- CREATE DIAGPACK
- CREATE FLASHCACHE
- CREATE FLASHLOG
- CREATE GRIDDISK
- CREATE KEY
- CREATE PMEMCACHE
- CREATE PMEMLOG
- CREATE QUARANTINE
- CREATE ROLE
- CREATE THRESHOLD
- CREATE USER

Related Topics

- About CellCLI Object Types
  There are serveral Oracle Exadata System Software object types that can be used with CellCLI object commands.

7.7.4.1 CREATE CELL

Purpose

The CREATE CELL command creates a cell object and assigns initial attributes to the object.

Syntax

```
CREATE CELL [name]
   interconnect1=intValue1 [, interconnect2=intValue2 ...]
   [, attributeName = attributeValue  ...]
```

Usage Notes

The attributes that can be set are shown as `modifiable` in Example 7-83.

- This command can be used to assign the `ASR` value to the `snmpSubscriber` attribute.

  When specifying the `snmpSubscriber` attribute, the community name cannot contain spaces or the following characters: = ' " \ / < >
- The default cell name is set to the network host name of the cell with hyphens in the network name replaced with underscores. You can display the network name with the `uname -n` command. If you change the cell name, then you must choose a unique cell name.
- One to four interconnects can be specified. The `interconnect1` attribute must be specified if the `interconnect2` attribute is specified. The `interconnect1` and `interconnect2` attributes must be specified if `interconnect3` is specified, and so on.
- Interconnects for RDMA over InfiniBand Network Fabric are `ib`*`N`*. Interconnects for RDMA over RoCE Network Fabric are `re`*`N`*.
- By default, the `CREATE CELL` command creates Exadata Smart Flash Cache cell disks on all flash LUNs, and creates Exadata Smart Flash Cache on the cell disks. Users can specify `FLASHCACHE=0` to bypass creation of flash cell disks and Exadata Smart Flash Cache on the cell disks. A non-zero value for Exadata Smart Flash Cache is taken as the total size for creating Exadata Smart Flash Cache. The size is divided evenly across the flash LUNs.
- The `CREATE CELL` command creates Exadata Smart Flash Logon the cell disks. Users can specify `FLASHLOG=0` to bypass creation of flash logs on the cell disks. A non-zero value for Exadata Smart Flash Log is taken as the total size for creating Exadata Smart Flash Log files. The size is divided evenly across the flash LUNs.
- The `CREATE CELL eighthRack` command enables or disables an Eighth Rack configuration on Oracle Exadata Database Machine X3-2 Quarter Racks or later. The options are `true` to enable the Eighth Rack configuration, and `false` to disable the Eighth Rack configuration. The `CREATE CELL eighthRack=true` command requires that there are no cell disks because enabling the Eighth Rack configures only half of the hard disks and flash capacity.
- The `CREATE CELL flashCacheCompress` command creates a cell with or without flash cache compression. Flash cache compression maximizes the amount of data in cache, and improves the cache hit rate. The options are `true` to create a cell with flash cache compression enabled, and `false` to create a cell without flash cache compression. The system must be restarted in order to change flash cache compression.

> **Note:**
>
> Flash cache compression is only supported on Oracle Exadata Database Machine X3 or Oracle Exadata Database Machine X4 racks

- Starting with Oracle Exadata System Software release 19.1.0, the `httpsAccess` attribute can be used to specify a list of IP addresses or IP subnet masks that control who can access the RESTful service via HTTPs. The value you specify for `httpsAccess` overwrites any previous value. You can use the following values for `httpsAccess`:
  - `ALL` — to allow access to all hosts (Default)
  - `NONE` — to disable the HTTPs port completely
  - `IP1, IP2,..., IP`*`n`* — to only allow access to hosts with IP addresses IP1, IP2,..., IP*n* where IP*n* is a valid IP address in IPv4, IPv4 subnet, IPv6 or IPv4-embedded IPv6 format. You can specify a maximum of 512 IP addresses for the access control list.

  Additionally, instead of a single IP address, you can use the `/` character to specify a range of IP addresses using a subnet mask. For example the range `'192.168.10.0/24'` corresponds to hosts having IP addresses from 192.168.10.1 to 192.168.10.255. If you specify an IP address range, you need to enclose the IP address string in quotes.

Examples

Example 7-63 Creating a Cell

This example shows how to create a cell. In the example, the interconnections are set to an existing RDMA over InfiniBand Network Fabric connection.

```
CellCLI> CREATE CELL cell22 interconnect1=bondib0
```

Example 7-64 Creating an Eighth Rack Configuration

This example shows how to create a cell in an Eighth Rack configuration.

```
CellCLI> CREATE CELL eighthRack=true
```

Example 7-65 Creating a Cell with Restricted HTTPs Access

This example shows how to create a cell that allows HTTPs port access only from hosts having IP addresses in the range 192.168.10.1 to 192.168.10.255.

```
CellCLI> CREATE CELL httpsAccess='192.168.10.0/24'
```

Related Topics

- Restrictions on Values of Common Attributes
  Review the following restrictions for the values of attributes used by multiple CellCLI objects.
- *Oracle Exadata Database Machine Maintenance Guide*

7.7.4.2 CREATE CELLDISK

Purpose

The `CREATE CELLDISK` command creates a cell disk object and assigns initial attributes to the object. You can use the `ALL` option to automatically create cell disks.

Syntax

```
CREATE CELLDISK  {
   ALL [FLASHDISK | HARDDISK]
  |cdisk1 attribute_name=value,[attribute_name=value]... [FORCE]
  |((name=cdisk2,attribute_name=value,[attribute_name=value]...)
  [,(name=cdisk3,attributename=value,[attributename=value]...)]...)
  }
```

Usage Notes

The attributes that can be specified during creation are the cell disk name (*cdiskN*), `comment`, `lun`, `size`, and `physicalDisk`.

- Either `lun` or `physicalDisk` is required when adding a specifically-named cell disk.
  - When a physical disk is provided, a single-disk LUN is created, and that LUN is used to create the cell disk. The LUN is flagged as automatically-created.
  - When a LUN is provided, that device is used to create the cell disk.
- You can use the `size` attribute when adding a new disk that is a different size than the existing disks. You must specify a value for `size` that is less than or equal to the maximum allowed cell disk size.
- `CREATE CELLDISK ALL` is a shortcut command to create all candidate cell disks for the cell. This operation occurs in two steps:
  - All physical disks that are not already configured as LUNs are configured as single-disk LUNs (SDLs). These LUNs are flagged as automatically-created LUNs.
  - All LUNs that are not configured as cell disks are used to create cell disks. These cell disks are initially named according to the template `CD_`*`lunname_cellname`*. This name can later be changed using the `ALTER CELLDISK` command.

    > **Note:**
    >
    > LUNs with a second or third extended file system (`ext2`/`ext3`) are ignored during the `CREATE CELLDISK ALL` operation.

  - You can include `size` to create all cell disks with the specified size, instead of attempting to use the entire disk. If you do not specify `size` with the `CREATE CELLDISK ALL` command, then:
    - On data disks, Management Server (MS) creates the cell disks with a size that is equal to the minimum physical disks size across all physical disks in a cell (which is also the maximum allowed cell disk size in a cell).

    * On system disks, MS creates the cell disks with a size that is equal to the maximum allowed cell disk size-size of system partitions.

- The FLASHDISK option limits the CREATE CELLDISK command to cell disks that are flash disks.
- The HARDDISK option limits the CREATE CELLDISK command to cell disks that are hard disks.
- The list form of CREATE CELLDISK enables you to add a series of cell disks in a single command.
- The FORCE keyword overrides the following error conditions:
  - The physical disk provided is already part of an existing LUN.
  - The LUN provided is already associated with a cell disk.

  FORCE causes the LUN to be reused to create the new cell disk. Any preexisting configuration is lost. FORCE is not an option for CREATE CELLDISK ALL or for the list form of CREATE CELLDISK.
- The INTERLEAVING option has been deprecated. Starting with Oracle Exadata System Software release 19.1.0, attempts to create interleaving grid disks will be automatically converted to normal grid disk creation. Interleaving grid disks created in earlier Oracle Exadata System Software releases will continue to operate normally.

Cell disks are created automatically using the CREATE CELLDISK ALL command. This command creates single-disk LUNs from all available physical disks, and then creates cell disks from all available LUNs.

When a cell disk is created, metadata describing the cell disk is written to the cell disk itself and to the cell configuration files. Approximately 48 MB of the cell disk is allocated for the cell disk metadata partition. On a subsequent restart, Cell Server (CELLSRV) attempts to rediscover the created cell disk by reading configuration data on the disk. Any hardware changes in the cell might cause a change in the LUN and device name for a cell disk. The rediscovery mechanism that runs during the cell restart process changes the cell disk configuration accordingly.

Example 7-66 Creating a Cell Disk

```
CellCLI> CREATE CELLDISK ALL

CellCLI> CREATE CELLDISK cdisk03 lun=0_3

CellCLI> CREATE CELLDISK cdisk04 physicalDisk='I2:1:2'


CellCLI> CREATE CELLDISK CD_08_cell06 lun=0_8, size=300M

CellCLI> CREATE CELLDISK CD_03_cell04 lun=0_3

CellCLI> CREATE CELLDISK CD_05_cell09 physicalDisk='2I:1:2'
```

Related Topics

- ALTER CELL

- Restrictions on Values of Common Attributes
  Review the following restrictions for the values of attributes used by multiple CellCLI objects.
- Create the Storage Cell Disks
  Use the `CREATE CELLDISK` command to create the cell disks.

7.7.4.3 CREATE DIAGPACK

Purpose

The `CREATE DIAGPACK` command creates a diagnostic package, which contains logs and traces that you can use to troubleshoot problems in your system. You can also send the generated package to Oracle Support Services, as needed.

Syntax

```
CREATE DIAGPACK packStartTime=time, [durationInHrs=duration]
```

or

```
CREATE DIAGPACK alertName=alertName
```

Usage Notes

When an alert occurs, a diagnostic package is created automatically. This package contains logs and traces related to the alert.

The `CREATE DIAGPACK` command enables you to generate diagnostic packages manually.

- The `packStartTime` parameter specifies when to start collecting the logs and traces. The format of `packStartTime` is *`yyyy_MM_dd`*`T`*`HH_mm_ss`*, For example: `2015_07_07T09_00_00`.

  You can also specify the keyword `now`for `packStartTime`. The `packStartTime` cannot be in the future and cannot be older than 7 days. The value of `packStartTime` is used as part of the name of the diagnostic package.
- The `durationInHrs` parameter specifies the number of hours of logs and traces to include in the diagnostic package. Valid values are from `1` (default) to `6`.

  Every diagnostic package includes logs 1 hour before and 1 hour after the `packStartTime`. For example, if you specify a time of 12_00_00, then logs will collected from 11_00_00 to 13_00_00, unless the end time is in the future.
- The `alertName` parameter specifies the alert name for which to create the diagnostic package. You can run the LIST ALERTHISTORY command to view the alert names.

Name of Diagnostic Packages

The name of the diagnostic package is formed as: *`hostname`*`_diag_`*`packStartTime`*`_`*`unique package ID`*

For example: `testcell1_diag_2015_07_07T09_00_00_3`

For alerts, the name of the diagnostic package is formed as: *hostname_timestamp of when the package was created_alert ID*. For example: `testcell1_2015_09_30T13_13_00_2_1`

Location of Diagnostic Packages

The location of the diagnostic packages is `$LOG_HOME`.

Status of Diagnostic Packages

You can run the LIST DIAGPACK command to get a list of diagnostic packages in your system, and their status.

Privileges Needed to Create, List, and Download Diagnostic Packages

There are certain privileges that are needed for working with diagnostic packages. Use CellCLI to grant the following privileges to a role:

- Privilege to create diagnostic packages:

  ```
  grant privilege CREATE ON DIAGPACK to ROLE role
  ```

- Privilege to list diagnostic packages and check their status:

  ```
  grant privilege LIST ON DIAGPACK to ROLE role
  ```

- Privilege to download diagnostic packages:

  ```
  grant privilege DOWNLOAD ON DIAGPACK to ROLE role
  ```

You can then grant the role to users. For example, if you named your role `diagpack_role`, the following command grants the role to `fred`.

```
CellCLI> GRANT ROLE diagpack_role TO USER fred
```

During deployment, Oracle Exadata Deployment Assistant (OEDA) creates an Exadata storage software user called `CELLDIAG`. You can use this user to connect to a cell remotely using ExaCLI or REST API. This user has privileges to create, list, and download diagnostic packages.

Downloading Diagnostic Packages

You can download diagnostic packages using any of the following methods. Note that you need the `DOWNLOAD ON DIAGPACK` privilege before you can download diagnostic packages.

- Using the REST API
  - To download the diagnostic package by name, use the following URL, where *hostname* specifies the host name of the cell and *diagpackname* specifies the name of the diagnostic package:

    ```
    https://hostname/diagpack/download?name=diagpackname
    ```

    If the user is not already logged in, the URL will prompt for a user name and password.

    Diagnostic packages can also be accessed at `https://hostname/diagpack` . For example: `https://cell1.example.com/diagpack`.

    The page then prompts the user to log in:

    ```
    User: fred
    Password: ********
    ```

    Based on the user's privileges, various sections of this page could be hidden:

    - The form to create a new diagpack will not be shown if the user does not have the `CREATE ON DIAGPACK` privilege.

    - Similarly, the list of alerts and their diagnostic packages will not be shown if the user does not have the `LIST ON DIAGPACK` privilege.

  - To download the diagnostic package by alert name, use the following URL, where *hostname* specifies the host name of the cell and *alertName* specifies the alert name of the diagnostic package:

    ```
    https://hostname/diagpack/download?alert=alertName
    ```

    The alert name is the same alert name that is used in AlertHistory. It looks like 1, 2, 3 for stateless alerts, and 1_1, 2_1, 3_1, 3_2 for stateful alerts.

- Using the `download` ExaCLI command

  ExaCLI enables you to run CellCLI commands on storage nodes remotely from compute nodes.

  To run the download command, run the following commands on a compute node:

  1. Start up ExaCLI and connect to the cell containing the diagnostic pack. For example, use a command similar to the following where *hostname* specifies the host name of the cell:

     ```
     exacli -l celladministrator -c hostname
     Password=********
     ```

  2. Run the download command using a command similar to the following where *name* specifies the name of the diagnostic package to download and *destinationFolder* specifies the directory where you want to save the downloaded diagnostic package:

     ```
     exacli> download diagpack name destinationFolder
     ```

- Getting the diagnostic package from the alert emails

  The alert emails include diagnostic packages for all alerts except INFO, CLEAR, and WARNING. Diagnostic packages are generated for critical alerts only.

Re-triggering Package Creation from the Web Page

You can use the following URL to re-trigger package creation:

```
https://hostname/diagpack
```

If the diagnostic package for an alert does not exist on disk, then the web page shows a **Create Package** link instead of a **Download** link.

Click the **Create Package** link to add the alert to the list for creating a diagnostic package. Once the diagnostic package has been created, and the web page is refreshed, the page will display a **Download** link that you can use to download the newly created diagnostic package.

Turning Off the Diagnostic Pack Attachment in Emails

To turn off the diagnostic pack attachment in emails, run `ALTER CELL diagPackEmailAttach=FALSE`. The diagnostic packs are still generated and stored on the storage servers. To download the diagnostic packs, see "Downloading Diagnostic Packages".

Examples

Example 7-67   Using "now" for packStartTime

This example creates a diagnostic package using `now` as the start time and the default duration of 1 hour.

The output is 1 compressed file under `$LOG_HOME`.

```
CellCLI> CREATE DIAGPACK packStartTime="now"
    Processing: scab01cel11_diag_2015_07_08T17_53_58_1
    Use 'list diagPack' to check its status.
```

Example 7-68   Specifying a duration

This example creates 3 diagnostic packages under `$LOG_HOME`:

The first package has a start time of `2015_07_07T09_00_00`.

The second package has a start time of `2015_07_07T10_00_00`.

The third package has a start time of `2015_07_07T11_00_00`.

```
CellCLI> CREATE DIAGPACK packStartTime="2015_07_07T09_00_00",
durationInHrs=3
    Processing: scab01cel11_diag_2015_07_07T09_00_00_1
    scab01cel11_diag_2015_07_07T10_00_00_1 (In queue...)
    scab01cel11_diag_2015_07_07T11_00_00_1 (In queue...)
    Use 'list diagPack' to check its status.
```

Related Topics

- About Oracle Auto Service Request
- About Automatic Diagnostic Repository
- Using the ExaCLI Utility

7.7.4.4 CREATE FLASHCACHE

Purpose

The `CREATE FLASHCACHE` command creates Exadata Smart Flash Cache on a cell for I/O requests.

Syntax

```
CREATE FLASHCACHE {[ALL [size = fc_size] | CELLDISK = 'cdisk1
[,cdisk2],,,'    \
   [, size=fc_size]} [attribute_name = attribute_value
   [, attribute_name = attribute_value] ...]
```

Usage Notes

Cell disks defined on Exadata Smart Flash Cache cannot be exported.

The `ALL` argument creates Exadata Smart Flash Cache on all flash cell disks. If the `ALL` argument is not specified, then the `CELLDISK` argument must be specified.

Using the `CELLDISK` argument, you can specify a list of flash cell disks to be used for flash cache. The names of the flash cell disks are comma-delimited. The `FLASHDISK` argument is not required.

The `size` argument specifies the total space used for flash cache. Similar to space in grid disks and flash logs, flash cache space is allocated in 16 MB units, referred to as allocation units. If the `size` attribute is specified when creating a flash cache, then the size allocated is the size of the largest multiple of allocation units less than or equal to the specified size. For example, if `300M` is specified for the `size` attribute, then 288 MB (16x18) is allocated because 288 is the largest multiple of 16 that is less than or equal to 300.

A minimum of 1 allocation unit is always allocated, so the minimum size for a flash cache is 16 MB. Any size value less than 16 MB is rounded up to 16 MB.

Before specifying the `size` attribute, ensure that you have first determined the available free space on each target flash cell disk with the `LIST FLASHCACHE` command. For example, `LIST FLASHCACHE ATTRIBUTES freespace`. If the `size` attribute is not specified, then the maximum size is allocated.

If the `size` attribute is not specified, then all available space on each cell disk in the list is used for Exadata Smart Flash Cache.

By default, 5 percent of space on Extreme Flash Exadata X5-2 storage servers is used for flash cache. To change the amount of space, it is necessary to drop the flash cache, and create it using the `CREATE FLASHCACHE SIZE =` command.

Examples

Example 7-69 Creating Exadata Smart Flash Cache

This example shows how to create Exadata Smart Flash Cache on a cell.

```
CellCLI> CREATE FLASHCACHE ALL

CellCLI> CREATE FLASHCACHE ALL SIZE=250g

CellCLI> CREATE FLASHCACHE CELLDISK='fd_01,fd_02,fd_03,fd_04'

CellCLI> CREATE FLASHCACHE CELLDISK='fd_01_mycell,fd_02_mycell', size = 64G
```

Related Topics

- ALTER FLASHCACHE

7.7.4.5 CREATE FLASHLOG

Purpose

The `CREATE FLASHLOG` command creates the Oracle Exadata Smart Flash Log on a cell for redo log I/O requests.

Syntax

```
CREATE FLASHLOG {[ALL {[FLASHDISK] |[size=log_size]} | CELLDISK='cdisk1 \
   [,cdisk2]...'  [, size=log_size]}
```

Usage Notes

The `CREATE FLASHLOG` command accepts a list of comma-delimited flash cell disks. If a size is specified in the command, then that size is divided evenly across the cell disks, and will total the specified size. If a size is not specified, then a default size of 512 MB is used.

The size of Oracle Exadata Smart Flash Log space on each flash disk must be less than 4 GB. If all 16 flash disks are available, then the total size of Oracle Exadata Smart Flash Log must be less than 64 GB.

Similar to space in grid disks and flash cache, flash log space is allocated in 16 MB units, referred to as allocation units. If the size attribute is specified when creating a flash log, then the size allocated is the size of the largest multiple of allocation units less than or equal to the specified size. For example, if 300M is specified for the size attribute, then 288 MB (16x18) is allocated because 288 is the largest multiple of 16 that is less than or equal to 300.

A minimum of 1 allocation unit is always allocated, so the minimum size for a flash log is 16 MB. Any size value less than 16 MB is rounded up to 16 MB.

The `ALL FLASHDISK` argument creates Oracle Exadata Smart Flash Log on all flash cell disks. If the `ALL` argument is not specified, then the `CELLDISK` argument must be specified. The `FLASHDISK` argument is not required.

> **Note:**
>
> The `CREATE FLASHCACHE` command, by default, uses all available space on each flash disk. Therefore, use the `CREATE FLASHLOG` command before creating the flash cache to ensure both objects consume the correct amount of flash disk space.

To change the size of the flash log, use the `DROP FLASHLOG` command to drop the flash log, and then use the `CREATE FLASHLOG` command to create a flash log with the new size.

Examples

The following example shows how to create Oracle Exadata Smart Flash Log on a cell.

Example 7-70 cCreating Oracle Exadata Smart Flash Log

```
CellCLI> CREATE FLASHLOG ALL

CellCLI> CREATE FLASHLOG ALL SIZE=1g

CellCLI> CREATE FLASHLOG ALL FLASHDISK

CellCLI> CREATE FLASHLOG CELLDISK='fd1,fd2,fd3,fd4'
```

7.7.4.6 CREATE GRIDDISK

Purpose

The `CREATE GRIDDISK` command creates a grid disk object on a specified cell disk or creates one grid disk on each cell disk on the cell. The command also assigns initial attributes to the new grid disks.

Syntax

```
CREATE GRIDDISK  {  ALL [FLASHDISK | HARDDISK] PREFIX=gdisk_name_prefix
   | gdisk CELLDISK = attribute_value  }
   [, attribute_name = attribute_value]...
```

Usage Notes

The attributes that can be specified are the grid disk name (*gdisk*), `CELLDISK`, `size`, and `comment`. The `CELLDISK` argument is required when a single grid disk is created.

- If an individual name is entered with the command, then the grid disk is created on the cell disk specified by the `CELLDISK` argument. You must ensure that the grid disk name is unique across all cells. If the disk name is not unique, then it might not be possible to add the grid disk to an Oracle ASM disk group.
- The length of a grid disk name is limited to 30 characters.
- The `FLASHDISK` option limits the `CREATE GRIDDISK` command to cell disks that are flash disks.
- The `HARDDISK` option limits the `CREATE GRIDDISK` command to cell disks that are hard disks.
- If the `ALL PREFIX` option is entered with the command, then one grid disk is created on each cell disk on the cell. `PREFIX` is required when `ALL` is used.

  The `PREFIX` option specifies the prefix assigned to the names of the created grid disks. The generated grid disk names are composed of the grid disk prefix followed by an underscore (_) and then the cell disk name in the following form:

  ```
  gdisk_name_prefix_cdisk_name
  ```

Choose a prefix for the grid disk name that matches the Oracle ASM disk group to which the grid disk belongs to help you identify which disks belong to a disk group. The generated grid disk (*gdisk_name_prefix_cdisk_name*) must follow the restrictions on the `name` value.

For example, if the Oracle ASM disk group name is `data01`, then `data01` is used as the prefix for the grid disk names. If `CREATE GRIDDISK ALL PREFIX=data01` is run on a cell with cell disks `CD_01_cell01`, `CD_02_cell01`, and `CD_03_cell01`, then grid disks are named `data01_CD_01_cell01`, `data01_CD_02_cell01`, and `data01_CD_03_cell01` are created on each cell disk respectively.

- The `CREATE GRIDDISK ALL` command skips disks which do not have enough free space for a minimum grid disk. A message stating which disks did not have enough free space appears, and the command continues.
- The `size` and `offset` attributes are optional attributes specified as a number in bytes, unless the suffix `M` (megabytes), `G` (gigabytes), or `T` (terabytes) is included with the number, such as `size=300M`, or `size=150G`.

  Grid disk space is allocated in 16 MB units, referred to as allocation units. If the `size` attribute is specified when creating a grid disk, then the size allocated is the size of the largest multiple of allocation units less than or equal to the specified size. For example, if `300M` is specified for the size attribute, then 288 MB (16x18) is allocated because 288 is the largest multiple of 16 that is less than or equal to 300.

  A minimum of 1 allocation unit is always allocated, so the minimum size for a grid disk is 16 MB. Any size value less than 16 MB is rounded up to 16 MB.

  Before specifying the `size` attribute, ensure that you have first determined the available free space on each target cell disk with the `LIST CELLDISK` command. For example, `LIST CELLDISK` *`cdisk`* `ATTRIBUTES freespace`. If the `size` attribute is not specified, then the maximum size is allocated.

  Offset determines the position on the disk where the grid disk is allocated. The outermost tracks have lower offset values, and these tracks have greater speed and higher bandwidth. Offset can be explicitly specified to create grid disks that are relatively higher performing than other grid disks. If offset is not specified, then the best (warmest) available offset is chosen automatically in chronological order of grid disk creation. You should first create the grid disks that are expected to contain the most frequently accessed (hottest) data, and then create the grid disks that contain the relatively colder data.

  > **Note:**
  >
  > Offset does not apply to flash storage on Extreme Flash storage servers.

- The value of the `availableTo` attribute is set to the names of the clients that you want to set up for DB-scoped security. These clients were assigned security keys that match keys in the configuration files on the hosts.

  The value entered for a client name is the unique name (`DB_UNIQUE_NAME`). The specified clients are those that are allowed to access the grid disk. If a value is entered for `availableTo`, then only the specified clients have access to the grid disk; otherwise, any client can have access.

- Do not edit the value of `idp.type` or `idp.boundary`. Oracle Exadata System Software passes a hint to Oracle ASM about the type of interleaved grid disk, either normal redundancy or high redundancy. Oracle ASM sets the default value for idp.type to static and idp.boundary to the type of redundancy used in the underlying grid disks. The default value of the idp.type attribute is static for Oracle Exadata Storage Server disk groups created on interleaved grid disks.

> **Note:**
>
> Interleaved grid disks are deprecated in Oracle Exadata System Software release 19.1.0.

- The cachingPolicy attribute can be set to default or none. The default option allows the data to be cached in flash cache for the grid disk. The none option means that flash cache is not used for data for the grid disk.

  Flash cache is not used with data files placed in a disk group composed of grid disks with their cachingPolicy set to none.

  Oracle Exadata Deployment Assistant (OEDA) configures the RECO disk group to have the cachingPolicy of its grid disks set to none; therefore flash cache is not used for any data files placed in the RECO disk group.
- The virtualSize attribute is used to create the SPARSE disk group. The maximum virtual size for a sparse grid disk is approximately 100 TB. Sparse grid disks are available for Oracle Exadata Database Machine X3-2 and later.

> **Note:**
>
> The Oracle Database and Oracle Grid Infrastructure software must be release 12.1.0.2.0 BP5 or later when using sparse grid disks.

Example 7-71 Creating a Grid Disk

This examples shows how to create grid disks.

```
CellCLI> CREATE GRIDDISK data1_CD_01_cell01 CELLDISK=CD_01_cell01,
size=200G

CellCLI> CREATE GRIDDISK data2_CD_02_cell01 CELLDISK=CD_02_cell01,
size=200G

CellCLI> CREATE GRIDDISK data3_CD_03_cell01 CELLDISK=CD_03_cell01

CellCLI> CREATE GRIDDISK ALL PREFIX=data01, -
         availableTo='+asm,db1,db2'

CellCLI> CREATE GRIDDISK hr7_CD_07_cell01 CELLDISK=CD_07_cell01, -
         availableTo='asm_hr,hrdb0'

CellCLI> CREATE GRIDDISK GD123 CELLDISK=RECO_CD123, size=100G,
cachingPolicy=none
```

Example 7-72 Creating a SPARSE Disk Group

```
CellCLI> CREATE GRIDDISK spar01 celldisk=CD_01_cel01, size=10G,
virtualsize=100G
```

Related Topics

- Restrictions on Values of Common Attributes
  Review the following restrictions for the values of attributes used by multiple CellCLI objects.
- ALTER GRIDDISK

7.7.4.7 CREATE KEY

Purpose

The `CREATE KEY` command creates and displays a random hexadecimal string to assign client keys. The use of `CREATE KEY` ensures that the security key is in the correct format. This command provides a way to generate a key in the correct format, and it can be run on any cell.

Syntax

```
CREATE KEY
```

Usage Notes

The security key must be entered in the `cellkey.ora` configuration file on the computer hosts that contain clients for which you want to authorize access to a cell.

The key is also assigned to clients that access grid disk storage.

The key must be copied manually to the hosts and cells.

Example 7-73 Creating a Key

This example shows the `CREATE` command with the `KEY` object.

```
CellCLI> CREATE KEY
         3452c64fec9a5800bbe48d4093269400
```

Related Topics

- About Security Keys
- ASSIGN KEY

7.7.4.8 CREATE PMEMCACHE

Purpose

The `CREATE PMEMCACHE` command creates a PMEM Cache for I/O requests.

Syntax

```
CREATE PMEMCACHE {[ALL [size = cache_size] | CELLDISK = 'cdisk1
[,cdisk2],,,'    \
   [, size=cache_size]}
```

Usage Notes

Cell disks used by PMEM Cache cannot be exported.

The `ALL` argument creates the PMEM Cache on all PMEM cell disks. If the `ALL` argument is not specified, then the `CELLDISK` argument must be specified.

Using the `CELLDISK` argument, you can specify a list of PMEM cell disks to be used for PMEM cache. *cdiskn* represents a cell disk name. The names of the PMEM cell disks are comma-delimited and enclosed in quotation marks.

The `size` argument specifies the total size of the PMEM cache. Similar to space in grid disks, PMEM cache space is allocated in 16 MB units, referred to as allocation units. If the `size` attribute is specified when creating a PMEM cache, then the size allocated is the size of the largest multiple of allocation units less than or equal to the specified size. For example, if `300M` is specified for the `size` attribute, then 288 MB (16x18) is allocated because 288 is the largest multiple of 16 that is less than or equal to 300.

A minimum of 1 allocation unit is always allocated, so the minimum size for a PMEM cache is 16 MB. If you specify a size less than 16 MB, then it is rounded up to 16 MB.

Before specifying the `size` attribute, ensure that you have first determined the available free space on each target PMEM cell disk with the `LIST CELLDISK ATTRIBUTES freespace WHERE disktype=PMEM` command.

If the `size` attribute is not specified, then the maximum size is allocated. All available space on each cell disk in the list is used for PMEM Cache.

Examples

Example 7-74 Creating PMEM Cache

The following command demonstrates how to create a 64 GB PMEM cache on specific PMEM cell disks.

```
CREATE PMEMCACHE celldisk='PM_01_mycell,PM_02_mycell', size = 64G
```

The following command demonstrates how to create a 64 GB PMEM cache using all PMEM cell disks.

```
CREATE PMEMCACHE ALL size = 64G
```

7.7.4.9 CREATE PMEMLOG

Purpose

The `CREATE PMEMLOG` command creates the PMEM log file on a cell for redo log I/O requests.

Syntax

```
CREATE PMEMLOG {ALL [size=log_size] | CELLDISK='cdisk1[,cdisk2]...\'
 [, size=log_size]}
```

Usage Notes

The `CREATE PMEMLOG` command accepts a list of comma-delimited PMEM cell disks. If a size is specified in the command, then that size is divided evenly across the PMEM cell disks. Each PMEM cell disk must be at least 16 MB in size. If a size is not specified, then a default size of 960 MB is used.

The `ALL` argument creates `PMEMLOG` on all PMEM cell disks. If the `ALL` argument is not specified, then the `CELLDISK` argument must be specified.

> **Note:**
>
> The `CREATE PMEMCACHE` command, by default, uses all available space on each PMEM cell disk. Therefore, use the `CREATE PMEMLOG` command before creating the PMEM cache to ensure both objects consume the correct amount of PMEM disk space.

To change the size of the PMEM log, use the `DROP PMEMLOG` command to drop the PMEM log, and then use the `CREATE PMEMLOG` command to create a PMEM log with the new size.

Examples

The following example shows how to create PMEM log file on a cell.

Example 7-75 Creating the PMEM log file

To create the PMEMLOG using the default size that uses all available PMEM cell disks, use the following command:

```
CellCLI> CREATE PMEMLOG ALL
```

To create the PMEMLOG of size 1 GB that uses all available PMEM cell disks, use the following command:

```
CellCLI> CREATE PMEMLOG ALL SIZE=1g
```

To create the PMEMLOG on specific PMEM cell disks, use the following command:

```
CellCLI> CREATE PMEMLOG CELLDISK='nv1,nv2,nv3,nv4'
```

To create the PMEMLOG that is 1 GB in size on specific PMEM cell disks, use the following command:

```
CellCLI> CREATE PMEMLOG CELLDISK='nv1,nv2,nv3,nv4', size=1G
```

7.7.4.10 CREATE QUARANTINE

Purpose

The `CREATE QUARANTINE` command allows a quarantine to be created manually.

Syntax

```
CREATE QUARANTINE quarantineType=value quarantinePlan="SYSTEM",
                  dbUniqueName=value[, attributename=value]...
```

Usage Notes

Manual creation of quarantines should be done in coordination with Oracle Support Services. In general, manual quarantines are created to proactively isolate SQL statements that are known to cause problems.

- `quarantineType` specifies the type of quarantine to be created, such as SQLID and SQL_PLAN.
- `quarantinePlan` must be set to `SYSTEM`. Oracle Support Services may specify other values.
- `dbUniqueName` specifies the name of the database that has the quarantine.

Example 7-76 Creating a Quarantine

This example shows the `CREATE` command with the `QUARANTINE` object.

```
CELLCLI> CREATE QUARANTINE  DBUG comment='For debugging quarantines"

CellCLI> CREATE QUARANTINE quarantineType="SQLID",
quarantinePlan="SYSTEM", -
         dbUniqueName="DB1", sqlid="5xnjp4cutc1s7";
```

Related Topics

- DESCRIBE QUARANTINE

7.7.4.11 CREATE ROLE

Purpose

The `CREATE ROLE` command creates a role for a user accessing a cell.

Syntax

```
CREATE ROLE  role_name1 [, role_name2, ...]
```

Usage Notes

The role name should be unique.

Example 7-77   Creating a Role

This example shows how to create a role named `gd_monitor`.

```
CellCLI>CREATE ROLE gd_monitor
```

7.7.4.12 CREATE THRESHOLD

Purpose

The `CREATE THRESHOLD` command creates a threshold object that specifies the conditions for generation of a metric alert.

Syntax

```
CREATE THRESHOLD name attributename=value [, attributename=value]...
```

Usage Notes

The attributes that can be specified are `comparison`, `critical`, `occurrences`, `observation`, and `warning`.

- The *name* argument is required. The name is comprised of a metric name and an object name with the format `metricName.objectName`, such as `db_io_rq_sm_sec.db123` or `ct_io_wt_rq.interactive`. Use the `LIST METRICCURRENT` *`metric`* command to display the available object name for metric. The object name is optional.
- When a object name is not specified, then the threshold is applied to all metric objects for the given metric.
- The `comparison` attribute is required with a condition value. The value must be `'<'`, `'<='`, `'='`, `'>='`, or `'>'`.
- The `occurrences` attribute specifies the number of consecutive measurements over the threshold value that trigger a state change.
- The `observation` attribute is the number of measurements over which measured values are averaged.
- A state change to the value set in `warning` or `critical` causes a stateful alert to be generated.
- The `GD_SP_PRCT_ALLOCATED` metric has a built-in threshold, and automatically sends alerts. Create thresholds for other metrics to receive alerts for those metrics.

When specifying occurrences and observations, you need the specified number of consecutive occurrences of sample averages over the number of observations to cause an alert. For example, if the following five observations (`observations=5`) happen on a cell, then the average sample would be 10 because the number of consecutive occurrences (`occurrences=2`) had values of 5 and 15.

```
Observation 1: 0
Observation 2: 30
Observation 3: 0
Observation 4: 5
Observation 5: 15
```

Example 7-78 Creating a Threshold

This example shows how to create a threshold.

```
CellCLI> CREATE THRESHOLD db_io_rq_sm_sec.db123 comparison='>',
critical=120

CellCLI> CREATE THRESHOLD ct_io_wt_sm.interactive warning=10, critical=20,
-
         comparison='=', occurrences=2, observation=5
```

Related Topics

- Monitoring and Tuning Oracle Exadata System Software
- *Oracle Database Administrator's Guide*

7.7.4.13 CREATE USER

Purpose

The `CREATE USER` command creates a user.

Syntax

```
CREATE USER name PASSWORD = *
```

Usage Notes

- The user name should be unique.
- `celladmin`, `cellmonitor`, and `root` are reserved user names that cannot be used with the CREATE USER command.
- The system prompts for a password for the new user. The password must have 12 to 40 alphanumeric characters or special characters !@#$%^&*() with at least one digit, one lowercase letter, and one uppercase letter. Starting with Oracle Exadata System Software release 18.1.0.0.0, the password can be 8 to 40 characters in length and can also utilize the special characters - and _.
- The new password cannot be the same as the current password for the user.

Example 7-79 Creating a User

This example shows how to create a user.

```
CellCLI> CREATE USER agarcia PASSWORD = *
password:
Confirm password: password
User agarcia successfully created.
```

7.7.5 DESCRIBE

Purpose

The `DESCRIBE` command displays a list of attributes for the object type that is provided as an argument. The list of attributes indicates whether each attribute can be modified.

Syntax

```
DESCRIBE object_type
```

Usage Notes

- The *object_type* is one of the object types supported by CellCLI.
- The list of attributes can be used as arguments in the `LIST` command.
- `DESCRIBE` does not display all of the attributes for the objects.
- DESCRIBE ACTIVEREQUEST
- DESCRIBE ALERTDEFINITION
- DESCRIBE ALERTHISTORY
- DESCRIBE CELL
- DESCRIBE CELLDISK
- DESCRIBE DATABASE
- DESCRIBE DISKMAP
- DESCRIBE FLASHCACHE
- DESCRIBE FLASHCACHECONTENT
- DESCRIBE FLASHLOG
- DESCRIBE GRIDDISK
- DESCRIBE IBPORT
- DESCRIBE IORMPLAN
- DESCRIBE KEY
- DESCRIBE LUN
- DESCRIBE METRICCURRENT
- DESCRIBE METRICDEFINITION
- DESCRIBE METRICHISTORY
- DESCRIBE OFFLOADGROUP
- DESCRIBE PHYSICALDISK
- DESCRIBE PLUGGABLEDATABASE
- DESCRIBE PMEMCACHE
- DESCRIBE PMEMLOG
- DESCRIBE QUARANTINE

- DESCRIBE ROLE
- DESCRIBE SOFTWAREHISTORY
- DESCRIBE SOFTWAREUPDATE
- DESCRIBE THRESHOLD
- DESCRIBE USER

7.7.5.1 DESCRIBE ACTIVEREQUEST

Purpose

The `DESCRIBE ACTIVEREQUEST` command displays a list of attributes for the `ACTIVEREQUEST` object type.

Syntax

```
DESCRIBE ACTIVEREQUEST
```

Usage Notes

The attributes for the `DESCRIBE ACTIVEREQUEST` command can include the following:

- `asmDiskGroupNumber`: Number of the Oracle ASM disk group
- `asmFileIncarnation`: Incarnation number of the Oracle ASM file
- `asmFileNumber`: Number of the Oracle ASM file
- `consumerGroupID`: Identifier of the consumer group
- `consumerGroupName`: Name of the consumer group
- `dbID`: Database unique name
- `dbName`: Database name
- `dbRequestID`: Identifier of the database request
- `fileType`: File type associated with the request
- `id`: Unique identifier of the active request
- `instanceNumber`: Instance number associated with the request
- `ioBytes`: Number of bytes of I/O against the grid disk in the current session
- `ioBytesSoFar`: Number of total bytes of I/O
- `ioGridDisk`: Grid disk used by a request
- `ioOffset`: Measure of the offset on the grid disk
- `ioReason`: Reason for I/O activity, such as a control-file read
- `ioType`: Type of active request, such as `file initialization`, `read`, `write`, `predicate pushing`, `filtered backup read`, or `predicate push read`
- `name`: Unique name of the active request
- `objectNumber`: Object number associated with the request
- `parentID`: Identifier of the parent request

- `pdbID`: Identifier of the pluggable database
- `requestState`: State of the active request, such as
  - Accessing Disk
  - Computing Result
  - Network Receive
  - Network Send
  - Queued Extent
  - Queued for Disk
  - Queued for File Initialization
  - Queued for Filtered Backup Read
  - Queued for Network Send
  - Queued for Predicate Pushing
  - Queued for Read
  - Queued for Write
  - Queued in Resource Manager
- `sessionID`: Identifier of the session
- `sessionSerNumber`: Serial number of the database session
- `sqlID`: Identifier of the SQL command associated with the request
- `tableSpaceNumber`: Tablespace number associated with the request

Example 7-80 Describing the ACTIVEREQUEST Object

This example shows the `DESCRIBE` command with the `ACTIVEREQUEST` object.

```
CellCLI> DESCRIBE ACTIVEREQUEST

         name
         asmDiskGroupNumber
         asmFileIncarnation
         asmFileNumber
         consumerGroupID
         consumerGroupName
         dbID
         dbName
         dbRequestID
         fileType
         id
         instanceNumber
         ioBytes
         ioBytesSofar
         ioGridDisk
         ioOffset
         ioReason
         ioType
         objectNumber
         parentID
         pdbID
         requestState
         sessionID
         sessionSerNumber
         sqlID
         tableSpaceNumber
```

7.7.5.2 DESCRIBE ALERTDEFINITION

Purpose

The DESCRIBE ALERTDEFINITION command displays a list of attributes for the ALERTDEFINITION object type.

Syntax

```
DESCRIBE ALERTDEFINITION
```

Usage Notes

The attributes for the DESCRIBE ALERTDEFINITION command can include the following:

- alertShortName: Abbreviated name for the alert. If the alert is based on a metric, then the short name is the same as the corresponding metric name attribute.
- alertSource: Source of the alert, such as BMC or ADR
- alertType: Type of the alert. Values are stateful or stateless.
  - Stateful alerts are automatically cleared on transition to normal.
  - Stateless alerts are never cleared. You can change the alert by setting the examinedBy attribute.
- description: Description for the alert
- metricName: Metric name if the alert is based on a metric
- name: Identifier for the alert

Example 7-81 Describing the ALERTDEFINITION Object

```
CellCLI> DESCRIBE ALERTDEFINITION

         name
         alertShortName
         alertSource
         alertType
         description
         metricName
```

Related Topics

- Displaying Alert Definitions
  Use the LIST ALERTDEFINITION command to display the alert definitions for the storage server.

7.7.5.3 DESCRIBE ALERTHISTORY

Purpose

The `DESCRIBE ALERTHISTORY` command displays a list of attributes for the `ALERTHISTORY` object type.

Syntax

```
DESCRIBE ALERTHISTORY
```

Usage Notes

The attributes for the `DESCRIBE ALERTHISTORY` command can include the following:

- `alertAction`: Recommended action to perform for this alert
- `alertDescription`: Description for the alert
- `alertMessage`: Brief explanation of the alert
- `alertSequenceID`: Unique sequence ID for the alert. When an alert changes its state, such as `warning` to `critical` or `critical` to `clear`, another occurrence of the alert is created with the same sequence number and a time stamp of the transition.
- `alertShortName`: Abbreviated name for the alert. If the alert is based on a metric, then the short name is the same as the corresponding metric `name` attribute.
- `alertType`: Type of the alert. Values are stateful or stateless.
  - Stateful alerts are automatically cleared on transition to `normal`.
  - Stateless alerts are never cleared. You can change the alert by setting the `examinedBy` attribute.
- `beginTime`: Time stamp when an alert changes its state
- `endTime`: Time stamp for the end of the period when an alert changes its state
- `examinedBy`: Administrator who reviewed the alert
- `failedMail`: Intended e-mail recipient when a notification failed
- `failedSNMP`: Intended SNMP subscriber when a notification failed
- `metricObjectName`: Object, such as cell disk or grid disk, for which a metric threshold has caused an alert
- `metricValue`: Value of the metric that caused the alert
- `name`: Unique identifier for the alert
- `notificationState`: Number indicating progress in notifying subscribers to alert messages:
  - 0, or "non-deliverable": Never tried
  - 1, or "sent": Sent successfully
  - 2, or "attempting delivery": Retrying, up to 5 times
  - 3, or "delivery failed": Retry failed 5 times

  - 4, or "creating diagpack": The diagnostic packaging and the outgoing alert email are still pending for this alert. It is used by MS to keep track of the alerts that have not been processed or are being processed. This enables MS, in the event of a restart, to continue processing alerts with state 4 and send the alert email.
- `sequenceBeginTime`: Time stamp when an alert sequence ID is first created
- `serviceRequestLink`: The URL to the service request associated with the alert
- `serviceRequestNumber`: The service request number associated with the alert
- `severity`: Severity level. Values are:
  - `clear`
  - `info`
  - `warning`
  - `critical`

Example 7-82 Describing the ALERTHISTORY Object

```
CellCLI> DESCRIBE ALERTHISTORY

         name
         alertAction
         alertDescription
         alertMessage
         alertSequenceID
         alertShortName
         alertType
         beginTime
         endTime
         examinedBy              modifiable
         failedMail
         failedSNMP
         metricObjectName
         metricValue
         notificationState
         sequenceBeginTime
         serviceRequestLink
         serviceRequestNumber
         severity
```

Related Topics

- DESCRIBE METRICDEFINITION

7.7.5.4 DESCRIBE CELL

Purpose

The `DESCRIBE CELL` command displays a list of attributes for the `CELL` object type.

Syntax

```
DESCRIBE CELL
```

Usage Notes

The following list contains the attributes for the `DESCRIBE CELL` command.

- `accessLevelPerm`: Specifies the access level at which the cell runs by default. Value is either `remoteLoginEnabled` or `remoteLoginDisabled`.
- `accessLevelTemp`: Duration of time in which the access level is temporarily changed from the setting of `accessLevelPerm`
- `accountLockInDays`: Number of days after a password expires before a user account is locked. Available with Oracle Exadata System Software release 19.1.0 or later.
- `bbuStatus`: Status of hard disk controller battery-backed unit (BBU)
- `cellsrvStatus`: Status of Cell Server
- `cellVersion`: Release number of the cell software
- `comment`: User-supplied text string
- `cpuCount`: Number of CPUs on the cell
- `dbPerfDataSuppress`: Specifies which databases should not have their statistics reported in Automatic Workload Repository (AWR) reports
- `diagHistoryDays`: Number of days ADR files are retained. The default is 7 days.
- `diagPackEmailAttach`: Whether a diagpack is included as an attachment in the alert email or not. The default is `true`.
- `diagPackUploadEnabled`: Whether the auto diagpack upload feature is enabled or not. The default is `true`.
- `doNotServiceLEDStatus`: Status of the cell DoNotService LED. The value can be `on` or `off`.
- `eighthRack`: Specifies whether Oracle Exadata Database Machine Eighth Rack configuration for storage cells is enabled or disabled
- `emailFormat`: File format for e-mail messages. The value can be `html` or `text`.
- `emailSubscriber`: List of names that subscribe to the alert notifications
- `events`: String for *events*++ that is passed to Cell Server for debugging and trace information purposes
- `exacliEnabled`: Specifies whether `exacli` is enabled or disabled
- `fanCount`: Count of working fans and total fans, displayed as working/total
- `fanStatus`: Status of the fan. The value can be `normal`, `warning`, or `critical`.
- `flashCacheCompress`: Setting for flash cache compression. The value can be `true` or `false`.
- `flashCacheMode`: Setting for flash cache. The value can be `writethrough` or `writeback`. The default is `writethrough`.

- `httpsAccess`: Control list of IP addresses for HTTPs port access to the Exadata RESTful Service. Available with Oracle Exadata System Software release 19.1.0 or later.
- `id`: Global unique identifier (GUID) supplied by the hardware vendor
- `interconnect1-8`: Interconnect1 to interconnect8 for the cell. For example: `bondeth0` or `bondib0`.
- `interconnectCount`: Number of active InfiniBand network interconnects
- `iormBoost`: Ratio of the cumulative number of positions in the I/O queue that were skipped because of IORM scheduling to the number of I/Os that where scheduled. This ratio is calculated by sampling the changes each minute in the two numbers.
- `IOTimeoutThreshold`: Specifies the timeout threshold. If cell I/O takes longer than the defined threshold, then the I/O is canceled, and Oracle ASM redirects the I/O to another mirror copy of the data.
- `ipaddress1-8`: `ipaddress1` to `ipaddress8` for the cell
- `kernelVersion`: Version of the host kernel software
- `location`: Physical location of the cell hardware supplied by the user
- `locatorLEDStatus`: Status of cell LOCATE LED. The value can be `on` or `off`.
- `makeModel`: Make and model of the cell hardware supplied by the vendor
- `memoryGB`: The memory in gigabytes for the cell
- `metricCollection`: Indicator for whether Management Server performs metric collection. Values are `TRUE` or `FALSE`. If set to `FALSE`, then all collection and alert mining is stopped. The default setting is `TRUE`.
- `metricHistoryDays`: Number of days metric history files are retained. The default is 7 days.
- `msStatus`: Status of Management Server
- `name`: Unique name for the cell
- `notificationMethod`: Notification method for alerts. The value should be `mail`, `snmp`, `none`, or both `mail` and `snmp`.
- `notificationPolicy`: Indicator for severity alerts to be sent to subscribers. The value for `notificationPolicy` should be `none` or any combination of `critical`, `warning`, and `clear`.
- `offloadGroupEvents`: Used only under the guidance of Oracle Support
- `powerCount`: Count of power supplies, displayed as working/total
- `powerStatus`: Status of the power. The value can be `normal`, `warning`, or `critical`.
- `pwdExpInDays`: The number of days before a user's password expires. Available with Oracle Exadata System Software release 19.1.0 or later.
- `pwdExpWarnInDays`: The number of days before a user's password expires that a warning message is issued during login attempts. Available with Oracle Exadata System Software release 19.1.0 or later.
- `rackName`: The name of the rack

- `ramCacheMaxSize`: The maximum allowable size of the Cell RAM Cache. Available with Oracle Exadata System Software release 18c (18.1.0) or later.
- `ramCacheMode`: Determines whether the Cell RAM Cache is enabled (`on`) or disabled (`off`). Available with Oracle Exadata System Software release 18c (18.1.0) or later.
- `ramCacheSize`: The size of the Cell RAM Cache. Available with Oracle Exadata System Software release 18c (18.1.0) or later.
- `releaseImageStatus`: Indicator for knowing whether imaging is successful
- `releaseTrackingBug`: Patch number for the cell software, such as 11223344
- `releaseVersion`: Release number for the cell software, such as 11.2.2.3.0
- `remotePwdChangeAllowed`: Whether or not a user password can be changed remotely through REST services. Available with Oracle Exadata System Software release 19.1.0 or later.
- `rescuePlan`: A list of commands that you can run after a server rescue to restore settings, such as IORM plans, thresholds, and notifications, to the last known values.
- `rollbackVersion`: The inactive image version that the cell maintains. If `patchmgr rollback` is invoked for the cell, the value displayed by `rollbackVersion` is the software version that will be reinstated.
- `rpmVersion`: The RPM version of the cell
- `rsStatus`: Status of Restart Server
- `securityCert`: The certified identity of the cell. Either CA-certified identity or the default self-certified identity.
- `siteName`: The site name for the cell
- `smtpFrom`: SMTP e-mail user name that appears in alert notifications
- `smtpFromAddr`: SMTP e-mail address from which alert notifications are sent
- `smtpPort`: SMTP e-mail server port used to send alert notifications
- `smtpPwd`: Password of the SMTP e-mail user that sends alert notifications
- `smtpServer`: SMTP e-mail server used to send alert notifications
- `smtpToAddr`: Address to which e-mail is sent. It can be a comma-delimited list in quotation marks to allow multiple subscribers to alerts.
- `smtpUser`: SMTP e-mail user name that is displayed in alert notifications
- `smtpUseSSL`: Specification to use Secure Socket Layer (SSL) authentication
- `snmpSubscriber`: List of hosts that subscribe to the SNMP alert notifications
- `snmpUser`: Defines the user who receives SNMP alerts
- `status`: Status of the cell
- `syslogConf`: Designates syslog messages that should be forwarded to a specified management server. Uses the following syntax for the attribute, where *selector* is the message type, and *node* is the specified server:

  ```
  syslogconf = (selector @node' [, 'selector @node']... )
  ```

Both *selector* and *node* follow `syslog.conf` standard syntax rules.

- `temperatureReading`: Current temperature (Celsius) of the cell obtained from the BMC
- `temperatureStatus`: Status of the temperature. The value can be `normal`, `warning`, or `critical`.
- `traceLevel`: Level for which trace messages are written. The default is `FINE`. The value can be:
  - A valid Java logging level
    - `SEVERE`
    - `WARNING`
    - `INFO`
    - `CONFIG`
    - `FINE`
    - `FINER`
    - `FINEST`
  - A valid Oracle Diagnostic Logging (ODL) logging level
    - `INCIDENT_ERROR:1`
    - `ERROR:1`
    - `WARNING:1`
    - `NOTIFICATION:1`
    - `NOTIFICATION:16`
    - `TRACE:1`
    - `TRACE:16`
    - `TRACE:32`
- `upTime`: Time (*days*, *hours:minutes*) since the system was restarted
- `usbStatus`: Status of the USB device

Examples

The following example shows the `DESCRIBE` command with the `CELL` object.

Example 7-83 Describing the CELL Object

```
CellCLI> DESCRIBE CELL

    name                    modifiable
    accessLevelPerm         modifiable
    accessLevelTemp         modifiable
    accountLockInDays       modifiable
    bbuStatus
    cellsrvStatus
    cellVersion
    comment                 modifiable
    cpuCount
```

```
dbPerfDataSuppress     modifiable
diagHistoryDays        modifiable
diagPackEmailAttach    modifiable
diagPackUploadEnabled  modifiable
doNotServiceLEDStatus
eighthRack             modifiable
emailFormat            modifiable
emailSubscriber        modifiable
events                 modifiable
exacliEnabled          modifiable
fanCount
fanStatus
flashCacheCompress     modifiable
flashCacheMode         modifiable
httpsAccess            modifiable
id
interconnect1          modifiable
interconnect2          modifiable
interconnect3          modifiable
interconnect4          modifiable
interconnect5          modifiable
interconnect6          modifiable
interconnect7          modifiable
interconnect8          modifiable
interconnectCount
iormBoost
IOTimeoutThreshold     modifiable
ipaddress1
ipaddress2
ipaddress3
ipaddress4
ipaddress5
ipaddress6
ipaddress7
ipaddress8
kernelVersion
location               modifiable
locatorLEDStatus
makeModel
memoryGB
metricCollection       modifiable
metricHistoryDays      modifiable
msStatus
notificationMethod     modifiable
notificationPolicy     modifiable
offloadGroupEvents     modifiable
powerCount
powerStatus
pwdExpInDays           modifiable
pwdExpWarnInDays       modifiable
rackName               modifiable
ramCacheMaxSize        modifiable
ramCacheMode           modifiable
ramCacheSize
releaseImageStatus
```

```
releaseTrackingBug
releaseVersion
remotePwdChangeAllowed modifiable
rescuePlan             hidden
rollbackVersion
rpmVersion
rsStatus
securityCert           modifiable
siteName               modifiable
smtpFrom               modifiable
smtpFromAddr           modifiable
smtpPort               modifiable
smtpPwd                modifiable
smtpServer             modifiable
smtpToAddr             modifiable
smtpUser               modifiable
smtpUseSSL             modifiable
snmpSubscriber         modifiable
snmpUser               modifiable
status
syslogConf             modifiable
syslogFormat           modifiable
temperatureReading
temperatureStatus
traceLevel             modifiable
upTime
usbStatus
```

Related Topics

- ALTER CELL
- CREATE CELL

7.7.5.5 DESCRIBE CELLDISK

Purpose

The DESCRIBE CELLDISK command displays a list of attributes for the CELLDISK object type.

Syntax

```
DESCRIBE CELLDISK
```

Usage Notes

The attributes displayed by the DESCRIBE CELLDISK command can include:

- comment: User comment for the cell disk.
- creationTime: Time stamp when the cell disk was created.
- deviceName: Operating system device name of the LUN used by the cell disk.
- devicePartition: Operating system device name of the partition that is used by the cell disk.

- diskType: The type of disk.
- errorCount: Number of errors that occurred on the cell disk.
- flushError: Errors reported by while flushing the flash cache.
- flushStatus: The current status of the flash cache flush operation.
- freeSpace: Amount of unused space available on the cell disk.
- id: Global unique identifier (GUID) that is generated when the cell disk is created.
- name: Unique name of the cell disk.
- physicalDisk: Name of the physical disk on which the cell disk is located.
- size: Total size of the cell disk.
- status: Current status of the cell disk, such as normal or importRequired.

Example 7-84 Describing the CELLDISK Object

This shows the DESCRIBE command with the CELLDISK object.

```
CellCLI> DESCRIBE CELLDISK

         name                   modifiable
         comment                modifiable
         creationTime
         deviceName
         devicePartition
         diskType
         errorCount
         flushError             hidden
         flushStatus            hidden
         freeSpace
         id
         physicalDisk
         size
         status
```

Related Topics

- ALTER CELLDISK
- CREATE CELLDISK

7.7.5.6 DESCRIBE DATABASE

Purpose

Displays the specified attributes for active databases.

Syntax

```
DESCRIBE DATABASE
```

Usage Notes

The attributes for the DESCRIBE DATABASE command include the following:

- databaseID: The unique identifier for the database.
- flashCacheLimit: Specifies a "soft" maximum size in flash cache; if the flash cache is not full, a database can exceed the flashCacheLimit value.
- flashCacheMin: The minimum size in the flash cache that is guaranteed for a database even if the blocks are cold.
- flashCacheSize: The size specified for the flash cache.
- iormShare: The IORM database share number.
- lastRequestTime: The time stamp of the last request from the database.
- name: The database name.
- pmemCacheLimit: Specifies a "soft" maximum size in PMEM cache; if the PMEM cache is not full, a database can exceed the pmemCacheLimit value.
- pmemCacheMin: The minimum size in the PMEM cache that is guaranteed for a database even if the blocks are cold.
- pmemCacheSize: The size specified for the PMEM cache.
- profile: The IORM profile associated with the database.

Examples

The following example shows the DESCRIBE command with the DATABASE object.

Example 7-85 Describing the DATABASE Object

```
CellCLI> DESCRIBE DATABASE
         name
         databaseID
         flashCacheLimit
         flashCacheMin
         flashCacheSize
         iormShare
         lastRequestTime
         pmemCacheLimit
         pmemCacheMin
         pmemCacheSize
         profile
```

Related Topics

- LIST DATABASE
- LIST IORMPROFILE

7.7.5.7 DESCRIBE DISKMAP

Purpose

Displays the grid disk attributes for a physical disk.

Syntax

```
DESCRIBE DISKMAP
```

Usage Notes

The attributes displayed by the DESCRIBE DISKMAP command can include:

- celldisk: The cell disk name
- devicePartition: The disk partition name
- gridDisks: The name of the grid disks associated with the disk
- name: The disk name.
- physicalSerial: The serial number of the disk
- physicalSize: The size of the disk
- slotNumber: The slot number of the disk
- status: The disk status

Examples

The following example shows the DESCRIBE command with the DISKMAP object.

Example 7-86 Describing the DISKMAP Object

```
CellCLI> DESCRIBE CELLDISK

         name
         celldisk
         devicePartition
         gridDisks
         physicalSerial
         physicalSize
         slotNumber
         status
```

7.7.5.8 DESCRIBE FLASHCACHE

Purpose

The DESCRIBE FLASHCACHE command displays a list of attributes for the FLASHCACHE object type.

Syntax

```
DESCRIBE FLASHCACHE
```

Usage Notes

The attributes displayed by the DESCRIBE FLASHCACHE command can include:

- cellDisk: Cell disk names that contain Exadata Smart Flash Cache.
- creationTime: Time stamp when the Exadata Smart Flash Cache was created.
- degradedCelldisks: List of cell disks configured for cache but not currently available..

- effectiveCacheSize: Usable cache size after deducting space on unavailable cell disks.
- id: Global unique identifier (GUID) that is generated when the Exadata Smart Flash Cache is created.
- name: Unique name of the Exadata Smart Flash Cache.
- size: Total size of the Exadata Smart Flash Cache.
- status: Current status of the Exadata Smart Flash Cache, such as normal, warning or critical.

Examples

The following example shows the DESCRIBE command with the FLASHCACHE object.

Example 7-87 Describing the FLASHCACHE Object

```
CellCLI> DESCRIBE FLASHCACHE

         name
         cellDisk                 modifiable
         creationTime
         degradedCelldisks
         effectiveCacheSize
         id
         size                     modifiable
         status
```

7.7.5.9 DESCRIBE FLASHCACHECONTENT

Purpose

The DESCRIBE FLASHCACHECONTENT command displays a list of attributes for the FLASHCACHECONTENT object type.

Syntax

```
DESCRIBE FLASHCACHECONTENT
```

Usage Notes

The attributes displayed by the DESCRIBE FLASHCACHECONTENT command can include:

- cachedKeepSize: Size, in bytes, cached in keep mode for this object.
- cachedSize: Size, in bytes, cached for this object.
- cachedWriteSize: Size, in bytes, of cached data for this object in write-back flash cache that has not yet been written to hard disk.
- columnarCacheSize: Size, in bytes, cached in Hybrid Columnar Compression (HCC) format for this object.
- columnarKeepSize: Size, in bytes, cached in Hybrid Columnar Compression (HCC) format that is in keep mode for this object.
- dbID: Database unique name identifier.

- `dbUniqueName`: Database unique name.
- `hitCount`: Number of I/Os which read data from flash cache for this object.
- `hoursToExpiration`: Time before this object is downgraded from `keep` section, if not accessed again.
- `missCount`: Number of I/Os which read data from disk for this object.
- `objectNumber`: Dictionary object number of the segment which contains the object in the database.
- `tableSpaceNumber`: Tablespace number for the object number.

Examples

The following example shows the `DESCRIBE` command with the `FLASHCACHECONTENT` object.

Example 7-88 Describing the FLASHCACHECONTENT Object

```
CellCLI> DESCRIBE FLASHCACHECONTENT

         cachedKeepSize
         cachedSize
         cachedWriteSize
         columnarCacheSize
         columnarKeepSize
         dbID
         dbUniqueName
         hitcount
         hoursToExpiration
         missCount
         objectNumber
         tableSpaceNumber
```

7.7.5.10 DESCRIBE FLASHLOG

Purpose

The `DESCRIBE FLASHLOG` command displays a list of attributes for the `FLASHLOG` object type.

Syntax

```
DESCRIBE FLASHLOG
```

Usage Notes

The attributes displayed by the `DESCRIBE FLASHLOG` command can include:

- `cellDisk`: Names of the cell disks that contain Oracle Exadata Smart Flash Log.
- `creationTime`: Timestamp when Oracle Exadata Smart Flash Log was created.
- `degradedCelldisks`: List of cell disks configured for Oracle Exadata Smart Flash Log, but not currently available.

- effectiveSize: Size of available Oracle Exadata Smart Flash Log after deducting space on unavailable cell disks.
- efficiency: Efficiency of Oracle Exadata Smart Flash Log expressed as a percentage.
- id: Global unique identifier (GUID) generated when Oracle Exadata Smart Flash Log is created.
- name: Unique name of Oracle Exadata Smart Flash Log.
- size: Total size of Oracle Exadata Smart Flash Log.
- status: Current status of Oracle Exadata Smart Flash Log, such as normal, warning or critical. Status normal indicates all flash disks are available. Status warning indicates some flash disks are not available. Status critical indicates all flash disks are unavailable.

Examples

The following example shows the DESCRIBE command with the FLASHLOG object.

Example 7-89 Describing the FLASHLOG Object

```
CellCLI> DESCRIBE FLASHLOG

         name
         cellDisk
         creationTime
         degradedCelldisks
         effectiveSize
         efficiency
         id
         size
         status
```

7.7.5.11 DESCRIBE GRIDDISK

Purpose

The DESCRIBE GRIDDISK command displays a list of attributes for the GRIDDISK object type.

Syntax

```
DESCRIBE GRIDDISK
```

Usage Notes

The attributes for the DESCRIBE GRIDDISK command include the following:

- asmDeactivationOutcome: Indicator whether a grid disk can be deactivated without loss of data. A value of YES indicates the grid disk can be deactivated without data loss.
- asmDiskgroupName: Name of the Oracle ASM disk group.
- asmDiskName: Name of the Oracle ASM disk.

- `asmDiskRepairTime`: The amount of time the grid disk can remain offline before it is dropped by Oracle ASM.
- `asmDiskSize`: Size of the Oracle ASM disk.

  This attribute is available in Oracle Exadata System Software release 12.1.2.3.0 and later.
- `asmFailGroupName`: Name of the Oracle ASM failure group.
- `asmModeStatus`: Indicator shows the current Oracle ASM usage of a grid disk. Statuses are `ONLINE`, `OFFLINE`, `DROPPED`, `UNUSED`, `SYNCING`, or `UNKNOWN`.
- `availableTo`: Names of the clients that can access this grid disk.
- `cachedBy`: The name of the flash disks that are currently caching data for this grid disk for write-back flash cache.
- `cachingPolicy`: The flash caching policy for this grid disk. Values are `default` or `none`.
  - `default` means data for this grid disk uses the flash cache.
  - `none` means data for this grid disk do not use flash cache.

  The caching policy can be set when creating a grid disk, or using the `ALTER GRIDDISK` command.
- `cellDisk`: Name of the cell disk that contains the grid disk.
- `comment`: User-supplied text string.
- `creationTime`: Time stamp when the grid disk was created.
- `diskType`: The type of disk.
- `errorCount`: Count of hardware errors detected by the cell disk containing this grid disk.
- `id`: Global unique identifier (GUID) that is generated when the grid disk is created.
- `name`: Unique name of the grid disk.
- `size`: Total size of the grid disk.
- `sparse`: Whether the grid disk is a sparse disk.
- `status`: Current status of the grid disk, such as `active`, `inactive`, `not present` or `importRequired`.
- `virtualSize`: The size of the disk group for the sparse grid disks.

> **Note:**
>
> The `asmDeactivationOutcome` and `asmModeStatus` attributes must be explicitly specified when using the `LIST GRIDDISK` command.

Example 7-90 Describing the GRIDDISK Object

This example shows the `DESCRIBE` command with the `GRIDDISK` object.

```
CellCLI> DESCRIBE GRIDDISK

         name                       modifiable
         asmDeactivationOutcome     hidden
         asmDiskgroupName
         asmDiskName
         asmDiskRepairTime          hidden
         asmDiskSize                hidden
         asmFailGroupName
         asmModeStatus              hidden
         availableTo                modifiable
         cachedBy
         cachingPolicy              modifiable
         cellDisk
         comment                    modifiable
         creationTime
         diskType
         errorCount
         id
         size                       modifiable
         sparse
         status
         virtualSize                modifiable
```

Related Topics

- CREATE GRIDDISK
- ALTER GRIDDISK

7.7.5.12 DESCRIBE IBPORT

Purpose

The `DESCRIBE IBPORT` command displays a list of attributes for the IBPORT object type on systems that use RDMA over InfiniBand Network Fabric.

> **Note:**
>
> This command does not apply to Oracle Exadata Database Machine X8M systems.

Syntax

```
DESCRIBE IBPORT
```

Usage Notes

The attributes for the `DESCRIBE IBPORT` command can include the following:

- `activeSlave`: Indicator whether the port is currently the active port for the bonded IP.
- `dataRate`: The data rate of the RDMA over InfiniBand Network Fabric port.
- `hcaFWVersion`: The version of the host channel adapter firmware.
- `id`: The Global unique identifier (GUID) of the RDMA over InfiniBand Network Fabric port.
- `lid`: The local identifier of the RDMA over InfiniBand Network Fabric port. It is unique within the subnet, and the 16-bit identifiers are used within a network by switches for routing.
- `linkDowned`: The number of times the port training state machine has failed the link error recovery process, and halted the link.
- `linkIntegrityErrs`: The number of link integrity errors.
- `linkRecovers`: The number of times the port training state machine has successfully completed the link error recovery process.
- `name`: The name of the RDMA over InfiniBand Network Fabric port.
- `physLinkState`: The physical link state.
- `portNumber`: The port number of the RDMA over InfiniBand Network Fabric port.
- `rcvConstraintErrs`: The number of received constraint errors experienced by the RDMA over InfiniBand Network Fabric port.
- `rcvData`: The number of 32-bit data words received by the RDMA over InfiniBand Network Fabric port.
- `rcvErrs`: The number of packets received at the RDMA over InfiniBand Network Fabric port containing an error.
- `rcvRemotePhysErrs`: The number of physical errors experienced at the RDMA over InfiniBand Network Fabric port.
- `status`: The link status.
- `symbolErrs`: The number of minor link errors experienced at the RDMA over InfiniBand Network Fabric port.
- `vl15Dropped`: The number of incoming VL15 packets dropped at the RDMA over InfiniBand Network Fabric port due to resource limitations, such as lack of buffers.
- `xmtConstraintErrs`: The number of transmitted constraint errors experienced at the RDMA over InfiniBand Network Fabric port.
- `xmtData`: The number of 32-bit data words transmitted on the RDMA over InfiniBand Network Fabric port.
- `xmtDiscards`: The number of outbound packets discarded by the RDMA over InfiniBand Network Fabric port because the port was down or congested.

Example 7-91 Describing the IBPORT Object

The following example shows the `DESCRIBE` command with the `IBPORT` object on systems that use RDMA over InfiniBand Network Fabric.

```
CellCLI> DESCRIBE IBPORT

        name
```

```
          activeSlave
          dataRate
          hcaFWVersion
          id
          lid
          linkDowned
          linkIntegrityErrs
          linkRecovers
          physLinkState
          portNumber
          rcvConstraintErrs
          rcvData
          rcvErrs
          rcvRemotePhysErrs
          status
          symbolErrs
          vl15Dropped
          xmtConstraintErrs
          xmtData
          xmtDiscards
```

7.7.5.13 DESCRIBE IORMPLAN

Purpose

The `DESCRIBE IORMPLAN` command displays a list of attributes for the `IORMPLAN` object type.

Syntax

```
DESCRIBE IORMPLAN
```

Usage Notes

The attributes for the `DESCRIBE IORMPLAN` command can include the following:

- `catPlan`: Allocation plan for the categories set up in the databases using the cell.
- `dbPlan`: Allocation plan for the databases using the cell.
- `name`: Unique name of the interdatabase plan. The `name` value is automatically set to *`cellname`*`_IORMPLAN`.
- `objective`: Optimization mode for IORM.
- `status`: Current status of the interdatabase plan, either `active` or `inactive`.

Examples

The following example shows the `DESCRIBE` command with the `IORMPLAN` object.

Example 7-92 Describing the IORMPLAN Object

```
CellCLI> DESCRIBE IORMPLAN

         name
         catPlan                 modifiable
         dbPlan                   modifiable
         objective                modifiable
         status
```

Related Topics

- ALTER IORMPLAN

7.7.5.14 DESCRIBE KEY

Purpose

The `DESCRIBE KEY` command displays a list of attributes for the `KEY` object type.

Syntax

```
DESCRIBE KEY
```

Usage Notes

The attributes for the `DESCRIBE KEY` command can include the following:

- `key`: Random hexadecimal string used to assign client keys.
- `name`: Name of the key. The value of this field is not displayed with `LIST`.
- `type`: The type of key.

Examples

The following example shows the `DESCRIBE` command with the `KEY` object.

Example 7-93 Describing the KEY Object

```
CellCLI> DESCRIBE KEY

         name
         key                      modifiable
         type                     modifiable
```

Related Topics

- CREATE KEY

7.7.5.15 DESCRIBE LUN

Purpose

The `DESCRIBE LUN` command displays a list of attributes for the `LUN` object type.

Syntax

```
DESCRIBE LUN
```

Usage Notes

The attributes for the DESCRIBE LUN command can include the following:

- cellDisk: The name of the flash disk, for example FD_02_rack1celadm10. Not used for hard disks.
- deviceName: Operating system device name for the LUN. For example, /dev/c1d5
- diskType: The type of disk.
- errorCount: Number of errors on this LUN.
- id: Identifier assigned by the system.
- isSystemLun: Indicator whether the disk is a system disk. If value is TRUE, then the disk is a system disk. If the value is FALSE, then the disk is not a system disk, and only has data on it.
- lunSize: Raw size of the LUN before being converted to a cell disk.
- lunUID: Unique identifier assigned by the system.
- lunWriteCacheMode: Status of LUN write cache. The status can be in Write Through Mode or Write Back Mode.
- name: Unique name assigned to the LUN. This might be different (or extended from) the LUN ID if the ID is not unique.
- overProvisioning: Indicator of the percentage of over-provisioned blocks in flash storage that are still available for a particular LUN. This attribute is only used for flash disks.
- physicalDrives: Physical disk names that form the LUN.
- raidLevel: Value of the RAID level that is used on the LUN. For example: RAID 0.
- status: Status of the LUN, which can be normal, warning, or critical.

Examples

The following example shows the DESCRIBE command with the LUN object.

Example 7-94 Describing the LUN Object

```
CellCLI> DESCRIBE LUN

         name
         cellDisk
         deviceName
         diskType
         errorCount
         id
         isSystemLun
         lunSize
         lunUID
         lunWriteCacheMode
         overProvisioning
         physicalDrives
         raidLevel
         status
```

Related Topics

- LIST DISKMAP

7.7.5.16 DESCRIBE METRICCURRENT

Purpose

The DESCRIBE METRICCURRENT command displays a list of attributes for the METRICCURRENT object type.

Syntax

```
DESCRIBE METRICCURRENT
```

Usage Notes

The attributes for the DESCRIBE METRICCURRENT command can include the following:

- alertState: Indicator of the alert state. Values are normal, warning, or critical.
- collectionTime: Time stamp when the metric value was collected
- metricObjectName: Name of the object, such as cell disk, grid disk, and consumer group, being measured
- metricType: Specification for how the statistic was created or defined
- metricValue: Value of the metric when it was collected
- name: Unique name of the current metric
- objectType: Type of object being measured. Values are:
  - CELL
  - CELL_FILESYSTEM
  - CELLDISK
  - FLASHCACHE
  - FLASHLOG
  - GRIDDISK
  - IBPORT
  - IORM_CATEGORY
  - IORM_CONSUMER_GROUP
  - IORM_DATABASE
  - IORM_PLUGGABLE_DATABASE
  - HOST_INTERCONNECT
  - SMARTIO

Examples

The following example shows the DESCRIBE command with the METRICCURRENT object.

Example 7-95 Describing the METRICCURRENT Object

```
CellCLI> DESCRIBE METRICCURRENT

         name
         alertState
         collectionTime
         metricObjectName
         metricType
         metricValue
         objectType
```

Related Topics

- DESCRIBE METRICDEFINITION
- Monitoring Oracle Exadata Storage Server with Metrics
  You can monitor a cell by viewing the Oracle Exadata Storage Server metrics.

7.7.5.17 DESCRIBE METRICDEFINITION

Purpose

The `DESCRIBE METRICDEFINITION` command displays a list of attributes for the `METRICDEFINITION` object type.

Syntax

```
DESCRIBE METRICDEFINITION
```

Usage Notes

The attributes for the `DESCRIBE METRICDEFINITION` command can include the following:

- `description`: Description of the metric.
- `metricType`: Indicator of how the statistic was created or defined. The options are as follows:
  - `cumulative`: Cumulative statistics since the metric was created.
  - `instantaneous`: Value at the time that the metric is collected.
  - `rate`: Rates computed by averaging statistics over observation periods.
  - `transition`: Transition metrics are collected at the time their value has changed and typically capture important transitions in hardware status.
- `name`: Unique name of the metric definition. (Details follow this list.)
- `objectType`: Type of object being measured. Values are:
  - `CELL`
  - `CELL_FILESYSTEM`
  - `CELLDISK`
  - `FLASHCACHE`

  - `FLASHLOG`
  - `GRIDDISK`
  - `IBPORT`
  - `IORM_CATEGORY`
  - `IORM_CONSUMER_GROUP`
  - `IORM_DATABASE`
  - `IORM_PLUGGABLE_DATABASE`
  - `HOST_INTERCONNECT`
  - `SMARTIO`
- `persistencePolicy`: Amount of time metric values are stored.
- `unit`: Unit for the metric explicitly, and is related to the metric collected:
  - Number
  - % (percentage)
  - F (Fahrenheit)
  - C (Celsius)
  - IO/sec
  - “IO requests”
  - KB
  - KB/sec
  - MB
  - MB/sec
  - /min
  - ms
  - ms/request
  - ms/sec
  - us (*microseconds*)
  - us/request
  - us/sec

The value of the `name` attribute is a composite of abbreviations. The attribute value starts with an abbreviation of the object type on which the metric is defined:

- `CD_` (cell disk)
- `CG_` (IORM consumer group, database-qualified)
- `CL_` (cell)
- `CT_` (IORM category)
- `DB_` (IORM database-level)
- `FC_` (flash cache)
- `FL_` (flash log)

- GD_ (grid disk)
- IORM
- N_ (network, IBPORT, HOST_INTERCONNECT)
- PDB_ (IORM pluggable database)
- SIO_ (Smart IO)

After the abbreviation of the object type, most of the name attributes contain one of the following combinations to identify the operation:

- IO_BY (I/O amount)
- IO_RQ (number of I/O requests)
- IO_TM (I/O latency)
- IO_WT (I/O wait time)
- FC_IO_BY (Flash cache I/O amount)
- FC_IO_RQ (Flash cache I/O requests)
- FD_IO_BY (Flash disk I/O amount)
- FD_IO_RQ (Flash disk I/O requests)
- FD_IO_TM (Flash disk latency)
- FD_IO_UTIL (Flash disk utilization percentage)

Next, in the name could be _R or _W for read or write. Following that in the name attribute value there might be _SM or _LG to identify small or large blocks, respectively. At the end of the name, there could be _SEC to signify per seconds or _RQ to signify per request.

For consumer group and category metrics, read or write details are omitted.

For example:

- CD_IO_RQ_R_SM is the number of requests to read small blocks on a cell disk.
- GD_IO_TM_W_LG is the microseconds of I/O latency writing large blocks on a grid disk.

Examples

The following example shows the DESCRIBE command with the METRICDEFINITION object.

Example 7-96 Describing the METRICDEFINITION Object

```
CellCLI> DESCRIBE METRICDEFINITION

         name
         description
         metricType
         objectType
         persistencePolicy
         unit
```

Related Topics

- LIST METRICDEFINITION
- Monitoring Oracle Exadata Storage Server with Metrics
  You can monitor a cell by viewing the Oracle Exadata Storage Server metrics.

7.7.5.18 DESCRIBE METRICHISTORY

Purpose

The DESCRIBE METRICHISTORY command displays a list of attributes for the METRICHISTORY object type.

Syntax

```
DESCRIBE METRICHISTORY
```

Usage Notes

The attributes for the DESCRIBE METRICHISTORY command can include the following:

- alertState: Indicator of the alert state. Values are normal, warning, or critical.
- collectionTime: Time stamp when the metric value was collected
- metricObjectName: Name of the object, such as cell disk, grid disk, and consumer group, being measured
- metricType: Specification for how the statistic was created or defined
- metricValue: Value of the metric when it was collected
- metricValueAvg: Average value of the metric
- metricValueMax: Maximum value of the metric
- metricValueMin: Minimum value of the metric
- name: Name of the current metric
- objectType: Type of object being measured. Values are:
  - CELL
  - CELL_FILESYSTEM
  - CELLDISK
  - FLASHCACHE
  - FLASHLOG
  - GRIDDISK
  - IBPORT
  - IORM_CATEGORY
  - IORM_CONSUMER_GROUP
  - IORM_DATABASE
  - IORM_PLUGGABLE_DATABASE
  - HOST_INTERCONNECT

  - SMARTIO

Examples

The following example shows the DESCRIBE command with the METRICHISTORY object.

Example 7-97 Describing the METRICHISTORY Object

```
CellCLI> DESCRIBE METRICHISTORY

         name
         alertState
         collectionTime
         metricObjectName
         metricType
         metricValue
         metricValueAvg
         metricValueMax
         metricValueMin
         objectType
```

7.7.5.19 DESCRIBE OFFLOADGROUP

Purpose

The DESCRIBE OFFLOADGROUP command displays a list of attributes for the OFFLOADGROUP object type.

Syntax

```
DESCRIBE OFFLOADGROUP
```

Usage Notes

The attributes for the DESCRIBE OFFLOADGROUP command can include the following:

- autoStart: Whether the offload server associated with the offload group is dynamically started. Value can be either true or false.
- comment: An optional comment
- creationTime: The time when the offload group was created
- id: An identifier for the offload group
- isSystemGroup: Whether the offload group was created by the system software. The value can be either true or false.
- name: The name of the offload group
- package:
- runtimeState: The current state of the offload group process. The value can be running or stopped.

Example 7-98 Describing the OFFLOADGROUP Object

The following example shows ....

```
CellCLI> DESCRIBE OFFLOADGROUP
        name
        autoStart
        comment                 modifiable
        creationTime
        id
        isSystemGroup
        package                 modifiable
        runtimeState
```

7.7.5.20 DESCRIBE PHYSICALDISK

Purpose

The `DESCRIBE PHYSICALDISK` command displays a list of attributes for the `PHYSICALDISK` object type.

Syntax

```
DESCRIBE PHYSICALDISK
```

Usage Notes

The attributes for the `DESCRIBE PHYSICALDISK` command can include the following:

- `ctrlFirmware`: The hard disk controller software version
- `ctrlHwVersion`: The hard disk controller hardware version
- `deviceID`: The ID for the physical disk
- `deviceName`: The name of the physical disk device, for example `/dev/sdx`
- `diskType`: Type of the disk, whether it is a `HardDisk`, `FlashDisk`, or `M2Disk`.
- `enclosureDeviceId`: Identifier for the hard disk enclosure. This attribute is only applicable to Oracle Exadata System Software on Oracle Exadata Storage Server.
- `errCmdTimeoutCount`: The count of execution of commands related to physical disks that timed out, for example, disk firmware upgrade, listing physical disks, and so on.
- `errHardReadCount`: Total count of read errors on a physical disk
- `errHardWriteCount`: Total count of write errors on a physical disk
- `errorCount`: The sum of all known error counts for a physical disk
- `errOtherCount`: Total error count of all other (unknown) errors for a physical disk
- `errSeekCount`: Total number of disk seek errors
- `flashLifeLeft`: The percentage of flash disk life left for a disk
- `hotPlugCount`: Total number of times a disk has been pulled out and reinserted (hot plugged)

- lastFailureReason: The reason for the last physical disk failure
- luns: List of LUNs converted from this disk. M.2 disks do not have LUNs.
- makeModel: Model description provided by the system
- name: Unique name of the physical disk
- notPresentSince: Date at which the disk was no longer detected
- physicalFirmware: The version of the firmware
- physicalInsertTime: Time that the disk was inserted
- physicalInterface: Interface type used by the hard disk. For example, SAS
- physicalPort: (Only appplicable for HP models) The physical disk port value
- physicalRPM: The RPM value of a physical hard disk. This attribute is also used to determine the disk type (SATA or SAS).
- physicalSerial: System-assigned unique ID
- physicalSize: Size of the disk in bytes
- physicalUseType: Intended use of the disk, for example, Data Drive
- sectorRemapCount: Total number of physical disk sectors that have been remapped because of sector failures
- slotNumber: Physical location of disk. This attribute is only applicable to Oracle Exadata System Software on Oracle Exadata Storage Server.
- status: Status of the physical disk. Values can be:
  - failed: The disk has failed. In earlier releases, this status was called critical.
  - normal: The disk is functioning normally
  - not present: The disk has been removed
  - peer failure: Flash disk failure only
  - poor performance: The disk is performing poorly
  - predictive failure: The disk is expected to fail
  - write-through caching: Flash disk caching only.

Example 7-99 Describing the PHYSICALDISK Object in Oracle Exadata Storage Server

```
CellCLI> DESCRIBE PHYSICALDISK
        name
        ctrlFirmware
        ctrlHwVersion
        deviceId
        deviceName
        diskType
        enclosureDeviceId
        errCmdTimeoutCount
        errHardReadCount
        errHardWriteCount
        errorCount
        errOtherCount
```

```
errSeekCount
flashLifeLeft
hotPlugCount
lastFailureReason
luns
makeModel
notPresentSince
physicalFirmware
physicalInsertTime
physicalInterface
physicalPort
physicalRPM
physicalSerial
physicalSize
physicalUseType
sectorRemapCount
slotNumber
status
```

7.7.5.21 DESCRIBE PLUGGABLEDATABASE

Purpose

The DESCRIBE PLUGGABLEDATABASE command displays a list of attributes for the PLUGGABLEDATABASE object type.

Syntax

```
DESCRIBE PLUGGABLEDATABASE
```

Usage Notes

The attributes for the DESCRIBE PLUGGABLEDATABASE command can include the following:

- asmClusterName: The Oracle ASM cluster name or alias. Available with Oracle Exadata System Software release 19.1.0 or later.
- containerName: The name of the container database (CDB)
- flashCacheLimit: The specified limit on the Flash cache for this pluggable database (PDB)
- flashCacheMin: The specified minimum size of the Flash cache for this PDB
- flashCacheSize: The size of the Flash cache specified for this PDB
- iormLimit: The disk I/O utilization limit for the PDB. Available with Oracle Exadata System Software release 19.1.0 or later.
- iormShare: The IORM share number for the PDB. Available with Oracle Exadata System Software release 19.1.0 or later.
- name: The name of the PDB
- pdbID: The ID for the PDB

- `pmemCacheLimit`: The specified limit on the PMEM cache for this pluggable database (PDB)
- `pmemCacheMin`: The specified minimum size of the PMEM cache for this PDB
- `pmemCacheSize`: The size of the PMEM cache specified for this PDB

Example 7-100 Describing the PLUGGABLEDATABASE Object

```
CellCLI> DESCRIBE PLUGGABLEDATABASE
        name
        asmClusterName
        containerName
        flashCacheLimit
        flashCacheMin
        flashCacheSize
        iormLimit
        iormShare
        pdbID
        pmemCacheLimit
        pmemCacheMin
        pmemCacheSize
```

7.7.5.22 DESCRIBE PMEMCACHE

Purpose

The `DESCRIBE PMEMCACHE` command displays a list of attributes for the `PMEMCACHE` object type.

Syntax

```
DESCRIBE PMEMCACHE
```

Usage Notes

The attributes displayed by the `DESCRIBE PMEMCACHE` command can include:

- `cellDisk`: Cell disk names used by the PMEM cache.
- `creationTime`: Time stamp when the PMEM cache was created.
- `degradedCelldisks`: List of cell disks configured for the cache but not currently available.
- `effectiveCacheSize`: Usable PMEM cache size after deducting space on unavailable cell disks.
- `id`: Global unique identifier (GUID) that is generated when the PMEM cache is created.
- `name`: Unique name of the PMEM cache.
- `size`: Total size of the PMEM cache.
- `status`: Current status of the PMEM cache, such as `normal`, `warning` or `critical`.

Examples

The following example shows the `DESCRIBE` command with the `PMEMCACHE` object.

Example 7-101 Describing the PMEMCACHE Object

```
CellCLI> DESCRIBE PMEMCACHE

         name
         cellDisk                 modifiable
         creationTime
         degradedCelldisks
         effectiveCacheSize
         id
         size                     modifiable
         status
```

7.7.5.23 DESCRIBE PMEMLOG

Purpose

The `DESCRIBE PMEMLOG` command displays a list of attributes for the `PMEMLOG` object type.

Syntax

```
DESCRIBE PMEMLOG
```

Usage Notes

The attributes displayed by the `DESCRIBE PMEMLOG` command can include:

- `cellDisk`: Names of the cell disks that contain PMEMLOG.
- `creationTime`: Timestamp when PMEMLOG was created.
- `degradedCelldisks`: List of cell disks configured for PMEMLOG that are not currently available.
- `effectiveSize`: Size of available PMEMLOG after deducting space on unavailable cell disks.
- `efficiency`: Efficiency of PMEMLOG expressed as a percentage.
- `id`: Global unique identifier (GUID) generated when PMEMLOG is created.
- `name`: Unique name of PMEMLOG.
- `size`: Total size of the PMEMLOG.
- `status`: Current status of PMEMLOG.
  - `normal`: All PMEM cell disks are available.
  - `warning`: Some PMEM cell disks are not available.
  - `critical`: All PMEM cell disks are unavailable.

Example 7-102 Describing the PMEMLOG Object

This example shows the `DESCRIBE` command with the `PMEMLOG` object.

```
CellCLI> DESCRIBE PMEMLOG
     name
     cellDisk
     creationTime
     degradedCelldisks
     effectiveSize
     efficiency
     id
     size
     status
```

7.7.5.24 DESCRIBE QUARANTINE

Purpose

The `DESCRIBE QUARANTINE` command displays a list of attributes for the `QUARANTINE` object type.

Syntax

```
DESCRIBE QUARANTINE
```

Usage Notes

The attributes for the `DESCRIBE QUARANTINE` command can include the following:

- `asmClusterId`: Identifier of the ASM cluster. This attribute is available in Exadata software 12.2.1.1.0 and later.
- `catDBPlan`: The name of the category plan
- `cellsrvChecksum`: Checksum of the CELLSRV binary
- `clientPID`: The process identifier for the client process which crashed the cell
- `comment`: Comment for the quarantine
- `conDbUniqueID`: The container database unique ID for the quarantine
- `conDbUniqueName`: The container database unique name for the quarantine
- `crashReason`: Reason for the crash
- `creationTime`: Quarantine creation time
- `dbUniqueID`: The database unique ID for the quarantine
- `dbUniqueName`: The database unique name for the quarantine
- `fineGrainControl`:
- `fineGrainValue`:
- `incidentID`: The incident identifier of the crash that caused the quarantine creation
- `interDBPlan`: The name of the interdatabase resource plan

- `intraDBPlan`: The name of the intradatabase resource plan
- `ioBytes`: The bytes of quarantined disk region. This is applicable to disk region quarantine only.
- `ioGridDisk`: The grid disk name for quarantined disk region. This is applicable to disk region quarantine only.
- `ioOffset`: The I/O offset for quarantined disk region. This is applicable to disk region quarantine only.
- `name`: Identifier of the quarantine
- `objectID`:
- `planLineID`: The SQL Plan Line identifier. This is applicable to SQL Plan quarantine only.
- `quarantineMode`:
- `quarantinePlan`: This is usually SYSTEM
- `quarantineReason`: The reason for creation of the quarantine
- `quarantineType`: The type of quarantine created
- `remoteHostName`: The host name of the remote host that ran the client process that crashed the cell
- `rpmVersion`: The RPM version of the cell being used when the cell crashed
- `sqlID`: The SQLID of the SQL statement that crashed a cell
- `sqlPlanHashValue`: The SQL Plan hash value. This is applicable to SQL Plan quarantine only.

Example 7-103   Describing the QUARANTINE Object

```
CellCLI> DESCRIBE QUARANTINE
        name
        asmClusterId
        catDBPlan
        cellsrvChecksum
        clientPID
        comment                modifiable
        conDbUniqueID
        conDbUniqueName
        crashReason
        creationTime
        dbUniqueID
        dbUniqueName
        fineGrainControl
        fineGrainValue
        incidentID
        interDBPlan
        intraDBPlan
        ioBytes
        ioGridDisk
        ioOffset
        objectID
        planLineID
        quarantineMode
```

```
               quarantinePlan
               quarantineReason
               quarantineType
               remoteHostName
               rpmVersion
               sqlID
               sqlPlanHashValue
```

7.7.5.25 DESCRIBE ROLE

Purpose

The DESCRIBE ROLE command displays a list of attributes for the ROLE object type.

Syntax

```
DESCRIBE ROLE
```

Usage Notes

The attributes for the DESCRIBE ROLE command can include the following:

- name: Unique name of the user assigned the role
- privileges: Privileges granted to the role

Example 7-104 Describing the ROLE Object

```
CellCLI> DESCRIBE ROLE
        name
        privileges
```

7.7.5.26 DESCRIBE SOFTWAREHISTORY

Purpose

The DESCRIBE SOFTWAREHISTORY command displays a list of attributes for the ALERTHISTORY object type.

Syntax

```
DESCRIBE SOFTWAREHISTORY
```

Usage Notes

The attributes for the DESCRIBE SOFTWAREHISTORY command can include the following:

- name: The name of the software update
- status: The status of the software update

Example 7-105 Describing the SOFTWAREHISTORY Object

```
CellCLI> DESCRIBE SOFTWAREHISTORY
        name
        status
```

7.7.5.27 DESCRIBE SOFTWAREUPDATE

Purpose

The DESCRIBE SOFTWAREUPDATE command displays a list of attributes for the SOFTWAREUPDATE object type.

Syntax

```
DESCRIBE SOFTWAREUPDATE
```

Usage Notes

The attributes for the DESCRIBE SOFTWAREUPDATE command can include the following:

- frequency: The time period in which this software update is performed automatically. The value can be none, daily, weekly, or biweekly. The value none is available in Oracle Exadata System Software release 19.1.0 or later.
- name: The name of the patch to use in the update, or unknown. If the name defaults to unknown, then when the software update is performed, the most recent patch is chosen for the upgrade.
- status: The status of this software update.
- store: The URL for the location of the software update file
- time: The specified date and time at which the software update should be performed
- timeLimitInMinutes: The number of minutes a cell will spend waiting to update the software before canceling and issuing an alert.

Example 7-106 Describing the SOFTWAREUPDATE Object

```
CellCLI> DESCRIBE SOFTWAREUPDATE
        name                    modifiable
        status
        store                   modifiable
        time                    modifiable
        timeLimitInMinutes      modifiable
```

7.7.5.28 DESCRIBE THRESHOLD

Purpose

The DESCRIBE THRESHOLD command displays a list of attributes for the THRESHOLD object type.

Syntax

```
DESCRIBE THRESHOLD
```

Usage Notes

The attributes displayed by the `DESCRIBE THRESHOLD` command can include:

- `comparison`: Operator for comparing the metric value to the threshold value (>, >=, =, <, <=) to determine whether the value violates the threshold.
- `critical`: Limit beyond which the metric value is considered to be in the critical state for generating alerts
- `name`: Unique name of the threshold
- `observation`: Number of measurements over which the rate metric is averaged before being compared with the threshold value
- `occurrences`: Number of consecutive violations of the threshold limit by the metric value before the appropriate alert is issued
- `warning`: Limit beyond which the metric value is considered to be in the warning state for generating alerts

Example 7-107   Describing the THRESHOLD Object

```
CellCLI> DESCRIBE THRESHOLD

         name
         comparison               modifiable
         critical                 modifiable
         observation              modifiable
         occurrences              modifiable
         warning                  modifiable
```

7.7.5.29 DESCRIBE USER

Purpose

The `DESCRIBE USER` command displays a list of attributes for the `USER` object type.

Syntax

```
DESCRIBE USER
```

Usage Notes

The attributes displayed by the `DESCRIBE USER` command can include:

- `name`: Unique name of the user
- `roles`: Roles assigned to the user

Example 7-108 Describing the USER Object

```
CellCLI> DESCRIBE USER
         name
         roles
```

7.7.6 DROP

Purpose

The DROP command removes the named objects from the cell or resets a cell.

Syntax

```
DROP object_type [object_name [, object_name]...] [options]
```

Usage Notes

- When multiple objects are the target of a DROP command, there is the possibility of partial success. If an error occurs, then the command is interrupted, and the remaining objects are not dropped.
- DROP ALERTHISTORY
- DROP CELL
- DROP CELLDISK
- DROP FLASHCACHE
- DROP FLASHLOG
- DROP GRIDDISK
- DROP PMEMCACHE
- DROP PMEMLOG
- DROP QUARANTINE
- DROP ROLE
- DROP SOFTWAREHISTORY
- DROP THRESHOLD
- DROP USER

7.7.6.1 DROP ALERTHISTORY

Purpose

The DROP ALERTHISTORY command removes alerts from the alert history of a cell.

Syntax

```
DROP ALERTHISTORY {ALL | alert1 {, alert2}, ...}
```

Usage Notes

- In the command, *alertN* is the name of the alert to be dropped from the history.

- When dropping stateful alerts, you must drop all members of the alert sequence at the same time. If you do not drop all members, then an error is issued by the system.

Examples

The following example shows the `DROP ALERTHISTORY` command.

Example 7-109   Dropping a Cell Alert History

```
CellCLI> DROP ALERTHISTORY 1, 2_1, 2_2
```

7.7.6.2 DROP CELL

Purpose

The `DROP CELL` command resets a cell to its original state.

Syntax

```
DROP CELL [ERASE = value] [FORCE]
```

Usage Notes

- This command is run from within the cell.
- All cell disks, grid disks, and thresholds are dropped. The interdatabase plan is reset to its default state. All cell attributes are set to default values.
- The `FORCE` option is required if the grid disks are configured on any cell disks when `DROP CELL` is issued. Otherwise, an error is reported.
- Flash cache compression must be disabled before securely erasing a drive.
- The `ERASE` option erases the content on the disk by overwriting the content. The values are as follows:
  - `1pass`: One pass, and the content is overwritten with zeros. This value is not available for flash drives.
  - `3pass`: Three passes, and the content is overwritten with set data patterns. This option follows the recommendations of NNSA. This value is not available for flash drives.
  - `7pass`: Seven passes, and the disk is overwritten with set data patterns. This option follows the recommendations from DOD.
- When dropping all cells using the `1pass` or `3pass` option, it necessary to drop the flash disks first using the `7pass` option, and then drop the cells. The following is an example of the commands:

  ```
  CellCLI> DROP CELLDISK ALL FLASHDISK ERASE=7pass
  CellCLI> DROP CELL ERASE=1pass
  ```

- As of Oracle Exadata System Software release 19.1.0, if you specify to erase hard disks or flash disks using `1pass`, `3pass`, or `7pass` method on Oracle Exadata Database Machine X5 or later, Oracle Exadata System Software automatically invokes Secure Eraser to erase the disks. Secure Eraser determines whether or not the disks can be erased using the better and faster cryptographic erasure method. If some of the disks are eligible, then the cryptographic erasure method is used to erase those disks, and the originally requested method (1/3/7 pass) is used on the other disks. This feature is not used on system disks.

See Table 7-3 for a list of the erasure methods available for each type of device.

The following table shows approximate time needed to securely erase a drive using the supported algorithms. When multiple grid disks or cell disks are dropped with the ERASE option, the command runs in parallel on all disks and flash drives. However, the recommended method of erasing data from a cell is to use Secure Erase. See Securely Erasing Database Servers and Storage Servers in *Oracle Exadata Database Machine Security Guide*.

Table 7-2 Estimated Erasure Times for Disks by Erasure Method

Type of Drive	1pass	3pass	7pass	Cryptographic
600 GB hard drive	1 hour	3 hours	7 hours	not available
1.2 TB hard drive	1.67 hours	5 hours	11.67 hours	not available
2 TB hard drive	5 hours	15 hours	35 hours	not available
3 TB hard drive	7 hours	21 hours	49 hours	not available
4 TB hard drive	8 hours	24 hours	56 hours	not available
8 TB hard drive	13.17 hours	39.5 hours	92.17 hours	1 min
10 TB hard drive	14 hours	42 hours	98 hours	1 min
14 TB hard drive	18 hours	54 hours	126 hours	1 min
22.875 GB flash drive	not available	not available	21 minutes	not available
93 GB flash drive	not available	not available	32 minutes	not available
186 GB flash drive	not available	not available	36 minutes	not available
1.6 TB flash drive	not available	not available	5.5 hours	1 min
3.2 TB flash drive	not available	not available	8 hours	1 min
4 GB Internal USB	not available	30 minutes	not available	not available
8 GB Internal USB	not available	1 hour	not available	not available
150 GB M.2 device	not available	not available	not available	1 minute
ILOM	not available	not available	not available	1 minute

Example 7-110 Dropping a Cell

```
CellCLI> DROP CELL FORCE
```

Related Topics

- ALTER CELL

7.7.6.3 DROP CELLDISK

Purpose

The DROP CELLDISK command removes all or the named cell disks from the cell.

This command is necessary if a cell disk fails, or it is replaced by a newer model.

Before dropping the cell disk, you should drop its grid disks and the corresponding Oracle ASM disks from the disk groups. The Oracle ASM disks should be dropped before dropping the grid disks.

Syntax

```
DROP CELLDISK { ALL [FLASHDISK | HARDDISK] | cdisk_name [, cdisk_name]... }
              [ERASE = value [NOWAIT]] [FORCE]
```

Usage Notes

- If individual cell disks are specified, then the named cell disks (*cdisk_name*) are dropped.
- If the LUN associated with the CELLDISK is flagged as automatically created, then that LUN is deleted along with the cell disk.
- If the ALL option is specified, then all the cell disks on the cell are removed.
- The FLASHDISK option limits the DROP CELLDISK command to cell disks that are flash disks.
- The HARDDISK option limits the DROP CELLDISK command to cell disks that are hard disks.
- If grid disks are configured on the cell disk when DROP CELLDISK is issued, then the FORCE option must be used or an error is reported. The FORCE option causes any grid disks to be dropped first, and then the cell disk is dropped.
- If the specified cell disk includes flash cache, and that flash cache is in writeback mode, then the cell disk cannot be dropped.
- As of Oracle Exadata System Software release 19.1.0, if you specify to erase hard disks or flash disks using 1pass, 3pass, or 7pass method on Oracle Exadata Database Machine X5 or later, Oracle Exadata System Software automatically invokes Secure Eraser to erase the disks. Secure Eraser determines whether or not the disks can be erased using the better and faster cryptographic erasure method. If some of the disks are eligible, then the cryptographic erasure method is used to erase those disks, and the originally requested method (1/3/7 pass) is used on the other disks. This feature is not used on system disks.
- The ERASE option erases the content on the disk by overwriting the content. The values are as follows:
  - 1pass: One pass, and the content is overwritten with zeros. This option is not applicable for flash drives.
  - 3pass: Three passes, and the content is overwritten with set data patterns. This option follows the recommendations from the National Nuclear Security Administration (NNSA). This option is not applicable for flash drives. This value is not available for flash drives.

  - `7pass`: Seven passes, and the disk is overwritten with set data patterns. This option follows the recommendations from the United States Department of Defense (DOD).

  See Table 7-2 for the approximate erasure times for each disk and erasure method.

- Use the `NOWAIT` option with the `ERASE` option to run the command asynchronously.
- When dropping all cell disks using the `1pass` or `3pass` option, you must drop the flash disks first using the `7pass` option, and then drop the cell disks, for example:

```
CellCLI> DROP CELLDISK ALL FLASHDISK ERASE=7pass
CellCLI> DROP CELLDISK ALL ERASE=1pass
```

The following table gives a summary of the secure erasure methods used for each device type. Hard drives, flash devices, and internal USBs are securely erased in parallel: the time required to erase one device is the same as that required for erasing multiple devices of the same kind.

Table 7-3 Methods Used to Securely Erase Various Devices

Component	Make or Model	Erasure Method
Hard drive	• 8 TB hard drives on Oracle Exadata Database Machine X5 • All hard drives on Oracle Exadata Database Machine X6 or later	Cryptographic erase
Hard drive	All other hard drives	1/3/7-Pass erase
Flash device	Flash devices on Oracle Exadata Database Machine X5 or later	Crypto erase
Flash device	All other flash devices	7-pass erase
M.2 device	Oracle Exadata Database Machine X7-2 or later	Crypto erase

Example 7-111 Examples of Dropping a Cell Disk

```
CellCLI> DROP CELLDISK CD_03_cell01

CellCLI> DROP CELLDISK CD_02_cell06 FORCE

CellCLI> DROP CELLDISK ALL

CellCLI> DROP CELLDISK CD_02_cell09 ERASE=1pass NOWAIT
CellDisk CD_02_cell09 erase is in progress
```

Related Topics

- DROP CELL
- Securely Erasing Database Servers and Storage Servers

7.7.6.4 DROP FLASHCACHE

Purpose

The `DROP FLASHCACHE` command removes Exadata Smart Flash Cache from a cell.

Syntax

```
DROP FLASHCACHE
```

Usage Notes

Before dropping flash cache, the data not synchronized with the grid disk (dirty data) must be flushed from flash cache to the grid disks. Not flushing dirty data may cause data loss.

Examples

The following example shows how to remove Exadata Smart Flash Cache from a cell.

Example 7-112 Removing Exadata Smart Flash Cache

```
CellCLI> DROP FLASHCACHE
```

Related Topics

- ALTER FLASHCACHE

7.7.6.5 DROP FLASHLOG

Purpose

The `DROP FLASHLOG` command removes Oracle Exadata Smart Flash Log from a cell.

Syntax

```
DROP FLASHLOG [FORCE]
```

Usage Notes

The `DROP FLASHLOG` command can be run at runtime, but the command does not complete until all redo data on the flash disk is written to hard disk.

If `FORCE` is not specified, then the `DROP FLASHLOG` command fails if there is any saved redo. If `FORCE` is specified, then all saved redo is purged, and Oracle Exadata Smart Flash Log is removed.

> **Caution:**
>
> If `DROP FLASHLOG` fails due to the existence of saved redo, then do not use the `FORCE` option unless you are sure that all saved redo is no longer needed for any databases to perform recovery. Contact Oracle Support Services for additional information.

Examples

The following example shows how to remove Exadata Smart Flash Cache from a cell.

Example 7-113   Removing Oracle Exadata Smart Flash Log from a Cell

```
CellCLI> DROP FLASHLOG

CellCLI> DROP FLASHLOG FORCE
```

7.7.6.6 DROP GRIDDISK

Purpose

The `DROP GRIDDISK` command removes the named grid disks from the cell or removes all the grid disks specified by the `ALL PREFIX` option.

> **Caution:**
>
> Before dropping a grid disk that belongs to an Oracle ASM disk group, ensure that the corresponding disk was dropped from the Oracle ASM disk group.

Syntax

```
DROP GRIDDISK {ALL [FLASHDISK | HARDDISK ] PREFIX=gdisk_name_prefix , |
gdisk_name
              [, gdisk_name]... } [ERASE = value [NOWAIT]]  [FORCE]
```

Usage Notes

- If the *gdisk_name* is entered, then the name identifies the individual grid disk to be removed. Multiple names can be entered.
- If the `ALL PREFIX` option is entered, then the *gdisk_name_prefix* option specifies the prefix assigned to the names of the grid disks to be dropped. The `PREFIX` option is required when `ALL` is used.

  > **Note:**
  >
  > A comma must be entered after the grid disk prefix when it is followed by the `ERASE` option.

- The `FLASHDISK` option limits the `DROP GRIDDISK` command to grid disks that are flash disks.
- The `HARDDISK` option limits the `DROP GRIDDISK` command to grid disks that are hard disks.
- If any of the grid disks are in use when `DROP GRIDDISK` is issued, then an error is reported. You can use `ALTER GRIDDISK` with the `INACTIVE` option to deactivate a

grid disk before dropping the grid disk. This action ensures that the grid disk is not in use.

- The FORCE option can be used to force the drop of a grid disk that is in use.
- If the grid disk being dropped was created on a cell disk of type FLASHDISK, then it does not re-create that area or any part of that grid disk or cell disk as FLASHCACHE automatically. Use the CREATE FLASHCACHE command to reuse any part of the dropped area for FLASHCACHE.
- The ERASE option erases the content on the disk by overwriting the content. The values are as follows:
  - 1pass: One pass, and the content is overwritten with zeros. This option is not applicable for flash drives. This value is not available for flash drives.
  - 3pass: Three passes, and the content is overwritten with set data patterns. This option follows the recommendations from NNSA. This option is not applicable for flash drives. This value is not available for flash drives.
  - 7pass: Seven passes, and the disk is overwritten with set data patterns. This option follows the recommendations from DOD.
- When dropping all grid disks using the 1pass or 3pass option, it necessary to drop the flash disks first using the 7pass option, and then drop the grid disks. The following is an example of the commands:

  ```
  CellCLI> DROP GRIDDISK ALL FLASHDISK PREFIX=data, ERASE=7pass
  CellCLI> DROP GRIDDISK ALL PREFIX=data, ERASE=1pass
  ```

- Use the NOWAIT option with the ERASE option to run the command asynchronously.

Example 7-114   Examples of Dropping a Grid Disk

```
CellCLI> ALTER GRIDDISK data01_CD_03_cell01 INACTIVE

CellCLI> DROP GRIDDISK data01_CD_03_cell01

CellCLI> DROP GRIDDISK ALL PREFIX=data01

CellCLI> DROP GRIDDISK data02_CD_04_cell01 FORCE

CellCLI> DROP GRIDDISK data02_CD_04_cell01 ERASE=1pass
GridDisk data02_CD_04_cell01 successfully dropped

CellCLI> DROP GRIDDISK ALL FLASHDISK PREFIX=DATA, ERASE=7pass
CellCLI> DROP GRIDDISK ALL PREFIX=DATA, ERASE=3pass
```

Related Topics

- DROP CELL
- Dropping a Disk from an Oracle ASM Disk Group
  You can drop a grid disk from a disk group.
- ALTER GRIDDISK

7.7.6.7 DROP PMEMCACHE

Purpose

The DROP PMEMCACHE command removes the PMEM Cache from a cell.

Syntax

```
DROP PMEMCACHE
```

Usage Notes

If the PMEM Cache is in write-back mode, then before dropping the PMEM cache, any data not synchronized with the grid disk (dirty data) must be flushed from PMEM cache to the disks. Failure to flush dirty data can cause data loss.

Example 7-115 Removing PMEM Cache from a Storage Server

```
DROP PMEMCACHE
```

7.7.6.8 DROP PMEMLOG

Purpose

The DROP PMEMLOG command removes PMEMLOG from a cell.

Syntax

```
DROP PMEMLOG [FORCE]
```

Usage Notes

You can use the DROP PMEMLOG command at run-time, but the command does not complete until all redo data on PMEM is flushed to disk.

If FORCE is not specified, then the DROP PMEMLOG command fails if there is any saved redo. If FORCE is specified, then all saved redo is purged, and PMEMLOG is removed.

> **Caution:**
>
> If you are unable to drop the PMEMLOG due to existing redo information in the log, then retry the command first, before using the FORCE option. The FORCE option should only be used in extreme circumstances because it may cause redo log copies to be out of sync, possibly resulting in database redo data corruption. Contact Oracle Support Services for additional information.

Example 7-116 Removing PMEMLOG from a Cell

This example shows how to remove PMEMLOG from a storage server.

```
CellCLI> DROP PMEMLOG
```

7.7.6.9 DROP QUARANTINE

Purpose

The `DROP QUARANTINE` command manually drops a quarantine.

Syntax

```
DROP QUARANTINE { ALL | quarantine1 [, quarantine2]... }
```

Usage Notes

In general, a quarantine can be removed if the quarantined entity is not expected to cause more problem to CELLSRV. For example, cell offload for problem SQL statements is disabled, or an Oracle Database patch is applied. Refer to the alert message for the quarantine for more details.

When a cell is patched, all quarantines are automatically dropped. It is not necessary to drop them manually.

Examples

The following example shows the `DROP QUARANTINE` command.

Example 7-117 Dropping Quarantines

```
CellCLI> DROP QUARANTINE 1
```

7.7.6.10 DROP ROLE

Purpose

The `DROP ROLE` command removes user roles from the cell.

Syntax

```
DROP ROLE  { ALL | role_name1 [, role_name2, ...]} [FORCE]
```

Usage Notes

The `FORCE` option drops the role when the role has been granted to a user.

Examples

The following example shows how to drop a role.

Example 7-118 Dropping a Role

```
CellCLI>DROP ROLE gd_monitor
```

7.7.6.11 DROP SOFTWAREHISTORY

Purpose

The `DROP SOFTWAREHISTORY` command removes all history or individual update history.

Syntax

```
DROP SOFTWAREHISTORY { ALL | 'update_name[,update_name...]'}
```

Example 7-119   Dropping the History of Scheduled Software Updates

```
CellCLI> DROP SOFTWAREHISTORY '12.2.1.2.0.170509,12.2.1.2.0.17052'
```

```
CellCLI> DROP SOFTWAREHISTORY ALL
```

7.7.6.12 DROP THRESHOLD

Purpose

The `DROP THRESHOLD` command removes all or the specified thresholds from the cell.

Syntax

```
DROP THRESHOLD { ALL |threshold_name [, threshold_name ...] }
```

Examples

The following example shows the `DROP THRESHOLD` command.

Example 7-120   Dropping Thresholds

```
CellCLI> DROP THRESHOLD ct_io_wt_rq.interactive
```

```
CellCLI> DROP THRESHOLD ALL
```

Related Topics

- DESCRIBE THRESHOLD

7.7.6.13 DROP USER

Purpose

The `DROP USER` command removes a user from a cell.

Syntax

```
DROP USER { ALL | user1 [, user2]... }
```

Examples

The following example shows how to drop a user.

Example 7-121 Dropping a User

```
CellCLI>DROP USER agarcia
```

7.7.7 EXIT

Purpose

The `EXIT` command exits from the CellCLI utility, and returns control to the operating system prompt.

Syntax

```
EXIT
```

`EXIT` has the same functionality as the `QUIT` command.

7.7.8 EXPORT CELLDISK

Purpose

The `EXPORT CELLDISK` command prepares all cell disks or a specified cell disk before moving (importing) the cell disk to a different cell.

Syntax

```
EXPORT CELLDISK { ALL | cdisk_name }
```

Usage Notes

To move a cell disk from one cell to another, use the `EXPORT CELLDISK` and `IMPORT CELLDISK` commands. Usually, all disks are moved to a new cell if the current cell is failing. First, export the cell disk on one cell. Then, import the exported cell disk using the CellCLI utility on the cell where you moved the physical drive that contains the cell disk.

When the `EXPORT CELLDISK` command is run:

- `ALL` exports all cell disks on the cells that have `normal` status.
- If the LUN associated with the cell disk is flagged as automatically-created, then that LUN is deleted as part of the export.
- A successfully exported cell disk has the `status` attribute set to `ImportRequired`, and the exported cell disk is displayed in the output of the `LIST CELLDISK` command.
- The following apply when a cell disk is exported (`status='ImportRequired'`) before it is imported:
  - You can change the `name` and `comment` attributes.
  - You can drop the cell disk.
  - You cannot create a new grid disk on the cell disk.
- When a disk is exported, any writes from the disk controller cache to the disk are cleared, and the disk is flagged to indicate that the disk was exported. The grid

disks on the disk are no longer visible to Oracle ASM. Any I/Os to the grid disks get errors.

Before exporting a cell disk, the data not synchronized with the grid disk (dirty data) must be flushed from flash cache to the grid disks. Not flushing dirty data may cause data loss.

Examples

The following example shows the `EXPORT CELLDISK` command.

Example 7-122 Exporting a Cell Disk

```
CellCLI> EXPORT CELLDISK CD_3_cell01

CellCLI> EXPORT CELLDISK ALL
```

Related Topics

- ALTER CELLDISK

7.7.9 GRANT

Purpose

The `GRANT` command sets attributes for privileges and roles.

Syntax

```
GRANT object_type [name] TO sub_object_type [sub_object_name]
```

Usage Notes

- *object_type* can be as follows:
  - `PRIVILEGE`
  - `ROLE`
- The following values can be used for `PRIVILEGE` object type:
  - *name* is in the following format:

    ```
    { ALL ACTIONS | action } ON { ALL OBJECTS | object }               \
    [{ ALL ATTRIBUTES | ATTRIBUTES attribute1 [, attribute2, ...] }]   \
    [{ WITH ALL OPTIONS | WITH OPTIONS option1 [, option2, ...] }]
    ```

  - The *sub_object_type* must be `ROLE`.
  - The *sub_object_name* is the name of the role.
- The following can be used for the `ROLE` object type:
  - *name* is the role name.
  - The *sub_object_type* must be `USER`.
  - The *sub_object_name* is the name of the user.
- GRANT PRIVILEGE
- GRANT ROLE

7.7.9.1 GRANT PRIVILEGE

Purpose

The GRANT PRIVILEGE command sets the access privileges for a role.

Syntax

```
GRANT PRIVILEGE { ALL ACTIONS | action } ON { ALL OBJECTS | object }   \
{ ALL ATTRIBUTES | ATTRIBUTES attribute1 [, attribute2, ...] }         \
{ WITH ALL OPTIONS | WITH OPTIONS option1 [, option2, ...] }           \
TO ROLE { ALL | role1 [, role2, ...] }
```

Usage Notes

- *action* is the command. Examples: ALTER, CREATE, DESCRIBE, DROP, EXPORT, IMPORT, LIST.

  Notes:

  - The GRANT and REVOKE commands cannot be granted.
  - CREATE USER and DROP USER cannot be granted.
  - CREATE ROLE and DROP ROLE cannot be granted.
- *object* is object type for the action. It can be any CellCLI object. Examples: CELL, THRESHOLD, PHYSICALDISK, ALERTHISTORY, ROLE.
- *attribute* are the attributes for the object. To get a list of attributes for an object, run the LIST *object_type* command.
- *option* are the options for the object. Examples: DETAIL, LIMIT, ORDER BY, WHERE.
- *role* is the name of the role to grant privileges.
- The ALL ACTIONS argument grants privileges for all actions.
- The ALL OBJECTS argument grants privileges for all objects.
- The ALL ATTRIBUTES argument grants privileges for all attributes.
- The WITH ALL OPTIONS argument grants privileges for all options.
- Specifying attributes and WITH OPTIONS is optional. If they are not specified, then all attributes and options are granted with the privilege.

Examples

Example 7-123 Granting Privileges to a Role

This example shows how to grant privileges to a role.

```
CellCLI> GRANT PRIVILEGE list on alerthistory ATTRIBUTES
alertAction,alertMessage  \
        WITH OPTIONS detail TO ROLE cellmonitor
```

Example 7-124 Granting All Attributes and Options to a Role

This example shows how to grant all attributes and options for a specified action and object to a role.

```
CellCLI> GRANT PRIVILEGE { ALL ACTIONS | action } ON { ALL OBJECTS |
object } to ROLE role1
```

Example 7-125 Granting All Options with Specified Action, Object and Attributes

This example shows how to grant all options with a specified action, object and attributes to a role.

```
CellCLI> GRANT PRIVILEGE { ALL ACTIONS | action } ON { ALL OBJECTS |
object }  \
ATTRIBUTES attribute1 [, attribute2, ...] to ROLE role1
```

Example 7-126 Granting All Attributes with Specified Action, Object and Options

This example shows how to grant all attributes with a specified action, object, and options to a role.

```
CellCLI> GRANT PRIVILEGE { ALL ACTIONS | action } ON { ALL OBJECTS |
object }   \
WITH OPTIONS option1 [, option, ...] to ROLE role1
```

7.7.9.2 GRANT ROLE

Purpose

The `GRANT ROLE` command sets the role for a user.

Syntax

```
GRANT ROLE { ALL | role1 [, role2, ...] } TO USER { ALL | user1 [, user2...] }
```

Usage Notes

- *role* is the name of the role.
- The `ALL` argument grants all roles to the user.
- The `TO USER ALL` argument grants the role to all users.

Examples

Example 7-127 Granting a Role to a User

This example shows how to grant a role to a user.

```
CellCLI> GRANT ROLE gd_monitor TO USER agarcia
```

7.7.10 HELP

Purpose

The HELP command displays syntax and usage descriptions for all CellCLI commands.

Syntax

```
HELP [help_topic]
```

If no topic argument is provided, HELP displays the name of all available topics. If a topic is specified, then detailed help text is displayed for that topic.

The following example shows examples of the HELP command.

Example 7-128 Display Help Text with the HELP Command

```
CellCLI> HELP
CellCLI> HELP ALTER
CellCLI> HELP ALTER CELL
```

7.7.11 IMPORT CELLDISK

Purpose

The IMPORT CELLDISK command reinstates all exported cell disks or an exported cell disk on a cell where you moved the physical drives that contain the cell disks.

The cell disk is typically imported to a different cell than the one from which the cell disk was exported. For example, the physical drive that contains the exported cell disk was moved to a different cell.

When you move a disk with cell disks and grid disks on it from one machine to another, be careful to ensure that the data on it is rebalanced, as per the ASM failure groups. If all disks from one cell are moved to another cell, then there is no need to perform a ASM rebalance, since the entire failure group is moved.

Syntax

```
IMPORT CELLDISK  { ALL  |  cdisk_name  LUN=lun_id  | cdisk_name |
LUN=lun_id }
    [, comment=comment_text] [FORCE]
```

Usage Notes

To move a cell disk from one cell to another, use the EXPORT CELLDISK and IMPORT CELLDISK commands. Usually, all disks are moved to a new cell if the current cell is failing. First, export the cell disk on one cell. Then, import the exported cell disk using the CellCLI utility on the cell where you moved the physical drive that contains the cell disk.

When the IMPORT CELLDISK command is run:

- Either ALL, the cell disk name, the LUN ID, or the cell disk name and LUN ID must be specified.

  - ALL imports cell disks that have ImportRequired status.
  - If the cell disk name is provided and the LUN ID is not provided, then you can import a cell disk by the specified name in cases where Management Server recognizes this cell disk. A recognized cell disk is displayed in the output of LIST CELLDISK with status equal to ImportRequired.
  - If the LUN ID is provided and the cell disk name is not provided, then the LUN is scanned, and the cell disk is imported. This variation of the command can be used to import a newly-inserted cell disk that was not recognized by Management Server and Cell Server.
  - If the LUN ID and cell disk name are both provided, then the LUN ID is used to import the cell disk, and the name is used to rename the imported cell disk.

- A new value can be entered for the comment attribute to update the existing cell disk comment.
- The cell disk name is verified to ensure that the name is unique within the cell. Cell disks can be renamed before import to ensure uniqueness.
- The grid disk names within a cell must be unique. If a physical disk is moved from one cell (cell_A) to another cell (cell_B) using the EXPORT and IMPORT commands, then there is a chance that the target cell (cell_B) could have two grid disks with identical names. In this case, the cell software automatically resolves the naming conflict by adding a temporary suffix (_duplicate_name, _duplicate_name2, _duplicate_name3, and so on) to the name of one of the grid disks. This additional suffix enables you to refer to a grid disk unambiguously in the CellCLI commands.

  It is recommended that you rename a duplicate grid disk on a cell (cell_B) with a new permanent unique name using the following command:

  ```
  ALTER GRIDDISK gdname_duplicate_name NAME=new_unique_name
  ```

  If you return the physical disk to the original cell (cell_A) or move the disk to another cell rather than renaming the disk, then the grid disk displays its original name.

- The LIST CELLDISK command shows which cell disks need to be imported. The command displays output similar to the following:

  ```
  CellCLI> list celldisk
           CD_01_cell00     normal
           CD_01_cell01     normal
           CD_01_cell02     importRequired
           CD_01_cell03     importForceRequired
           CD_01_cell04     importRequired
           CD_01_flash0     normal
           CD_01_flash1     normal
           CD_01_flash2     normal
           CD_01_log00      normal
           CD_01_log01      normal
  ```

- If the cell disk was not successfully exported and moved between cells, then the FORCE option must be specified with IMPORT or an error occurs.Oracle recommends contacting Oracle Support Services before using the FORCE option.

- The IMPORT command checks the disk to determine if it was exported. If it was exported, then the IMPORT command makes the grid disk visible to Oracle ASM. If the disk was not exported, then the FORCE option should be used with the IMPORT command to reconstruct the grid disks on the disk, and make them visible to Oracle ASM.

Example 7-129 Importing a Cell Disk

This example shows the IMPORT CELLDISK command. The LUN ID is provided with the IMPORT command to identify the cell disk, and the cell disk name is used to rename the cell disk on the cell where it was imported.

```
CellCLI> IMPORT CELLDISK CD_7_cell04 lun=3

CellCLI> IMPORT CELLDISK ALL
```

7.7.12 LIST

Purpose

The LIST command displays attributes for Oracle Exadata System Software objects. Objects displayed are identified by name or by filters. The attributes displayed for each object are determined by the specified attribute list.

Syntax

```
LIST object_type  [ name | attribute_filters] [attribute_list] [DETAIL]  \
[ORDER BY attribute [ASC| DESC][, attribute [ASC| DESC], ...] \
[LIMIT integer]
```

Usage Notes

- Using LIST with only an *object_type* (without the DETAIL option or an attribute list) displays the names of the existing objects of this type and a default list of attributes.
  - For an object type that has a status attribute, the object name and the status are displayed.
  - For the METRICHISTORY object type, the collection time, the object name, and value are displayed.
  - For the PHYSICALDISK and LUN object types, the ID attribute is displayed.
  - For the ALERTHISTORY object type, the time and alert message are displayed.
  - For the KEY object type, the key value is displayed.
- The attributes displayed for each object are determined by the specified attribute list. Attribute values that are strings with embedded blank spaces or tabs must be enclosed in quotation marks.
- Attribute filters determine the specific objects that are displayed. Because of the amount of metrics, you should use filters when using the LIST METRICCURRENT or LIST METRICHISTORY commands to narrow the output of the command.

- In the default format without the DETAIL option, each object is displayed on a separate line, with successive attribute values separated by tabs in the order of the specified list of attributes.
- In the DETAIL format, each attribute of a specific object is displayed on a separate line, with an attribute name followed by its value. If no attribute list is provided, then all attributes that have values are displayed. Blank lines separate each object in the display. DETAIL is similar to the ATTRIBUTES ALL option, only the format is different.
- Attributes that are not set are not listed with the DETAIL option. However, attributes that are set to an empty value are listed with the DETAIL option.
- The ORDER BY option orders attributes in ascending or descending order. The default is ASC.
- The LIMIT option sets a limit on the number of displayed attributes. The maximum value is 100 when LIMIT is used with the ORDER BY option.

- LIST ACTIVEREQUEST
- LIST ALERTDEFINITION
- LIST ALERTHISTORY
- LIST CELL
- LIST CELLDISK
- LIST DATABASE
- LIST DIAGPACK
- LIST DISKMAP
- LIST FLASHCACHE
- LIST FLASHCACHECONTENT
- LIST FLASHLOG
- LIST GRIDDISK
- LIST IBPORT
- LIST IORMPLAN
- LIST IORMPROFILE
- LIST KEY
- LIST LUN
- LIST METRICCURRENT
- LIST METRICDEFINITION
- LIST METRICHISTORY
- LIST OFFLOADGROUP
- LIST PHYSICALDISK
- LIST PLUGGABLEDATABASE
- LIST PMEMCACHE
- LIST PMEMLOG
  The LIST PMEMLOG command displays the specified attributes for the PMEMLOG.

- LIST QUARANTINE
- LIST ROLE
- LIST SOFTWAREHISTORY
- LIST SOFTWAREUPDATE
- LIST THRESHOLD
- LIST USER

Related Topics

- Attribute Lists in LIST Command
  You can specify which attributes to display for the `LIST` command with the optional `ATTRIBUTES` clause.
- Attribute Filters in LIST Commands
  You can specify which objects to display for the `LIST` command with the optional `WHERE` clause.

7.7.12.1 LIST ACTIVEREQUEST

Purpose

The `LIST ACTIVEREQUEST` command displays specified attributes for the outstanding active requests for the cell.

Syntax

```
LIST ACTIVEREQUEST  [ name |  attribute_filters ]  [attribute_list] [DETAIL]
```

Usage Notes

The list of attributes that can be displayed is shown in the following example.

Example 7-130   Listing ACTIVEREQUEST Attributes

This example shows the `LIST` command with the `ACTIVEREQUEST` object.

```
CellCLI> LIST ACTIVEREQUEST 5 DETAIL

         name:                    5
         ID:                      5
         ParentID:                5
         dbName:                  "test DB"
         InstNum:                 5
         ConsumerGrp:             "test group"
         SessID:                  5
         SerialNum:               5
         AsmFileNum:              5
         AsmDGNum:                5
         FileIncNum:              5
         ObjNum:                  5
         TsNum:                   5
         SqlID:                   5
         FileType:                "Oracle db data file"
         IoReason:                "test io"
         IoType:                  "test read"
         State:                 "Queued for Test"
         GdList:                gdName=testGrid,gdOffset=0,gdSize=524288000
```

7.7.12.2 LIST ALERTDEFINITION

Purpose

The `LIST ALERTDEFINITION` command displays all available sources of the alerts on the cell.

Syntax

```
LIST ALERTDEFINITION [ name |  attribute_filters ]  [attribute_list]
[DETAIL]
```

Usage Notes

The list of attributes that can be displayed is shown in the following example.

Example 7-131 Listing ALERTDEFINITION Attributes

This example shows the `LIST` command with the `ALERTDEFINITION` object.

```
CellCLI> LIST ALERTDEFINITION StatefulAlert_CG_IO_RQ_LG DETAIL

         name:                    StatefulAlert_CG_IO_RQ_LG
         alertShortName:          CG_IO_RQ_LG
         alertSource:             Metric
         alertType:               Stateful
         description:             "Threshold Alert"
         metricName:              CG_IO_RQ_LG
```

7.7.12.3 LIST ALERTHISTORY

Purpose

The `LIST ALERTHISTORY` command displays all alerts that occurred on the cell.

Syntax

```
LIST ALERTHISTORY [ name |  attribute_filters ]  [attribute_list]  [DETAIL]
```

Usage Notes

The list of attributes that can be displayed is shown in Example 7-82.

A `WHERE` clause can include the `ageInMInutes` attribute to specify the list is limited to those alerts which have the specified age. For example, the following command would show the alerts created in the previous 15 minutes:

```
CellCLI> LIST ALERTHISTORY WHERE ageInMinutes < 15
```

The `alerthistory` attribute can be used to check for stateful and stateless alerts.

Examples

Example 7-132 shows the LIST command with the ALERTHISTORY object.

Example 7-133 shows open stateful and stateless alerts.

Example 7-134 shows only open stateful alerts. The closed stateful alerts are filtered out.

Example 7-135 shows alerts that have not been cleared.

Example 7-132   Listing ALERTHISTORY Attributes

```
CellCLI>  LIST ALERTHISTORY 1671443714 DETAIL
          name:                1671443714
          alertSequenceID:     1671443714
          sequenceBeginTime:   1179185707672
          beginTime:           "Sat May 18 10:14:16 PDT 2009"
          endTime:             "Sat May 25 10:14:16 PDT 2009"
          severity:            critical
          alertMessage:        "Errors in file svtrc_2840_10.trc
(incident=13):"
          alertShortName:      ADR
          alertNotified:       0
          examinedBy:          johndoe
          alertType:           stateless

CellCLI> LIST ALERTHISTORY WHERE begintime > 'Jun 1, 2009 11:37:00 AM PDT'

         39      2009-10-02T12:26:53-07:00       "ORA-07445: exception
                 encountered: core dump [__kerne l_vsyscall()+5] [6]
                 [0x408C] [] [] []"
         40      2009-10-06T23:28:06-07:00       "RS-7445
[unknown_function]
                 [signum: 6] [] [] [] [] [ ] []"
         41      2009-10-07T00:50:42-07:00       "RS-7445 [Serv MS not
responding]
                 [It will be restart ed] [] [] [] [] [] []"
         42      2009-10-07T02:21:19-07:00       "RS-7445
[unknown_function]
                 [signum: 6] [] [] [] [] [ ] []"

CellCLI> LIST ALERTHISTORY 7 DETAIL
         name:                   7
         alertMessage:           "Flash cache mode is set to WriteBack
because
                                 there is dirty data in the flash cache."
         alertSequenceID:        7
         alertShortName:         Software
         alertType:              Stateless
         beginTime:              2012-09-10T13:22:38-07:00
         examinedBy:
         metricObjectName:       FlashCache
         notificationState:      0
         sequenceBeginTime:      2012-09-10T13:22:38-07:00
         severity:                info
         alertAction:             "If the newly-assigned mode for flash
cache is
                                  not wanted, then change it using the
ALTER CELL
                                  command as described in the Oracle
Exadata user's
                                  guide."
```

Example 7-133 Listing Open Stateful and Stateless Alerts

```
CellCLI> LIST ALERTHISTORY ATTRIBUTES alertsequenceid,name,alerttype    \
         WHERE endtime=null

         1       1       Stateless
         3       3       Stateless
         11      11_1    Stateful
```

Example 7-134 Listing Open Stateful Alerts

```
CellCLI> LIST ALERTHISTORY WHERE endtime=null AND alerttype=stateful
```

Example 7-135 Listing Alerts That Have Not Cleared

```
CellCLI> LIST ALERTHISTORY WHERE endtime=null

1       2014-11-11T11:08:15-08:00  info      "Factory defaults restored for
 Adapter 0"
3       2014-11-11T11:27:06-08:00  critical   "RS-700 [No IP found in
Exadata
 config file] [Check cellinit.ora]
                                               [] [] [] [] [] [] [] [] []
[]"
11_1    2014-12-19T12:01:06-08:00  critical    "The HDD disk controller
battery
 has failed. All disk drives have been placed in WriteThrough caching
mode. Disk
 write performance may be reduced. The flash drives are not affected.
Battery
 Serial Number : 1142  Battery Type          : ibbu08  Battery
Temperature   : 39
 C  Full Charge Capacity  : 773 mAh  Relative Charge       : 83%  Ambient
 Temperature   : 32 C"
```

7.7.12.4 LIST CELL

Purpose

The `LIST CELL` command displays specified attributes of the cell.

Syntax

```
LIST CELL [ATTRIBUTES attribute_list] [DETAIL]
```

Usage Notes

- The list of attributes that can be displayed is shown in Example 7-83.
- `LIST CELL` only lists the local cell. Name and filter options on `LIST CELL` are not required.
- To monitor the status of cell components, use the `LIST` command to verify the value of `status`, `fanStatus`, `temperatureStatus`, and `powerStatus`.

Examples

Example 7-136 shows the `LIST` command with the `CELL` object, and the corresponding output.

Example 7-137 shows how to display the status of cell components.

Example 7-138 shows how to display the values of the `snmpSubscriber` attribute.

Example 7-139 shows how to display the value of the `emailFormat` attribute.

Example 7-140 shows how to display the value of the `locateLEDStatus` attribute.

Example 7-141 shows how to display the value of the `doNotServiceLEDStatus` attribute.

Example 7-142 shows how to display the value of the `bbuLearnCycleTime` attribute.

Example 7-143 shows how to display the value of the `rescuePlan` attribute.

Example 7-144 shows how to retrieve the value of the `httpsAccess` attribute.

Example 7-136 Listing Cell Information

```
CellCLI> LIST CELL

         cell01        online
```

Example 7-137 Displaying the Status of Cell Components

```
CellCLI> LIST CELL ATTRIBUTES name, status, location, -
         fanStatus, temperatureStatus, powerStatus

         cell01      online  rack5:shelf1     normal  normal  normal
```

Example 7-138 Displaying the snmpSubscriber Attribute

```
CellCLI> LIST CELL ATTRIBUTES snmpSubscriber

((host=server1.example.com,port=3873,community=public, type=asr))
```

Example 7-139 Displaying E-mail Format

```
CellCLI> LIST CELL ATTRIBUTES emailFormat
         html
```

Example 7-140 Displaying locateLEDStatus

```
CellCLI> LIST CELL ATTRIBUTES locateLEDStatus
         off
```

Example 7-141 Displaying doNotServiceLEDStatus

```
CellCLI> LIST CELL ATTRIBUTES doNotServiceLEDStatus
         on
```

Example 7-142 Listing the bbuLearnCycleTime Attribute

```
CellCLI> LIST CELL ATTRIBUTES bbuLearnCycleTime
```

Example 7-143 Displaying rescuePlan

```
CellCLI> LIST CELL ATTRIBUTES rescuePlan

CREATE ROLE "admin"

GRANT PRIVILEGE all actions ON diagpack all attributes WITH all options TO
ROLE "admin"

CREATE ROLE "diagRole"

GRANT PRIVILEGE download ON diagpack all attributes WITH all options TO
ROLE "diagRole"

GRANT PRIVILEGE create ON diagpack all attributes WITH all options TO ROLE
"diagRole"

GRANT PRIVILEGE list ON diagpack all attributes WITH all options TO ROLE
"diagRole"

ALTER CELL accessLevelPerm="remoteLoginEnabled", diagHistoryDays="7",
metricHistoryDays="7", notificationMethod="mail,snmp",
notificationPolicy="warning,critical,clear",
snmpSubscriber=((host="localhost", port=162, community="public",
type=asr)), bbuLearnCycleTime="2016-10-17T02:00:00-07:00",
bbuLearnSchedule="MONTH 1 DATE 17 HOUR 2 MINUTE 0",
alertSummaryStartTime="2016-09-21T17:00:00-07:00",
alertSummaryInterval=weekly, hardDiskScrubInterval=biweekly,
hardDiskScrubFollowupIntervalInDays="14"

ALTER IORMPLAN objective=basic
```

Example 7-144 Displaying the HTTPs Access Control List

This example shows how to view the HTTPs access control list for the Exadata RESTful service.

```
CellCLI> LIST CELL ATTRIBUTES httpsAccesss
         ALL
```

The value of `ALL` is the default value and allows access to all hosts.

7.7.12.5 LIST CELLDISK

Purpose

The `LIST CELLDISK` command displays attributes for cell disks determined by the specified attributes and filters.

Syntax

```
LIST CELLDISK [ name | attribute_filters ] [attribute_list] [DETAIL]
```

Usage Notes

The list of attributes that can be displayed is shown in Describing the CELLDISK Object.

Examples

The following example shows the `LIST` command with the `CELLDISK` object, and the corresponding output.

Example 7-145 Listing Cell Disk Attributes

```
CellCLI> LIST CELLDISK CD_01_cell05 ATTRIBUTES size

         557.859375G

CellCLI> LIST CELLDISK WHERE status!=normal ATTRIBUTES name

         CD_01_cell03

CellCLI> LIST CELLDISK WHERE DEVICENAME LIKE '/dev/c0d[2-5]' -
         ATTRIBUTES name, size

         CD_01_cell05             557.859375G

CellCLI> LIST CELLDISK CD_01_cell05 DETAIL

         name:                   CD_01_cell05
         comment:
         creationTime:           2018-03-21T13:39:15-04:00
         deviceName:             /dev/sdi
         devicePartition:        /dev/sdi
         diskType:               HardDisk
         errorCount:             0
         freespace:                0
         id:                       00000117-84d2-ed2c-0000-000000000000
         physicalDisk:             K7N5JJ
         size:                     557.859375G
         status:                   normal
```

7.7.12.6 LIST DATABASE

Purpose

Displays the specified attributes for active databases.

Syntax

```
LIST DATABASE [name | attribute_filters] [attribute_list]  [DETAIL]
```

Usage Notes

The filters option is an expression that determines which active databases should be listed by the command.

The list of attributes that can be displayed is shown in Example 7-85.

Examples

The following example shows the `LIST` command with the `DATABASE` object, and the corresponding output.

Example 7-146 Listing Database Attributes

```
CellCLI> LIST DATABASE
         DB01


CellCLI> LIST DATABASE DETAIL
         name:                     DB01
         asmClusterName:           SALESDBS_ASMCLUSTER
         databaseID:               1234567656
         lastRequestTime:          2016-10-27T07:46:36-07:00
         profile:                  GOLD
         flashCacheMin:            4.00390625G
         flashCacheLimit:          4.19921875G
         flashCacheSize:           0
         pmemCacheMin:             2.001953125G
         pmemCacheLimit:           2.099609375G
         pmemCacheSize:            0


CellCLI> LIST DATABASE ATTRIBUTES NAME, PROFILE
         ASM
         TEST50          GOLD
         TEST100         GOLD
         TEST150         SILVER
         TEST20          GOLD
         TEST200         BRONZE
         TEST180          SILVER
         TEST175          SILVER
         TEST225          BRONZE
         TEST230          BRONZE
         TEST300
         TEST280
         TEST245          BRONZE

CellCLI> LIST DATABASE ATTRIBUTES NAME, DATABASEID WHERE PROFILE = 'GOLD'
         TEST50           50
         TEST100          100
         TEST20           20
```

7.7.12.7 LIST DIAGPACK

Purpose

The `LIST DIAGPACK` command lists the diagnostic packages in your system, along with their status.

Syntax

```
LIST DIAGPACK [DETAIL]
```

Usage Notes

The location of the diagnostic packages is $LOG_HOME.

Examples

Example 7-147 Output of the "list diagpack" Command

This example shows the output of the LIST DIAGPACK command.

```
CellCLI> LIST DIAGPACK
scab01cel04_diag_2015_09_30T13_29_06_1.tar.bz2    (7 minutes ago)
scab01cel04_2015_09_30T13_13_00_2_1.tar.bz2    (23 minutes ago for alert: 2_1)
scab01cel04_2015_09_30T13_07_10_1_1.tar.bz2    (28 minutes ago for alert: 1_1)
```

Example 7-148 Output of the "list diagpack" command with the DETAIL option

This example shows the output of the LIST DIAGPACK command with the DETAIL option.

```
CellCLI> LIST DIAGPACK DETAIL
Name:                scab01cel04_diag_2015_09_30T13_29_06_1.tar.bz2
Time:                Wed, 30 Sep 2015 13:29:06 (7 minutes ago)
Type:                Custom package

Name:                scab01cel04_2015_09_30T13_13_00_2_1.tar.bz2
Time:                Wed, 30 Sep 2015 13:13:00 (23 minutes ago)
Alert ID:            2_1
Alert description:   InfiniBand Port HCA-1:2 indicates invalid state.

Name:                scab01cel04_2015_09_30T13_07_10_1_1.tar.bz2
Time:                Wed, 30 Sep 2015 13:07:10 (28 minutes ago)
Alert ID:            1_1
Alert description:   File system "/" is 84% full
```

Related Topics

- CREATE DIAGPACK

7.7.12.8 LIST DISKMAP

Purpose

Displays the specified grid disk attributes for a physical disk.

Syntax

```
LIST DISKMAP
```

Usage Notes

The list of attributes that can be displayed is shown in Example 7-86.

Examples

The following example shows the `LIST` command with the `DISKMAP` object, and the corresponding output.

Example 7-149 Listing Grid Disk Attributes for a Physical Disk

```
CELLCLI> LIST DISKMAP

Name   PhysicalSerial  SlotNumber    Status  PhysicalSize  CellDisk  DevicePartition  GridDisks
27:0   E0XH34          0             normal   559G         CD_00_sgrcel2   /dev/sda3
"DATA_CD_00_sgrcel2, RECO_CD_00_sgrcel2"

27:1   E0XH2S          1             normal   559G         CD_01_sgrcel2   /dev/sdb3
"DATA_CD_01_sgrcel2, RECO_CD_01_sgrcel2"

27:2   E0Z0CS           2            normal   559G         CD_02_sgrcel2   /dev/sdc
"DATA_CD_02_sgrcel2, DBFS_CD_02_sgrcel2, RECO_CD_02_sgrcel2"
.
.
.
```

7.7.12.9 LIST FLASHCACHE

Purpose

The `LIST FLASHCACHE` command displays attributes for the Exadata Smart Flash Cache determined by the specified attributes.

Syntax

```
LIST FLASHCACHE [attribute_list] [DETAIL]
```

Usage Notes

The list of attributes that can be displayed is shown in Example 7-87.

Examples

The following example shows the `LIST` command with the `FLASHCACHE` object, and the corresponding output.

Example 7-150 Listing Exadata Smart Flash Cache Attributes

```
CellCLI> LIST FLASHCACHE

         raw_FLASHCACHE normal

CellCLI> LIST FLASHCACHE DETAIL

         name:                   raw_FLASHCACHE
         cellDisk:               c9FLASH0,FD_FLASH1_raw,FD_FLASH2_raw
         creationTime:           2012-08-04T15:42:42-07:00
         degradedCelldisks:
         effectiveCacheSize:     192M
         id:                     8a0adc84-9088-4c4e-8e1c-b6bcbd5cb1ba
         size:                   192M
         status:                 normal
```

7.7.12.10 LIST FLASHCACHECONTENT

Purpose

The `LIST FLASHCACHECONTENT` command displays attributes for the Exadata Smart Flash Cache entries determined by the specified attributes.

Syntax

```
LIST FLASHCACHECONTENT [attribute_filters] [attribute_list] [DETAIL]
```

Usage Notes

The list of attributes that can be displayed is shown in Example 7-88.

Examples

Example 7-151 Listing Exadata Smart Flash Cache Content Attributes

This example shows the `LIST` command with the `FLASHCACHECONTENT` object, and the corresponding output.

```
  CellCLI> LIST FLASHCACHECONTENT DETAIL

           cachedKeepSize:             8192
           cachedSize:                16384
           dbID:                 3557170052
           dbUniqueName:              ACME1
           hitCount:                      4
           hoursToExpiration:            24
           missCount:
           objectNumber:                267
           tableSpaceNumber:              1

           cachedKeepSize:                0
           cachedSize:                 8192
```

```
         dbID:                4325252357
         dbUniqueName:             MYODB
         hitCount:                     1
         hoursToExpiration:           23
         missCount:                    1
         objectNumber:             65523
         tableSpaceNumber:             4
```

Example 7-152   Listing Exadata Smart Flash Cache Content by Database Object

This example shows a database query for an object in a partitioned table, and then lists the flash cache for the same object. In the example, a partitioned table is created in the database, and then queried for the data object numbers of the partitions. The flash cache on Oracle Exadata Storage Server is then queried.

```
CREATE TABLE parttabl (c1 number) PARTITION BY RANGE(c1)
(
  PARTITION partt1 VALUES LESS THAN (100),
  PARTITION partt2 VALUES LESS THAN (200)
);

SQL> SELECT SUBSTR(OBJECT_NAME, 0 , 10) OBJ_NAME, SUBOBJECT_NAME,
DATA_OBJECT_ID
     FROM user_objects WHERE OBJECT_NAME LIKE ('PARTT%');

OBJ_NAME   SUBOBJECT_NAME                 DATA_OBJECT_ID
---------- ------------------------------ --------------
PARTTABL
PARTTABL   PARTT1                                  63197
PARTTABL   PARTT2                                  63198

CellCLI> LIST FLASHCACHECONTENT WHERE objectNumber=63197 DETAIL
         cachedKeepSize:         0
         cachedSize:             24576
         dbID:                   3722937438
         dbUniqueName:           VIEW6
         hitCount:               0
         missCount:              4
         objectNumber:           63197
         tableSpaceNumber:       0


  CellCLI> LIST FLASHCACHECONTENT WHERE objectNumber=63198 DETAIL
         cachedKeepSize:         0
         cachedSize:             16384
         dbID:                   3722937438
         dbUniqueName:           VIEW6
         hitCount:               0
         missCount:              2
         objectNumber:           63198
         tableSpaceNumber:       0
```

Example 7-153 Listing Exadata Smart Flash Cache Content for an ASMCLUSTER Client

Starting with Oracle Exadata System Software release 19.1.0, the dbUniqueName attribute is qualified with the ASMCLUSTER client name if ASM-scoped security is configured. This example shows the partial output of the LIST FLASHCACHECONTENT command for database instances associated with Oracle ASM clusters.

```
  CellCLI> LIST FLASHCACHECONTENT WHERE dbuniquename LIKE 'ASM.*' DETAIL

         cachedKeepSize:         0
         cachedSize:             65536
         cachedWriteSize:        65536
         columnarCacheSize:      0
         columnarKeepSize:       0
         dbID:                   3334479949
         dbUniqueName:           ASM1.DB1.CDB$ROOT
         hitCount:               0
         missCount:              0
         objectNumber:           75307
         tableSpaceNumber:       1

...

         cachedKeepSize:         0
         cachedSize:             2957312
         cachedWriteSize:        0
         columnarCacheSize:      0
         columnarKeepSize:       0
         dbID:                   1238079488
         dbUniqueName:           ASM1.DB1.PDB1
         hitCount:               4
         missCount:              47
         objectNumber:           4294967294
         tableSpaceNumber:       1

...

         cachedKeepSize:         0
         cachedSize:             17326080
         cachedWriteSize:        0
         columnarCacheSize:      0
         columnarKeepSize:       0
         dbID:                   1757889862
         dbUniqueName:           ASM2.DB2
         hitCount:               9
         missCount:              255
         objectNumber:           4294967294
         tableSpaceNumber:       5
```

7.7.12.11 LIST FLASHLOG

Purpose

The `LIST FLASHLOG` command displays attributes for the Oracle Exadata Smart Flash Log entries determined by the specified attributes.

Syntax

```
LIST FLASHLOG
```

Usage Notes

The list of attributes that can be displayed is shown in Example 7-89.

Examples

The following example shows the `LIST` command with the `FLASHLOG` object, and the corresponding output.

Example 7-154 Listing Oracle Exadata Smart Flash Log Attributes

```
CellCLI> LIST FLASHLOG

         raw_FLASHLOG normal

CellCLI> LIST FLASHLOG DETAIL

         name:                   raw_FLASHLOG
         cellDisk:               c9FLASH0,FD_FLASH1_raw,FD_FLASH2_raw
         creationTime:           2011-01-23T12:34:56-05:00
         degradedCelldisks:
         effectiveSize:          512M
         efficiency:             100.0
         id:                     8a0aadc84-908804c4e08e1c-b6bcbd5cb1ba
         size:                   512M
         status:                 normal
```

7.7.12.12 LIST GRIDDISK

Purpose

The `LIST GRIDDISK` command displays attributes for one or more Oracle Exadata Storage Server grid disks determined by the specified attributes and filters.

Syntax

```
LIST GRIDDISK [ name |  attribute_filters ]  [attribute_list] [DETAIL]
```

Usage Notes

The list of attributes that can be displayed is shown in Example 7-90.

The `asmDeactivationOutcome` attribute can be used to determine if a grid disk can be deactivated without loss of data. This attribute is not included in the list of attributes shown by the `DESCRIBE GRIDDISK` command. When using this attribute, a `YES` in the output means the grid disk can be deactivated.

The `asmModeStatus` attribute can be used to determine the current usage of a grid disk. This attribute is not included in the list of attributes shown by the `DESCRIBE GRIDDISK` command. The possible values for this attribute are as follows:

- `ONLINE`: Oracle ASM is actively using this grid disk.
- `OFFLINE`: Oracle ASM has taken this grid disk offline.
- `DROPPED`: Oracle ASM has dropped this grid disk.
- `UNUSED`: No Oracle ASM instance has used this grid disk on the storage cell.
- `SYNCING`: Oracle ASM has started to set this grid disk to online.
- `UNKNOWN`: Oracle ASM instances that use the grid disk are not available to query, or Oracle ASM has rejected the query because it is not in a currently-mounted disk group.

When the `cachingPolicy` attribute is set to none, the associated flash cache is used for write I/O latency capping and logging. It is not used for caching.

Examples

Example 7-155
Example 7-156
Example 7-157
Example 7-158
Example 7-159
Example 7-160

Example 7-155 Listing Grid Disk Attributes

This example shows the `LIST` command with the `GRIDDISK` object, and the corresponding output.

```
CellCLI> LIST GRIDDISK WHERE cellDisk = 'CD_01_cell05' -
         ATTRIBUTES name, status

         DATA_CD_01_cell05        active
         RECO_CD_01_cell05        active

CellCLI> LIST GRIDDISK DATA_CD_01_cell05 DETAIL

         name:                   DATA_CD_01_cell05
         status:                 active
         comment:
         id:                     00000117-84d9-0096-0000-000000000000
         creationTime:           2009-01-16T17:04:49-06:00
         cellDisk:               CD_01_cell05
         offset:                 0
         availableTo:            CLUSTER-C1
         size:                   10G
         errorCount:             0
         diskType:               HardDisk
         cachedBy:               FD_01_FLASH, FD02_FLASH, FD03_FLASH
         cachingPolicy:          default

CellCLI> LIST GRIDDISK DATA_CD_01_cell05 ATTRIBUTES size

         136.640625G

CellCLI> LIST GRIDDISK WHERE status!=active ATTRIBUTES name

         data_CD_01_1_abcd2x3

CellCLI> LIST GRIDDISK data4_CD_09_cell01 DETAIL

         name:                   data4_CD_09_cell01
         availableTo:
         cellDisk:               CD_09_cell01
         comment:
         creationTime:           2009-07-26T17:09:46-07:00
         diskType:               HardDisk
         errorCount:             0
         id:                     00000122-b98a-a47a-0000-000000000000
         offset:                 27.546875G
         size:                   75G
         status:                 active
```

Example 7-156 Determining if a Grid Disk can be Deactivated

This example shows the `asmDeactivationOutcome` attribute being used to determine if a grid disk can be deactivated.

```
CellCLI> LIST GRIDDISK ATTRIBUTES name, asmDeactivationOutcome

         QUAL_CD_00_cell01       Yes
         PROD_CD_02_cell01       Cannot de-activate due to other offline
disks in
                                 the diskgroup
         TEST_CD_03_cell01       Yes
         DATA_CD_04_cell01       Yes
         DATA_CD_05_cell01       Yes
         DATA_CD_06_cell01       Yes
         RECO_CD_01_cell01       Cannot de-activate due to other offline
disks in
                                 the diskgroup
         DATA_CD_08_cell01       Yes
         DATA_CD_09_cell01       Yes
         DATA_CD_10_cell01       Yes
         DATA_CD_11_cell01       Yes
```

Example 7-157 Viewing the Current Usage of a Grid Disk

This example shows the asmModeStatus attribute being used to check the current usage of a grid disk.

```
CellCLI> LIST GRIDDISK ATTRIBUTES name, asmModeStatus

         QUAL_CD_00_cell01        UNUSED
         RECO_CD_01_cell01        OFFLINE
         PROD_CD_02_cell01        SYNCING
         TEST_CD_03_cell01        UNKNOWN
         DATA_CD_04_cell01        ONLINE
         DATA_CD_05_cell01        ONLINE
         DATA_CD_06_cell01        ONLINE
         DATA_CD_07_cell01        ONLINE
         DATA_CD_08_cell01        ONLINE
         DATA_CD_09_cell01        ONLINE
         DATA_CD_10_cell01        ONLINE
         DATA_CD_11_cell01        ONLINE
```

Example 7-158 Checking the Status Using the LIST GRIDDISK Command

This example shows the LIST GRIDDISK command being used to check the status of the resize process when the NOWAIT option is used with the ALTER GRIDDISK command.

```
CellCLI> LIST GRIDDISK DETAIL

         name:                   gd0
         availableTo:
         cellDisk:               c9standby0
         comment:
         creationTime:           2009-07-09T09:07:36-07:00
         diskType:               HardDisk
         errorCount:             0
         id:                     00000122-6045-173b-0000-000000000000
         resizeStatus:            Resize in progress
         offset:                 48M
         size:                   48M
         status:                 active
```

Example 7-159 Checking the Status of Secure Erase

This example shows the LIST GRIDDISK command being used to check the status of grid disks that are being erased.

```
CellCLI> LIST GRIDDISK
         DATA_CD_00_cell01     active
         DATA_CD_05_cell01     active
         DATA_CD_06_cell01     erase in progress
         DATA_CD_07_cell01     erase in progress
```

Example 7-160 Listing Grid Disk Attributes Using ORDER BY and LIMIT

This example shows the `LIST GRIDDISK` command with the `ORDER BY` and `LIMIT` options.

```
CellCLI> LIST GRIDDISK attributes name, size ORDER BY size,name desc LIMIT
6
         standby0        208M
         controlfile0    304M
         logfile1        800M
         logfile0        800M
         datafile4       1.453125G
         datafile3       1.453125G
```

7.7.12.13 LIST IBPORT

Purpose

The `LIST IBPORT` command displays attributes for InfiniBand ports determined by the specified attributes and filters.

Syntax

```
LIST IBPORT  [ name |  attribute_filters ]  [attribute_list] [DETAIL]
```

Usage Notes

The list of attributes that can be displayed is shown in Example 7-91.

If the `activeSlave` attributes for both InfiniBand ports on a server are not listed, then active-active bonding is being used. If an `activeSlave` attribute is `TRUE`, then active-passive bonding is being used.

Examples

The following example shows the `LIST` command with the `IBPORT` object, and the corresponding output.

Example 7-161 Listing IBPORT Attributes

```
CellCLI> LIST IBPORT
         HCA-1:1         Active
         HCA-1:2         Active

CellCLI> LIST IBPORT DETAIL
         name:                   HCA-1:1
         activeSlave:            TRUE
         dataRate:               "40 Gbps"
         hcaFWVersion:           2.7.0
         id:                     0x00212800013e8c67
         lid:                    20
         linkDowned:             0
         linkIntegrityErrs:      0
         linkRecovers:           0
         physLinkState:          LinkUp
         portNumber:             1
         rcvConstraintErrs:      0
         rcvData:                84653709
```

```
rcvErrs:                0
rcvRemotePhysErrs:      0
status:                 Active
symbolErrs:             0
vl15Dropped:            0
xmtConstraintErrs:      0
xmtData:                84572496
xmtDiscards:            0

name:                   HCA-1:2
activeSlave:            FALSE
dataRate:               "40 Gbps"
hcaFWVersion:           2.7.0
id:                     0x00212800013e8c68
lid:                    21
linkDowned:             0
linkIntegrityErrs:      0
linkRecovers:           0
physLinkState:          LinkUp
portNumber:             2
rcvConstraintErrs:      0
rcvData:                79355427
rcvErrs:                0
rcvRemotePhysErrs:      0
status:                 Active
symbolErrs:             0
vl15Dropped:            0
xmtConstraintErrs:      0
xmtData:                79274016
xmtDiscards:            0
```

7.7.12.14 LIST IORMPLAN

Purpose

The `LIST IORMPLAN` command lists the current plan of the local cell.

Syntax

```
LIST IORMPLAN [attribute_list]  [DETAIL]
```

Usage Notes

The list of attributes that can be displayed is shown in Example 7-92.

Examples

The following example shows the `LIST` command with the `IORMPLAN` object, and the corresponding output.

Example 7-162 Listing IORMPLAN Attributes

```
CellCLI> LIST IORMPLAN ATTRIBUTES status

         active

CellCLI> LIST IORMPLAN DETAIL

         name:                   cell01_IORMPLAN
         catPlan:             name=administrative,level=1,allocation=80
                              name=interactive,level=2,allocation=90
                              name=batch,level=3,allocation=80
                              name=maintenance,level=4,allocation=50
                              name=other,level=4,allocation=50
         dbPlan:              name=sales_prod,level=1,allocation=80
                              name=finance_prod,level=1,allocation=20
                              name=sales_dev,level=2,allocation=100
                              name=sales_test,level=3,allocation=50
                              name=other,level=3,allocation=50
         objective:           balanced
         status:              active
```

7.7.12.15 LIST IORMPROFILE

Purpose

The `LIST IORMPROFILE` command enables you to list IORM profiles. To see which databases are associated with an IORM profile, use the LIST DATABASE command with the new `PROFILE` attribute.

Syntax

```
LIST IORMPROFILE [name | filters]
```

Usage Notes

name specifies the IORM profile to display.

filters specifies an expression that determines which IORM profiles to display.

If *name* and *filters* are omitted, all the IORM profiles are displayed.

Examples

The following example shows the `LIST IORMPROFILE` command, and the corresponding output.

Example 7-163   LIST IORMPROFILE

```
CellCLI> LIST IORMPROFILE
GOLD
SILVER
BRONZE
```

Related Topics

- About I/O Resource Management Profiles
  I/O Resource Management (IORM) interdatabase plans support profiles to ease management, and configuration of interdatabase plans for hundreds of databases.

7.7.12.16 LIST KEY

Purpose

The LIST KEY command displays key values for clients determined by the specified attributes and filters.

Syntax

```
LIST KEY [name | filters] [attribute_list] [DETAIL]

LIST KEY [FOR {ASMCLUSTER | CELL | LOCAL CELL | REMOTE CELL}] [DETAIL]
```

Usage Notes

- The key value assigned to a client must match the keys in the cellkey.ora files on cells, and the Oracle ASM and database host computers.
- The output of the command might show a type attribute, which was introduced in Oracle Exadata System Software release 12.2.1.1.0. Values for this attribute include ASMCLUSTER, LOCAL-CELL, REMOTE-CELL, and CELL. See the examples below.
- The FOR [LOCAL | REMOTE] CELL and FOR ASMCLUSTER clauses were also introduced in Oracle Exadata System Software release 12.2.1.1.0. These clauses indicate that the command is to display only the keys with the specified type.
- *name* specifies the key to display.

Examples

The following example shows the LIST command with the KEY object, and the corresponding output.

Example 7-164 Listing KEY Attributes

```
CellCLI>  LIST KEY db1 DETAIL
         name:                   db1
         key:                    b67d5587fe728118af47c57ab8da650a


CellCLI>  LIST KEY
         db1     b67d5587fe728118af47c57ab8da650a
         db456   118af47c57ab8da650ab67d5587fe728
         asm1    118af47c57ab8da650ab67d5587fe728   ASMCLUSTER


CellCLI>  LIST KEY asm1 DETAIL
         name:         asm1
         key:          b67d5587fe728118af47c57ab8da650a
         type:         ASMCLUSTER


CellCLI>  LIST KEY FOR CELL DETAIL
```

```
         key: fa292e11b31b210c4b7a24c5f1bb4d32
         type: CELL
```

Related Topics

- About Security Keys

7.7.12.17 LIST LUN

Purpose

The `LIST LUN` command displays attributes for LUNs determined by the specified attributes and filters.

Syntax

```
LIST LUN [ name |  attribute_filters ]  [attribute_list] [DETAIL]
```

Usage Notes

The list of attributes that can be displayed is shown in Example 7-94.

Examples

The following example shows the `LIST` command with the `LUN` object, and the corresponding output.

Example 7-165   Listing LUN Attributes

```
CellCLI> LIST LUN
         0_0     0_0     normal
         0_1     0_1     normal
         0_2     0_2     normal
         0_3     0_3     normal
         0_4     0_4     normal
         0_5     0_5     normal
         0_6     0_6     normal
         0_7     0_7     normal
         0_8     0_8     normal
         0_9     0_9     normal
         0_10    0_10    normal
         0_11    0_11    normal
         1_0     1_0     normal
         1_1     1_1     normal
         1_2     1_2     normal
         1_3     1_3     normal
         2_0     2_0     normal
         2_1     2_1     normal
         2_2     2_2     normal
         2_3     2_3     normal
         4_0     4_0     normal
         4_1     4_1     normal
         4_2     4_2     normal
         4_3     4_3     normal
         5_0     5_0     normal
         5_1     5_1     normal
         5_2     5_2     normal
         5_3     5_3     normal

CellCLI> LIST LUN 0_0 DETAIL
         name:                   0_0
         cellDisk:               CD_00_sgsas1
         deviceName:             /dev/sda
         diskType:               HardDisk
         id:                     0_0
         isSystemLun:            TRUE
         lunAutoCreate:          TRUE
         lunSize:                558.40625G
         lunUID:                 0_0
         physicalDrives:         20:0
         raidLevel:              0
         status:                 normal

CellCLI> LIST LUN 1_0 DETAIL
         name:                   1_0
         cellDisk:               FD_00_sgsas1
         deviceName:             /dev/sdr
         diskType:               FlashDisk
         id:                     1_0
         isSystemLun:            FALSE
         lunAutoCreate:          FALSE
         lunSize:                22.8880615234375G
         overProvisioning:       100.0
         physicalDrives:         [9:0:0:0]
         status:                 normal
```

7.7.12.18 LIST METRICCURRENT

Purpose

The LIST METRICCURRENT command displays a list of collections of all metrics.

Syntax

```
LIST METRICCURRENT [ name |  attribute_filters ]  [attribute_list]
[DETAIL]
```

Usage Notes

- The list of attributes that can be displayed is shown in Example 7-95.
- To reduce the size of the output when you run the LIST METRICCURRENT command, use filters.

Examples

Example 7-166 shows the LIST command with filters to display information about the METRICCURRRENT object, and the corresponding output.

Example 7-167 shows the `LIST METRICCURRENT` command with the `ORDER BY` and `LIMIT` options.

Example 7-166 Listing METRICCURRENT Attributes

```
CellCLI> LIST METRICCURRENT WHERE objectType = 'CELLDISK'

         CD_IO_TM_W_SM_RQ        c9controlfile0  205.5 us/request
         CD_IO_TM_W_SM_RQ        c9datafile0     93.3 us/request
         CD_IO_TM_W_SM_RQ        c9datafile1     0.0 us/request
         CD_IO_TM_W_SM_RQ        c9datafile2     110.5 us/request
         CD_IO_TM_W_SM_RQ        c9datafile3     0.0 us/request
         CD_IO_TM_W_SM_RQ        c9datafile4     541.5 us/request
         CD_IO_TM_W_SM_RQ        c9logfile0      181.2 us/request
         CD_IO_TM_W_SM_RQ        c9logfile1      0.0 us/request
         CD_IO_TM_W_SM_RQ        c9standby0      130.4 us/request

CellCLI> LIST METRICCURRENT WHERE name = CD_IO_TM_W_SM_RQ   -
         AND metricObjectName = c9datafile4 DETAIL

         name:                   CD_IO_TM_W_SM_RQ
         alertState:             normal
         collectionTime:         2009-07-01T15:19:25-07:00
         metricObjectName:       c9datafile4
         metricType:             Rate
         metricValue:            0.0 us/request
         objectType:             CELLDISK

CellCLI> LIST METRICCURRENT CG_IO_UTIL_LG
         CG_IO_UTIL_LG   RDB1.BATCH_GROUP                                0
%
         CG_IO_UTIL_LG   RDB1.INTERACTIVE_GROUP                          0
%
         CG_IO_UTIL_LG   RDB1.OTHER_GROUPS                               0
%
         CG_IO_UTIL_LG   RDB2.BATCH_GROUP                               0
%
         CG_IO_UTIL_LG   RDB2.INTERACTIVE_GROUP                         0
%
         CG_IO_UTIL_LG   RDB2.OTHER_GROUPS                              0
%
```

Example 7-167 Listing METRICCURRENT Attributes Using ORDER BY and LIMIT

```
CellCLI> LIST METRICCURRENT attributes name, metricObjectName,
alertState,   \
         metricValue ORDER BY metricValue desc, metricObjectName
asc,        \
         name desc LIMIT 3

CD_IO_TM_R_LG   c9FLASH0   normal    160,514,088   us
CD_IO_TM_R_LG   c9FLASH1   normal    156,659,463   us
DB_IO_TM_SM     ASM        normal    33,111,890    us
```

Related Topics

- Attribute Filters in LIST Commands
  You can specify which objects to display for the LIST command with the optional WHERE clause.
- Monitoring Oracle Exadata Storage Server with Metrics
  You can monitor a cell by viewing the Oracle Exadata Storage Server metrics.

7.7.12.19 LIST METRICDEFINITION

Purpose

The LIST METRICDEFINITION command displays a list of metric definitions on the cell.

Syntax

```
LIST METRICDEFINITION [ name |  attribute_filters ]  [attribute_list]
[DETAIL]
```

Usage Notes

The list of attributes that can be displayed is shown in Example 7-96.

Examples

The following example shows the LIST command with the METRICDEFINITION object, and the corresponding output.

Example 7-168 Listing the Metric Definitions for a Specific Object

```
CellCLI> LIST metricDefinition WHERE objectType=cell
         CL_BBU_CHARGE
         CL_BBU_TEMP
         CL_CPUT
         CL_CPUT_CS
         CL_CPUT_MS
         CL_FANS
         CL_IO_RQ_NODATA
         CL_IO_RQ_NODATA_SEC
         CL_MEMUT
         CL_MEMUT_CS
         CL_MEMUT_MS
         CL_RUNQ
         CL_SWAP_IN_BY_SEC
         CL_SWAP_OUT_BY_SEC
         CL_SWAP_USAGE
         CL_TEMP
         CL_VIRTMEM_CS
         CL_VIRTMEM_MS
         IORM_MODE
         N_HCA_MB_RCV_SEC
         N_HCA_MB_TRANS_SEC
         N_NIC_KB_RCV_SEC
         N_NIC_KB_TRANS_SEC
```

Example 7-169 Listing the Metric Definition Detail for a Specific Metric

```
CellCLI> LIST metricDefinition WHERE name=cl_swap_in_by_sec DETAIL
         name:                   CL_SWAP_IN_BY_SEC
         description:            "Amount of swap pages read in KB per 
second"
         metricType:             Instantaneous
         objectType:             CELL
         unit:                   KB/sec
```

Related Topics

- Monitoring Oracle Exadata Storage Server with Metrics
  You can monitor a cell by viewing the Oracle Exadata Storage Server metrics.

7.7.12.20 LIST METRICHISTORY

Purpose

The `LIST METRICHISTORY` command displays a list of individual metrics.

Syntax

```
LIST METRICHISTORY [ name | attribute_filters ]  [attribute_list]
                   {over_specification] [MEMORY] [DETAIL]
```

Usage Notes

- The list of attributes that can be displayed is shown in Example 7-97.
- The retention period for metric history entry files is specified by the `metricHistoryDays` cell attribute. You can modify this setting with the CellCLI `ALTER CELL` command.
- The *over_specification* syntax is as follows:

  ```
  OVER number [aggregation_type [aggregation_type]...]
  ```

  In the preceding syntax, *number* is amount of time in minutes for the aggregation, and *aggregation_type* can be `max`, `min`, or `avg`.
- A `WHERE` clause can include the `ageInMinutes` attribute to specify the list is limited to those metrics which have the specified age. For example, the following command would show the metrics created in the previous 15 minutes:

  ```
  CellCLI> LIST METRICHISTORY WHERE ageInMinutes < 15
  ```

Examples

Example 7-170 Listing METRICHISTORY Attributes

This example shows the `LIST` command with the `METRICHISTORY` object, and the corresponding output. To reduce the size of the output when you run the `LIST METRICHISTORY` command, use filters.

```
CellCLI> LIST METRICHISTORY WHERE name like 'CL_.*'             -
            AND collectionTime > '2009-07-01T15:28:36-07:00'

         CL_RUNQ         stbcr03_2       6.0     2009-07-01T15:28:37-07:00
         CL_CPUT         stbcr03_2       47.6 %  2009-07-01T15:29:36-07:00
         CL_FANS         stbcr03_2       1       2009-07-01T15:29:36-07:00
         CL_TEMP         stbcr03_2       0.0 C   2009-07-01T15:29:36-07:00
         CL_RUNQ         stbcr03_2       5.2     2009-07-01T15:29:37-07:00
```

Example 7-171 Listing METRICHISTORY Using the OVER and MEMORY Attributes

This example shows the `LIST METRICHISTORY` command with the `OVER` and `MEMORY` options.

```
CellCLI> LIST METRICHISTORY cl_cput OVER 10 MIN MAX MEMORY
        CL_CPUT         firstcell       55.0 %  2009-11-15T06:00:17-08:00      55.0 %
57.1 %
        CL_CPUT         firstcell       54.7 %  2009-11-15T06:10:17-08:00      54.7 %
56.3 %
        CL_CPUT         firstcell       54.8 %  2009-11-15T06:20:18-08:00      54.7 %
57.2 %
        CL_CPUT         firstcell       55.0 %  2009-11-15T06:30:18-08:00      54.3 %
55.9 %
        CL_CPUT         firstcell       55.0 %  2009-11-15T06:40:18-08:00      54.9 %
57.0 %
        CL_CPUT         firstcell       55.1 %  2009-11-15T06:50:18-08:00      54.8 %
56.4 %
        CL_CPUT         firstcell       58.0 %  2009-11-15T07:00:18-08:00      55.2 %
58.0 %
        CL_CPUT         firstcell       55.5 %  2009-11-15T07:10:18-08:00      55.5 %
67.5 %
```

Example 7-172 Listing METRICHISTORY Attributes Using ORDER BY and LIMIT

This example shows the `LIST METRICHISTORY` command with the `ORDER BY` and `LIMIT` options.

```
CellCLI> LIST METRICHISTORY WHERE name like '.*IO_RQ.*' DETAIL ORDER BY  -
         metricValue desc, metricObjectName desc LIMIT 4

name:                   CT_FD_IO_RQ_SM
alertState:             normal
collectionTime:         2014-05-23T10:59:06-07:00
metricObjectName:       OTHER
metricType:             Cumulative
```

```
metricValue:            3,211,568 IO requests
objectType:             IORM_CATEGORY

name:                   CT_FD_IO_RQ_SM
alertState:             normal
collectionTime:         2014-05-23T10:58:06-07:00
metricObjectName:       OTHER
metricType:             Cumulative
metricValue:            3,211,568 IO requests
objectType:             IORM_CATEGORY

name:                   CT_FD_IO_RQ_SM
alertState:             normal
collectionTime:         2014-05-22T17:23:45-07:00
metricObjectName:       OTHER
metricType:             Cumulative
metricValue:            3,211,568 IO requests
objectType:             IORM_CATEGORY

name:                   CT_FD_IO_RQ_SM
alertState:             normal
collectionTime:         2014-05-22T17:21:41-07:00
metricObjectName:       OTHER
metricType:             Cumulative
metricValue:            3,211,568 IO requests
objectType:             IORM_CATEGORY
```

Example 7-173 Listing METRICHISTORY Attributes with ASM-scoped security configured

This example for Oracle Exadata System Software release 19.1.0 shows the `LIST METRICHISTORY` command with the `DETAIL` option for a system with ASM-scoped security configured.

```
CellCLI> LIST METRICHISTORY WHERE name like 'DB_IO_RQ_SM' DETAIL ORDER BY
-
         metricObjectName LIMIT 4

name:                   DB_IO_RQ_SM
alertState:             normal
collectionTime:         2018-08-23T07:34:12-05:00
metricObjectName:       ASM
metricType:             Cumulative
metricValue:            *.
objectType:             IORM_DATABASE

name:                   DB_IO_RQ_SM
alertState:             normal
collectionTime:         2018-08-23T07:33:38-05:00
metricObjectName:       ASM1.PRODDB
metricType:             Cumulative
metricValue:            *.
objectType:             IORM_DATABASE

name:                   DB_IO_RQ_SM
```

```
alertState:             normal
collectionTime:         2018-08-22T14:27:45-05:00
metricObjectName:       ASM1.PRODDB
metricType:             Cumulative
metricValue:            *.
objectType:             IORM_DATABASE

name:                   DB_IO_RQ_SM
alertState:             normal
collectionTime:         2018-08-22T14:25:06-05:00
metricObjectName:       _OTHER_DATABASE_
metricType:             Cumulative
metricValue:            *.
objectType:             IORM_DATABASE
```

Related Topics

- ALTER CELL
- DESCRIBE CELL
- Attribute Filters in LIST Commands
  You can specify which objects to display for the `LIST` command with the optional `WHERE` clause.

7.7.12.21 LIST OFFLOADGROUP

Purpose

The `LIST OFFLOADGROUP` command displays the attributes for offload groups.

Syntax

```
LIST OFFLOADGROUP [name | filters] [attribute_list] [DETAIL]
```

Usage Notes

- The *name* and *filters* parameters specify the offload groups for which you want to display the attributes.
  - *name* specifies the name of an offload group.
  - *filters* specifies an expression to match one or more offload groups.
  - If neither *name* nor *filters* is specified, the command lists the attributes for all offload groups.
- The *attribute_list* parameter specifies one or more attributes for which you want to view. The *attribute_list* begins with the `ATTRIBUTES` keyword. If specifying more than one attribute, separate the attributes with a comma:

  ```
  ATTRIBUTES { attr1 [, attr2]... }
  ```

  If the *attribute_list* parameter is omitted, the command displays all attributes.
- If the `DETAIL` parameter is used, the command displays an attribute descriptor for each attribute.

Examples

Example 7-174 Displaying All Offload Groups in Detail

The following command lists all the attributes for all offload groups.

```
LIST OFFLOADGROUP DETAIL
```

Example 7-175 Displaying the Attributes for an Offload Group

The following command lists all the attributes for an offload group named "offloadgroup1".

```
LIST OFFLOADGROUP offloadgroup1
```

Example 7-176 Displaying Specific Attributes

The following command lists the name and package attributes for all offload groups.

```
LIST OFFLOADGROUP ATTRIBUTES name, package
```

Related Topics

- ALTER OFFLOADGROUP

7.7.12.22 LIST PHYSICALDISK

Purpose

The `LIST PHYSICALDISK` command displays attributes for one or more physical disks determined by the specified attributes and filters.

Syntax

```
LIST PHYSICALDISK [ name |  attribute_filters ]  [attribute_list]  [DETAIL]
```

Usage Notes

The list of attributes that can be displayed is shown in Example 7-99.

When a physical disk is performing a power cycle, the status of the disk is included in the output. The status options are `Normal-DiskPoweredOn`, `Normal-DiskPoweredOff`, `ProactiveFailure-DiskPoweredOn`, and `ProactiveFailure-DiskPoweredDown`. If physical disk status shows as failed, perform a power cycle on the disk to verify the status of the physical disk.

Example 7-177 Listing Physical Disk Attributes

This example shows the `LIST` command with the `PHYSICALDISK` object, and the corresponding output.

```
CellCLI> LIST PHYSICALDISK
         20:0            K68DWJ          normal
         20:1            K7YXUJ          normal
         20:2            K7TYEJ          normal
         20:3            K7BJMJ          normal
         20:4            K5B4SM          normal
         20:5            KEBTDJ          normal
         20:6            K4URJJ          normal
         20:7            K5E1DM          normal
         20:8            K7VL6J          normal
         20:9            K7N5NJ          normal
         20:10           K7Z3KJ          normal
         20:11           K504ZM          normal
         FLASH_1_0       1030M03RK1      normal
         FLASH_1_1       1030M03RJN      normal
         FLASH_1_2       1030M03RJH      normal
         FLASH_1_3       1030M03RJD      normal
         FLASH_2_0       1027M03N6X      normal
         FLASH_2_1       1027M03NMN      normal
         FLASH_2_2       1027M03N6Y      normal
         FLASH_2_3       1027M03N6W      normal
         FLASH_4_0       1025M03EJ3      normal
         FLASH_4_1       1025M03EJ2      normal
         FLASH_4_2       1025M03EHU      normal
         FLASH_4_3       1025M03EKE      normal
         FLASH_5_0       1028M03QP8      normal
         FLASH_5_1       1028M03QNA      normal
         FLASH_5_2       1028M03QKU      normal
         FLASH_5_3       1028M03QHN      normal

CellCLI> LIST PHYSICALDISK 20:0 DETAIL
         name:                   20:0
         deviceId:               46
         deviceName:             /dev/sda
         diskType:               HardDisk
         enclosureDeviceId:      20
         errOtherCount:          0
         luns:                   0_0
         makeModel:              "HITACHI HUS1560SCSUN600G"
         physicalFirmware:       A8C0
         physicalInsertTime:     2017-07-27T07:03:00-04:00
         physicalInterface:      sas
         physicalSerial:         K68DWJ
         physicalSize:           558.9120712280273G
         slotNumber:             0
         status:                 normal

CellCLI> LIST PHYSICALDISK FLASH_5_3 DETAIL
         name:                   FLASH_5_3
         deviceName:             /dev/sdx
         diskType:               FlashDisk
         luns:                   5_3
         makeModel:              "Sun Flash Accelerator F20 PCIe Card"
         physicalFirmware:       D21Y
         physicalInsertTime:     2017-07-27T07:03:01-04:00
         physicalSerial:         1028M03QHN
         physicalSize:           22.8880615234375G
         slotNumber:             "PCI Slot: 5; FDOM: 3"
         status:                 normal
```

Related Topics

- ALTER PHYSICALDISK
- About Leading Zeros in Object Names
  If an object name begins with a 0, you must use quotes to avoid parsing errors.

7.7.12.23 LIST PLUGGABLEDATABASE

Purpose

Displays the specified attributes for active pluggable databases.

Syntax

```
LIST PLUGGABLEDATABASE [name | attribute_filters] [attribute_list]
[DETAIL]
```

Usage Notes

The filters option is an expression that determines which active pluggable databases should be listed by the command.

Examples

The following example shows the `LIST PLUGGABLEDATABASE` command and the corresponding output.

Example 7-178 Listing Pluggable Database Attributes

```
CellCLI> LIST PLUGGABLEDATABASE
         PDB$SEED
         CDB1_PDB1
         NEWPDB1
         NEWPDB2

CellCLI> LIST PLUGGABLEDATABASE DETAIL
         name:                   PDB$SEED
         asmClusterName:         SALESDBS_ASMCLUSTER
         containerName:          CDB1
         flashCacheLimit:        515M
         flashCacheMin:          0
         flashCacheSize:         0
         iormLimit:              0.0
         iormShare:              2
         pdbID:                  385656752
         pmemCacheMin:           0
         pmemCacheLimit:         256M
         pmemCacheSize:          0


         name:                   CDB1_PDB1
         asmClusterName:         SALESDBS_ASMCLUSTER
         containerName:          CDB1
         flashCacheLimit:        315G
         flashCacheMin:          200G
         flashCacheSize:         0
         iormLimit:              0.0
         iormShare:              5
         pdbID:                  2850864136
         pmemCacheMin:           0
         pmemCacheLimit:         256M
         pmemCacheSize:          0


         name:                   NEWPDB1
         asmClusterName:         SALESDBS_ASMCLUSTER
         containerName:          CDB1
         flashCacheLimit:        157G
         flashCacheMin:          20G
         flashCacheSize:         0
         iormLimit:              80.0
         iormShare:              1
         pdbID:                  167491455
         pmemCacheMin:           0
         pmemCacheLimit:         256M
         pmemCacheSize:          0

         name:                   NEWPDB2
         asmClusterName:         SALESDBS_ASMCLUSTER
         containerName:          CDB1
         flashCacheLimit:        157G
         flashCacheMin:          20G
         flashCacheSize:         0
         iormLimit:              60.0
         iormShare:              2
         pdbID:                  2392787216
         pmemCacheMin:           0
         pmemCacheLimit:         256M
         pmemCacheSize:          0
```

7.7.12.24 LIST PMEMCACHE

Purpose

The `LIST PMEMCACHE` command displays the specified attributes for the PMEM cache.

Syntax

```
LIST PMEMCACHE [attribute_list] [DETAIL]
```

Usage Notes

The list of attributes that can be displayed is shown in DESCRIBE PMEMCACHE.

Examples

The following example shows the `LIST` command with the `PMEMCACHE` object, and the corresponding output.

Example 7-179 Listing PMEM Cache Attributes

```
CellCLI> LIST PMEMCACHE
         dbm01celadm08_PMEMCACHE      normal

CellCLI> LIST PMEMCACHE DETAIL
         name:               dbm01celadm08_PMEMCACHE
         cellDisk:
PM_10_dbm01celadm08,PM_11_dbm01celadm08,PM_06_dbm01celadm08,PM_01_dbm01cela
dm08,

PM_00_dbm01celadm08,PM_03_dbm01celadm08,PM_08_dbm01celadm08,PM_02_dbm01cela
dm08,

PM_09_dbm01celadm08,PM_07_dbm01celadm08,PM_04_dbm01celadm08,PM_05_dbm01cela
dm08
         creationTime:        2019-09-12T11:37:00-07:00
         degradedCelldisks:
         effectiveCacheSize:  1.474365234375T
         id:                  0ba69a26-b02d-46a8-a5a6-82e699a6ac88
         size:                1.474365234375T
         status:              normal
```

7.7.12.25 LIST PMEMLOG

The `LIST PMEMLOG` command displays the specified attributes for the PMEMLOG.

Syntax

```
LIST PMEMLOG [attribute_list] [DETAIL]
```

Usage Notes

The list of attributes that can be displayed is shown in DESCRIBE PMEMLOG.

Examples

The following example shows the `LIST` command with the `PMEMLOG` object, and the corresponding output.

Example 7-180 Listing the Current PMEMLOG

The following example shows the `LIST PMEMLOG` command with the default values.

```
CellCLI> LIST PMEMLOG

         raw_PMEMLOG    normal
```

Example 7-181 Listing All PMEMLOG Attributes

```
CellCLI> LIST PMEMLOG DETAIL

         name:                   raw_PMEMLOG
         cellDisk:               NV_00_raw,NV_01_raw,NV_02_raw,NV_03_raw
         creationTime:           2019-11-23T12:34:56-05:00
         degradedCelldisks:
         effectiveSize           1G
         efficiency:             100.0
         id:                     8a0adc84-9088-4c4e-8e1c-b6bcbd5cb1ba
         size:                   1G
         status:                 normal
```

7.7.12.26 LIST QUARANTINE

Purpose

The `LIST QUARANTINE` command displays specified attributes for quarantines.

Syntax

```
LIST QUARANTINE [ name |  attribute_filters ]  [attribute_list] [DETAIL]
```

Examples

The following example shows the `LIST` command with the `QUARANTINE` object.

Example 7-182   Listing QUARANTINE Attributes

```
CellCLI> LIST QUARANTINE  DETAIL

CellCLI> LIST QUARANTINE where comment like 'added.*'
```

Related Topics

- DESCRIBE QUARANTINE

7.7.12.27 LIST ROLE

Purpose

The `LIST ROLE` command displays the specified attributes for a role.

Syntax

```
LIST ROLE [name | filters]
          [ATTRIBUTES {ALL | attr1 [,attr2]...}] [DETAIL]
```

Usage Notes

- Use *name* to display information about a specific role.
- Use an expression in place of *filters* to display information about the roles that satisfy the expression.
- Use the `ATTRIBUTES` keyword to display information about one or more attributes of the roles. The `ALL` option can be used to display all attributes.
- Use the keyword `DETAIL` to format the output as an attribute on each line, with an attribute descriptor preceding each value.

Example 7-183 Displaying Roles

This example shows how to display detailed information about all roles.

```
CellCLI> LIST ROLE DETAIL
         name:                   admin
         privileges:             object=all objects, verb=all actions,
attributes=all attributes, options=all options

         name:                   gd_monitor
         privileges:             object=griddisk, verb=0, attributes=all
attributes, options=all options
                                 object=griddisk, verb=list,
attributes=all attributes, options=all options
```

Example 7-184 Displaying Roles Using an Expression

This example shows how to display roles using an expression.

```
CellCLI> LIST ROLE WHERE name>'ad' AND name<'ba'
         admin
```

7.7.12.28 LIST SOFTWAREHISTORY

Purpose

The `LIST SOFTWAREHISTORY` command displays a list of final states for past software updates.

Syntax

```
LIST SOFTWAREHISTORY [attribute_filters] [attribute_list] [DETAIL]
```

Example 7-185 Displaying the History of a Scheduled Software Update

By default, only the update name and status are shown.

```
CellCLI> LIST SOFTWAREHISTORY
  12.2.1.2.0.170509  Last update completed at: 2017-05-20T08:00:57-07:00
  12.2.1.2.0.170520  Last update completed at: 2017-05-21T06:39:54-07:00
  12.2.1.2.0.17052   Last update completed at: 2017-06-08T08:56:45-07:00
  12.2.1.2.0.170603  Last update completed at: 2017-06-08T16:03:17-07:00
```

Example 7-186 Displaying the Detailed History of a Specific Software Update

This example shows the detailed software update history for a particular update which is referenced by its name.

```
CellCLI> LIST SOFTWAREHISTORY WHERE name='12.2.1.2.0.170808.1' DETAIL
         name:                   12.2.1.2.0.170808.1
         status:                 Upgrade failed. See alerts at:
2017-08-10T10:56:15-07:00
```

7.7.12.29 LIST SOFTWAREUPDATE

Purpose

The `LIST SOFTWAREUPDATE` command displays the status of the most recently scheduled update.

Syntax

```
LIST SOFTWAREUPDATE [attribute_list] [DETAIL]
```

Usage Notes

The possible states for the software update are:

- `Ready to update at: update_time`
- `Downloading`
- `Checking prerequisites`
- `Prerequisites failed. See alerts.`
- `Last update completed at: update_time`
- `Running`
- `Upgrade failed. See alerts.`

Example 7-187 Displaying the Status of a Scheduled Software Update

By default, only the update name and status are shown.

```
CellCLI> LIST SOFTWAREUPDATE
  12.2.1.2.0.170603   Last update completed at: 2017-06-08 16:03:17 -0700
```

Example 7-188 Displaying the Detailed Status of a Scheduled Software Update

By default, only the update name and status are shown.

```
CellCLI> LIST SOFTWAREUPDATE DETAIL
         name:                   12.2.1.2.0.170808.1
         status:                 Upgrade failed. See alerts
         store:                  https://mystore_url:4443
         time:                   2017-08-10T10:35:00-07:00
```

7.7.12.30 LIST THRESHOLD

Purpose

The `LIST THRESHOLD` command displays attributes for one or more thresholds determined by the specified attributes and filters.

Syntax

```
LIST THRESHOLD [ name |  attribute_filters ]  [attribute_list] [DETAIL]
```

Usage Notes

The list of attributes that can be displayed is shown in Example 7-107.

Examples

The following example shows the `LIST` command with the `THRESHOLD` object, and the corresponding output.

Example 7-189   Listing Threshold Attributes

```
CellCLI> LIST THRESHOLD

         ct_io_wt_rq.interactive
         db_io_rq_sm_sec.db123
         ....

CellCLI> LIST THRESHOLD ct_io_wt_rq.interactive DETAIL

         comparison:             =
         critical:               20.0
         name:                   ct_io_wt_rq.interactive
         observation:            5
         occurences:             2
         warning:                10.0

CellCLI> LIST THRESHOLD db_io_rq_sm_sec.db123 DETAIL

         comparison:             >
         name:                   db_io_rq_sm_sec.db123
         critical:               120.0
```

7.7.12.31 LIST USER

Purpose

The `LIST USER` command displays the specified attributes for a user.

Syntax

`LIST USER [`*`name`*` | `*`filters`*`] [`*`attribute_list`*`] [DETAIL]`

Usage Notes

- *name* is the user name.
- *filters* is an expression that determines which users are displayed.
- attribute_list is the attributes to display. The `ALL` option can be used to display all attributes.
- The `DETAIL` option formats the output as an attribute on each line, with an attribute descriptor preceding each value.

Examples

The following example shows the `LIST USER` command.

Example 7-190 Using the LIST USER Command

```
CellCLI> LIST USER DETAIL

CellCLI> LIST USER where name like 'agarcia' DETAIL
         name:                  agarcia
         roles:                 role=gd_monitor
         Privileges:            object=griddisk
                                verb=list
                                attributes=all attributes
                                options= all options
```

7.7.13 QUIT

Purpose

The QUIT command exits from the CellCLI utility, and returns control to the operating system prompt.

Syntax

```
QUIT
```

QUIT has the same functionality as the EXIT command.

7.7.14 REVOKE

Purpose

The REVOKE command removes privileges and roles.

Syntax

```
REVOKE object_type [name] FROM sub_object_type [sub_object_name]
```

Usage Notes

- *object_type* can be as follows:
  - PRIVILEGE
  - ROLE
- The following can be used for PRIVILEGE object type:
  - *name* is in the following format:

    ```
    { ALL ACTIONS | action } ON { ALL OBJECTS | object }  { ALL ATTRIBUTES | \
    ATTRIBUTES attribute1 [, attribute2, ...] }  { WITH ALL OPTIONS |        \
    WITH OPTIONS option1 [, option2, ...] }
    ```

  - The *sub_object_type* must be ROLE.
  - The *sub_object_name* is the name of the role.
- The following can be used for the ROLE object type:
  - *name* is the role name.
  - The *sub_object_type* must be USER.

  - The *sub_object_name* is the name of the user.
- REVOKE PRIVILEGE
- REVOKE ROLE

Related Topics

- REVOKE PRIVILEGE
- REVOKE ROLE

7.7.14.1 REVOKE PRIVILEGE

Purpose

The REVOKE PRIVILEGE command revokes privileges from a role.

Syntax

```
REVOKE PRIVILEGE { ALL ACTIONS | action } ON { ALL OBJECTS | object }
{ ALL ATTRIBUTES | ATTRIBUTES attribute1 [, attribute2...] } { WITH ALL
OPTIONS | WITH OPTIONS option1 [, option2, ...] } FROM ROLE { ALL | role1
[, role2, ...] }
```

Usage Notes

- *action* is the command.
- *object* is object type for the action.
- *attribute* are the attributes for the object.
- *option* are the options for the object.
- *role* is the name of the role from which to revoke privileges.
- The ALL ACTIONS argument revokes privileges for all actions.
- The ALL OBJECTS argument revokes privileges for all objects.
- The ALL ATTRIBUTES argument revokes privileges for all attributes.
- The WITH ALL OPTIONS argument revokes privileges for all options.

Examples

The following example shows the REVOKE PRIVILEGE command.

Example 7-191 Revoking a Privilege

```
CellCLI> REVOKE PRIVILEGE ALL ACTIONS ON ALL OBJECTS ALL ATTRIBUTES -
         WITH ALL OPTIONS FROM ROLE ALL

CellCLI> REVOKE PRIVILEGE list on griddisk ATTRIBUTES name,size     -
         WITH OPTIONS detail FROM ROLE gd_monitor
```

7.7.14.2 REVOKE ROLE

Purpose

The `REVOKE ROLE` command revokes the role for a user.

Syntax

```
REVOKE ROLE { ALL | role1 [, role2, ...] } FROM USER { ALL | user1 [,
user2...] }
```

Usage Notes

- *role* is the name of a role.
- *user* is the name of a user.
- The `ALL` argument revokes all roles from the user.
- The `FROM USER ALL` argument revokes the role from all users.

Examples

The following example shows how to revoke a role from a user.

Example 7-192   Revoking a Role From a User

```
CellCLI> REVOKE ROLE gd_monitor FROM USER jdoe
```

7.7.15 SET

Purpose

The `SET` command sets parameter options in the CellCLI environment.

Syntax

```
SET DATEFORMAT {LOCAL | STANDARD}
SET ECHO [ON | OFF]
```

The `SET DATEFORMAT` command controls the format of displayed dates. For commands that accept dates, the standard date-time format is recommended. The local format is also accepted. The standard format is recommended for scripts because that format is less sensitive to the time zone, region, and locale changes that might occur when running a script.

The `SET ECHO` command controls whether to echo commands in a script that is run with `@` or `START`. The `ON` option displays the commands on screen. The `OFF` option suppresses the display. The `SET ECHO` command does not affect the display of commands entered interactively or redirected from the operating system.

Example 7-193 Setting the Date Format with the SET Command

This example shows an example of the SET command.

```
SET DATEFORMAT STANDARD
```

7.7.16 SPOOL

Purpose

The SPOOL command writes (spools) the results of commands to the specified file on the cell file system.

Syntax

```
SPO[OL] [file_name [ CRE[ATE] | REP[LACE] | APP[END] ] | OFF]
```

If you issue SPOOL *file_name* with no option, then the output is spooled to that file whether or not the file already exists. The REPLACE option is the default behavior.

The SPOOL options are described in the following table.

Table 7-4 SPOOL Options

Option	Description
APPEND	Adds the results to the end of the file specified.
CREATE	Creates a new file with the name specified, and raises an error if the file exists.
file_name	Names the file to which the results are written. It can be specified with a fully-qualified path name, or with a partially-qualified path name relative to the current directory.
no option	Displays the name of the current spool target file, if any.
OFF	Stops writing (spooling) output to the file.
REPLACE	Replaces the contents of an existing specified file. If the file does not exist, then REPLACE creates the file. This is the default behavior.

7.7.17 START and @

Purpose

The START or @ command runs the CellCLI commands in the specified script file.

Syntax

```
STA[RT] file_name
@file_name
```

The START and @ both require the option *file_name*. It is the name of the script file that contains the CellCLI commands. If the file name does not include a fully-qualified path, then the CellCLI utility searches for the file relative to the current directory.

The START or @ command is useful when entering long or multiple CellCLI commands. For example, all the commands in Example 7-5 or Example 7-6 can be entered in a text file named alter_cell, then run with START alter_cell, assuming that the alter_cell file is in the current directory.

8
Using the dcli Utility

The dcli utility facilitates centralized management across an Oracle Exadata System Software realm.

dcli automates the execution of CellCLI commands on a set of cells and returns the output to the centralized management location where the dcli utility was run.

- Overview of the dcli Utility
  The dcli utility runs commands on multiple cells in parallel threads, but does not support an interactive session with a remote application on a cell.
- dcli Syntax
  This topic describes the syntax for the dcli utility.
- dcli Examples
  This section contains examples of the using the dcli utility.
- Setting Up SSH User-Equivalence on Oracle Exadata Storage Server
  Setting user-equivalence enables you to issue commands to remote cells without having to enter the password for the cell.

8.1 Overview of the dcli Utility

The dcli utility runs commands on multiple cells in parallel threads, but does not support an interactive session with a remote application on a cell.

To use the dcli utility, copy the utility from the `bin` directory on a cell to a host computer from which central management can be performed. You can issue a command to be run on multiple cells, or use files that can be copied to cells and then run. The cells are referenced by their host name or IP address.

The dcli utility requires Python version 2.3 or later. You can determine the version of Python by running the `python -V` command. In addition, use of this tool assumes prior setup of SSH user-equivalence to a cell. You can use the dcli utility initially with the `-k` option to set up SSH user-equivalence to a cell. Also, you can manually set up SSH user-equivalence to cells.

Command output (`stdout` and `stderr`) is collected and displayed after the copy and command execution is finished on the specified cells. The dcli options allow command output to be abbreviated to minimize non-error output, such as messages showing normal status.

Related Topics

- Setting Up SSH User-Equivalence on Oracle Exadata Storage Server
  Setting user-equivalence enables you to issue commands to remote cells without having to enter the password for the cell.

8.2 dcli Syntax

This topic describes the syntax for the dcli utility.

Syntax

```
dcli [options] [command]
```

Command Arguments

- *options*: command options
- *command:* Any command that can be run from an operating system prompt.

Command Options

Table 8-1   dcli Options

Option	Description
`-c` *`cells`*	Specifies a comma-delimited list of target cells to which commands are sent.
`-d` *`destfile`*	Specifies the target destination directory or file on remote cells to be used when copying files or directories using the `-f` option.
`-f` *`file`*	Specifies the files or file template to be copied to the cells. These files are not run. These files can be script files to be run later. The files are copied to the default home directory of the user on the target cell.
`-g` *`groupFile`*	Specifies a file containing a list of target cells to which commands are sent. The cells can be identified by cell names or IP addresses.
`-h`, `--help`	Displays help text and then exits.
`--hidestderr`	Hide standard error messages (STDERR) for commands run remotely using SSH.
`-k`	Sets up SSH user-equivalence for the current user to the cells specified with the `-c` or `-g` option by appending public key files to the `authorized_keys` file on cells.
`-l` *`userId`*	Identifies the user to log in as on remote cells. The default is the `celladmin` user.
`-n`	Abbreviates nonerror output. Cells that return normal output (return code of `0`) only have the cell name listed. The `-n` and `-r` options cannot be used together.
`-r` *`regexp`*	Abbreviates the output lines that match a regular expression. All output lines with that pattern are deleted from output, and the cells names from those output lines are listed on one line. The `-r` and `-n` options cannot be used together.
`-s` *`sshOptions`*	Passes a string of options to SSH.
`--scp=` *`scpOptions`*	Passes a string of options to `scp` if different from *`sshoptions`*.
`--serial`	Serializes the process over Oracle Exadata Storage Servers.
`--showbanner`, `--sh`	Show the banner of the remote node when using SSH.
`-t`	Displays the target cells that are named with the `-c` option or in the *`groupfile`* identified by the `-g` option.

Table 8-1 (Cont.) dcli Options

Option	Description
`--unkey`	Drops keys from the target `authorized_keys` file on Oracle Exadata Storage Servers.
`-v`	Prints the verbose version of messages to `stdout`.
`--version`	Shows the version number of the program and then exits.
`--vmstat=VMSTATOPS`	Displays view process, virtual memory, disk, trap, and CPU activity information, depending on the switches.
`-x` *`execFile`*	Specifies the command file to be copied and run on the cells. The specified file contains a list of commands. A file with the `.scl` extension is run by the CellCLI utility. A file with a different extension is run by the operating system shell on the cell. The file is copied to the default home directory of the user on the target cell.

Usage Notes

For commands that contain punctuation that would be interpreted by the local shell, enclose the command in double quotation marks. If the command includes the following characters, then outer quotation marks and escape characters are required:

- $ (dollar sign)
- ' (quotation mark)
- < (less than)
- > (greater than)
- ( ) (parentheses)

The backslash (\) is the escape character that allows the characters to be passed to the CellCLI utility without being interpreted by the remote shell.

If the command is complex in terms of punctuation that need escape characters, then it may require that the command be put in a script, and run using the `-x` option. Within a script, the escape character is not required.

Troubleshooting

If the local dcli process is terminated, then remote commands might continue, but their output and status is unknown.

Return values from the dcli utility are:

- 0: The file or command was copied, and run successfully on all cells.
- 1: One or more cells could not be reached or remote execution returned a nonzero status.
- 2: A local error prevented any command execution.

If any cells are down or do not respond, then a message is written to `stderr` listing the unresponsive cells. The operations continue on the other cells, and the return code after completion is 1.

8.3 dcli Examples

This section contains examples of the using the dcli utility.

- Using dcli to Set up SSH User-equivalence for a Current User
  This example shows how to set SSH user-equivalence for a current user using the `-k` option.
- Using dcli with the -n Option
  This example shows how to run the CellCLI command `ALTER IORMPLAN OBJECTIVE='basic'`, and abbreviates non-error output.
- Using dcli with the -r Option
  This example shows how to run the CellCLI command `LIST GRIDDISK`, and deletes the lines in the output that contain `normal`.
- Using dcli with the -v Option
  This example shows how to use the verbose (`-v`) option with SSH.
- Using dcli with the -t Option
  This example shows how to use the `-t` option to list target cells.
- Using dcli with the -f Option
  This example shows how to use the `-f` option.
- Using dcli with the --vmstat Option
  This example shows how to use the `--vmstat` option of dcli.
- Using dcli with the --hidestderr Option
  This example shows how to use the `--hidestderr` option.
- Using dcli with the --showbanner Option
  This example shows how to use the `--showbanner` option.
- Using dcli to Change an IORM Plan
  This example shows a CellCLI command that changes the IORMPLAN to active on the target cells in the `mycells` group file.
- Using dcli with a Script
  This example shows the CellCLI commands in the `reConfig.scl` file on the target cells.
- Using dcli to List Grid Disk Status
  This example shows how to run a CellCLI command that lists the name and status of grid disks on the target cells.
- Using dcli to List Alert History Information
  This example shows a CellCLI command that lists alert history name, examined by, severity on the target cells.
- Using dcli to List Alert History Where examinedby is Not Set
  This example shows a CellCLI command that lists alert history where examined by has not been set on the target cells.
- Using dcli to List Current Metric Alert State
  This example shows a CellCLI command that retrieves metric current objects for the number of MB read in large blocks on a grid disk for a group of cells.
- Using dcli to List Specific Metric Current Objects in a Group
  This example shows a CellCLI command that retrieves metric current objects for the number of requests to read or write blocks on a grid disk.

- Using dcli to List Specific Metric Current Objects
  This example shows a CellCLI command that lists metric current objects with name equal to `cl_put` (cell CPU utilization) on the target cells.
- Using dcli to List Physical Disks
  This example shows a CellCLI command that lists physical disks where status is not equal to normal on the target cells.
- Using dcli to List Cell Disks with Free Space
  This example shows a CellCLI command that lists cell disks where free space is less than 100 MB on the target cells.
- Using dcli to View Alert History
  This example shows a CellCLI command to view the alert history from a particular period.

8.3.1 Using dcli to Set up SSH User-equivalence for a Current User

This example shows how to set SSH user-equivalence for a current user using the `-k` option.

Example 8-1 Setting up SSH User-equivalence for a Current User

```
$ ./dcli -k -g mycells
```

The `-k` option assumes the user has accepted the default key file names for the SSH protocol, version 2. These file names are `id_dsa.pub` or `id_rsa.pub`, and are located in the `~/.ssh` directory.

You may be prompted to acknowledge cell authenticity, and may be prompted for the remote user password. The `-k` key exchange is done serially over the cells to prevent the user from getting password prompts from all cells simultaneously. After the `-k` option is used once, subsequent commands to the same cells do not require the `-k` option and do not require passwords for that user from the host.

8.3.2 Using dcli with the -n Option

This example shows how to run the CellCLI command `ALTER IORMPLAN OBJECTIVE='basic'`, and abbreviates non-error output.

Example 8-2 Using the -n Option

```
$ ./dcli -g mycells -l celladmin -n "cellcli -e alter iormplan \
objective=\'basic\'"
```

The abbreviated output would be similar to the following:

```
OK: ['abcd2x3']
stsd2s2:
stsd2s2: CELL-02619: Current IORMPLAN state is not 'active'.
```

8.3.3 Using dcli with the -r Option

This example shows how to run the CellCLI command `LIST GRIDDISK`, and deletes the lines in the output that contain `normal`.

The command is run on the target cells listed in the `mycells` group file.

Example 8-3 Using the -r Option

```
$ ./dcli -l celladmin -r '.*normal' -g mycells "cellcli -e list celldisk"
```

The output would be similar to the following:

```
.*normal: ['stsd2s2', 'abcd2x3']
abcd2x3: CD_06_abcd2x3    importRequired
```

8.3.4 Using dcli with the -v Option

This example shows how to use the verbose (`-v`) option with SSH.

Example 8-4 Using the -v Option

```
$ ./dcli -s "-v" -c mycell date
```

8.3.5 Using dcli with the -t Option

This example shows how to use the `-t` option to list target cells.

The `-t` option should be used with `-c` or `-g` option.

Example 8-5 Using the -t Option

```
$ ./dcli -t -c exa01celadm09 date
Target cells: ['exa01celadm09']
exa01celadm09: Fri Jul 17 15:37:31 PDT 2019
```

8.3.6 Using dcli with the -f Option

This example shows how to use the `-f` option.

Example 8-6 Using the -f Option

In this example, a set of files is copied to the storage servers listed in the `mycells` file.

```
dcli -g mycells -f '*.bin'
```

8.3.7 Using dcli with the --vmstat Option

This example shows how to use the `--vmstat` option of dcli.

Example 8-7   Using the --vmstat Option

```
$ ./dcli -g 123 -l sage --vmstat="-a 3 5"

procs -----------memory---------- ---swap-- -----io----  --system-- ...
13:43:03: r  b   swpd   free  inact active   si   so    bi    bo    in
cs us
abcd2x1: 2  0      0  22656 178512 792272    0    0     1    21     7
2  2
abcd2x2: 0  0 452304  21432 108760 867712    0    0   178   269     2
0  2
abcd2x3: 0  0  49252 912164  70156  49996    1    1    74   249     2
2  1
Minimum: 0  0      0  21432  70156  49996    0    0     1    21     2
0  1
Maximum: 2  0 452304 912164 178512 867712    1    1   178   269     7
2  2
Average: 0  0 167185 318750 119142 569993    0    0    84   179     3
1  1
```

8.3.8 Using dcli with the --hidestderr Option

This example shows how to use the `--hidestderr` option.

This option can only be used when SSH is used for remotely run commands.

Example 8-8   Using the --hidestderr Option

This first command does not use the `--hidestderr` option, so the errors are shown.

```
$ ./dcli -l root -g cell "ls -1 unknown_file; cellcli -e list cell"

exam08cel01: ls: unknown_file: No such file or directory
exam08cel01: exam08cel01         online
exam08cel02: ls: unknown_file: No such file or directory
exam08cel02: exam08cel02         online
```

This second command uses the `--hidestderr` option, so the errors are not shown.

```
$ ./dcli -l root -g cell --hidestderr -l root "ls -l unknown_file;\
cellcli -e list cell"

exam08cel01: exam08cel01         online
exam08cel02: exam08cel02         online
```

8.3.9 Using dcli with the --showbanner Option

This example shows how to use the `--showbanner` option.

The banner of the remote cell replaces `******`*`BANNER`*`******` shown in the example.

Example 8-9 Using the --showbanner Option

```
$ ./dcli --showbanner -l root -g cell "cellcli -e list cell"
exam08cel01: ******BANNER******
exam08cel01:
exam08cel01: ******BANNER******
exam08cel01: exam08cel01         online
exam08cel02: ******BANNER******
exam08cel02:
exam08cel02: ******BANNER******
exam08cel02: exam08cel02         online
```

8.3.10 Using dcli to Change an IORM Plan

This example shows a CellCLI command that changes the IORMPLAN to active on the target cells in the `mycells` group file.

The `-t` option displays the cells that are in the `mycells` group file.

Example 8-10 Using dcli to Change an IORM Plan

```
$ ./dcli -g mycells -l root -t "cellcli -e alter iormplan active"
```

8.3.11 Using dcli with a Script

This example shows the CellCLI commands in the `reConfig.scl` file on the target cells.

The target cells are contained in the `mycells` group file. The command is run as the default `celladmin` user.

Example 8-11 Using dcli with a Script

```
$ ./dcli -g mycells -x reConfig.scl
```

8.3.12 Using dcli to List Grid Disk Status

This example shows how to run a CellCLI command that lists the name and status of grid disks on the target cells.

The target cells are contained in the `mycells` group file. The command is run as the default `celladmin` user. Output lines that contain `active` as the status are deleted.

Example 8-12 Using dcli to List Grid Disk Status

```
$ ./dcli -r '.*active' -g mycells "cellcli -e list griddisk"
```

8.3.13 Using dcli to List Alert History Information

This example shows a CellCLI command that lists alert history name, examined by, severity on the target cells.

The target cells are contained in the mycells group file. The command is run as the default celladmin user. Output lines that contain clear for severity are deleted.

Example 8-13 Using dcli to List Alert History Information

```
$ ./dcli -r '.*clear' -g mycells \
   "cellcli -e list alerthistory attributes name, examinedby, severity"
```

8.3.14 Using dcli to List Alert History Where examinedby is Not Set

This example shows a CellCLI command that lists alert history where examined by has not been set on the target cells.

The target cells are contained in the allcells group file. The command is run as the default celladmin user.

Example 8-14 Using dcli to List Alert History where examinedby is not Set

```
$ ./dcli -g allcells -l celladmin \
   "cellcli -e list alerthistory where examinedby=\'\' "
```

8.3.15 Using dcli to List Current Metric Alert State

This example shows a CellCLI command that retrieves metric current objects for the number of MB read in large blocks on a grid disk for a group of cells.

The command lists the current metric alert state and metric value for the metric GD_IO_BY_R_LG on the target cells. The target cells are contained in the mycells group file. The command is run as the default celladmin user.

Example 8-15 Using dcli to List Current Metric Alert State

```
$ ./dcli -g mycells "cellcli -e list metriccurrent GD_IO_BY_R_LG \
  attributes alertstate, metricvalue"
```

8.3.16 Using dcli to List Specific Metric Current Objects in a Group

This example shows a CellCLI command that retrieves metric current objects for the number of requests to read or write blocks on a grid disk.

The CellCLI command lists metric current objects for names that begin with GD_IO_RQ on the target cells. The target cells are contained in the mycells group file. The command is run as the default celladmin user.

Example 8-16 Using dcli to List Specific Metric Current Objects in a Group

```
$ ./dcli -g mycells "cellcli -e list metriccurrent where name like
\'GD_IO_RQ.*\'"
```

8.3.17 Using dcli to List Specific Metric Current Objects

This example shows a CellCLI command that lists metric current objects with name equal to `cl_put` (cell CPU utilization) on the target cells.

The target cells are contained in the `mycells` group file. The command is run as the default `celladmin` user.

Example 8-17   Using dcli to List Specific Metric Current Objects

```
$ ./dcli -g mycells "cellcli -e list metriccurrent cl_cput"
```

8.3.18 Using dcli to List Physical Disks

This example shows a CellCLI command that lists physical disks where status is not equal to normal on the target cells.

The target cells are contained in the `mycells` group file. The command is run as the default `celladmin` user.

Example 8-18   Using dcli to List Physical Disks

```
$ ./dcli -g allcells "cellcli -e list physicaldisk where status not =
normal"
```

8.3.19 Using dcli to List Cell Disks with Free Space

This example shows a CellCLI command that lists cell disks where free space is less than 100 MB on the target cells.

The target cells are contained in the `mycells` group file. The command is run as the default `celladmin` user.

Example 8-19   Using dcli to List Cell Disks with Free Space

In this example, the backslash (\) is an escape character that allows the greater than character (>) to be passed to the CellCLI utility without being interpreted by the remote shell.

```
$ ./dcli -g allcells "cellcli -e list celldisk where freespace \> 100M"
```

8.3.20 Using dcli to View Alert History

This example shows a CellCLI command to view the alert history from a particular period.

Example 8-20 Using dcli to View Alert History

In this example, the backslash (\) is an escape character that allows the greater than character (>) and the quotation marks to be passed to the CellCLI utility without being interpreted by the remote shell.

```
dcli -g lab.cells "cellcli -e  list alerthistory where begintime \> \'Aug
4, 2009 12:06:38 PM\'"
```

8.4 Setting Up SSH User-Equivalence on Oracle Exadata Storage Server

Setting user-equivalence enables you to issue commands to remote cells without having to enter the password for the cell.

- To set up SSH user-equivalence for use with the `dcli` utility, use the `-k` option.

  See Using dcli to Set up SSH User-equivalence for a Current User.

9 Setting up Oracle Exadata Storage Snapshots

Exadata Snapshots are ideal for creating space-efficient read-only or read-write clones of an Oracle database that you can use for development, testing, or other non-production purposes.

- Overview of Exadata Snapshots
- Prerequisites for Exadata Snapshot Databases
  Before creating Exadata snapshot databases, check that your environment meets these requirements.
- Exadata Snapshot Concepts
  There are various object you need when creating and using Exadata Snapshots.
- Sparse Disk Sizing and Allocation Methodology
  To create Exadata Snapshots, you must have sparse grid disks created with an Oracle ASM disk group created based on those disks.
- Refresh Considerations, or Lifecycle of Exadata Snapshots
  Refresh cycles can influence how you use and create Exadata Snapshots.
- Using an Oracle Data Guard Standby Database as the Test Master
  In a situation where the test master is either a non-container database or a full CDB that needs to be refreshed regularly (that is, the test master is not a PDB), Oracle recommends creating the test master database as an Oracle Data Guard physical standby dedicated to this purpose.
- Managing Exadata Snapshots
  To create and manage Exadata Snapshots, you need to perform these procedures:
- Managing Sparse Griddisks
  You can resize, recreate, or monitor the activity of sparse griddisks.

9.1 Overview of Exadata Snapshots

Traditionally, to clone databases in a production system, you would create test master and snapshot databases on a non-Exadata system (Figure 9-1). In some cases, these databases are a full copy that consumes as much storage as its source (Figure 9-2).

Figure 9-1 Traditional Clone to a Non-Exadata System

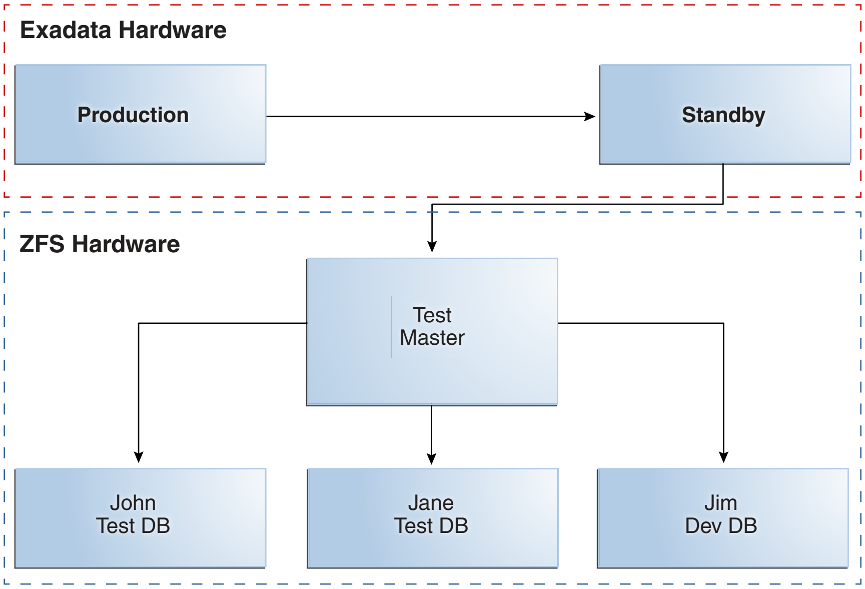
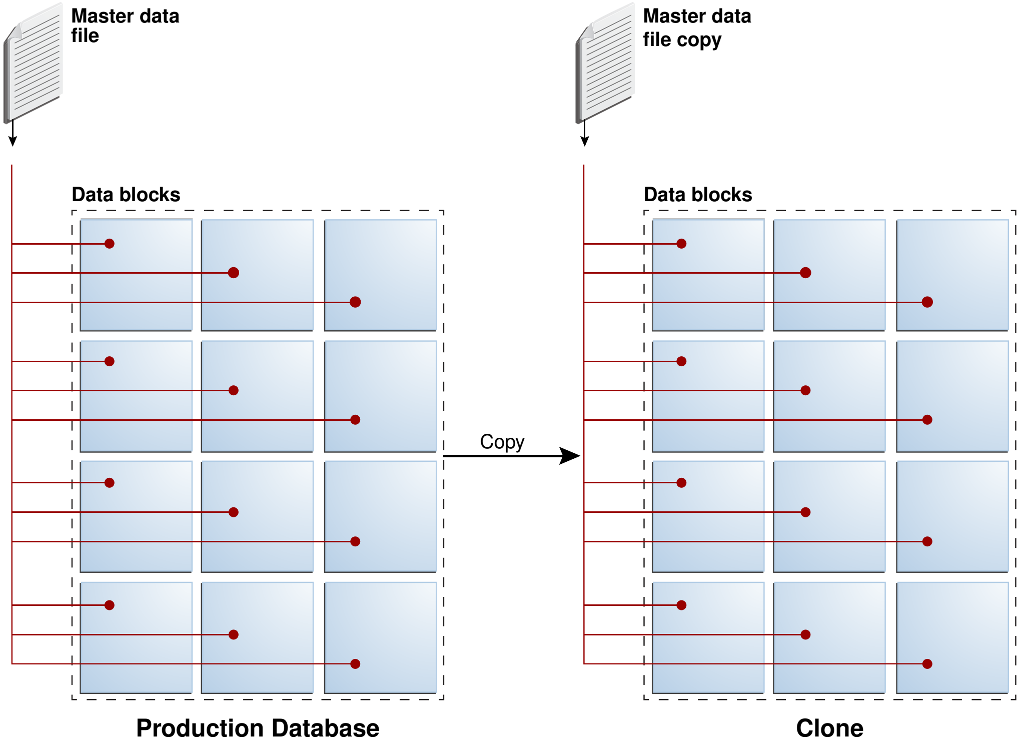


Figure 9-2 Database Clone That Is a Full Copy of Its Source



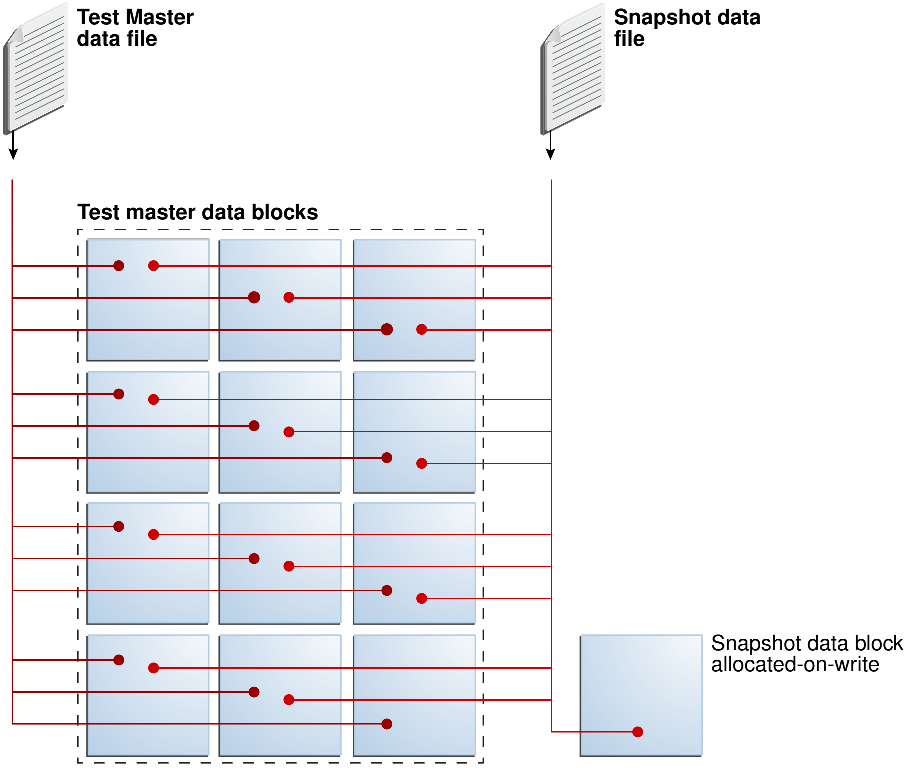
If the clones are a full copy of the production database, this is expensive in terms of the amount of storage consumed and the time it takes to create the clones. Imagine creating ten clones for a multi-terabyte database and it is easy to see why this approach does not scale.

Another drawback to this approach is that Oracle Exadata System Software features such as Smart Scan, Smart Logging, and Smart Flash are not available on non-Exadata systems.

To solve these problems, you can use *Exadata Snapshots*. Exadata Snapshots are ideal for creating space-efficient read-only or read-write clones of an Oracle database that you can use for development, testing, or other non-production purposes, and when multiple clones are required because of disk space and time savings. The following image depicts the space required for an Exadata Snapshot.

> **Note:**
>
> Exadata Snapshots should be used only for development and testing purposes. They should not be used in production environments.

Figure 9-3 An Exadata Snapshot



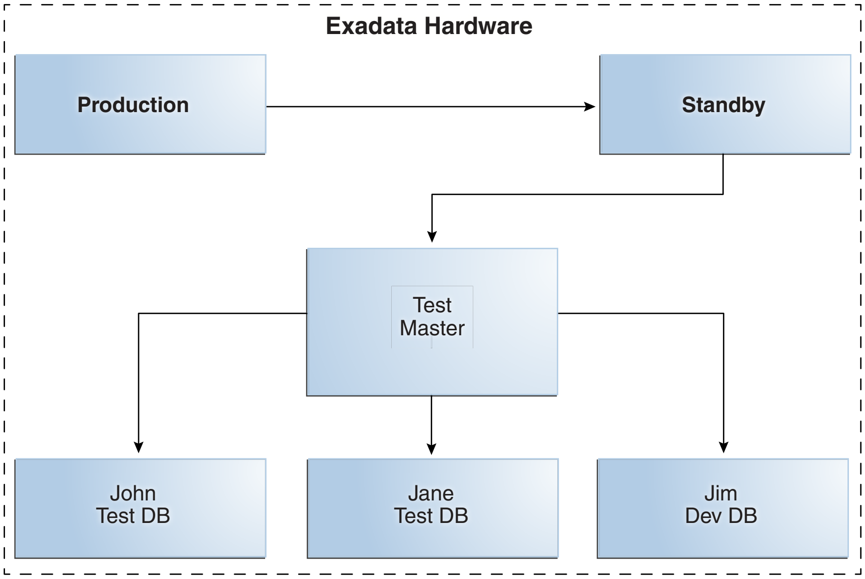
An Exadata Snapshot is based on a *test master*, which is a full clone of the source database. The test master is the only full copy of the source database. From a single test master you can create multiple Exadata Snapshots with minimal additional storage and minimal effort. Each Exadata Snapshot uses a small fraction of the disk space required for the test master and can be created or dropped in seconds. Each Exadata Snapshot is a logical copy of the test master.

Before creating Exadata Snapshots from the test master, you can modify the data in the test master, if required. For example, you can delete or mask sensitive data in the test master before making it available to non-privileged users.

Creating an Exadata Snapshot from the test master is as simple as recording the parent file name in the child file header, an operation that completes in seconds and requires minimal disk space. Additional disk space is consumed only when the user of the snapshot begins to change data. Only new data is written to data blocks that are allocated to the snapshot on write. All requests for data that has not changed are serviced by the data blocks of the test master.

Multiple users can create independent snapshots from the same test master. This enables multiple development and test environments to share space while maintaining independent databases for each user. The following image shows an Exadata environment with three Exadata Snapshots that use the same test master.

Figure 9-4 An Exadata Environment with 3 Exadata Snapshots from the Same Test Master



Hierarchical and Read-Write Snapshots

Oracle Exadata System Software release 12.2.1.1.0 introduces hierarchical and read-write snapshots. Hierarchical snapshots enable you to create snapshots from snapshots. You might want to do this if you are working on your snapshot and wish to save a copy before you make additional changes to it. In hierarchical snapshots, you can make a snapshot at any level descended from the test master. Exadata Snapshots are writable. A snapshot points to the parent's blocks for the data. If you edit a snapshot, then the snapshot will point to the new data. For the unchanged data, it will point to the parent's blocks.

If you have taken a snapshot from a snapshot, and you edit the parent snapshot, then you have to delete all snapshots that are dependent on that snapshot.

- Exadata Snapshot Support of Exadata Features
  In addition to space and time savings, Exadata Snapshots provide cost-effective development, quality assurance and test environments on Oracle Exadata Database Machine.
- Separate Test/Development and Production Environments
  Oracle recommends that test and development environments be hosted on a separate physical Oracle Exadata Rack from the rack hosting the production database.
- Types of Exadata Snapshots
  You can create two types of Exadata Snapshots, depending on the current setup of your environment.
- Hierarchical Snapshot Databases
  Hierarchical snapshots enable you to create snapshot databases from other snapshot databases.
- Sparse Test Masters
  With the introduction of hierarchical snapshots, you can now create Sparse Test Masters.

9.1.1 Exadata Snapshot Support of Exadata Features

In addition to space and time savings, Exadata Snapshots provide cost-effective development, quality assurance and test environments on Oracle Exadata Database Machine.

Exadata Snapshots can be used by developers and testers who need to validate functionality. You can also use Exadata Snapshot to practice maintenance and operational steps in a fully functional Exadata environment (for example, Exadata Smart Flash Cache, Exadata Smart Scan Offload, amd Exadata Hybrid Columnar Compression).

9.1.2 Separate Test/Development and Production Environments

Oracle recommends that test and development environments be hosted on a separate physical Oracle Exadata Rack from the rack hosting the production database.

An Oracle Exadata Database Machine system dedicated to development and test is ideal. Test masters and their associated Exadata Snapshots would be hosted on this system. Alternatively it may be an Oracle Exadata Database Machine system that hosts Oracle Data Guard standby databases for high availability, disaster recovery, or other purposes as permitted by capacity. Test masters and their snapshots may reside in either physical or virtual machines on an Oracle Exadata Database Machine system.

Related Topics

- Types of Exadata Snapshots
  You can create two types of Exadata Snapshots, depending on the current setup of your environment.

9.1.3 Types of Exadata Snapshots

You can create two types of Exadata Snapshots, depending on the current setup of your environment.

- **A test master from a pluggable database (PDB).**

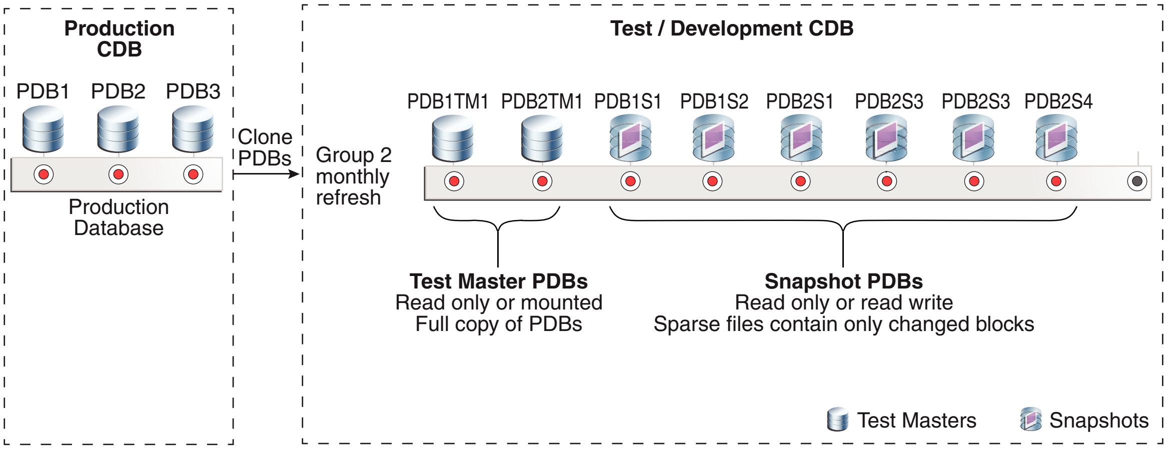
  You can create **Exadata Snapshot PDBs** from individual PDBs within a container database (CDB). You can clone individual PDBs to create the test master PDB, from which Exadata Snapshot PDBs are generated.

  You can move Exadata Snapshot PDBs from one CDB to another in the same Exadata cluster. You can also create an Exadata Snapshot PDB in one container from a test master PDB residing in another container as long as both CDBs are in the same Exadata cluster.

  The test master database and their Exadata Snapshots must be in the same Oracle Automatic Storage Management (Oracle ASM) cluster environment.

  The following image shows a production CDB with three PDBs. Two of the PDBs (PDB1 and PDB2) have been cloned to create test master PDBs, which were then unplugged and plugged into another CDB on the test Oracle Exadata Database Machine. In this figure, six Exadata Snapshot PDBs have been created from the test master PDBs.

Figure 9-5 Exadata Snapshot PDBs



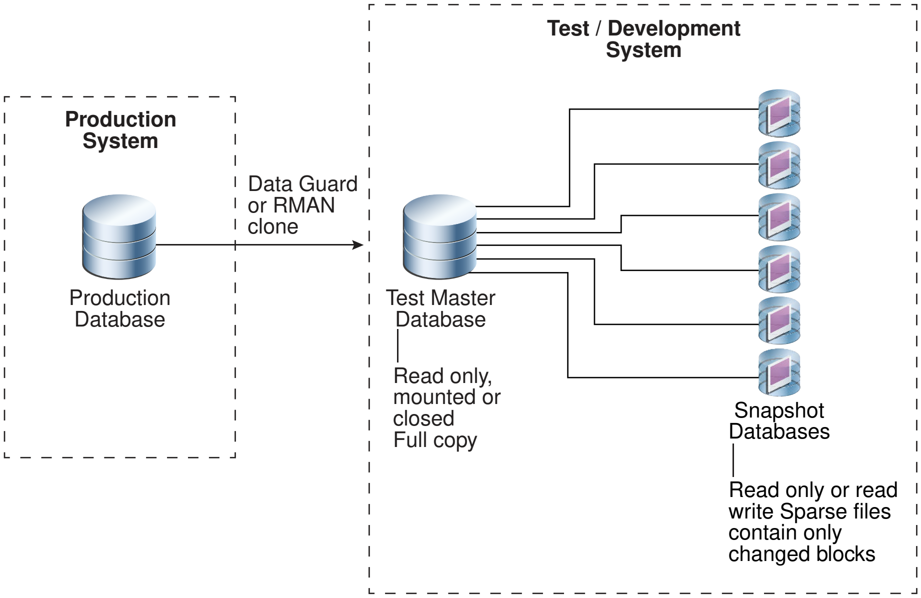
- **A test master from a non-container database**

  An *Exadata Snapshot database* can be a non-container database (non-CDB).

  The next figure depicts a full clone of the production database. The clone, which is the test master database, is hosted on a separate test/development system, and it is associated with six Exadata Snapshots. The test master database is created using either an Oracle Data Guard standby database (recommended if the test master will be refreshed on a regular basis) or Oracle Recovery Manager (RMAN).

  The test master database and their Exadata Snapshots must be in the same Oracle ASM cluster environment.

Figure 9-6 Exadata Snapshot Databases in a Non-Multitenant Environment

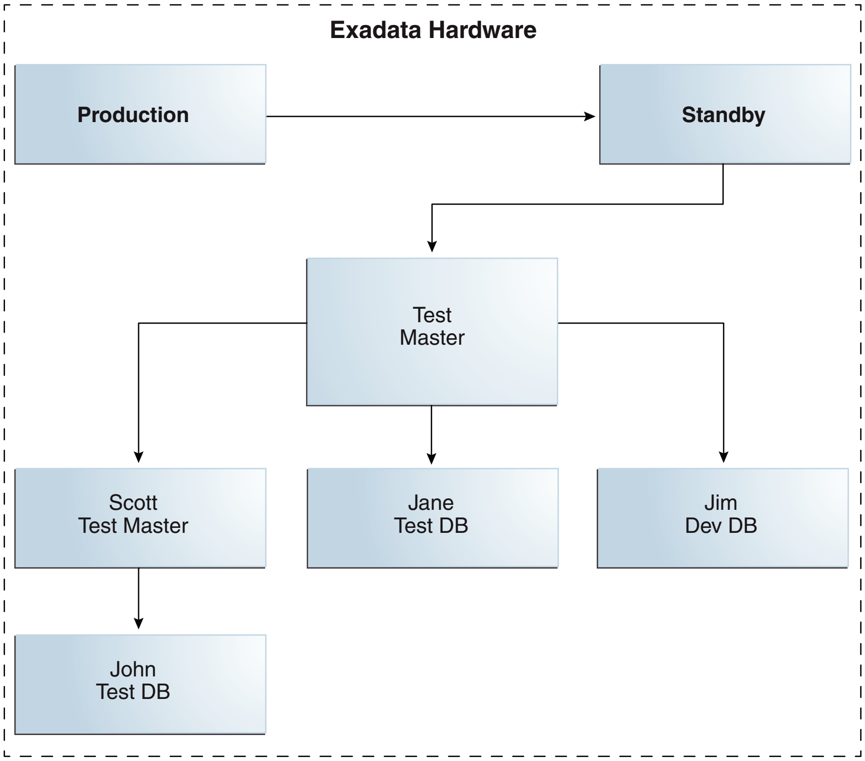

Production System
Production Database
Data Guard or RMAN clone
Test / Development System
Test Master Database
Read only, mounted or closed Full copy
Snapshot Databases
Read only or read write Sparse files contain only changed blocks

9.1.4 Hierarchical Snapshot Databases

Hierarchical snapshots enable you to create snapshot databases from other snapshot databases.

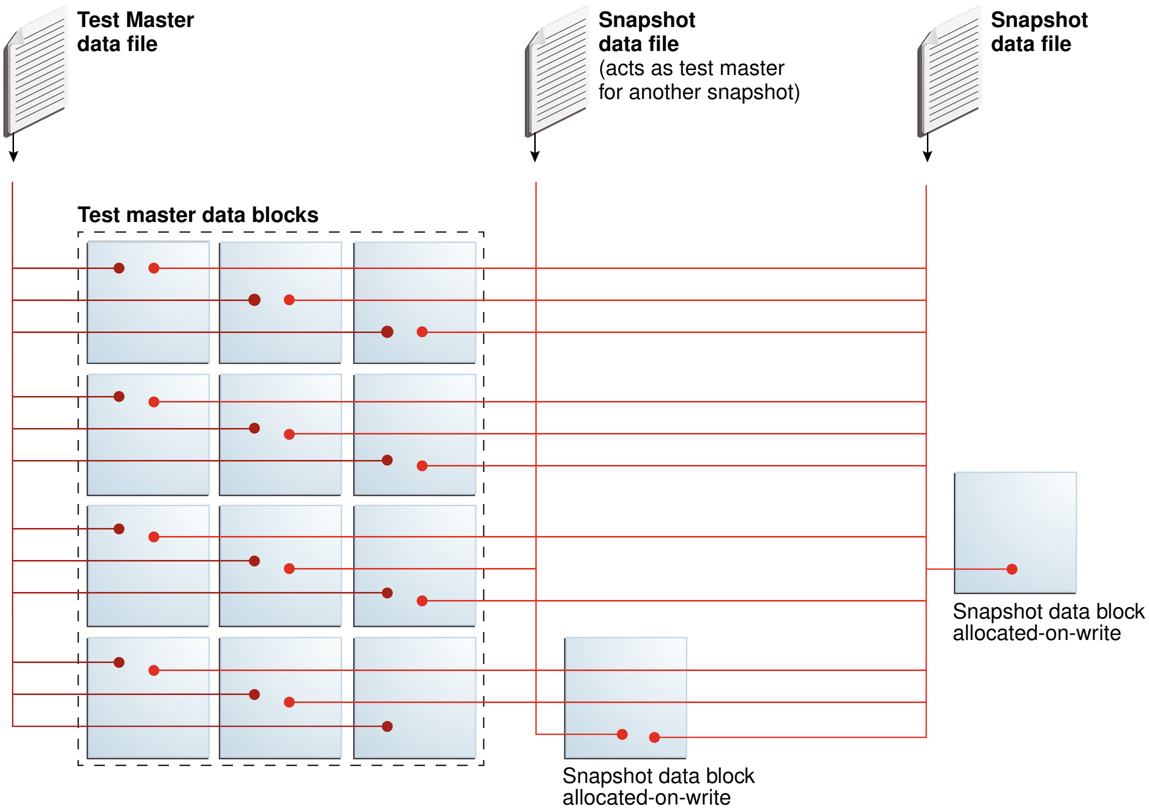
Oracle Exadata System Software release 12.2.1.1.0 introduced hierarchical snapshots. You might want to use Hierarchical snapshots if you are working on your snapshot database and you want to save a copy before you make additional changes to it. You can make as many levels of snapshot databases as you want, but for performance reasons, you might want to limit it at 10 levels.

Figure 9-7   Hierarchical Snapshot Databases



A snapshot database points to its parent’s blocks for data. If you make a change to a snapshot database, then the snapshot database allocates a new block for the changed data. For the unchanged data, the snapshot points to the parent’s blocks. A snapshot that is several levels descended from the original test master will retrieve its data by traversing up the tree starting with the snapshot database from which it was created.

Figure 9-8 Allocation of Blocks in Hierarchical Snapshot Databases


Test Master data file
Snapshot data file (acts as test master for another snapshot)
Snapshot data file
Test master data blocks
Snapshot data block allocated-on-write
Snapshot data block allocated-on-write

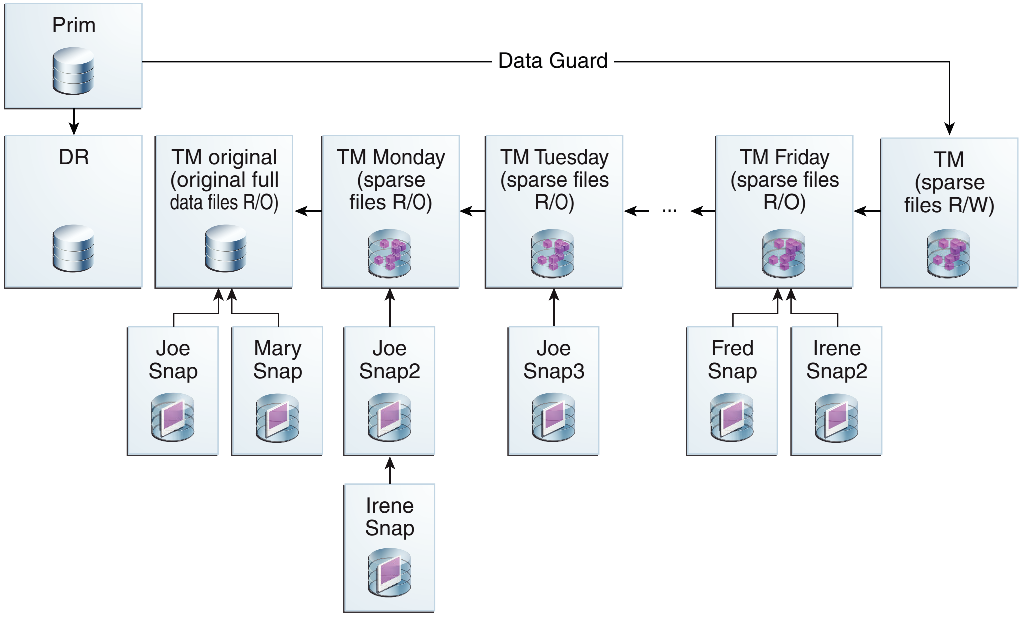
If you have taken a snapshot database from another snapshot database, and you make a change to the parent snapshot database, then you have to drop all snapshot databases that depend on that snapshot database. The parent snapshot database becomes read-only when child snapshots are created from it. If you want to write to the parent snapshot again, you must drop all child snapshots.

9.1.5 Sparse Test Masters

With the introduction of hierarchical snapshots, you can now create Sparse Test Masters.

Hierarchical snapshot databases provide the capability to create Sparse Test Masters. With a sparse test master there is one full copy of data to provide the parent file data for multiple levels of snapshots. When you are ready to create snapshots, you create an additional set of sparse files to receive updates from your production database via a replication technology such as Oracle Data Guard. When you need to create a new test master of current data, rather than cloning the full production database again as of the new point in time, you mark the previous additional sparse files read only creating a sparse test master and create a new set of sparse files to be kept current. Additional snapshots on this sparse test master could then be created to be used by application developers for testing. You can repeat this process up to 9 times total to allow one full copy of the production database to support snapshots from multiple points in time. Creating the new sparse test master is very quick (5 minutes or less) allowing you to easily keep current with production.

Figure 9-9   Example configuration with Sparse Test Masters


Prim
Data Guard
DR
TM original (original full data files R/O)
TM Monday (sparse files R/O)
TM Tuesday (sparse files R/O)
...
TM Friday (sparse files R/O)
TM (sparse files R/W)
Joe Snap
Mary Snap
Joe Snap2
Joe Snap3
Fred Snap
Irene Snap2
Irene Snap

The sparse test master database, `TM`, applies to a non-container database (CDB).

Related Topics

- Creating SparseTest Masters from a Single Full Database Copy
  You can create multiple sparse test master from a single copy of a full database.

9.2 Prerequisites for Exadata Snapshot Databases

Before creating Exadata snapshot databases, check that your environment meets these requirements.

- Storage servers must be Oracle Exadata Database Machine X3-2 or later
- Oracle Exadata System Software 12.1.2.1.0 or later for Oracle Exadata Storage Servers and Oracle Exadata Database Servers

  You cannot downgrade to an earlier version with sparse grid disks on a cell.
- Oracle Grid Infrastructure software release 12.1.0.2.0 BP5 or later

  The Oracle ASM disk group that contains the sparse Oracle ASM grid disks must have both `COMPATIBLE.RDBMS` and `COMPATIBLE.ASM` set to 12.1.0.2 or later.

  The parent disk group can be 11.2 compatible.
- For disk groups hosting test master files, `COMPATIBLE.RDBMS` must be set to 11.2.0.0.0 or later.
- Oracle Database software release 12.1.0.2.0 BP5 or later

  The parent database and the snapshot database must be 12.1.0.2 compatible.
- The data files for the snapshot database and the parent database must be on the same Oracle ASM cluster.

- The `db_block_size` database initialization parameter must be at least 4096 and be a multiple of 4096.
- If you are using hierarchical snapshot databases, sparse test master databases, the new `--sparse` option for the Oracle ASM `cp` command, or the new `setsparseparent` Oracle ASM command, then you need Oracle Database and Oracle Grid Infrastructure software release 12.2.0.1.0 and Oracle Exadata System Software release 12.2.1.1.0 or later.

Related Topics

- setsparseparent Command

9.3 Exadata Snapshot Concepts

There are various object you need when creating and using Exadata Snapshots.

- Sparse Database and Sparse Files
  In a sparse database, such as an Exadata Snapshot database, its data files are sparse files.
- Sparse Grid Disks
  A sparse grid disk has a virtual size attribute as well as physical size.
- Sparse Disk Groups
  Exadata Snapshots utilize Oracle ASM sparse disk groups.

9.3.1 Sparse Database and Sparse Files

In a sparse database, such as an Exadata Snapshot database, its data files are sparse files.

A database consists of the following files:

- control files
- online redo logs
- temp files
- data files

A **sparse file** contains only changes made to blocks from the parent file (the parent file remains unchanged) and maintains a pointer to the parent file for access to unchanged data.

The Exadata Snapshot has its own copy of the other database files (control files, online redo logs, and temp files). These other database files are not sparse files.

Related Topics

- Overview of Exadata Snapshots

9.3.2 Sparse Grid Disks

A sparse grid disk has a virtual size attribute as well as physical size.

A sparse Oracle ASM disk group is composed of sparse grid disks.

The maximum physical size that can be allocated to sparse grid disks from a single cell disk is 4 TB. The maximum allowed virtual size is 100 TB.

You create sparse grid disks before creating a sparse Oracle ASM disk group. If you are deploying a new system or re-deploying an existing system, set aside some space from each cell disk for use by the disk group just as you would for any other disk group.

If you want to use an existing system without reimaging, see My Oracle Support note 1467056.1 for instructions on resizing existing grid disks.

Related Topics

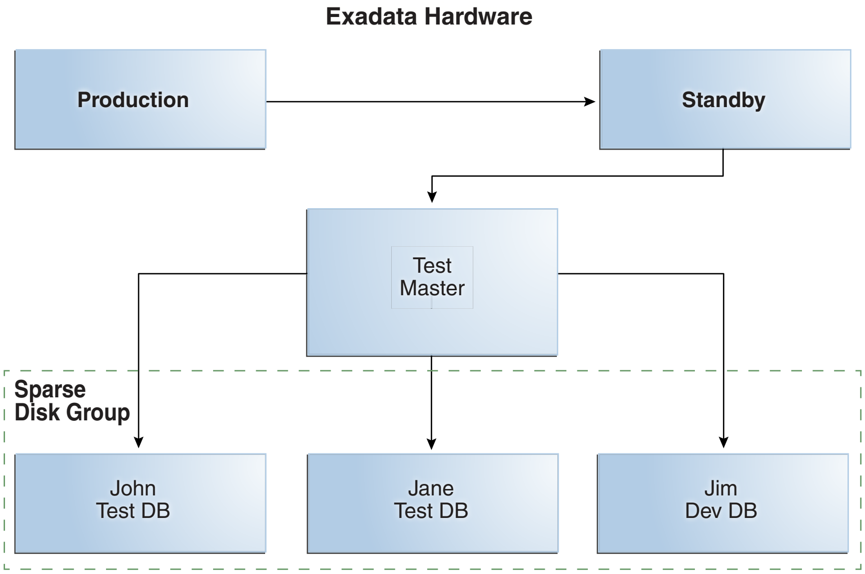
- Creating Sparse Grid Disks
- Resizing Grid Disks in Exadata: Examples (My Oracle Support Doc ID 1467056.1)

9.3.3 Sparse Disk Groups

Exadata Snapshots utilize Oracle ASM sparse disk groups.

Sparse data files can be created only in Oracle Automatic Storage Management (Oracle ASM) sparse disk groups. The following figure shows a sparse disk group containing three Exadata Snapshots.

Figure 9-10 Sparse Disk Group Contains Exadata Snapshots



A sparse Oracle ASM disk group can store both sparse and non-sparse files. You can create full database objects in a sparse Oracle ASM disk group. For example, you can create your test master database in your sparse Oracle ASM disk group along with the Exadata Snapshots. Note that there will be no space savings for the test master database because it will be a full copy of the source database.

A sparse disk group has the following attributes:

- `compatible.asm` must be set to `12.1.0.2` or higher.

- compatible.rdbms must be set to 12.1.0.2 or higher.
- cell.sparse_dg must be set to allsparse. This attribute identifies the disk group to Oracle ASM as being made up of sparse grid disks.
- appliance.mode must be set to true.
- A sparse disk group uses 16 times the size of an extent. For 4M allocation unit (AU), each extent is 64M.
- A sparse disk group uses Virtually Allocated Metadata.

For example, the following SQL command creates a sparse disk group:

```
SQL> CREATE DISKGROUP SPARSE
  NORMAL REDUNDANCY
  DISK 'o/*/SPARSE_*'
  ATTRIBUTE
    'compatible.asm'          = '12.1.0.2',
    'compatible.rdbms'        = '12.1.0.2',
    'cell.smart_scan_capable' = 'true',
    'cell.sparse_dg'          = 'allsparse',
    'au_size'                 = '4M';
```

9.4 Sparse Disk Sizing and Allocation Methodology

To create Exadata Snapshots, you must have sparse grid disks created with an Oracle ASM disk group created based on those disks.

It is not possible to directly modify an existing disk group and convert it into a sparse disk group; you must create the grid disks with the sparse attributes. If you want to use an existing disk group, then the disk group and associated grid disks must be dropped and re-created.

If you want to create a new sparse disk group and all of your space is currently allocated to existing disk groups, then you must resize one or more existing disk groups to free up space. Use the following steps to perform this process.

- Sizing Steps for New Sparse Disk Groups
  These steps describe how to determine the space needed for spare disk groups and how to allocate that space.

9.4.1 Sizing Steps for New Sparse Disk Groups

These steps describe how to determine the space needed for spare disk groups and how to allocate that space.

When sizing, remember that you can also put non-sparse databases into the sparse disk group. Non-sparse databases and files will use up the full physical space.

The maximum physical size that can be allocated to sparse grid disks from a single cell disk is 4 TB.

The maximum virtual size that can be allocated per sparse grid disk is 100 TB. However, to avoid additional performance impact Oracle does not recommend creating large virtual sizes unless it is truly required.

1. Determine the amount of physical space for your snapshots. Use the formula found in Calculating the Physical Size for Grid Disks.

   Note that a good maximum starting point for physical space allocation for sparse grid disk is 15% of your total space. This can be tuned up or down as required based on future utilization patterns and needs.

2. Determine the amount of virtual space you will require for your snapshots. Use the formula found in Calculating the Virtual Size for Grid Disks.

   Note that the virtual size can be modified online by just modifying the Oracle ASM disk group without making any changes to the underlying grid disks.

3. Using the number of snapshots you plan on having at any one time and how much changes you expect to make, use the process defined in Determine the Amount of Available Space to determine which existing disk group(s) you want to shrink to release space for the sparse disk group. Based on your current allocations, you may need to shrink multiple disk groups to get the space you require for the sparse disk group. Leave a minimum of 15% free space in the original disk groups to allow for rebalancing during the resize operation.

   If there is not enough free space available in your current disk groups to provide desired space for the sparse disk group:

   - Rethink your sparse usage in this environment and downsize accordingly
   - Relocate objects and databases from this environment to another with available free space
   - Add storage

4. Once you have decided which disk group to resize, follow the steps in Shrink the Oracle ASM Disks in the Donor Disk Group and Shrink the Grid Disks in the Donor Disk Group to resize the disks.

   This process allows you to resize disks while they are online and only require a single rebalance operation per disk group. Oracle ASM may run more than one rebalance operation concurrently, depending on the number of Oracle ASM instances available in the cluster.

5. Create the sparse grid disk from the space just released using the commands outlined in Creating Sparse Grid Disks.

6. Create the sparse Oracle ASM disk group using the commands outlined in Resizing the Physical Space.

7. To monitor sparse grid disk activity see Monitoring Sparse Disk Group Activity.

8. If you find that you need to resize your sparse grid disks:

   - If more physical space is required, follow steps 1 through 4 in this section to free up space and then follow the steps in Resizing the Physical Space. The maximum physical size that can be allocated to sparse grid disks from a single cell disk is 4 TB.
   - If more virtual space is required, follow the steps in Resizing the Virtual Space. The maximum virtual size that can be allocated to sparse grid disks from a single cell disk is 100 TB.

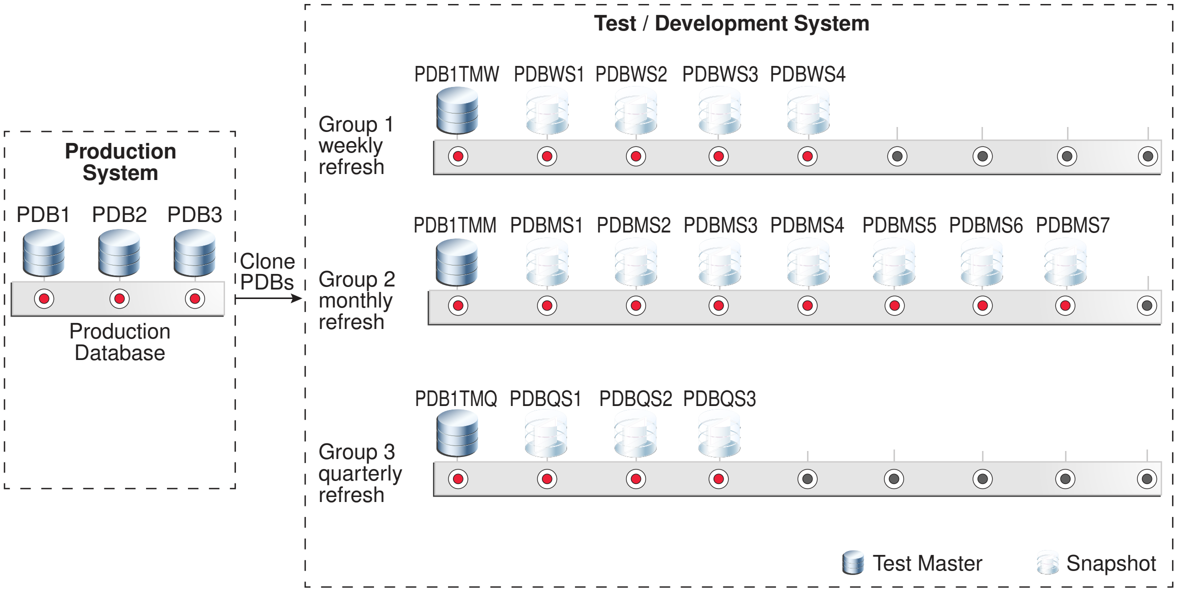
9.5 Refresh Considerations, or Lifecycle of Exadata Snapshots

Refresh cycles can influence how you use and create Exadata Snapshots.

The test master must remain in either `READONLY` state (if open) or closed while it has Exadata Snapshots associated with it. If the test master needs to be refreshed, all Exadata Snapshots dependent on the test master must be dropped and re-created.

If different groups of Exadata Snapshot users have different refresh cycle requirements, you may need to maintain multiple test masters. The following figure shows three test masters, each with its own refresh schedule.

Figure 9-11 Three Developer Groups for PDB1, Each With a Different Refresh Cycle



9.6 Using an Oracle Data Guard Standby Database as the Test Master

In a situation where the test master is either a non-container database or a full CDB that needs to be refreshed regularly (that is, the test master is not a PDB), Oracle recommends creating the test master database as an Oracle Data Guard physical standby dedicated to this purpose.

There are multiple benefits when using this approach:

- Easier refresh

  Oracle Data Guard is a proven solution for synchronizing multiple physical replicas of a production database used for disaster recovery, read-only offload from production, backup offload, and test. This same functionality can be used to maintain a copy of the production database dedicated to serving as a test master database that is more easily refreshed on a periodic basis. The benefit of using an Oracle Data Guard physical standby increases as the size of the test master database and the number of times it must be refreshed increases. These benefits far outweigh the small incremental effort required to create an Oracle Data Guard replica compared to simply cloning a database from an Oracle Recovery Manager (RMAN) backup.

- Minimal impact to primary

  During the time the Oracle Data Guard replica is being used as the test master database, Oracle Data Guard redo transport and apply are disabled. There is zero impact to the production database. When it is time to refresh, only the deltas generated since the Oracle Data Guard replica was converted to a test master database are taken from the production database and used to resynchronize the test master database.

  > **Note:**
  >
  > Since transport and apply for this Oracle Data Guard replica will be stopped while it functions as a test master, it should not be used for disaster recovery or for any purpose other than as test master. If you are already using Oracle Data Guard for high availability or disaster protection, Oracle recommends creating an Oracle Data Guard replica(s) to use as test master databases for Exadata Snapshot databases.

- Easy to scrub prior to creating snapshot clones

  Oracle Data Guard makes it easy to modify the test master database before making it available to createExadata Snapshots. For example, an Oracle Data Guard replica can be opened read-write and data can be masked or scrubbed prior to creatingExadata Snapshots. Later, when testing is complete, the test master database can be converted back into an Oracle Data Guard replica discarding any modifications made to the original copy and refreshing it using only the deltas from the production database. Note that after refreshing the Oracle Data Guard replica, you need to re-scrub the database before you can use it as a test master again.

  If you are using an RMAN backup database, and you mask or scrub the data, when you need to refresh the test master, you have to create another backup as the test master and rescrub it to make it current.

Related Topics

- Refreshing the (Read-only) Test Master Database
  To refresh a read-only test master database, it must be converted temporarily to a read-write test master.
- Creating a Physical Standby Database

9.7 Managing Exadata Snapshots

To create and manage Exadata Snapshots, you need to perform these procedures:

- Creating Sparse Grid Disks
- Creating an ASM Disk Group for the Sparse Grid Disks

- Setting Up the Test Master
  You can create the test master using one of two methods:
- Creating the Snapshot Databases from a (Read-Only) Test Master Database
  You can create an Exadata Snapshot database of a pluggable database (PDB) or of a full database.
- Refreshing the (Read-only) Test Master Database
  To refresh a read-only test master database, it must be converted temporarily to a read-write test master.
- Creating a Snapshot Database from Another Snapshot Database
- Creating SparseTest Masters from a Single Full Database Copy
  You can create multiple sparse test master from a single copy of a full database.
- Creating Sparse Test Masters for PDBs
  This procedure creates a hierarchical snapshot tree or sparse test masters manually for a pluggable database (PDB) in an Oracle Multitenant database.
- Doing a Sparse Copy

9.7.1 Creating Sparse Grid Disks

When creating sparse grid disks, you need to specify the physical size and the virtual size.

- Calculating the Physical Size for Grid Disks
- Calculating the Virtual Size for Grid Disks
- Creating a Sparse Grid Disk

9.7.1.1 Calculating the Physical Size for Grid Disks

You can use the following formula to get a rough estimate of the total physical space to set aside for a sparse ASM disk group:

```
Total physical space =
   (SUM(size of all test masters in the sparse ASM disk group) +
    SUM(approximate size of all updates to the snapshot databases))
   * ASM Redundancy
```

In the formula above, ASM redundancy takes into account ASM mirroring of extents. Exadata requires ASM redundancy set to either normal redundancy (double mirror the extents) or high redundancy (triple mirror the extents). If the sparse ASM disk group will use normal redundancy, expect to double the space used. If using high redundancy, expect to triple the space used.

For example, if you want 2 test masters in the sparse ASM disk group created with normal redundancy with a combined total space of 500 GB (250 GB each) and each test master will have 5 Exadata snapshots with the expectation that each snapshot will modify 20% of the blocks, then the total physical space that will be needed can be calculated as follows:

```
Space for 2 test masters:   2 * 250 GB =                              500 GB
Space for 5 snapshots per test master, for a total of 10 snapshots:
    10 * 250 GB * 20% =                                               500 GB
Subtotal                                                             1000 GB
Normal redundancy: 2 * 1000 GB =                                     2000 GB
```

Divide this value by the number of disks to determine the size parameter for each disk. ASM grid disks should be allocated on 16 MB boundaries. If the size parameter in MB for each grid disk is not evenly divisible by 16, adjust up to a 16 MB boundary.

Note that you should set aside additional space to use for multiple projects and through multiple iterations.

Also to accommodate any disk rebalancing operations, you should add a 15% space cushion on top of the space used for snapshots and test masters.

9.7.1.2 Calculating the Virtual Size for Grid Disks

You can use the following formula to get a rough estimate of the virtual size to assign for a sparse ASM disk group:

```
Virtual size required for sparse disks =
   (SUM(full virtual size of all Exadata snapshots) + Physical space allocated)
   * ASM Redundancy
```

To continue with the example from the previous section, you have 10 Exadata snapshots. If they were full copies of the test master, they would be 250 GB each.

The following shows the calculation for the total virtual space:

```
Full size for 5 snapshots per test master, for a total of 10 snapshots:
    10 * 250 GB =                                                    2500 GB
Size of the 2 test masters: 2 * 250 GB =                              500 GB
Subtotal                                                             3000 GB
Normal redundancy: 2 * 3000 GB =                                     6000 GB
```

Divide this value by the number of disks to determine the virtualsize parameter for each disk. The virtual size for each grid disk should be allocated on 16 MB boundaries. If the virtualSize parameter in MB for each grid disk is not evenly divisible by 16, adjust up to a 16 MB boundary.

Note that you should set aside additional space to use for multiple projects and through multiple iterations.

9.7.1.3 Creating a Sparse Grid Disk

To create the sparse ASM grid disk, log in to each cell (or use dcli) in CellCLI and run a command similar to the following, changing size values where appropriate:

```
CellCLI> create griddisk all harddisk prefix=SPARSE, size=56G, virtualsize=560G
```

This creates a grid disk of physical size 56 GB but presents to ASM as a 560 GB grid disk. The size parameter should match the actual physical size of the ASM grid disk while the virtualsize parameter should be at least the physical size of the ASM grid disk.

Attributes for the ASM grid disk created above would look like the following from the "LIST GRIDDISK DETAIL" command:

```
CellCLI> LIST GRIDDISK DETAIL
size:        56G
sparse:      TRUE
virtualSize: 560G
```

size displays the actual physical size of the grid disk.

`sparse` has a value of `TRUE`.

`virtualSize` displays the virtual size of the grid disk.

9.7.2 Creating an ASM Disk Group for the Sparse Grid Disks

After you have created the sparse grid disks, you create an ASM disk group to enable those disks to be accessible to a database on the cluster. To create a disk group, log in to an ASM instance using SQL*Plus as `sysasm` and run a command similar to the following:

```
SQL> create diskgroup SPARSE high redundancy disk 'o/*/SPARSE_*' attribute
'compatible.asm'='12.1.0.2',
'compatible.rdbms'='12.1.0.2',
'au_size'='4M',
'cell.smart_scan_capable'='true',
'cell.sparse_dg'='allsparse',
'appliance.mode' = 'TRUE';
```

`compatible.asm` must be set to 12.1.0.2 or higher.

`compatible.rdbms` must be set to 12.1.0.2 or higher.

`cell.sparse_dg` must be set to "`allsparse`". This identifies the disk group to ASM as being made up of sparse grid disks.

`appliance.mode` must be set to `true`.

9.7.3 Setting Up the Test Master

You can create the test master using one of two methods:

After creating the test master, perform the tasks described in Setting the Ownership of the Test Master Data Files.

- Create a New Test Master - Full Clone on a Disk Group with ASM ACL Enabled
- Converting an Existing Full Clone or Standby Database to a Test Master
  If you already have a full clone or a standby database that you want to repurpose as a test master, then you can convert that database to a test master.
- Setting the Ownership of the Test Master Data Files
  After you have cloned the database to create a test master database, configure permissions on the disk group and data files.

9.7.3.1 Create a New Test Master - Full Clone on a Disk Group with ASM ACL Enabled

You can create a full clone of your database using RMAN backup/restore, data pump, or any of the other methods typically used to create a full clone of a database.

After creating the full clone, remove write permissions on all the data files to help prevent accidental overwrite.

SQL commands in ASM instance only allow to set file permissions to read only. You cannot remove write permissions in SQL.

```
SQL> ALTER DISKGROUP DATA set permission owner=read ONLY, group=read ONLY,
other=none for file 'FILENAME';
```

9.7.3.2 Converting an Existing Full Clone or Standby Database to a Test Master

If you already have a full clone or a standby database that you want to repurpose as a test master, then you can convert that database to a test master.

Standby databases cannot be running redo apply while serving as a test master.

1. If you are using an Oracle Data Guard standby database, perform the following steps:
   a. For initial creation of the Oracle Data Guard replica, use the steps outlined in My Oracle Support note 1617946.1.

      The Data Guard copy must have enough redo applied that it can be opened in a READ ONLY state.
   b. If the test master is a physical standby database and you need to make any modifications to the test master, for example, deleting or masking sensitive data, then perform the following steps:
      i. Convert the standby database into a snapshot standby.

         > **Note:**
         >
         > An Oracle Data Guard snapshot standby is different from an Oracle Exadata snapshot. An Oracle Data Guard snapshot standby is a complete copy of the source database that is open read-write. Conversion to an Oracle Data Guard snapshot standby is a simple operation using a single command. Oracle Data Guard snapshot standby facilitates making modifications to the test master and refreshing it for subsequent rounds of testing. See *Oracle Data Guard Concepts and Administration* (referenced at the end of this topic) for more information on Oracle Data Guard snapshot standby databases.

      ii. Modify the standby database as required.
   c. When the standby database is at a consistent state and can be opened in `READ ONLY` mode, stop log transport to the standby and disable redo apply on the standby.

      ```
      DGMGRL> edit database TESTMASTER set property logshipping=OFF;
      Property "logshipping" updated
      ```

      If you have not converted the physical standby database into a snapshot standby, then stop redo apply.

      ```
      DGMGRL> edit database TESTMASTER set state=APPLY-OFF;
      Succeeded
      ```
2. If access control is not already enabled on the disk group that contains the test master's data files, then enable access control on the disk group.

The disk group must be on Oracle Exadata storage servers.

```
SQL> ALTER DISKGROUP DATA SET ATTRIBUTE 'ACCESS_CONTROL.ENABLED' =
'TRUE';
```

3. Grant ownership to all data files.

   See Setting the Ownership of the Test Master Data Files for details.

4. Remove write permissions on all the data files to help prevent accidental overwrite.

   SQL commands in an Oracle ASM instance only allow you to set file permissions to read only. You cannot remove write permissions in SQL.

   ```
   SQL> ALTER DISKGROUP DATA set permission owner=read ONLY, group=read
   ONLY, other=none for file 'FILENAME';
   ```

   This allows snapshots to be created and owned by users other than the owner of the base files.

Related Topics

- Creating a Physical Standby using RMAN Duplicate (RAC or Non-RAC) (My Oracle Support Doc ID 1617946.1)
- Managing a Snapshot Standby Database

9.7.3.3 Setting the Ownership of the Test Master Data Files

After you have cloned the database to create a test master database, configure permissions on the disk group and data files.

Set an operating system user as the owner of the disk group, and make the operating system user the owner of the test master's data files.

You can do this by running SQL commands manually in SQL*Plus, or by running a script:

- Running Commands Manually
  You can use SQL*Plus to manually run the commands to set the ownership of the test master data files.
- Running from a Script
  You can also set the ownership of the test master data files using a SQL script.

9.7.3.3.1 Running Commands Manually

You can use SQL*Plus to manually run the commands to set the ownership of the test master data files.

The following commands are run in SQL*Plus.

1. If the operating system user you are granting access to is not added as a user on the disk group, then add the user.

> **Note:**
>
> When enabling access control, all software owners that are running databases must be added as a user to the disk group.

For example, to add the user SCOTT as an owner of the DATA disk group, use the following command:

```
SQL> ALTER DISKGROUP DATA ADD USER 'scott';
```

2. Make the operating system user the owner of the test master's data files:

```
SQL> ALTER DISKGROUP DATA SET OWNERSHIP OWNER='scott' FOR FILE
  '+DATA/TESTMASTER/DATAFILE/system.257.865863315';

SQL> ALTER DISKGROUP DATA SET OWNERSHIP OWNER='scott' FOR FILE
  '+DATA/TESTMASTER/DATAFILE/sysaux.258.865863317';

SQL> ALTER DISKGROUP DATA SET OWNERSHIP OWNER='scott' FOR FILE
  '+DATA/TESTMASTER/DATAFILE/sysext.259.865863317';

SQL> ALTER DISKGROUP DATA SET OWNERSHIP OWNER='scott' FOR FILE
  '+DATA/TESTMASTER/DATAFILE/tbs_1.256.865863315';
```

Related Topics

- *Oracle Automatic Storage Management Administrator's Guide*

9.7.3.3.2 Running from a Script

You can also set the ownership of the test master data files using a SQL script.

The following procedure is equivalent to the commands in the previous topic, but it queries V$DATAFILE for the filenames:

1. Add an operating system user as owner of the disk group.

```
SQL> ALTER DISKGROUP DATA ADD USER 'scott';
```

2. Generate a script called set_owner.sql to set the owner of the test master's data files.

   - If the test master is a full database, run the following in the test master database:

```
set newpage 0
set linesize 999
set pagesize 0
set feedback off
set heading off
set echo off
set space 0
set tab off
set trimspool on
```

```
spool set_owner.sql
select 'ALTER DISKGROUP DATA set ownership
owner='||''''||'scott'||''''||' for file '||''''||name||''''||';'
from v$datafile;
exit
```

- If the test master is a PDB, run the following in the CDB$ROOT of the test master PDB:

  In the `select` statement below, the example assumes the test master PDB has a con_id of `10`.

  ```
  set newpage 0
  set linesize 999
  set pagesize 0
  set feedback off
  set heading off
  set echo off
  set space 0
  set tab off
  set trimspool on
  spool set_owner.sql
  select 'ALTER DISKGROUP DATA set ownership
  owner='||''''||'scott'||''''||' for file '||''''||name||''''||';' -
  from v$datafile where con_id=10;
  exit
  ```

3. Remove extra lines in `set_owner.sql`.

   ```
   sed -i '/SQL/d' set_owner.sql
   ```

4. Run the script in the ASM instance.

   ```
   SQL> @set_owner
   ```

9.7.4 Creating the Snapshot Databases from a (Read-Only) Test Master Database

You can create an Exadata Snapshot database of a pluggable database (PDB) or of a full database.

When you create the snapshot database from a test master database, the test master database becomes read-only.

The first example addresses Exadata Snapshot PDBs, the second example addresses snapshot databases (non-CDB only).

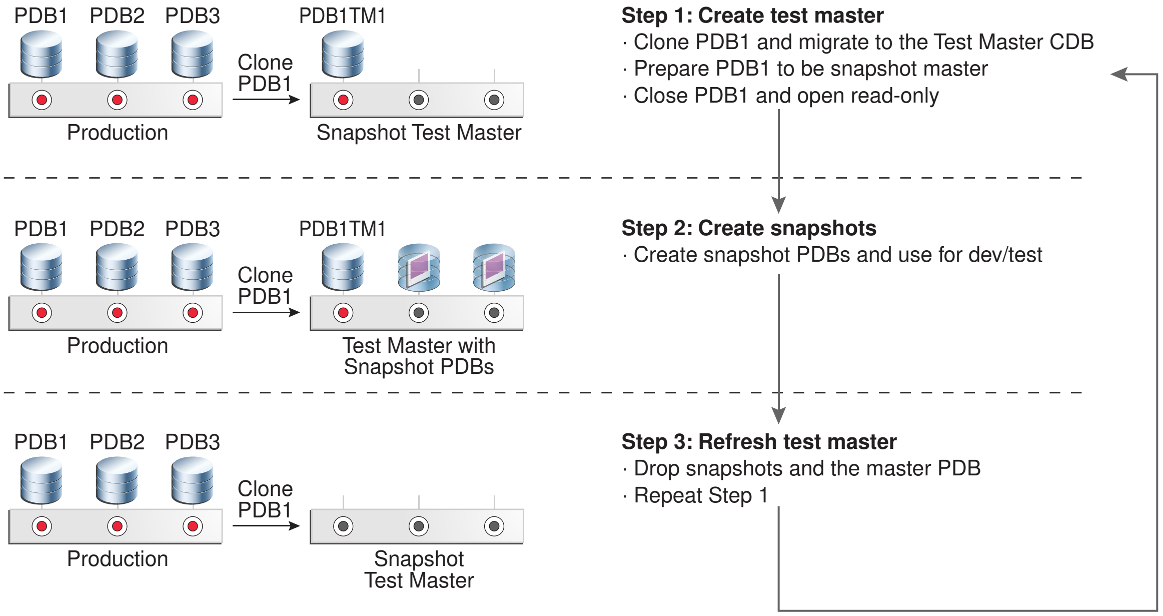
- Creating a Snapshot of a Pluggable Database
- Creating a Snapshot of a Full Database
  You create an Exadata snapshot of a full database if your test master database is a non-CDB.

9.7.4.1 Creating a Snapshot of a Pluggable Database

Creating an Exadata snapshot of a pluggable database (PDB) is the simplest method for creating a snapshot because it requires no additional manual steps. Two new clauses to the CREATE PLUGGABLE DATABASE statement were added to identify the PDB as an Exadata snapshot. The snapshot creation process alters the permissions on the files on the test master PDB to READONLY to ensure that it is not modified.

Creating individual Exadata snapshot PDBs is best used when creating snapshots for a smaller number of PDBs within a given CDB. The following figure shows a high-level example of a typical lifecycle for a PDB with two PDB snapshots.

Figure 9-12 Lifecycle of Exadata Snapshots Using PDBs


PDB1
PDB2
PDB3
PDB1TM1
Clone PDB1
Production
Snapshot Test Master
Step 1: Create test master
· Clone PDB1 and migrate to the Test Master CDB
· Prepare PDB1 to be snapshot master
· Close PDB1 and open read-only
PDB1
PDB2
PDB3
PDB1TM1
Clone PDB1
Production
Test Master with Snapshot PDBs
Step 2: Create snapshots
· Create snapshot PDBs and use for dev/test
PDB1
PDB2
PDB3
Clone PDB1
Production
Snapshot Test Master
Step 3: Refresh test master
· Drop snapshots and the master PDB
· Repeat Step 1

One of the benefits of Oracle Multitenant and PDBs is the ability to easily clone an existing PDB to create a test master and move it from one CDB to another. Oracle recommends that you clone your source PDB to create the test master and then migrate it to your test environment where you can perform any data scrubbing that may be needed. Once complete, the test master PDB can be used to create any number of Exadata snapshot PDBs.

NOTE: An Exadata snapshot PDB is created as a PDB in the CDB. You are limited to a total of 252 PDBs in a single CDB. All PDBs in that CDB, whether they are test masters, snapshots or PDBs that are not being used for snapshots, are included in that limit. If you need to create more PDBs than can fit in a single CDB, you can create the Exadata snapshot PDB in another CDB on the same cluster.

After creating the test master PDB, perform the following steps to create an Exadata snapshot PDB:

1. In SQL*Plus connect to the cdb$root.
2. Close the test master PDB in all instances.

   ```
   SQL> alter pluggable database PDB1TM1 close instances=all;
   ```

3. Open the test master PDB in the local instance in read only mode.

   ```
   SQL> alter pluggable database PDB1TM1 open read only;
   ```

4. Create an Exadata snapshot PDB of the test master.

   ```
   SQL> create pluggable database PDB1S1 from PDB1TM1 tempfile reuse
   create_file_dest='+SPARSE' snapshot copy;
   ```

   `create_file_dest` must specify the name of a sparse disk group; this ensures the files for the Exadata snapshot are created in the correct disk group. The "`snapshot copy`" clause creates the PDB as a snapshot rather than a full PDB clone.

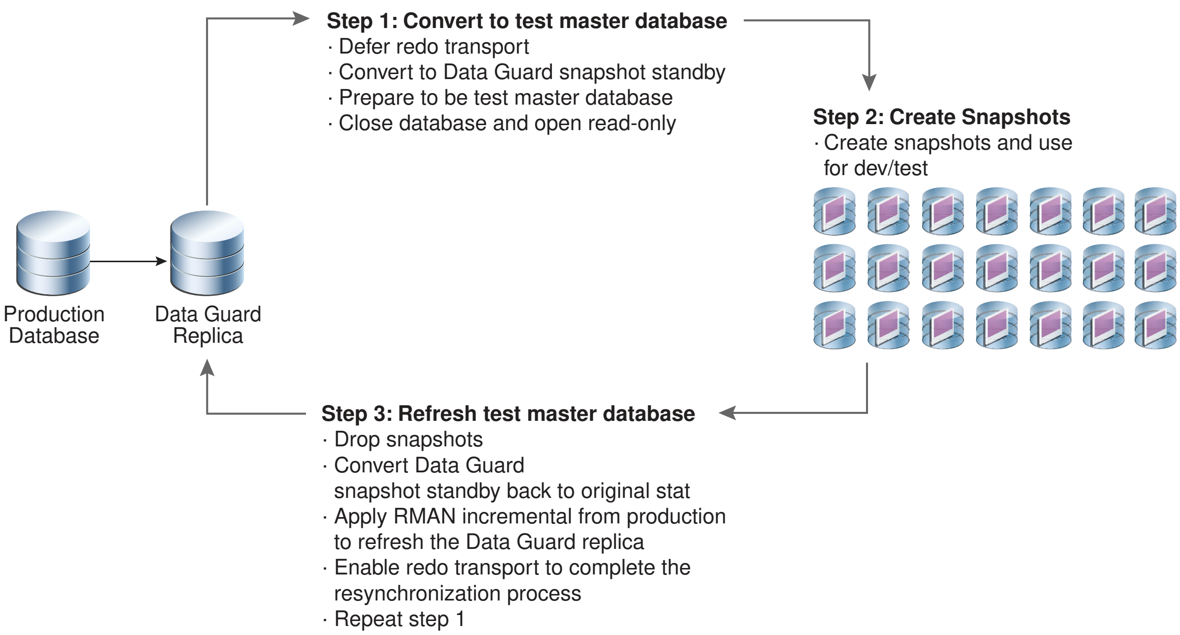
   NOTE: When an Exadata snapshot PDB is created, the command changes file permissions on the test master PDB, marking the files as READONLY in ASM.

9.7.4.2 Creating a Snapshot of a Full Database

You create an Exadata snapshot of a full database if your test master database is a non-CDB.

The following figure shows the lifecycle of an Exadata test master database and snapshot databases.

Figure 9-13 Lifecycle of Test Master and Snapshot Databases


Step 1: Convert to test master database
· Defer redo transport
· Convert to Data Guard snapshot standby
· Prepare to be test master database
· Close database and open read-only
Step 2: Create Snapshots
· Create snapshots and use for dev/test
Production Database
Data Guard Replica
Step 3: Refresh test master database
· Drop snapshots
· Convert Data Guard snapshot standby back to original stat
· Apply RMAN incremental from production to refresh the Data Guard replica
· Enable redo transport to complete the resynchronization process
· Repeat step 1

Note that the test master database cannot be used as a failover target to provide high availability or disaster recovery (a Data Guard configuration may have multiple replicas that can each serve different purposes). Similar to test master PDBs, test master databases cannot be modified while Exadata snapshots exist against them.

The test master database cannot be a read-only physical standby database that is in recovery mode (for example, Active Data Guard in Real Time Apply).

The test master database and their Exadata snapshots must be in the same Oracle ASM cluster environment.

> **Note:**
>
> If the test master is an Oracle RAC database, you have to do one of the following:
>
> - Create redo logs for all threads in the `CREATE CONTROLFILE` statement for the sparse clone, OR
> - Specify Oracle Managed Files for redo logs using the `ONLINE_LOG_CREATE_DEST_1` initialization parameter in the SPFILE of the sparse clone to have the redo logs created automatically.

1. In the test master database, create a sample control file script to use for your Exadata snapshot databases by backing up the existing control file to trace.

   Connect to the test master database via SQL*Plus as SYSDBA and do the following:

   a. Determine name and location of any trace file to be generated by your session:

   ```
   SQL> SELECT value FROM v$diag_info WHERE name = 'Default Trace
   File';

   VALUE
   ---------------------------------------------------------------------
   -------------
   /u01/app/oracle/diag/rdbms/TESTMASTER/TESTMASTER1/trace/
   TESTMASTER1_ora_26756.trc
   ```

   b. Run the BACKUP CONTROLFILE TO TRACE command to place the CREATE CONTROLFILE command into the trace file.

   ```
   SQL> ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
   ```

   c. Retrieve the file shown for the Default Trace File.

2. In the test master database, determine the existing file names for the rename that will happen in step 10.

   Log into SQL*Plus as SYSDBA and run the following:

   ```
   SET newpage 0
   SET linesize 999
   SET pagesize 0
   SET feedback off
   SET heading off
   SET echo off
   SET space 0
   ```

```
SET tab off
SET trimspool on
SPOOL rename_files.sql
SELECT 'EXECUTE dbms_dnfs.clonedb_renamefile -
('||''''||name||''''||','||''''||REPLACE(REPLACE(REPLACE(name,'.','_'),-
'TESTMASTER','JOHNTEST'),'+DATA','+SPARSE')||''''||');' FROM v$datafile;
EXIT
```

The previous query builds a file called `rename_files.sql` that contains statements for each data file similar to the following:

```
EXECUTE dbms_dnfs.clonedb_renamefile (
'+DATA/TESTMASTER/DATAFILE/system.257.865863315',
'+SPARSE/JOHNTEST/DATAFILE/system_257_865863315');
```

The REPLACE function in the query:

- Replaces periods in the original file name with underscores
- Replaces the original database name of TESTMASTER with JOHNTEST
- Replaces the original disk group name of +DATA with +SPARSE

3. Shut down the test master.

```
SQL> shutdown;
```

4. Create an `init.ora` file for the Exadata snapshot database.

   You can use the `init.ora` file of the test master as a template, but make sure to change the `db_name` and `control_files` entries. For this procedure, the `init.ora` file for the Exadata snapshot database is referenced as `administrator` in commands and examples.

```
$ cp init_TestMaster.ora snap_init.ora
```

   Modify `administrator` with the new database name, new control file name, and audit file destination, for example:

```
db_name = JohnTest
control_files = '+DATA/JOHNTEST/control1.f'
audit_file_dest=/u01/app/oracle/admin/johntest/adump
```

5. Edit the trace file generated in step 1.

   Modify the trace file to create a control file for the Exadata snapshot database and create a SQL file called `crt_ctlfile.sql`. You will use this script later in step 9.

   The control file should be created with the Exadata snapshot database name, new log file names, and the data file names of the test master.

   The example below shows a control file creation script. `JohnTest` is the Exadata snapshot database. The `LOGFILE` lines specify the new log file locations, and the `DATAFILE` lines specify the data file locations for the test master database. The new log files can be in any disk group that has enough space, but they should not be created in the sparse Oracle ASM disk group.

```
SQL> CREATE CONTROLFILE REUSE SET DATABASE JohnTest RESETLOGS ARCHIVELOG
      MAXLOGFILES 32
      MAXLOGMEMBERS 2
      MAXINSTANCES 1
      MAXLOGHISTORY 908
  LOGFILE
      GROUP 1 '+DATA/JOHNTEST/t_log1.f' SIZE 100M BLOCKSIZE 512,
      GROUP 2 '+DATA/JOHNTEST/t_log2.f' SIZE 100M BLOCKSIZE 512
  DATAFILE
      '+DATA/TESTMASTER/DATAFILE/system.257.865863315',
      '+DATA/TESTMASTER/DATAFILE/sysaux.258.865863317',
      '+DATA/TESTMASTER/DATAFILE/sysext.259.865863317',
      '+DATA/TESTMASTER/DATAFILE/tbs_1.256.865863315'
  CHARACTER SET WE8DEC;
```

6. Create the `audit_file_dest` directory on all nodes on which the snapshot will be running.

   ```
   $ mkdir -p /u01/app/oracle/admin/johntest/adump
   ```

7. Create the directories in Oracle ASM for the snapshot data files.

   Use ASMCDM to execute the following commands:

   ```
   $ asmcmd -p
   ASMCMD > cd +SPARSE
   ASMCMD [+sparse] > mkdir JOHNTEST
   ASMCMD [+sparse] > cd JOHNTEST
   ASMCMD [+sparse/johntest] > mkdir DATAFILE
   ```

8. Start a database instance pointing to the Exadata snapshot database `init.ora` file using the following commands:

   ```
   $ sqlplus / as sysdba
   SQL> startup nomount pfile=snap_init.ora
   ```

9. Create the Exadata snapshot control file using the script created in step 5.

   In the following example the script is named `crt_ctlfile.sql`.

   ```
   SQL> @crt_ctlfile
   ```

10. Run the script you modified in step 2.

    All the files must be renamed prior to opening the Exadata snapshot database.

    Connect using SQL*Plus as SYSDBA to the Exadata snapshot database and run the following command:

    ```
    SQL> @rename_files
    ```

This script modifies the permissions of the test master database files stored in Oracle ASM, marking them as READONLY.

The `dbms_dnfs.clonedb_renamefile` procedure, which is called by `rename_files.sql`, sets up the parent-child relationship between the test master database and the snapshot database, and renames the file names in the snapshot database's control file.

11. Open the Exadata snapshot database with the RESETLOGS option:

    ```
    SQL> ALTER DATABASE OPEN RESETLOGS;
    ```

12. Confirm that the Exadata snapshot files are child files of the test master database. Connect using SQL*Plus as SYSASM to the Exadata snapshot, and run the following command:

    ```
    SQL> SELECT filenumber num, clonefilename child, snapshotfilename parent
    FROM x$ksfdsscloneinfo;
    ```

    The following is an example of the output from the query:

    ```
    NUM  CHILD
    ---- ---------------------------------------------
    PARENT
    -----------------
    1     +SPARSE/JOHNTEST/DATAFILE/system_257_865863315
    +DATA/TESTMASTER/DATAFILE/system.257.865863315

    2     +SPARSE/JOHNTEST/DATAFILE/sysaux_258_865863317
    +DATA/TESTMASTER/DATAFILE/sysaux.258.865863317

    3     +SPARSE/JOHNTEST/DATAFILE/sysext_259_865863317
    +DATA/TESTMASTER/DATAFILE/sysext.259.865863317

    4     +SPARSE/JOHNTEST/DATAFILE/tbs_1_256_865863315
     +DATA/TESTMASTER/DATAFILE/tbs_1.256.865863315
    ```

13. Log in using SQL*Plus to the Exadata snapshot database, and add temp files to the TEMP tablespace. This is a full size temp file, not a sparse temp file.

    ```
    SQL> ALTER TABLESPACE temp ADD TEMPFILE '+DATA' SIZE 10G;
    ```

9.7.5 Refreshing the (Read-only) Test Master Database

To refresh a read-only test master database, it must be converted temporarily to a read-write test master.

To refresh the (read-only) test master database, the main steps are:

- Drop the Snapshot Databases
  Delete the Exadata snapshot databases that are children of the test master database you want to refresh.

- Change the Permissions on the Test Master to Read-Write
  To modify the data files from read-only to read-write you can use SQL to generate a script of SQL commands.
- Convert the Test Master Database Back to a Data Guard Replica
  Now that the data files for the test master are writable, convert the read-only test master database into an Oracle Data Guard replica.
- Update the Test Master Database
- Close the Test Master and Make All Test Master Data Files Read-Only
  After the Test Master has been updated, you can revert it to a read-only test master.
- Re-create All Snapshots
  After the Test Master has been updated and made read-only again, re-create all the snapshot databases to get the latest updates.

9.7.5.1 Drop the Snapshot Databases

Delete the Exadata snapshot databases that are children of the test master database you want to refresh.

You can delete the snapshot databases using RMAN.

1. Connect to the Exadata snapshot database using RMAN with the Exadata snapshot as the target.

   Connect as a user that has the necessary privileges to start and drop a database, such as SYSBACKUP or SYSDBA.

   ```
   RMAN> CONNECT TARGET "user@snapdb_name AS SYSDBA"
   ```

2. Start the snapshot database in restricted mode.

   ```
   RMAN> STARTUP FORCE MOUNT DBA
   ```

3. Delete the snapshot database.

   ```
   RMAN> DROP DATABASE;
   ```

> **Note:**
>
> Failure to drop an Exadata snapshot database will have no impact on the state of a test master database. However, an Exadata snapshot can behave unpredictably if its test master database is dropped or refreshed.

9.7.5.2 Change the Permissions on the Test Master to Read-Write

To modify the data files from read-only to read-write you can use SQL to generate a script of SQL commands.

Before starting this part of refreshing the read-only test master database, you must first drop the snapshot databases.

1. After all Exadata snapshot databases have been deleted, start up the test master database in mount mode.
2. Create a script to reset permissions on the data files for the test master database.

   Connect to the test master using SQL*Plus and run the following script to create a new SQL script that contains the commands to reset permissions on the data files belonging to the test master database.

   ```
   set newpage 0
   set linesize 999
   set pagesize 0
   set feedback off
   set heading off
   set echo off
   set space 0
   set tab off
   set trimspool on
   spool change_perm.sql
   select 'ALTER DISKGROUP DATA set permission owner=read write,
   group=read write, other=none for file '||''''||name||''''||';' from
   v$datafile;
   exit
   ```

3. Remove the extra lines from the generated script.

   Run the following `sed` command to remove extra lines from the `change_perm.sql` script that you created in the previous step. Run this command from an operating system prompt, in the directory that contains the `change_perm.sql` script.

   ```
   $ sed -i '/SQL/d' change_perm.sql
   ```

4. Use the generated script to change the file permissions.

   Use SQL*Plus to connect to an Oracle ASM instance as a SYSASM user. Run the `change_perm.sql` script. This script changes the permissions of the test master's data files to make them writable.

   ```
   SQL> @change_perm
   ```

9.7.5.3 Convert the Test Master Database Back to a Data Guard Replica

Now that the data files for the test master are writable, convert the read-only test master database into an Oracle Data Guard replica.

If you had originally prepared the test master database using Oracle Data Guard snapshot standby, then convert it back to its original state as an Oracle Data Guard replica using the `CONVERT` command. This command discards any changes previously made to the replica to prepare it to be the test master. It also makes it possible to refresh the test master using just incremental changes from the source database instead of a complete restore from a current backup.

9.7.5.4 Update the Test Master Database

You have two options for refreshing the test master database:

- Allow Oracle Data Guard to refresh the test master database

  If the Oracle Data Guard replica has been used as a test master database for only a short period of time and you have all the redo generated during this time in archive logs on disk at the source database, then you can enable redo shipping and start redo apply. The test master database will use regular Oracle Data Guard protocols to retrieve archive logs and apply the logs until it is caught up with the primary database. Once the Oracle Data Guard replica is as current as you need it to be, disable redo shipping, stop redo apply and repeat the test master and snapshot creation cycle described in Setting Up the Test Master and Creating the Snapshot Databases from a (Read-Only) Test Master Database.

  This option has the benefit of being able to stop redo apply at some intermediate point rather than bringing the test master database totally current.

  To let Oracle Data Guard refresh the standby, enable log shipping to the standby and redo apply on the standby:

  ```
  DGMGRL> edit database TESTMASTER set property logshipping=ON;
  Property "logshipping" updated
  DGMGRL> edit database TESTMASTER set state=apply-on;
  Succeeded
  ```

- Use RMAN `RECOVER...FROM SERVICE` to roll forward the test master database

  If the Oracle Data Guard replica has been used as a test master database for a long period of time or if you no longer have the redo available on disk to enable Oracle Data Guard to automatically refresh the test master database, use RMAN to perform live incremental apply over the network.

  A major advantage to using this method is that no additional disk space is required. RMAN will bring changed blocks to the standby from the primary over the network and apply them directly. Also RMAN greatly simplifies the process by determining which blocks need to be retrieved based on the SCN of the data files on the test master. With this method you cannot recover to an intermediate point in time; the refresh will bring the test master database current with the primary. For more information on this method refer to Performing RMAN Recovery: Advanced Scenarios in *Oracle Database Backup and Recovery User's Guide*.

To refresh the test master database using RMAN Network Incrementals:

1. Prepare Oracle Net Services for the RMAN connections.

   These steps need to be performed only once.

   a. Create a `listener.ora` entry for the test master database (the Oracle Data Guard replica).

      The listener entry allows RMAN to connect to the target using the SID because the service is not started when the database is opened in `NOMOUNT` mode. The following is an example of the entry:

      ```
      SID_LIST_LISTENER =
        (SID_LIST =
          (SID_DESC =
            (SID_NAME = TESTMASTER1)
            (ORACLE_HOME = /u01/app/oracle/product/12.1.0.2/dbhome_1)
          )
        )
      ```

b. Reload the listener to pick up the changes to the `listener.ora`.

```
$ lsnrctl reload listener
```

c. Create a TNS entry on the test master environment pointing to the SID of the local test master instance.

   The entry should use the local host name rather than the SCAN name to ensure the connection request goes to the correct host.

```
TESTMASTER1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = standbydb01.example.com)
(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = TESTMASTER1)
      (UR=A)
    )
  )
```

2. Connect via RMAN to the test master database and save the `CURRENT_SCN` for later.

   This value will be used to determine if newly created files since the last refresh need to be restored from the source database.

```
RMAN> select current_scn from v$database;
CURRENT_SCN#
------------------
          17081990
```

3. List the names and group identifiers of the redo log files.

   The names of the online redo log files and standby redo log files of the Oracle Data Guard replica might be required in a later step.

```
RMAN> SELECT type, group#, member FROM v$logfile;
```

4. Refresh the standby control file of the Oracle Data Guard replica from the source database to make the control file current.

   a. Reconnect to the Oracle Data Guard replica as the RMAN target.

   b. Restart the target in `NOMOUNT` mode.

```
RMAN> startup nomount force;
```

   c. Restore the standby control file by using the control file on the source database.

      The following example restores the control file on the Oracle Data Guard replica by using the database control file from `SOURCEMASTER`, the source database.

```
RMAN> RESTORE STANDBY CONTROLFILE FROM SERVICE SOURCEMASTER;
```

   d. Mount the Oracle Data Guard replica.

   ```
   RMAN> ALTER DATABASE MOUNT;
   ```

5. Update the names of the data files and the temp files in the standby control file.

   If you are not using an RMAN catalog, the names of files in the standby control file are the names that were used in the source database, not the standby.

   Use the `CATALOG` command and the `SWITCH` command to update all the data file names. The `SWITCH` command will be used after restoring any newly created files from the source database in step 7.

   In the following example, `+DATA/TESTMASTER/DATAFILE/` is the location of the data files on the Oracle Data Guard replica. All data files must be stored in this location.

   ```
   RMAN> CATALOG START WITH '+DATA/TESTMASTER/DATAFILE/';
   ```

6. Determine if new files were added that need to be restored from the source database.

   Use the `CURRENT_SCN` from step 2.

   ```
   RMAN> SELECT file# FROM v$datafile WHERE creation_change# >= 17081990;
    FILE#
   ----------
            9
           10
   ```

7. If there are files returned by the previous query, restore those data files from the source database.

   Execute an RMAN `RUN` block similar to the following using the list of `FILE#` values returned by the previous step. If no `FILE#` values were returned, then skip this step.

   ```
   RMAN> run{
   2> set newname for database to '+DATA';
   3> restore datafile 9,10 from service SOURCEMASTER;
   4> }
   ```

8. If not using an RMAN catalog, rename the data files in the standby control file.

   Switch to the copies cataloged in step 5.

   ```
   RMAN> SWITCH DATABASE TO COPY;
   ```

9. Update the names of the online redo logs and standby redo logs in the standby control file.

   Use one of the following methods:

   - Use the `ALTER DATABASE CLEAR` command to clear the log files in all redo log groups of the Oracle Data Guard replica. RMAN then recreates all the standby redo logs and the online redo log files.

Note:

Clearing log files is recommended only if the Oracle Data Guard replica does not have access to the online redo log files and standby redo log files of the source database. If the Oracle Data Guard replica has access to the redo log files of the source database and the redo log file names of the source database are OMF names, then the `ALTER DATABASE` command will delete log files on the source database.

Also, the clearing of the log files will create new log files. Any existing log files are not used because the control file is not aware of those existing files. To conserve space, delete the existing log files from Oracle ASM prior to running the `ALTER DATABASE CLEAR` commands.

The `GROUP#` column of the `V$LOGFILE` view queried in step 5 provides the redo log group identifiers of the log groups that must be cleared. Use separate `ALTER DATABASE CLEAR` commands to clear each redo log group.

For example, the following command clears the redo log group with identifier 2.

```
SQL> ALTER DATABASE CLEAR LOGFILE GROUP 2;
```

- Use the `ALTER DATABASE RENAME FILE` command to rename the redo log files. Use a separate command to rename each log file listed in step 5.

  To rename log files, the `STANDBY_FILE_MANAGEMENT` initialization parameter must be set to `MANUAL`. Renaming log files is recommended when the number of online redo logs files and standby redo log files is the same in the source database and the Oracle Data Guard replica.

10. Use RMAN `RECOVER....FROM SERVICE` to roll forward the data files to current state.

    No additional space is required for this operation. Note that this process can only bring the files totally current; it cannot bring the files to a previous point in time. Connect via RMAN to the Oracle Data Guard replica as target using the TNS entry created in step 3. The service specified should point to the primary.

```
RMAN> recover database noredo from service SOURCEMASTER;
```

11. Enable redo shipping to the Oracle Data Guard replica and start redo apply.

    This is necessary to update the control file with the blocks applied as part of step 10.

```
DGMGRL> edit database TESTMASTER set property logshipping=ON;
Property "logshipping" updated
DGMGRL> edit database TESTMASTER set state=apply-on;
Succeeded.
```

12. After redo has been applied, repeat the process you used to convert the Oracle Data Guard replica into a test master database and then create Exadata database snapshots.

    Remember to once again disable log shipping and redo apply at the standby.

9.7.5.5 Close the Test Master and Make All Test Master Data Files Read-Only

After the Test Master has been updated, you can revert it to a read-only test master.

Complete one of the tasks described in "Setting the Ownership of the Test Master Data Files".

9.7.5.6 Re-create All Snapshots

After the Test Master has been updated and made read-only again, re-create all the snapshot databases to get the latest updates.

You can create an Exadata snapshot database of a pluggable database (PDB) or of a full database as described in "Creating the Snapshot Databases from a (Read-Only) Test Master Database."

9.7.6 Creating a Snapshot Database from Another Snapshot Database

To create a snapshot from a snapshot:

1. Create a first level snapshot. In the following example, the snapshot is called PDB1S1.

```
create pluggable database PDB1S1
  from PDB1TM1
  create_file_dest='+SPARSE'
  snapshot copy;
```

2. Open and close the PDB, so you can re-open it as read-only in the next step.

```
alter pluggable database PDB1S1 open;
alter pluggable database PDB1S1 close;
```

3. Open the PDB in read-only mode so it can serve as a test master.

```
alter pluggable database PDB1S1 open read only;
```

4. Create a snapshot from the snapshot created in step 1. In the following example, the second level snapshot is called PDB1S1_A.

```
create pluggable database PDB1S1_A
  from PDB1S1
  create_file_dest='+SPARSE'
  snapshot copy;

alter pluggable database PDB1S1_A open;
```

Related Topics

- Hierarchical Snapshot Databases
  Hierarchical snapshots enable you to create snapshot databases from other snapshot databases.

9.7.7 Creating SparseTest Masters from a Single Full Database Copy

You can create multiple sparse test master from a single copy of a full database.

The source for the test master is a full copy of a Data Guard physical standby database. This standby database should not be used as a target for switchover or failover; it should only be used as the source for the test masters defined in this process.

This process takes advantage of the hierarchical snapshot functionality to allow redo to be shipped and applied, keeping the standby database current with production while also providing files to be used as source for test masters to be used by Exadata storage snapshots. The physical standby database begins as a full copy of the primary database. When you are ready to create storage snapshots, sparse data files are created pointing to the full database files to apply redo shipped from the primary. These sparse files are then used in the standby database instance to apply redo. You can also open the sparse data files in Active Data Guard mode to supply read only access of current data.

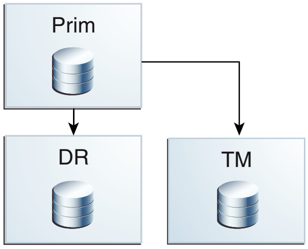
When additional snapshots are required at different points in time, you repeat the process of creating new sparse files on top of the previously created sparse files to apply redo and keep the data current. This allows you to use a single full copy of the data files to use as multiple test masters from different points in time. Also, you can create a new test master in a matter of minutes because you do not have to drop the existing snapshots.

Restrictions

This process can be used for only non-container databases.

The following tasks assume that a physical standby database has already been created to be used as the source for the test masters.

Figure 9-14 Starting Configuration



- Task 1: Prepare the Standby Database to Be Used as a Sparse Test Master
- Task 2: Configure the Sparse Test Master and Sparse Files on the Standby Site
  In this task you convert the standby into a test master and create sparse files to receive and apply redo from the primary database.
- Task 3: Create Full Database Snapshots Using the New Sparse Test Master
  Create full snapshots using the new sparse test master.

- Task 4: Create a New Sparse Test Master Using a Previously Created Sparse Test Master
  Create a new set of snapshots to provide a new test master and new redo apply files.

Related Topics

- Using an Oracle Data Guard Standby Database as the Test Master
  In a situation where the test master is either a non-container database or a full CDB that needs to be refreshed regularly (that is, the test master is not a PDB), Oracle recommends creating the test master database as an Oracle Data Guard physical standby dedicated to this purpose.

> **See Also:**
>
> "Creating a PDB in a Primary Database" in *Oracle Data Guard Concepts and Administration*

9.7.7.1 Task 1: Prepare the Standby Database to Be Used as a Sparse Test Master

The existing files for the standby database are used to support snapshots. You create a series of sparse files pointing to the existing files of the standby. Redo received from the primary database is applied to these files. These sparse files allow the standby database to be used as a sparse test master and also kept current with the primary database.

1. Stop redo apply at the standby.

   To ensure that the structure of the database is at a quiesced state for creating the supporting files to build snapshots, redo apply should be turned off at the standby database.

   ```
   DGMGRL> edit database tm_standby set state='APPLY-OFF';
   ```

2. Prepare the current standby database data files to be used as a test master.

   Data files must meet the pre-requisites to be used as a test master:

   a. The disk group the files reside on must have the `access_control.enabled` attribute set to `TRUE`.

      This example sets the attribute on the +DATA disk group. This step only needs to be done one time for the disk group.

      As SYSASM, log into an Oracle ASM instance using SQL*Plus and issue the following command:

      ```
      SQL> alter diskgroup DATA set attribute
      'ACCESS_CONTROL.ENABLED'='TRUE';
      ```

   b. The database owner user must be added as an explicit user of the disk group the files reside on.

      This example grants access to the +DATA disk group to the user SCOTT. This step only needs to be done one time per user per disk group.

Using SQL*Plus while connected as the SYSASM user, issue the following command:

```
SQL> alter diskgroup DATA add user 'scott';
```

c. The files that going to be used must have explicit permissions granted to the database owner user.

   You must perform this step for all users that will be creating snapshots using these files and for all files that will be referenced by the snapshot. The following script can be used to build SQL statements to perform the set ownership. Run the script while connected to the standby database using SQL*Plus. If the standby is a contained database (CDB) you must be connected to the cdb$root container:

```
set newpage 0
set linesize 999
set pagesize 0
set feedback offset heading off
set echo off
set space 0
set tab off
set trimspool on
spool set_owner.sql
select 'ALTER DISKGROUP DATA set ownership
owner='||''''||'scott'||''''||'
 for file '||''''||name||''''||';' from v$datafile;
exit
```

   After building the statement, login to an Oracle ASM instance using SQL*Plus as the SYSASM user, and execute the `set_owner.sql` script.

```
SQL> @set_owner
```

3. Create a backup of the controlfile.

   All snapshots are created using the current state of the standby database, so they need to know all of the files that make up the standby. Create a binary backup of the control file to allow future creation of the CREATE CONTROLFILE script required for additional snapshots

```
SQL> ALTER DATABASE BACKUP CONTROLFILE TO '/home/oracle/snap_tm/
control_monday_tm.ctl';
```

4. Create the `rename_files.sql` script to create the sparse data files for the snapshot.

   This script builds a series of RENAME statements to create the sparse data files to be used for the snapshot to apply redo received from the primary. Use a SQL statement similar to the following. Note that this statement uses the same directory structure as the original files, but the files will be created in the SPARSE disk group. The new file names will be created replacing '_' (underscores) with '.' (periods).

```
set newpage 0
set linesize 999
set pagesize 0
set feedback off
set heading off
set echo offset space 0
set tab off
set trimspool on
spool rename_files.sql
select 'EXECUTE dbms_dnfs.clonedb_renamefile ('||''''||name||''''||
','||''''||replace(replace(name,'.','_'),'DATA/','SPARSE/')||''''||
');' from v$datafile;
exit
```

This script produces output similar to the following:

```
EXECUTE dbms_dnfs.clonedb_renamefile ('+DATA/TM_STANDBY/DATAFILE/system.
515.9304
75939','+SPARSE/TM_STANDBY/DATAFILE/system_515_930475939');

EXECUTE dbms_dnfs.clonedb_renamefile ('+DATA/TM_STANDBY/
429CE0836E0166ACE05382C8
E50A1154/DATAFILE/system.567.930475945','+SPARSE/TM_STANDBY/
429CE0836E0166ACE053
82C8E50A1154/DATAFILE/system_567_930475945');

EXECUTE dbms_dnfs.clonedb_renamefile ('+DATA/TM_STANDBY/DATAFILE/sysaux.
571.9304
75939','+SPARSE/TM_STANDBY/DATAFILE/sysaux_571_930475939');

EXECUTE dbms_dnfs.clonedb_renamefile ('+DATA/TM_STANDBY/
429CE0836E0166ACE05382C8
E50A1154/DATAFILE/sysaux.516.930475945','+SPARSE/TM_STANDBY/
429CE0836E0166ACE053
82C8E50A1154/DATAFILE/sysaux_516_930475945');

EXECUTE dbms_dnfs.clonedb_renamefile ('+DATA/TM_STANDBY/DATAFILE/
undotbs1.497.93
0475939','+SPARSE/TM_STANDBY/DATAFILE/undotbs1_497_930475939');

EXECUTE dbms_dnfs.clonedb_renamefile ('+DATA/TM_STANDBY/
429CE0836E0166ACE05382C8
E50A1154/DATAFILE/undotbs1.564.930475945','+SPARSE/TM_STANDBY/
429CE0836E0166ACE0
5382C8E50A1154/DATAFILE/undotbs1_564_930475945');
```

5. Using ASMCMD, create directories for all of the directories identified in the script `rename_files.sql`.

   When the `dbms_dnfs_clonedb_renamefile` command is executed, it requires that all directory structures used for the files already exist in ASM. Use the output from the previous step to determine the structures required and then create them as needed. You can use ASMCMD to create the directories as in the following example:

```
cd ASMCMD [+] > cd sparse
ASMCMD [+sparse] > ls
ASMCMD [+sparse] > mkdir tm_standby
ASMCMD [+sparse] > cd tm_standby
ASMCMD [+sparse/tm_standby] > mkdir datafile
ASMCMD [+sparse/tm_standby] > mkdir 429DC0E1BCBD1B90E05382C8E50A8E80
ASMCMD [+sparse/tm_standby] > mkdir 429CE0836E0166ACE05382C8E50A1154
ASMCMD [+sparse/tm_standby] > cd 429DC0E1BCBD1B90E05382C8E50A8E80
ASMCMD [+sparse/tm_standby/429DC0E1BCBD1B90E05382C8E50A8E80] > mkdir 
datafile
ASMCMD [+sparse/tm_standby/429DC0E1BCBD1B90E05382C8E50A8E80] > cd ../
429CE0836E0166ACE05382C8E50A1154
ASMCMD [+sparse/tm_standby/429CE0836E0166ACE05382C8E50A1154] > mkdir 
datafile
```

9.7.7.2 Task 2: Configure the Sparse Test Master and Sparse Files on the Standby Site

In this task you convert the standby into a test master and create sparse files to receive and apply redo from the primary database.

> **Note:**
>
> During this process you will not be creating a full snapshot database, you will be using portions of the existing standby database and adding sparse data files to the standby. The standby database controlfile will be modified to use the sparse files that are added. Going forward, the same standby instance is used, but redo apply will use the sparse files to store changes, leaving the original standby data files to serve as a sparse test master files for snapshots.

The existing data files are used to support full database snapshots with data as of the point in time the process was run.

1. Shutdown all instances of the TM_STANDBY database.

   ```
   $ srvctl stop db -d tm_standby -o abort
   ```

2. Using SQL*Plus, start one of the TM_STANDBY instances in mount mode.

   ```
   SQL> startup mount
   ```

3. Change the DB_CREATE_FILE_DEST setting in the standby instance to point to the SPARSE disk group.

This ensures all new data files that are created will reside in the SPARSE disk group. To perform this step you must disable standby file management.

```
SQL> alter system set standby_file_management='MANUAL';
SQL> alter system set db_create_file_dest='+SPARSE';
```

4. Run the `rename_files.sql` script created in Task 1 against the standby database.

   Running the script renames the data files in the TM_STANDBY controlfile and creates the sparse files for the snapshot.

   ```
   SQL> @rename_files
   ```

5. Re-enable standby file management.

   Completing this step ensures that all new data files added to the primary will automatically be created by the standby when it receives the redo to create the data file.

   ```
   SQL> alter system set standby_file_management='AUTO';
   ```

6. Enable redo apply on TM_STANDBY.

   Completing this step applies redo to the snapshot, keeping it current and preparing for the next round of snapshot creation.

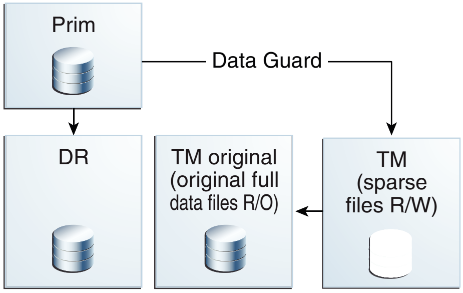
   ```
   DGMGRL> edit database tm_standby set state='APPLY-ON';
   ```

7. Restart the remaining instances in mount mode.

   ```
   $ srvctl start db -d tm_standby -o mount
   ```

> **Note:**
>
> If you plan to clone a local pluggable database (PDB) at the primary database, then enable Active Data Guard mode at the standby. This requires an Active Data Guard license.

Figure 9-15 Configuration with Test Master Files and Sparse Files for Redo Apply



9.7.7.3 Task 3: Create Full Database Snapshots Using the New Sparse Test Master

Create full snapshots using the new sparse test master.

At this time you can create full snapshots against the original files of the standby database as described in "Creating a Snapshot of a Full Database".

You must use the backup controlfile created in step 2 of Task 1: Prepare the Standby Database to Be Used as a Sparse Test Master to build the CREATE CONTROLFILE statement. To use the file you can create a temporary database instance to mount the controlfile and execute the backup controlfile to trace command.

1. Create a small PFILE file for the instance to use.

   At a minimum, the PFILE file should contain the following parameters:

   ```
   control_files='/home/oracle/snap_tm/control_monday_tm.ctl'  # This
   should be the control file name created above
   db_name=primary             # This should be the db_name used in the
   Data Guard configuration
   db_unique_name=temp         # This should be a unique name for a
   database instance on this host
   sga_target=5g               # Provide enough memory to start the
   instance
   ```

2. Set your environment to point to a unique ORACLE_SID.

   ```
   $ export ORACLE_SID=temp
   ```

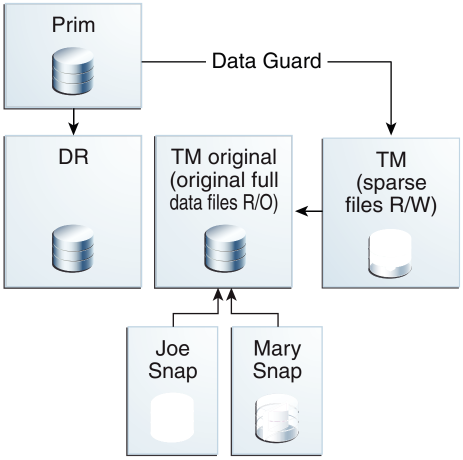
3. Using SQL*Plus, start the instance in mount mode using the PFILE created in Step 1.

   ```
   SQL> startup mount pfile='/home/oracle/snap_tm/pfile.ora'
   ```

4. Build the create controlfile statement and the rename files script.

   Use steps 1 and 2 in "Creating a Snapshot of a Full Database" to build the CREATE CONTROLFILE statement and the rename files script. The rename files script created in step 4 of "Task 1: Prepare the Standby Database to Be Used as a Sparse Test Master" can be used, but you must modify the directory structure of the sparse files to be created.

Figure 9-16 Configuration After Creating Snapshots



9.7.7.4 Task 4: Create a New Sparse Test Master Using a Previously Created Sparse Test Master

Create a new set of snapshots to provide a new test master and new redo apply files.

It is possible that at periodic intervals you will want to use the standby database to create additional snapshots without having to build a complete copy of the test master. You can repeat the process performed in the previous three tasks to do just that, taking advantage of the hierarchical snapshot functionality. The new test master is built on top of the latest existing snapshot that is applying redo. This snapshot becomes read-only and a new snapshot is built to continue the redo apply processing.

Do the following to configure the standby for the new test master time line:

1. Repeat the steps from "Task 1: Prepare the Standby Database to Be Used as a Sparse Test Master " with the following changes:

   This process is the same as before, except now you are using files that are sparse files for the new snapshots being created.

   a. In Step 2, change all the commands that alter the +DATA disk group to instead alter the +SPARSE disk group.

      The commands for setting access control and adding the database owner user to the disk group only need to be performed one time.

   b. In Step 4, supply a different name for the snapshot.

      You are creating a new snapshot, so the files need unique names from what was previously used. As a suggestion, you can append an identifier to the end of the file name to help identify it with the snapshot to be built. For example, if this was the original command:

      ```
      EXECUTE dbms_dnfs.clonedb_renamefile ('+SPARSE/TM_STANDBY/DATAFILE/
      system_515
      _930475939','+SPARSE/TM_STANDBY/DATAFILE/system_515_930475939');
      ```

You can add an identifier to the end of the file name to create a unique file name, as shown here:

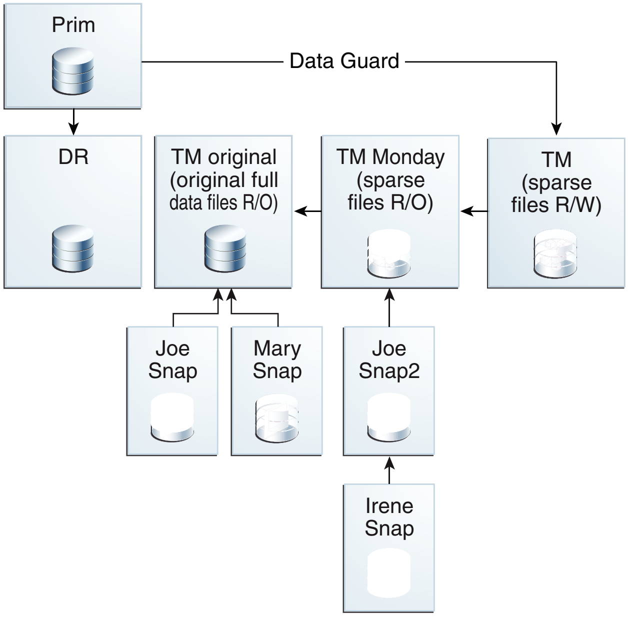
```
EXECUTE dbms_dnfs.clonedb_renamefile ('+SPARSE/TM_STANDBY/DATAFILE/
system_515
_930475939','+SPARSE/TM_STANDBY/DATAFILE/
system_515_930475939_Dec_15_16');
```

This step must be repeated for each statement for the `rename_files.sql` script.

2. Repeat the steps from "Task 2: Configure the Sparse Test Master and Sparse Files on the Standby Site."

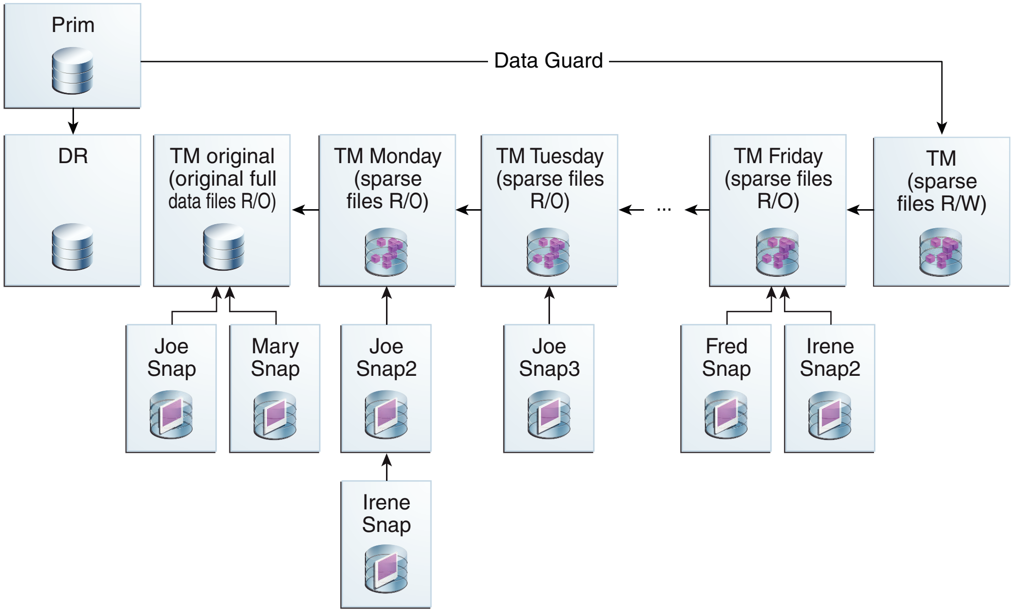
   There should be no changes required to the steps.

Figure 9-17 Configuration After Repeating the Process



The process can be repeated up to 9 times which would create an environment 10 levels deep (the original standby data files and 9 hierarchical snapshots). When repeating the process for the 9th time, do not create a new snapshot to receive the redo from the primary database.

Figure 9-18 Potential Future Configuration


Prim
Data Guard
DR
TM original (original full data files R/O)
TM Monday (sparse files R/O)
TM Tuesday (sparse files R/O)
...
TM Friday (sparse files R/O)
TM (sparse files R/W)
Joe Snap
Mary Snap
Joe Snap2
Joe Snap3
Fred Snap
Irene Snap2
Irene Snap

If you have reached the maximum 10 levels, you have multiple options:

- If you have enough space to maintain multiple copies of the standby database and snapshots, refresh the standby database and create a new tree of hierarchical snapshots. The original complete files and snapshots can remain as long as required.
- If you do not have enough space to maintain multiple copies of the standby database and snapshots, then delete all data files and snapshots, refresh the standby and create a new tree of hierarchical snapshots.
- Create a new standby database on a different environment and create a new tree of hierarchical snapshots.

Related Topics

- Task 1: Prepare the Standby Database to Be Used as a Sparse Test Master
- Task 2: Configure the Sparse Test Master and Sparse Files on the Standby Site

9.7.8 Creating Sparse Test Masters for PDBs

This procedure creates a hierarchical snapshot tree or sparse test masters manually for a pluggable database (PDB) in an Oracle Multitenant database.

The test master must be closed while making the daily reference snapshot. The downtime is very short (less than 5 minutes). You can use a replication mechanism, such as Oracle GoldenGate, to keep the sparse test master current with the production PDB. For more information about configuring Oracle GoldenGate with PDBs, see Configuring Oracle GoldenGate in a Multitenant Container Database in *Oracle GoldenGate Oracle Installation and Setup Guide*. The following example assumes you are using Oracle GoldenGate.

STEP 1: Create First Test Master PDB From the PROD PDB

This is a traditional PDB clone operation to instantiate the test master PDB. Once the clone completes, you configure Oracle GoldenGate to extract changes from the PRODPDB1 PDB in production and replicate these changes to the test master TMPDB1 PDB.

1. Run the following commands on the PROD container database (CDB) root:

   ```
   PRODCDB> alter pluggable database prodpdb1 close;

   PRODCDB> alter pluggable database prodpdb1 open read only;
   ```

2. Run the following commands from the test master CDB root:

   ```
   TMCDB> create database link PROD_DBLINK
   connect to system identified by password using 'PROD_CDB';

   TMCDB> create pluggable database TMPDB1
   from PRODPDB1@PROD_DBLINK;

   TMCDB> alter pluggable database TMPDB1 open;
   ```

3. Configure Oracle GoldenGate so that the changes made at the PRODPDB1 PDB will be extracted, replicated and applied to the TMPDB1 PDB. After configuring the extract and replicat and starting the extract process, open PRODPDB1 PDB in read write mode.

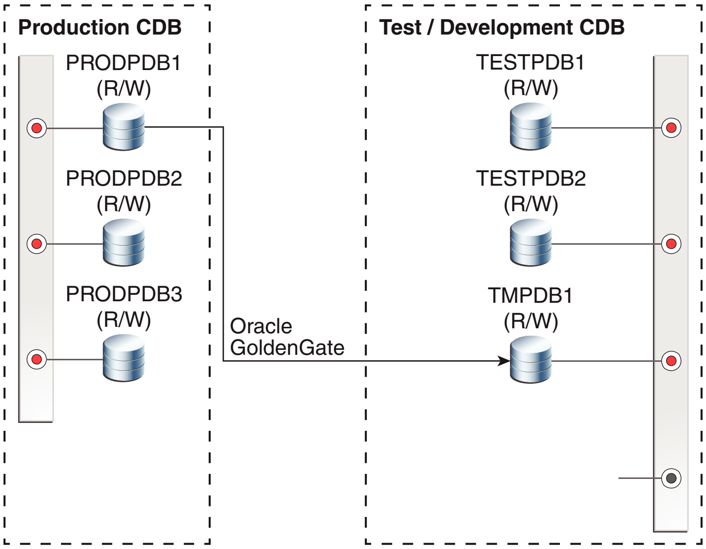
   > **Note:**
   >
   > The PRODPDB1 PDB cannot be opened for changes until after Oracle GoldenGate has been configured and the extract process started.

   ```
   PRODCDB> alter pluggable database PRODPDB1 close;
   PRODCDB> alter pluggable database PRODPDB1 open;
   ```

At this point, you have a full copy of PRODPDB1 on test master as TMPDB1 receiving all data changes made at PRODPDB1.

> **Note:**
>
> Oracle GoldenGate does not replicate data dictionary changes such as CREATE TABLESPACE or ADD DATAFILE. Only schema changes are replicated from PRODPDB1 to TMPDB1.

Figure 9-19 TMPDB1 Created From the PRODPDB1 Pluggable Database



Although `TMPDB1` can be opened in read/write mode, you should leave it in read-only mode because the only changes it should receive are from `PRODPDB1` through Oracle GoldenGate.

To create a snapshot from `TMPDB1`, the test master PDB must be opened in read-only mode. To provide a test master PDB from which you can create snapshots, and a test master PDB that is kept current with its source, you need two PDBs. The next step shows how you can accomplish this.

STEP 2: Create Daily Read-Only Snapshot and Move TMPDB1 PDB to a New Sparse Test Master PDB

This step creates a read-only snapshot PDB that acts as a test master. You can then create read/write snapshot PDBs from this read-only snapshot PDB every day. The main steps are:

- Create a (daily) read-only snapshot PDB that you can make available to private read/write clients.
- Create a new sparse `TMPDB1` PDB pointing back to the read-only daily snapshot PDB. The new `TMPDB1` PDB also accepts and applies changes from `PRODPDB1`.

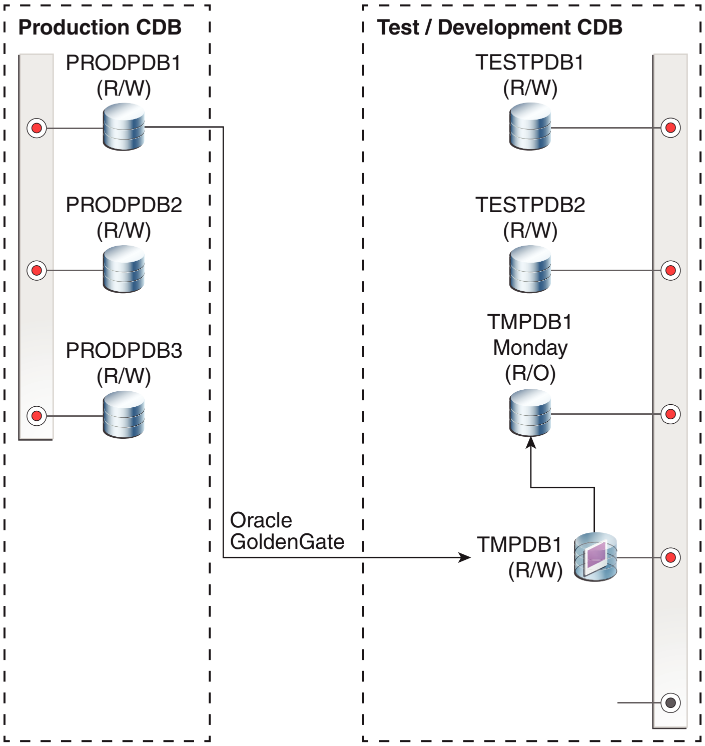
Connect to the `TMPDB1` PDB, then run the following commands:

```
TMCDB> alter session set container = CDB$ROOT;

# Stop the Oracle GoldenGate replicat process at the Test Master
database.  This allows
# all changes made at PRODPDB1 to continue to be extracted and then
applied to
# TMPDB1 when the replicat process is restarted.

# Close the test master PDB.
TMCDB> alter pluggable database TMPDB1 close;
```

```
# Write the test master PDB metadata to an XML file.
TMCDB> alter pluggable database TMPDB1 unplug into
       '/home/oracle/snapshot/TMPDB1_monday.XML';

# Drop the test master PDB, but keep the data files.
TMCDB> drop pluggable database TMPDB1 keep datafiles;

# Create a TMPDB1_MONDAY PDB using the XML file you just created.
#Use the NOCOPY clause to reuse the original data files.

TMCDB> create pluggable database TMPDB1_MONDAY using
       '/home/oracle/snapshot/TMPDB1_monday.XML' nocopy;

# Open the new TMPDB1_MONDAY PDB. The PDB must be opened
# once in read/write mode to complete the creation process.

TMCDB> alter pluggable database TMPDB1_MONDAY open;
TMCDB> alter pluggable database TMPDB1_MONDAY close;
TMCDB> alter pluggable database TMPDB1_MONDAY open read only;

# Create the new TMPDB1 PDB to receive changes from PRODPDB1. This PDB
# must have the same name as the original test master PDB to ensure no
# changes are required to the Oracle GoldenGate configuration.

TMCDB> create pluggable database TMPDB1 from TMPDB1_MONDAY
        create_file_dest='+SPARSE'
        snapshot copy;

# Open the new TMPDB1 PDB. The PDB must be opened once in read/write
# mode to complete the PDB creation process.

TMCDB> alter pluggable database TMPDB1 open;
TMCDB> alter pluggable database TMPDB1 close;
TMCDB> alter pluggable database TMPDB1 open read only;


# Restart the Oracle GoldenGate replicat process to the new TMPDB1
# PDB. The Oracle GoldenGate replicat process now applies changes from
# PRODPDB1 to the TMPDB1 snapshot and all changes are written to
# sparse files.
```

The following figure shows the `TMPDB1` created from `TMPDB1_MONDAY`. The original TMPDB1 has been renamed to TMPDB1_Monday as part of the `DROP PLUGGABLE DATABASE/CREATE PLUGGABLE DATABASE` steps listed above. The new TMPDB1 is a sparse snapshot pluggable database that, until any changes are made to TMPDB1, looks exactly like TMPDB1_Monday. Oracle GoldenGate applies redo to the new TMPDB1 snapshot without having to make any changes to the replicat configuration

Figure 9-20   TMPDB1 Created From TMPDB1_MONDAY



STEP 3: Create Read/Write Snapshot From TMPDB1_MONDAY

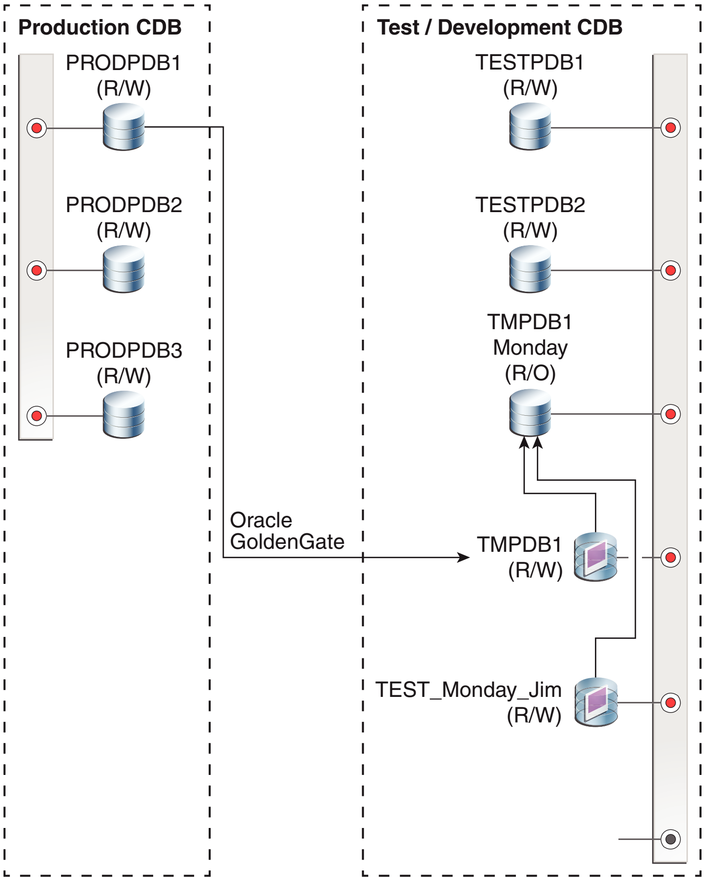
You create the snapshots from `TMPDB1_MONDAY`, not from `TMPDB1`. This allows `TMPDB1` to continue receiving and applying changes from `PRODPDB1`.

Connect to the `TMPDB1_MONDAY` PDB, then run the following commands:

```
TMCDB> alter session set container = cdb$ROOT;

TMCDB> create pluggable database TEST_MONDAY_JIM from TMPDB1_MONDAY
   create_file_dest='+SPARSE'
   snapshot copy;

TMCDB> alter pluggable database TEST_MONDAY_JIM open;
```

The following figure shows the TEST_MONDAY_JIM snapshot PDB created from TMPDB1_MONDAY. TEST_MONDAY_JIM uses TMPDB1_MONDAY as its parent so all data in TMPDB1_MONDAY_JIM is that same as the data in TMPDB1_MONDAY until Jim begins making changes to his snapshot PDB. Oracle GoldenGate continues to receive and apply redo to TMPDB1.

Figure 9-21   TEST_MONDAY_JIM Created From TMPDB1_MONDAY



When you need to create another test master and snapshot, you just need to repeat Step 2. For example, to create a test master on Tuesday, you can do the following:

Start a SQL*Plus session for the TMPDB1 PDB.

```
TMCDB> alter session set container = CDB$ROOT;

# Stop the Oracle GoldenGate replicat process from applying changes to
# TMPDB1

# Close the test master PDB
TMCDB> alter pluggable database TMPDB1 close;

# Write the test master PDB metadata to an XML file
TMCDB> alter pluggable database TMPDB1 unplug into
'/home/oracle/snapshots/TMPDB1_tuesday.XML';

# Drop the test master PDB, but keep the data files
TMCDB> drop pluggable database TMPDB1 keep datafiles;

# Create a TMPDB1_TUESDAY PDB from the XML file
TMCDB> create pluggable database TMPDB1_TUESDAY using
'/home/oracle/snapshot/TMPDB1_tuesday.XML' nocopy;
```

```
# Open the new TMPDB1_TUESDAY PDB
TMCDB> alter pluggable database TMPDB1_TUESDAY open;
TMCDB> alter pluggable database TMPDB1_TUESDAY close;
TMCDB> alter pluggable database TMPDB1_TUESDAY open read only;

# Create the new TMPDB1 PDB as a snapshot PDB
TMCDB> create pluggable database TMPDB1 from TMPDB1_TUESDAY
   create_file_dest='+SPARSE'
   snapshot copy;

# Open the TMPDB1 PDB
TMCDB> alter pluggable database TMPDB1 open;
TMCDB> alter pluggable database TMPDB1 close;
TMCDB> alter pluggable database TMPDB1 open read only;

# Restart the Oracle GoldenGate replicat process to apply changes to
# the new TMPDB1
```

You can now create read/write snapshot PDBs from TMPDB1_TUESDAY, similar to Step 3 above. The same as with full database sparse test masters, you can repeat this process up to 9 times in total before needing to either create a new TMPDB1 test master or drop and recreate the original TMPDB1 to begin building a new hierarchical snapshot tree.

9.7.9 Doing a Sparse Copy

The ASM cp command copies a sparse file to a new destination. However, this operation copies the sparse file with all the blocks instantiated from the parent. The “sparse copy” feature enables you to do a sparse copy of a file.

You can have multiple ASM instances running at the same time. If an operation involves a source or a destination on a different ASM instance other than the one it is executing on, it is treated as a remote ASM instance. You can do a sparse copy on a local ASM instance, or between a local and a remote ASM instance. However, sparse copy does not work between two remote ASM instances.

To do a sparse copy, you use the new --sparse option in the existing ASM cp command. The syntax looks like the following:

```
ASMCMD> cp --sparse <src_sparse_file> <tgt_file>
```

A new ASM command called setsparseparent enables you to set the parent of a sparse file. If you do a sparse copy of a file to a sparse destination on a local ASM instance, its parent is set as part of the sparse copy operation. However, if the destination is on a remote ASM instance, you have to set its parent explicitly using the setsparseparent command.

The setsparseparent command requires sparse child file and parent file as parameters. It sets the parent of the sparse child file to the new parent file. The syntax looks like the following:

```
ASMCMD> setsparseparent <sparse_file> <parent_file>
```

The cp ASM command performs the following validations before doing a sparse copy operation. The operation is allowed only if it satisfies the following criteria:

- The source file must exist and must be a sparse file.
- If you specify multiple source sparse files, all of them must be on the same ASM instance.
- Copying multiple sparse files on a remote ASM instance to a destination on a local ASM instance and vice versa is allowed provided all source files are on the same ASM instance.
- Destination file should be backed by a sparse disk group. However, it can be a non-sparse file if event “KFTST_KFPKG_CP_SPARSE” is set. This event is required to validate sparse copy operation by merging and copying the files to a non-sparse destination.
- Both source and destination cannot be on a remote ASM instance. However, either source or destination can be on a remote ASM instance.
- If the destination is on a remote ASM instance, its file type cannot be validated and you have to ensure that it is backed by a sparse disk group. You also have to set the parent explicitly using the ASM setsparseparent command.
- If the destination is a non-sparse file and you run the setsparseparent command, the command will fail because the child file should be sparse. This is a second-level validation if the destination is a non-sparse file.

The setsparseparent ASM command performs the following validations before it sets the parent. The operation is allowed only if it satisfies the following criteria:

- The child file must exist and must be a sparse file.
- The parent file must exist. It can be a sparse or a non-sparse file.
- Parent and child files must be present on same ASM instance.

Note:

You have to ensure that the files you specify in the setsparseparent ASM command have a valid parent-child relationship. The command cannot perform this check for files on remote ASM instances. If the files do not have a valid parent-child relationship, then data integrity and corruption issues are likely to occur.

Example 1: The following ASM command copies sparse file “TBS_1.264.908376549” to the destination “+SPARSEDG/child_1”.

```
ASMCMD> cp --sparse +SPARSEDG/MERGE/DATAFILE/TBS_1.264.908376549 +SPARSEDG/child_1
```

Example 2: The following ASM command sets parent “tbs_1.269.908374993” for the sparse file “remote_child_10”.

```
ASMCMD> setsparseparent +SPARSEDG/remote_child_10 +DATAFILE/DATAFILE/
tbs_1.269.908374993
```

Example 3: The following command copies sparse child files child_1, child_2 and child_3 to the destination directory +SPARSEDG.

```
ASMCMD> cp --sparse +SPARSEDG/DATAFILE/child_1 +SPARSEDG/DATAFILE/child_2 +SPARSEDG/
DATAFILE/child_3 +SPARSEDG/
```

9.8 Managing Sparse Griddisks

You can resize, recreate, or monitor the activity of sparse griddisks.

- Resizing the Virtual Space
  When V$ASM_DISKGROUP.FREE_MB or V$ASM_DISK.FREE_MB is running low, you need to increase the virtual address space.
- Resizing the Physical Space
  When the grid disks are running out of physical space, then you need to increase the physical size of the grid disks.
- Monitoring Sparse Disk Group Activity
- Repurposing Sparse Griddisks
  You can change sparse griddisks back to normal griddisks.

9.8.1 Resizing the Virtual Space

When V$ASM_DISKGROUP.FREE_MB or V$ASM_DISK.FREE_MB is running low, you need to increase the virtual address space.

1. To increase the size of the virtual space:

   a. Run the following command on the cells, specifying all the grid disks for the SPARSE disk group:

      ```
      CellCLI> alter griddisk
      SPARSE_CD_00_CELL01,SPARSE_CD_01_CELL01,....,SPARSE_CD_11_CELL01
      virtualSize=newSize
      ```

      For example, on the first cell:

      ```
      CellCLI> alter griddisk
      SPARSE_CD_00_CELL01,SPARSE_CD_01_CELL01,SPARSE_CD_02_
      CELL01,SPARSE_CD_03_CELL01,SPARSE_CD_04_CELL01,SPARSE_CD_05_CELL01,S
      PARSE_CD_
      06_CELL01,SPARSE_CD_07_CELL01,SPARSE_CD_08_CELL01,SPARSE_CD_09_CELL0
      1,SPARSE_
      CD_10_CELL01,SPARSE_CD_11_CELL01 virtualSize=12000G

      GridDisk SPARSE_CD_00_CELL01 successfully altered
      GridDisk SPARSE_CD_01_CELL01 successfully altered
      ...
      ```

      For example, on the next cell:

      ```
      CellCLI> alter griddisk
      SPARSE_CD_00_CELL02,SPARSE_CD_01_CELL02,SPARSE_CD_02_
      CELL02,SPARSE_CD_03_CELL02,SPARSE_CD_04_CELL02,SPARSE_CD_05_CELL02,S
      PARSE_CD_
      06_CELL02,SPARSE_CD_07_CELL02,SPARSE_CD_08_CELL02,SPARSE_CD_09_CELL0
      ```

```
2,SPARSE_
CD_10_CELL02,SPARSE_CD_11_CELL02 virtualSize=12000G

GridDisk SPARSE_CD_00_CELL02 successfully altered
GridDisk SPARSE_CD_01_CELL02 successfully altered
...
```

> **Note:**
>
> You must change the size of the grid disks on all cells before making any changes in Oracle ASM.

b. On an ASM instance, resize the disk group to this new size:

```
SQL> alter diskgroup SPARSE resize all size newSize;
```

For example:

```
SQL> alter diskgroup SPARSE resize all size 12000G;
```

2. To decrease the size of the virtual space:

a. On an ASM instance, resize the disk group to this new size:

```
SQL> alter diskgroup SPARSE resize all size newSize;
```

For example:

```
SQL> alter diskgroup SPARSE resize all size 8000G;
```

b. Run the following command on the cells, specifying all the grid disks for the SPARSE disk group:

```
CellCLI> alter griddisk
SPARSE_CD_00_CELL01,SPARSE_CD_01_CELL01,....,SPARSE_CD_11_CELL01
virtualSize=newSize
```

For example, on the first cell:

```
CellCLI> alter griddisk
SPARSE_CD_00_CELL01,SPARSE_CD_01_CELL01,SPARSE_CD_02_
CELL01,SPARSE_CD_03_CELL01,SPARSE_CD_04_CELL01,SPARSE_CD_05_CELL01,S
PARSE_CD_
06_CELL01,SPARSE_CD_07_CELL01,SPARSE_CD_08_CELL01,SPARSE_CD_09_CELL0
1,SPARSE_
CD_10_CELL01,SPARSE_CD_11_CELL01 virtualSize=8000G

GridDisk SPARSE_CD_00_CELL01 successfully altered
GridDisk SPARSE_CD_01_CELL01 successfully altered
...
```

For example, on the next cell:

```
CellCLI> alter griddisk
SPARSE_CD_00_CELL02,SPARSE_CD_01_CELL02,SPARSE_CD_02_
CELL02,SPARSE_CD_03_CELL02,SPARSE_CD_04_CELL02,SPARSE_CD_05_CELL02,S
PARSE_CD_
06_CELL02,SPARSE_CD_07_CELL02,SPARSE_CD_08_CELL02,SPARSE_CD_09_CELL0
2,SPARSE_
CD_10_CELL02,SPARSE_CD_11_CELL02 virtualSize=8000G

GridDisk SPARSE_CD_00_CELL02 successfully altered
GridDisk SPARSE_CD_01_CELL02 successfully altered
...
```

Related Topics

- *Oracle Automatic Storage Management Administrator's Guide*

9.8.2 Resizing the Physical Space

When the grid disks are running out of physical space, then you need to increase the physical size of the grid disks.

You can determine the amount of physical space left by comparing the values of the `TOTAL_MAT_MB` and `ALLOCATED_MAT_MB` columns in `V$ASM_DISK_SPARSE`. When the values in these two columns are close in size, then you need to increase the physical size of the grid disks.

1. To increase the size of the physical disk space:
   a. Before increasing the physical size of grid disks ensure there is free space available on the respective celldisks.

      ```
      [root@exa01adm01 tmp]# dcli -g ~/cell_group -l root "cellcli -e
      list celldisk \
        attributes name,freespace"
      exa01celadm01: CD_00_exa01celadm01 0
      exa01celadm01: CD_01_exa01celadm01 0
      exa01celadm01: CD_02_exa01celadm01 0
      exa01celadm01: CD_03_exa01celadm01 0
      exa01celadm01: CD_04_exa01celadm01 0
      exa01celadm01: CD_05_exa01celadm01 0
      exa01celadm01: CD_06_exa01celadm01 0
      exa01celadm01: CD_07_exa01celadm01 0
      exa01celadm01: CD_08_exa01celadm01 0
      exa01celadm01: CD_09_exa01celadm01 0
      exa01celadm01: CD_10_exa01celadm01 0
      exa01celadm01: CD_11_exa01celadm01 0
      ...
      ```

      If there is no available free space, then you have to free up disk space being used by the Oracle ASM disk groups.

b. Run the following command on the cells, specifying the grid disk to resize:

```
CellCLI> alter griddisk
data01_CD_00_exa01celadm01,data01_CD_01_exa01celadm01,...
,data01_CD_11_exa01celadm01 size=newPhysicalSize
```

For example, on the first cell:

```
CellCLI> alter griddisk
data01_CD_00_exa01celadm01,data01_CD_01_exa01celadm01,
data01_CD_02_exa01celadm01,data01_CD_03_exa01celadm01,data01_CD_04_e
xa01celadm01,
data01_CD_05_exa01celadm01,data01_CD_06_exa01celadm01,data01_CD_07_e
xa01celadm01,
data01_CD_08_exa01celadm01,data01_CD_09_exa01celadm01,data01_CD_10_e
xa01celadm01,
data01_CD_11_exa01celadm01 size=12000G
```

And then on the next cell:

```
CellCLI> alter griddisk
data01_CD_00_exa01celadm02,data01_CD_01_exa01celadm02,
data01_CD_02_exa01celadm02,data01_CD_03_exa01celadm02,data01_CD_04_e
xa01celadm02,
data01_CD_05_exa01celadm02,data01_CD_06_exa01celadm02,data01_CD_07_e
xa01celadm02,
data01_CD_08_exa01celadm02,data01_CD_09_exa01celadm02,data01_CD_10_e
xa01celadm02,
data01_CD_11_exa01celadm02 size=12000G
```

c. You do not need to do anything on the Oracle ASM instance.

2. To decrease the size of the physical disk space:

   a. You do not need to do anything on the Oracle ASM instance.

   b. Make sure you are not using more space in the Oracle ASM disk group than the size to which you are shrinking the physical disk space.

```
SQL> SELECT sum(allocated_mat_mb) FROM v$asm_disk_sparse
      WHERE group_number = group_number_of_diskgrp_to_shrink;
```

   If the amount of used physical space is more than the size you plan to shrink the disk down to, then you must drop objects from the sparse disk group until the used space is below the new physical space threshold.

   c. Shrink the size of the disks by running the following command on the cells, specifying the grid disks to resize:

```
CellCLI> alter griddisk
data01_CD_00_exa01celadm01,data01_CD_01_exa01celadm01,...
,data01_CD_11_exa01celadm01 size=newPhysicalSize
```

For example, on the first cell:

```
CellCLI> alter griddisk
data01_CD_00_exa01celadm01,data01_CD_01_exa01celadm01,
data01_CD_02_exa01celadm01,data01_CD_03_exa01celadm01,data01_CD_04_e
xa01celadm01,
data01_CD_05_exa01celadm01,data01_CD_06_exa01celadm01,data01_CD_07_e
xa01celadm01,
data01_CD_08_exa01celadm01,data01_CD_09_exa01celadm01,data01_CD_10_e
xa01celadm01,
data01_CD_11_exa01celadm01 size=4000G
```

And then on the next cell:

```
CellCLI> alter griddisk
data01_CD_00_exa01celadm02,data01_CD_01_exa01celadm02,
data01_CD_02_exa01celadm02,data01_CD_03_exa01celadm02,data01_CD_04_e
xa01celadm02,
data01_CD_05_exa01celadm02,data01_CD_06_exa01celadm02,data01_CD_07_e
xa01celadm02,
data01_CD_08_exa01celadm02,data01_CD_09_exa01celadm02,data01_CD_10_e
xa01celadm02,
data01_CD_11_exa01celadm02 size=4000G
```

Related Topics

- Resizing Grid Disks
  You can resize grid disks and Oracle ASM disk groups to shrink one with excess free space and increase the size of another that is near capacity.

9.8.3 Monitoring Sparse Disk Group Activity

The V$ASM_DISK and V$ASM_DISKGROUP views contain information about the virtual size and utilization of the sparse ASM disk group. New views V$ASM_DISK_SPARSE and V$ASM_DISKGROUP_SPARSE contain information about the actual size and utilization of the sparse ASM disk group. V$ASM_DISK_SPARSE also contains performance and usage metrics.

The following table describes the columns in V$ASM_DISK_SPARSE:

Table 9-1 V$ASM_DISK_SPARSE

Column	Description
GROUP_NUMBER	Number of the disk group containing the disk
DISK_NUMBER	Number assigned to the disk within this disk group
INCARNATION	Incarnation number for the disk
ALLOCATED_MAT_MB	Total used physical capacity of this disk
TOTAL_MAT_MB	Total physical capacity of this disk
SPARSE_READS	Number of read requests on sparse regions of this disk
SPARSE_BYTES_READ	Bytes read from sparse regions of this disk

Table 9-1 (Cont.) V$ASM_DISK_SPARSE

Column	Description
SPARSE_READ_TIME	Time taken by sparse read I/Os

The following table describes the columns in V$ASM_DISKGROUP_SPARSE:

Table 9-2 V$ASM_DISKGROUP_SPARSE

Column	Description
GROUP_NUMBER	Cluster-wide number assigned to the disk group
ALLOCATED_MAT_MB	Total used physical capacity of this disk group
TOTAL_MAT_MB	Total physical capacity of this disk group

The following example shows the used space and the total space for some disks.

```
SQL> select
     DISK_NUMBER          dsk_num,
     ALLOCATED_MAT_MB     alloc,
     TOTAL_MAT_MB         total
from V$ASM_DISK_SPARSE
where GROUP_NUMBER = 5;

DSK_NUM    ALLOC      TOTAL
---------- ---------- ----------
         0       5536      57336
         1       5424      57336
         2       5532      57336
         3       5424      57336
         4       5424      57336
```

In the following example, sparse ASM grid disks were created with an actual size of 56 GB and a virtual size of 560 GB. When you query V$ASM_DISK's OS_MB and TOTAL_MB columns, you can see the virtual size of 573440 MB (573440 MB / 1024 = 560 GB).

```
SQL> select os_mb, total_mb from v$asm_disk where group_number=4;

     OS_MB   TOTAL_MB
---------- ----------
    573440     573440
    573440     573440
    573440     573440
```

Querying V$ASM_DISK_SPARSE for TOTAL_MB, you can see the actual size of the ASM grid disk available for use. Note that each ASM grid disk stores metadata information of approximately 2 MB per 16 GB of space allocated to the sparse ASM grid disk. For 56 GB allocated per grid disk in this example, 8 MB of space is reserved for sparse disk metadata (57336 MB + 8 MB = 57344 MB / 1024 = 56 GB).

```
SQL> select total_mb from v$asm_disk_sparse where group_number=4;

 TOTAL_MB
---------
 57336
```

```
57336
57336
```

9.8.4 Repurposing Sparse Griddisks

You can change sparse griddisks back to normal griddisks.

If you previously created sparse griddisks, but now want to use them as normal griddisks, you can drop and recreate the disks.

1. Drop the snapshot database that is currently using the sparse griddisks.

   ```
   RMAN> startup mount force;
   RMAN> delete database;
   ```

2. Using SQL*Plus or ASMCMD, drop the Oracle ASM disk group that contains the sparse griddisks.

   ```
   SQL> DROP DISKGROUP sparse INCLUDING CONTENTS force;
   ```

3. Using CellCLI, drop the griddisks in the storage cells.

   ```
   cellcli -e drop griddisk all harddisk prefix=SPARSEC1
   ```

4. Recreate the griddisks.

   When you create the griddisks, use a similar size as the other disks and add back to the disk group of your choice. Do not specify the sparse attribute. See CREATE GRIDDISK for the command syntax.

5. Add the recreated griddisks to an Oracle ASM disk group.

   Use the SQL `ALTER DISKGROUP` command with the `ADD DISK` clause to add the disk to the Oracle ASM disk group using syntax similar to the following:

   ```
   SQL> ALTER DISKGROUP disk_group_name ADD DISK 'o/cell_IPaddress/data*';
   ```

Related Topics

- Creating a Sparse Grid Disk

A

Upgrading Oracle Exadata System Software

Oracle Exadata System Software, operating systems, and component firmware are upgraded and patched using Oracle utilities.

Patches include all required software updates, operating system updates, and firmware updates. Use the following utilities to upgrade and patch Oracle Exadata Database Machine:

- The `patchmgr` utility is used for Oracle Exadata System Software, and RDMA Network Fabric switches.
- The DB Node Update utility (`dbnodeupdate.sh` utility) is used for database servers.

In addition, Oracle recommends running the Oracle EXAchk utility before and after performing planned maintenance.

Upgrade and patching information is available from My Oracle Support note 888828.1. Additional information is available in the patch README, and associated support notes.

For detailed information about updating the Oracle Exadata System Software, operating systems, and component firmware, refer to Updating Exadata Software.

Related Topics

- Exadata Database Machine and Exadata Storage Server Supported Versions (My Oracle Support Doc ID 888828.1)

B
Installation Information

This appendix contains information about Oracle Exadata Storage Server installation. Refer to this installation information if the Oracle Exadata Storage Server hardware and software are not already installed.

- Installation of Oracle Exadata Storage Server
  This section describes the installation process for a storage server.
- Background Processes in the Oracle Exadata Storage Server Environment

Related Topics

- Configuring Oracle Exadata System Software
  This chapter describes the major steps to configure Oracle Exadata System Software.

B.1 Installation of Oracle Exadata Storage Server

This section describes the installation process for a storage server.

- Installing Oracle Exadata Storage Server Hardware
  Use these steps to complete the hardware installation necessary for standalone Oracle Exadata Storage Servers connecting to Oracle Exadata Database Machine.
- Installing the Oracle Exadata System Software on a Cell
  This procedure describes how to install the Oracle Exadata System Software on an Oracle Exadata Storage Server.
- Updates Made During Installation
- Online Documentation

B.1.1 Installing Oracle Exadata Storage Server Hardware

Use these steps to complete the hardware installation necessary for standalone Oracle Exadata Storage Servers connecting to Oracle Exadata Database Machine.

1. Unpack, situate, and set up the storage server.
2. Read the printed vendor documentation supplied with the storage servers.
3. Connect the management and Integrated Lights Out Manager (ILOM) Ethernet ports to the management network.
4. Connect the RDMA Network Fabric ports of the Oracle Exadata Storage Server to the RDMA Network Fabric network of Oracle Exadata Database Machine.

> **Note:**
>
> Both the Oracle Exadata Database Machine and additional Oracle Exadata Storage Servers must use the same type of RDMA Network Fabric.

5. Attach a computer or terminal to one of the new storage servers so you can configure the cell as described in Configuring Oracle Exadata System Software for Your Location. You must configure each new cell.
6. Connect the power supply on the new storage servers.
7. Power on the storage servers.
8. Assign IP addresses for the network interconnections of the new storage servers.

Related Topics

- Assigning IP Addresses for Oracle Exadata Database Machine
  This topic summarizes the Oracle Exadata Database Machine network preparation before installing the new storage server.

B.1.2 Installing the Oracle Exadata System Software on a Cell

This procedure describes how to install the Oracle Exadata System Software on an Oracle Exadata Storage Server.

1. Power on the Oracle Exadata Storage Server to boot its operating system.
2. Log in to the Oracle Exadata Storage Server with super user (`root`) privileges.

> **Note:**
>
> If you do not have the password for the root user, then contact Oracle Support Services.

3. Change the `root` password on the cell.
4. Verify that the Oracle Exadata Storage Server meets the software requirements.
5. Copy the self-extracting binary to the Oracle Exadata Storage Server, if necessary.

   You can use the `ftp` command to copy the binary to the Oracle Exadata Storage Server from a network location, or you can copy the binary from a portable storage device.
6. Install the self-extracting binary on the Oracle Exadata Storage Server if it is not already installed.

   ```
   ./cell_package_version.bin
   ```
7. Review the `/opt/oracle/cell/.install_log.txt` file to verify that the installation completed correctly. The installation process adds two new users, `celladmin` and `cellmonitor`. The `celladmin` user should be used to run all services on the cell. The `cellmonitor` user is for monitoring purposes.

8. After successful installation of the Oracle Exadata System Software, follow the instructions at "Preparing the Servers".

Related Topics

- Preparing the Servers
  Use the following steps to prepare the database servers and storage servers for use.

B.1.3 Updates Made During Installation

Changes that are made to `snmpd` files during installation include the following:

- Updates to the public community string
- New entries for `trapdestination`
- Updates to the kernel configuration files (limits for `aio`, `udprecv`, and number of open files)

B.1.4 Online Documentation

The documentation can be found online at https://docs.oracle.com/.

B.2 Background Processes in the Oracle Exadata Storage Server Environment

The background processes for the database and Oracle ASM instance for an Oracle Exadata Storage Server environment are the same as other environments, except for the following background processes:

- diskmon Process
- XDMG Process
- XDWK Process

B.2.1 diskmon Process

The `diskmon` process is a fundamental component of Oracle Exadata System Software, and is responsible for implementing I/O fencing. The process is located on the database server host computer, and is part of Oracle Clusterware Cluster Ready Services (CRS). This process is important for Oracle Exadata System Software and should not be modified.

The log files for `diskmon` are located in the `$CRS_HOME/log/hostname/diskmon` directory.

See Also:

- *Oracle Clusterware Administration and Deployment Guide* for additional information about the following:
  - Oracle Clusterware Diagnostic and Alert Log Data
  - Overview of Oracle Clusterware Platform-Specific Software Components
- *Oracle Database Concepts* for information about Oracle Database processes
- *Oracle Database Reference* for a description of the `V$BGPROCESS` view that displays information about background processes

B.2.2 XDMG Process

The `XDMG` (Exadata Automation Manager) process initiates automation tasks used for monitoring storage. This background process monitors all configured Oracle Exadata Storage Servers for state changes, such as replaced disks, and performs the required tasks for such changes. Its primary task is to watch for inaccessible disks and cells, and to detect when the disks and cells become accessible. When the disks and cells are accessible, the `XDMG` process initiates the `ASM ONLINE` process, which is handled by the `XDWK` background process. The `XDMG` process runs in the Oracle ASM instances.

B.2.3 XDWK Process

The `XDWK` (Exadata Automation Worker) process performs automation tasks by requested by the `XDMG` background process. The `XDWK` process begins when asynchronous actions, such as `ONLINE`, `DROP` or `ADD` for an Oracle ASM disk are requested by the `XDMG` process. The `XDWK` process stops after 5 minutes of inactivity. The `XDWK` process runs in the Oracle ASM instances.

C

Automatic Workload Repository Views for Oracle Exadata System Software

The Automatic Workload Repository (AWR) views in this appendix are specific to Oracle Exadata System Software. The views are static, because they change only when a change is made to the data dictionary, such as when a new table is created or when a user is granted new privileges.

- DBA_HIST_ASM_BAD_DISK
- DBA_HIST_ASM_DISKGROUP
- DBA_HIST_ASM_DISKGROUP_STAT
- DBA_HIST_CELL_CONFIG
- DBA_HIST_CELL_CONFIG_DETAIL
- DBA_HIST_CELL_DB
- DBA_HIST_CELL_DISKTYPE
- DBA_HIST_CELL_DISK_NAME
- DBA_HIST_CELL_DISK_SUMMARY
- DBA_HIST_CELL_GLOBAL
- DBA_HIST_CELL_GLOBAL_SUMMARY
- DBA_HIST_CELL_IOREASON
- DBA_HIST_CELL_IOREASON_NAME
- DBA_HIST_CELL_METRIC_DESC
- DBA_HIST_CELL_NAME
- DBA_HIST_CELL_OPEN_ALERTS

C.1 DBA_HIST_ASM_BAD_DISK

The DBA_HIST_ASM_BAD_DISK view displays historic information about non-online Oracle Automatic Storage Management (Oracle ASM) disks. This view contains snapshots of V$ASM_DISK.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot identifier.
DBID	NUMBER	NOT NULL	Database identifier for the snapshot.
GROUP_NUMBER	NUMBER	NOT NULL	Number of the disk group containing the disk.
NAME	VARCHAR2(128)	NOT NULL	Name of the disk.

Column	Datatype	NULL	Description
PATH	VARCHAR2(256)		Operating system path name portion of the name returned by discovery.
STATUS	VARCHAR2(8)	NOT NULL	Global status for the disk. Only non-online disks are stored in this view.
CON_DBID	NUMBER		The database identifier.
CONI_D	NUMBER		The identifier of the container identified by CON_DBID. Possible values are as follows: • 0: This value is used for non-container database (CDB). • 1: This value is used for the root container of the CDB.

C.2 DBA_HIST_ASM_DISKGROUP

The DBA_HIST_ASM_DISKGROUP view contains information about Oracle ASM disk groups. This retrieves a subset of columns from V$ASM_DISKGROUP.

Column	Datatype	NULL	Description
GROUP_NUMBER	NUMBER	NOT NULL	Cluster-wide number assigned to the disk group.
NAME	VARCHAR2(128)	NOT NULL	Name of the disk group.
TYPE	VARCHAR2(6)		Redundancy type for the disk group.
CON_DBID	NUMBER		The database identifier.
CONI_D	NUMBER		The identifier of the container identified by CON_DBID. Possible values are as follows: • 0: This value is used for non-container database (CDB). • 1: This value is used for the root container of the CDB.

C.3 DBA_HIST_ASM_DISKGROUP_STAT

The DBA_HIST_ASM_DISKGROUP_STAT view displays historic information about Oracle ASM disk groups.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot identifier.
DBID	NUMBER	NOT NULL	Database identifier for the snapshot.
GROUP_NUMBER	NUMBER	NOT NULL	Cluster-wide number assigned to the disk group.
TOTAL_MB	NUMBER		Total capacity of the disk group in MB.

Column	Datatype	NULL	Description
FREE_MB	NUMBER		Unused capacity in MB of the disk group.
NUM_DISK	NUMBER		Number of disks in the disk group.
CON_DBID	NUMBER		The database identifier.
CONI_D	NUMBER		The identifier of the container identified by CON_DBID. Possible values are as follows: • 0: This value is used for non-container database (CDB). • 1: This value is used for the root container of the CDB.

C.4 DBA_HIST_CELL_CONFIG

The DBA_HIST_CELL_CONFIG view contains information about the configuration of the cells. This data is from V$CELL_CONFIG_INFO.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database identifier for the snapshot.
CELLNAME	VARCHAR2(256)	NOT NULL	Unique identifier for the cell.
CELLHASH	NUMBER	NOT NULL	Hash number to uniquely identify the cell.
CONFTYPE	VARCHAR2(15)	NOT NULL	Configuration type. The configuration type determines the type of information stored in the CONFVAL column. • AWRXML: configuration information pertinent to AWR reports. • CELL: configuration information about the cell. • CELLDISK: configuration information about cell disks. • GRIDDISK: configuration information about grid disks. • OFFLOAD: configuration information for offload servers.
CONFVAL	CLOB		XML data associated with CONFTYPE.
CON_DBID	NUMBER		The database identifier.
CONI_D	NUMBER		The identifier of the container identified by CON_DBID. Possible values are as follows: • 0: This value is used for non-container database (CDB). • 1: This value is used for the root container of the CDB.

C.5 DBA_HIST_CELL_CONFIG_DETAIL

The `DBA_HIST_CELL_CONFIG_DETAIL` view displays historic information about the configuration of the cells. This view has the same descriptions as `DBA_HIST_CELL_CONFIG`, but includes the data for each snapshot, such as a historic view.

Column	Datatype	NULL	Description
`SNAP_ID`	NUMBER	NOT NULL	Unique snapshot identifier.
`DBID`	NUMBER	NOT NULL	Database identifier for the snapshot.
`CELLNAME`	VARCHAR2(256)	NOT NULL	Unique identifier for the cell.
`CELLHASH`	NUMBER	NOT NULL	Hash number to uniquely identify the cell.
`CONFTYPE`	VARCHAR2(15)	NOT NULL	Configuration type. The configuration type determines the type of information stored in the CONFVAL column. • AWRXML: configuration information pertinent to AWR reports. • CELL: configuration information for the cell. • CELLDISK: configuration information about cell disks. • GRIDDISK: configuration information about grid disks. • OFFLOAD: configuration information for offload servers.
`CONFVAL`	CLOB		XML data associated with the CONFTYPE.
`CON_DBID`	NUMBER		The database identifier.
`CONI_D`	NUMBER		The identifier of the container identified by CON_DBID. Possible values are as follows: • 0: This value is used for non-container database (CDB). • 1: This value is used for the root container of the CDB.

C.6 DBA_HIST_CELL_DB

The `DBA_HIST_CELL_DB` view displays historic information about the databases consuming resources on the cells. Only the top 10 databases are stored in each AWR snapshot. This is a snapshot of `V$CELL_DB`.

Column	Datatype	NULL	Description
`SNAP_ID`	NUMBER	NOT NULL	Unique snapshot identifier.
`DBID`	NUMBER	NOT NULL	Database identifier for the snapshot.

Column	Datatype	NULL	Description
CELL_HASH	NUMBER	NOT NULL	Hash number to uniquely identify the cell.
INCARNATION_NUM	NUMBER	NOT NULL	Incarnation number of the cell. Each cell reboot will increment the incarnation number.
SRC_DBID	NUMBER	NOT NULL	DBID performing the I/Os.
SRC_DBNAME	VARCHAR2(256)		DB Name performing the I/Os.
DISK_REQUESTS	NUMBER		Number of disk I/O requests performed by the SRC_DBID.
DISK_BYTES	NUMBER		Number of disk I/O bytes performed by the SRC_DBID.
FLASH_REQUESTS	NUMBER		Number of flash I/O requests performed by the SRC_DBID.
FLASH_BYTES	NUMBER		Number of flash I/O bytes performed by the SRC_DBID.
CON_DBID	NUMBER		The database identifier.
CONI_D	NUMBER		The identifier of the container identified by CON_DBID. Possible values are as follows: • 0: This value is used for non-container database (CDB). • 1: This value is used for the root container of the CDB.

C.7 DBA_HIST_CELL_DISKTYPE

The DBA_HIST_CELL_DISKTYPE view displays historic information about cells, including the types of disks and the capacity of the disks. This view is derived from DBA_HIST_CELL_CONFIG and DBA_HIST_CELL_CONFIG_DETAIL.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot identifier.
DBID	NUMBER	NOT NULL	Database identifier for the snapshot.
CELL_HASH	NUMBER	NOT NULL	Hash number to uniquely identify the cell.
CELL_NAME	VARCHAR2(4000)		User-readable cell name.
HARD_DISK_TYPE	VARCHAR2(4000)		The type of hard disk. The format of which is H/*size*, where *size* is suffixed by G, or T.
FLASH_DISK_TYPE	VARCHAR2(4000)		The type of flash, the format of which is F/*size*, where *size* is suffixed by G, or T.
NUM_CELL_DISKS	NUMBER		Number of cell disks.

Column	Datatype	NULL	Description
NUM_GRID_DISKS	NUMBER		Number of grid disks.
NUM_HARD_DISKS	NUMBER		Number of hard disks.
NUM_FLASH_DISKS	NUMBER		Number of flash disks.
MAX_DISK_IOPS	NUMBER		The maximum number of IOPs for the hard disk type.
MAX_FLASH_IOPS	NUMBER		The maximum number of IOPs for the flash disk type.
MAX_DISK_MBPS	NUMBER		The maximum I/O throughput for the hard disk type, in megabytes per second.
MAX_FLASH_MBPS	NUMBER		The maximum I/O throughput for the flash disk type, in megabytes per second.
MAX_CELL_DISK_IOPS	NUMBER		The maximum number of hard disk IOPs for the cell. This is calculated using MAX_DISK_IOPS*NUM_HARD_DISKS.
MAX_CELL_FLASH_IOPS	NUMBER		The maximum number of flash IOPs for the cell. This is calculated using MAX_FLASH_IOPS*NUM_FLASH_DISKS.
MAX_CELL_DISK_MBPS	NUMBER		The maximum I/O throughput of hard disk for the cell., in megabytes per second. This is calculated using MAX_DISK_MBPS*NUM_HARD_DISKS.
MAX_CELL_FLASH_MBPS	NUMBER		The maximum I/O throughput flash for the cell, in megabytes per second. This is calculated using MAX_FLASH_MBPS*NUM_FLASH_DISKS.
CON_DBID	NUMBER		The database identifier.
CONI_D	NUMBER		The identifier of the container identified by CON_DBID. Possible values are as follows: • 0: This value is used for non-container database (CDB). • 1: This value is used for the root container of the CDB.

C.8 DBA_HIST_CELL_DISK_NAME

The DBA_HIST_CELL_DISK_NAME view displays historic information about disk names on cells. This view is derived from DBA_HIST_CELL_CONFIG and DBA_HIST_CELL_CONFIG_DETAIL.

Column	Datatype	NULL	Description
`SNAP_ID`	NUMBER	NOT NULL	Unique snapshot identifier.
`DBID`	NUMBER	NOT NULL	Database identifier for the snapshot.
`CELL_HASH`	NUMBER	NOT NULL	Hash number to uniquely identify the cell.
`DISK_ID`	NUMBER	NOT NULL	Unique identifier of the disk.
`DISK_NAME`	VARCHAR2(4000)		User-readable cell name.
`DISK`	VARCHAR2(4000)		Disk type, either FlashDisk or HardDisk.
`CON_DBID`	NUMBER		The database identifier.
`CONI_D`	NUMBER		The identifier of the container identified by CON_DBID. Possible values are as follows: • 0: This value is used for non-container database (CDB). • 1: This value is used for the root container of the CDB.

C.9 DBA_HIST_CELL_DISK_SUMMARY

The `DBA_HIST_CELL_DISK_SUMMARY` view displays historic information about the performance of disks on cells. For each AWR snapshot, a summary of the per-minute metrics is visible in this view. Each metric stores the sum and the sum of squares, similar to `DBA_HIST_SYSMETRIC_SUMMARY`. This view is a summary of `V$CELL_DISK_HISTORY`.

Column	Datatype	NULL	Description
`SNAP_ID`	NUMBER	NOT NULL	Unique snapshot identifier.
`DBID`	NUMBER	NOT NULL	Database identifier for the snapshot.
`CELL_HASH`	NUMBER	NOT NULL	Hash number to uniquely identify the cell.
`DISK_ID`	NUMBER	NOT NULL	Unique identifier of the disk.
`NUM_SAMPLES`	NUMBER		Number of one-minute samples the summary is based on.
`DISK_UTILIZATION_SUM`	NUMBER		Sum of the per-minute disk utilization metrics. This is based on OS statistics.
`READS_SUM`	NUMBER		Sum of the per-minute read requests metrics per second. This is based on OS statistics.
`READ_MB_SUM`	NUMBER		Sum of the per-minute read metrics, in megabytes per second. This is based on OS statistics.
`WRITES_SUM`	NUMBER		Sum of the per-minute write requests metrics, per second. This is based on OS statistics.

Column	Datatype	NULL	Description
WRITE_MB_SUM	NUMBER		Sum of the per-minute write metrics, in megabytes per second. This is based on OS statistics.
IO_REQUESTS_SUM	NUMBER		Sum of the per-minute IOPs. This is based on OS statistics.
IO_MB_SUM	NUMBER		Sum of the per-minute I/O metrics, in megabytes per second. This is based on OS statistics.
SERVICE_TIME_SUM	NUMBER		Sum of the per-minute service time metrics. This is based on OS statistics.
WAIT_TIME_SUM	NUMBER		Sum of the per-minute wait time metrics. This is based on OS statistics.
SMALL_READS_SUM	NUMBER		Sum of the per-minute small read requests metrics per second. This is from cell server statistics.
SMALL_WRITES_SUM	NUMBER		Sum of the per-minute small write requests metrics per second. This is from cell server statistics.
LARGE_READS_SUM	NUMBER		Sum of the per-minute large read requests metrics per second. This is from cell server statistics.
LARGE_WRITES_SUM	NUMBER		Sum of the per-minute large write requests metrics per second. This is from cell server statistics.
SMALL_READ_BYTES_SUM	NUMBER		Sum of the per-minute small read bytes metrics per second. This is from cell server statistics.
SMALL_WRITE_BYTES_SUM	NUMBER		Sum of the per-minute small write bytes metrics per second. This is from cell server statistics.
LARGE_READ_BYTES_SUM	NUMBER		Sum of the per-minute large read bytes metrics per second. This is from cell server statistics.
LARGE_WRITE_BYTES_SUM	NUMBER		Sum of the per-minute large write bytes metrics per second. This is from cell server statistics.
SMALL_READ_LATENCY_SUM	NUMBER		Sum of the per-minute small read latency metrics. This data is from cell server statistics.
SMALL_WRITE_LATENCY_SUM	NUMBER		Sum of the per-minute small write latency metrics. This data is from cell server statistics.
LARGE_READ_LATENCY_SUM	NUMBER		Sum of the per-minute large read latency metrics. This data is from cell server statistics.
LARGE_WRITE_LATENCY_SUM	NUMBER		Sum of the per-minute large write latency metrics. This data is from cell server statistics.

Column	Datatype	NULL	Description
APP_IO_REQUESTS_SUM	NUMBER		Sum of the per-minute IOPs metrics. This data is from cell server statistics.
APP_IO_BYTES_SUM	NUMBER		Sum of the per-minute I/O metrics, in megabytes per second. This data is from cell server statistics.
APP_IO_LATENCY_SUM	NUMBER		Sum of the per-minute I/O latency metrics. This data is from cell server statistics.
*_SUMX2	NUMBER		All the previous columns from DISK_UTILIZATION to APP_IO_LATENCY are repeated with the SUMX2 suffix. This is the sum of squares of the per-minute metrics.
*_AVG	NUMBER		All the previous columns from DISK_UTILIZATION to APP_IO_LATENCY are repeated with the AVG suffix. This is the average of the per-minute metrics.
CON_DBID	NUMBER		The database identifier.
CONI_D	NUMBER		The identifier of the container identified by CON_DBID. Possible values are as follows: • 0: This value is used for non-container database (CDB). • 1: This value is used for the root container of the CDB.

C.10 DBA_HIST_CELL_GLOBAL

The DBA_HIST_CELL_GLOBAL view displays historic information about cell performance statistics. This is a snapshot of V$CELL_GLOBAL.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot identifier.
DBID	NUMBER	NOT NULL	Database identifier for the snapshot.
CELL_HASH	NUMBER	NOT NULL	Hash number that uniquely identifies the cell.
INCARNATION_NUM	NUMBER	NOT NULL	Incarnation number of the cell.
METRIC_ID	NUMBER	NOT NULL	Metric_id that identifies the metric.
METRIC_NAME	VARCHAR2(257)		Name of the metric.
METRIC_VALUE	NUMBER		The value of the metric.
CON_DBID	NUMBER		The database identifier.

Column	Datatype	NULL	Description
CONI_D	NUMBER		The identifier of the container identified by CON_DBID. Possible values are as follows: • 0: This value is used for non-container database (CDB). • 1: This value is used for the root container of the CDB.

C.11 DBA_HIST_CELL_GLOBAL_SUMMARY

The DBA_HIST_CELL_GLOBAL_SUMMARY view displays historic information about the performance of cells. For each AWR snapshot, a summary of the per-minute metrics is visible in this view. This is a summary of V$CELL_GLOBAL_HISTORY.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database identifier for the snapshot.
CELL_HASH	NUMBER	NOT NULL	Hash number that uniquely identifies the cell.
INCARNATION_NUM	NUMBER	NOT NULL	Incarnation number of the cell.
NUM_SAMPLES	NUMBER		Number of 1 minute samples the summary is based on.
CPU_USAGE_SUM	NUMBER		Sum of the per-minute CPU usage percentage metrics.
SYS_USAGE_SUM	NUMBER		Sum of the per-minute system CPU usage percentage metrics.
USER_USAGE_SUM	NUMBER		Sum of the per-minute user CPU usage percentage metrics.
NETWORK_BYTES_RECD_SUM	NUMBER		Sum of the per-minute network received metrics, in bytes per second.
NETWORK_BYTES_SENT_SUM	NUMBER		Sum of the per-minute network sent metrics, in bytes per second.
*_SUMX2	NUMBER		All previous columns from CPU_USAGE to NETWORK_BYTES_SENT are repeated with the SUMX2 suffix. This is the sum of squares of the per-minute metrics.
*_AVG	NUMBER		All previous columns from CPU_USAGE to NETWORK_BYTES_SENT are repeated with the SUMX2 suffix. This is the average of the per-minute metrics.

Column	Datatype	NULL	Description
CON_DBID	NUMBER		The database identifier.
CONI_D	NUMBER		The identifier of the container identified by CON_DBID. Possible values are as follows: • 0: This value is used for non-container database (CDB). • 1: This value is used for the root container of the CDB.

C.12 DBA_HIST_CELL_IOREASON

The DBA_HIST_CELL_IOREASON view displays historic information about the reasons for performing I/Os on the cells. This is a snapshot of V$CELL_IOREASON.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot identifier.
DBID	NUMBER	NOT NULL	Database identifier for the snapshot.
CELL_HASH	NUMBER	NOT NULL	Hash number that uniquely identifies the cell.
INCARNATION_NUM	NUMBER	NO T NULL	Incarnation number of the cell.
REASON_ID	NUMBER	NOT NULL	Unique identifier for the I/O reason.
REASON_NAME	VARCHAR2(257)		User readable I/O reason.
REQUESTS	NUMBER		Number of I/O requests performed for the reason.
BYTES	NUMBER		Number of IO Bytes performed for the reason.
CON_DBID	NUMBER		The database identifier.
CONI_D	NUMBER		The identifier of the container identified by CON_DBID. Possible values are as follows: • 0: This value is used for non-container database (CDB). • 1: This value is used for the root container of the CDB.

C.13 DBA_HIST_CELL_IOREASON_NAME

The DBA_HIST_CELL_IOREASON_NAME view displays the reasons for different I/Os. This data is from V$CELL_IOREASON_NAME.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database identifier for the snapshot.
REASON_ID	NUMBER	NOT NULL	Unique identifier for an I/O reason.

Column	Datatype	NULL	Description
REASON_NAME	VARCHAR2(257)		User-readable description of the I/O reason.
CON_DBID	NUMBER		The database identifier.
CONI_D	NUMBER		The identifier of the container identified by CON_DBID. Possible values are as follows: • 0: This value is used for non-container database (CDB). • 1: This value is used for the root container of the CDB.

C.14 DBA_HIST_CELL_METRIC_DESC

The DBA_HIST_CELL_METRIC_DESC view displays information about cell metrics. This data is from V$CELL_METRIC_DESC.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database identifier for the snapshot.
METRIC_ID	NUMBER	NOT NULL	Unique identifier for the metric
METRIC_NAME	VARCHAR2(257)		User-readable metric name
METRIC_TYPE	VARCHAR2(17)	NOT NULL	Unit for the metrics, if NULL this is a count.
CON_DBID	NUMBER		The database identifier.
CONI_D	NUMBER		The identifier of the container identified by CON_DBID. Possible values are as follows: • 0: This value is used for non-container database (CDB). • 1: This value is used for the root container of the CDB.

C.15 DBA_HIST_CELL_NAME

The DBA_HIST_CELL_NAME view displays historic information about the names of cells. This view is derived from DBA_HIST_CELL_CONFIG and DBA_HIST_CELL_CONFIG_DETAIL.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot identifier.
DBID	NUMBER	NOT NULL	Database identifier for the snapshot.
CELL_HASH	NUMBER	NOT NULL	Hash number that uniquely identifies the cell.
CELL_NAME	VARCHAR2(4000)		User-readable cell name.
CON_DBID	NUMBER		The database identifier.

Column	Datatype	NULL	Description
CONI_D	NUMBER		The identifier of the container identified by CON_DBID. Possible values are as follows: - 0: This value is used for non-container database (CDB). - 1: This value is used for the root container of the CDB.

C.16 DBA_HIST_CELL_OPEN_ALERTS

The DBA_HIST_CELL_OPEN_ALERTS view displays the open alerts on cells. Only the 10 most-recent alerts for each cell are stored in AWR. This data is from V$CELL_OPEN_ALERTS.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot identifier.
DBID	NUMBER	NOT NULL	Database identifier for the snapshot.
CELL_HASH	NUMBER	NOT NULL	Hash number that uniquely identifies the cell.
BEGIN_TIME	DATE	NOT NULL	Time that the alert was opened.
SEQ_NO	NUMBER	NOT NULL	Sequence number of the alert.
MESSAGE	VARCHAR2(1024)		Message describing the alert.
STATEFUL	VARCHAR2(1)		Whether or not the alert has state.
SEVERITY	VARCHAR2(64)		The severity of the alert. Possible values are as follows: - critical - warning - info
CON_DBID	NUMBER		The database identifier.
CONI_D	NUMBER		The identifier of the container identified by CON_DBID. Possible values are as follows: - 0: This value is used for non-container database (CDB). - 1: This value is used for the root container of the CDB.

D

Oracle Exadata System Software Accessibility Recommendations

Use these tips for using screen readers and screen magnifiers with Oracle Exadata System Software utilities.

Oracle Exadata System Software includes tools such as `dcli` and `cellcli` that you can run from the command line.

- Tips on Using Screen Readers and Braille Displays
  Use the following tips when using screen readers with Oracle Exadata System Software
- Tips on Using Screen Magnifiers
  Use the following tips when using screen magnifiers with Oracle Exadata System Software
- Tips on Using Exawatcher Charts
  Exawatcher is an Exadata specific tool that collects performance data from Exadata storage cells.
- Oracle Exadata Deployment Assistant (OEDA) Web interface Accessibility
  The OEDA Web interface conforms with the Web Content Accessibility Guidelines version 2.0 at the AA level (WCAG 2.0 AA).

D.1 Tips on Using Screen Readers and Braille Displays

Use the following tips when using screen readers with Oracle Exadata System Software

Examples of screen readers include JAWS, SuperNova, NVDA, and Window-Eyes. Each of these provides text-to-speech output and supports braille displays.

> **Note:**
>
> Window-Eyes is now a part of ZoomText Fusion by Ai Squared.

- Use a character mode based terminal such as Putty or Cygwin. Do not use an X-Windows-based VNC.
- For screen reader users, we recommend installing "screen" in order to get multiple session support. The Linux based screen program allows for multiple sessions in different windows. You can access each session with keyboard based commands, for example, `Ctrl-a`. Screen allows you to detach or re-attach to a given window session. Like VNC, if you get disconnected when running Oracle ExaCHK, or patchmgr, or other program, you can re-attach to and resume that session.

The screen package is not installed by default on Exadata. You will need to install it using yum. See the "How To Use Linux Screen" tutorial at https://www.rackaid.com/blog/linux-screen-tutorial-and-how-to/ for details.

- In the settings of the terminal software, set the cursor type to "block" cursor, not blinking or flashing.
- The output of the commands can generate a significant amount of information and might spill off the terminal window, and the virtual window or braille display. For example, the following command can generate a long alert history output:

  ```
  dcli -g cell_group -l root cellcli list alerthistory
  ```

  To display the output one screen-full at a time, pipe the output through the more command, as in the following:

  ```
  dcli -g cell_group -l root cellcli list alerthistory | more
  ```

  You can then use the space bar key to page through the output.
- When exachk or dbnodeupdate.sh is launched interactively, do not pipe its output to the more or page commands. As it runs, it displays informational messages on the terminal. The messages pause when exachk requires user input, then resume after input is received. Important messages, user input, errors, and check results are logged in various files. The results from exachk are written to an HTML report. All you need to do is to transfer the HTML report to a computer that runs your assistive technology and open the HTML report in a browser that you can access with your assistive technology.
- If you are running the patchmgr utility, and it is performing a task that takes some time to complete, the output displays a "spinner" and a countdown clock. The "spinner" cycles through the \, |, and / characters in-place, and the countdown clock is updated periodically. When the task is done, the output displays a "success" or "error" message, depending on the outcome. The output messages are also logged in a log file.
- A few recommended screen reader settings include the following (JAWS is used here just as an example):
  - Set the JAWS cursor to "All". Use the key combination of Insert + s until you hear "All".
  - You may need to turn off virtual cursor. If you are using JAWS, you can do this using the key combination of Insert + z.
  - Use the virtual window to capture text. If you are using JAWS, you can do this using the key combination of Insert + Alt + w.

D.2 Tips on Using Screen Magnifiers

Use the following tips when using screen magnifiers with Oracle Exadata System Software

Examples of screen magnifiers include ZoomText, MAGic, and SuperNova.

- Screen magnifiers can support both character-based terminals and X-Window-based VNC.

- If you are using the screen reader function of the screen magnifier (ZoomText screen reader), then you should use a character-based terminal as described above.
- If you are using a VNC, decide your preference for a window display, for example, TWM or ICE. A display setting for ICE can be done with the following:

  ```
  vncserver -geometry 1600x950 :2
  ```

  `1600x950` specifies the display size, and `:2` specifies the VNC display number.

D.3 Tips on Using Exawatcher Charts

Exawatcher is an Exadata specific tool that collects performance data from Exadata storage cells.

The data collected by Exawatcher can be graphed and presented in web pages. The web pages are based on Oracle JavaScript Extension Toolkit (JET), HTML and some use of ARIA. The Exawatcher pages can present charts which can be navigated using only a keyboard.

D.4 Oracle Exadata Deployment Assistant (OEDA) Web interface Accessibility

The OEDA Web interface conforms with the Web Content Accessibility Guidelines version 2.0 at the AA level (WCAG 2.0 AA).

The OEDA Web interface was built using Oracle JET, which provides support for:

- Keyboard and touch navigation

  Oracle JET components follow the Web Accessibility Initiative - Accessible Rich Internet Application (WAI-ARIA) guidelines. You can find the Oracle JET hotkey information at Oracle JavaScript Extension Toolkit (JET) Keyboard and Touch Reference.
- Zoom

  Oracle JET supports browser zooming up to 200%. For example, on the Firefox browser, you can choose **View**, then **Zoom**, and then **Zoom In**.
- Screen reader

  Oracle JET supports screen readers such as JAWS, Apple VoiceOver, and Google Talkbalk by generating content that complies with WAI-ARIA standards, and no special mode is needed.
- Oracle JET component roles and names

  Each component has an appropriate role, such as `button`, `link`, and so on, and each component supports an associated name (label), if applicable.
- Sufficient color contrast

  Oracle JET provides the Alta theme which is designed to provide a luminosity contrast ratio of at least 4.5:1.

Related Topics

- WAI-ARIA: Developing a Keyboard Interface

Index