Oracle® Database

High Availability Overview

11*g* Release 1 (11.1)

B28281-05

August 2011

ORACLE®

Oracle Database High Availability Overview, 11*g* Release 1 (11.1)

B28281-05

Primary Author: Lawrence To, Viv Schupmann

Contributors: Andrew Babb, Tammy Bednar, Larry Carpenter, Immanuel Chan, Ray Dutcher, Rajkumar Irudayaraj, Frank Kobylanski, Barb Lundhild, Rahim Mau, Joe Meeks, Valarie Moore, Michael Nowak, Darryl Presley, Ashish Ray, Michael T. Smith, Vinay Srihari, Lawrence To, Douglas Utzig, James Viscusi, Shari Yamaguchi, Frances Zhao

Contents

3 Determining Your High Availability Requirements

Preface

This book introduces you to Oracle best practices for deploying a highly available database environment. It provides an overview of high availability and helps you to determine your high availability requirements. It describes the Oracle Database products and features that are designed to support high availability and describes the primary database architectures that can help your business achieve high availability.

This preface contains these topics:

Audience

This book is intended for chief technology officers, information technology architects, database administrators, system administrators, network administrators, and application administrators who perform the following tasks:

- Plan data centers
- Implement data center policies
- Maintain high availability systems
- Plan and build high availability solutions

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 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

Knowledge of Oracle Database, Oracle RAC, and Oracle Data Guard concepts and terminology is required to understand the configuration and implementation details described in this book. For more information, see the Oracle Database documentation set. These books may be of particular interest:

- *Oracle Database High Availability Best Practices*

  This book typically lags behind the *Oracle Database High Availability Overview* because extensive testing is required to determine the best practices. Until the release 11.2 book is available, you may find some of the methodologies in the *Oracle Database High Availability Best Practices* for release 11.1.0.7 to be useful.
- *Oracle Database Administrator's Guide*
- *Oracle Database 2 Day + Real Application Clusters Guide*
- *Oracle Clusterware Administration and Deployment Guide*
- *Oracle Real Application Clusters Administration and Deployment Guide*
- *Oracle Automatic Storage Management Administrator's Guide*
- *Oracle Data Guard Concepts and Administration*
- *Oracle Database Backup and Recovery User's Guide*

Many books in the documentation set use the sample schemas of the seed database, which is installed by default when you install Oracle. See *Oracle Database Sample Schemas* for information about using these schemas.

Also, you can download the Oracle MAA best practice white papers at

`http://www.oracle.com/goto/maa`

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, 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.

What's New in High Availability?

Oracle Database 11*g* Release 1 (11.1) offers an integrated suite of high availability (HA) solutions and Maximum Availability Architecture (MAA) best practrices that provide:

- High availability to eliminate or minimize both planned and unplanned downtime
- Unmatched data protection
- Flexible deployment options that enhance overall performance and scalability
- Reduced cost using low cost platforms, without compromising HA
- Maximum return on investment by using all HA components for productive work at all times

For an overview of all of the high availability features available with the Oracle Database, see the "High Availability" chapter in *Oracle Database Concepts*.

For a list of new high availability features introduced in Oracle Database 11*g* Release 1 (11.1), see the "Availability" section in Chapter 1 of the *Oracle Database New Features Guide*.

1 Overview of High Availability

This chapter contains the following sections:

1.1 Introduction to High Availability

Databases and the Internet have enabled worldwide collaboration and information sharing by extending the reach of database applications throughout organizations and communities. This reach emphasizes the importance of high availability in data management solutions. Both small businesses and global enterprises have users all over the world who require access to data 24 hours a day. Without this data access, operations can stop, and revenue is lost. Users, who have become more dependent upon their solutions, now demand service-level agreements from their Information Technology (IT) departments and solution providers. Increasingly, availability is measured in dollars, euros, and yen, not just in time and convenience.

Enterprises have used their IT infrastructure to provide a competitive advantage, increase productivity, and empower users to make faster and more informed decisions. However, with these benefits has come an increasing dependence on that infrastructure. If a critical application becomes unavailable, then the business can be in jeopardy. Revenue and customers can be lost, penalties can be owed, and bad publicity can have a lasting effect on customers and a company's stock price. It is important to examine the factors that determine how your data is protected and maximize availability to your users.

1.2 What is Availability?

Availability is the degree to which an application, service, or function is accessible on demand. Availability is measured by the perception of an application's end user. End users experience frustration when their data is unavailable or the computing system is not performing within certain expectations, and they do not understand or care to differentiate between the complex components of an overall solution. Performance failures due to higher than expected usage create the same havoc as the failure of critical components in the solution.

Reliability, recoverability, timely error detection, and continuous operations are primary characteristics of a highly available solution:

- **Reliability**: Reliable hardware is one component of a high availability solution. Reliable software—including the database, Web servers, and applications—is just as critical to implementing a highly available solution. A related characteristic is resilience. For example, low-cost commodity hardware, combined with software such as Oracle RAC, can be used to implement a very reliable system, because the *resilience* of an Oracle RAC database allows processing to continue even though individual servers may fail.
- **Recoverability**: Because there may be many choices for recovering from a failure, it is important to determine what types of failures may occur in your high availability environment and how to recover from those failures in a timely manner that meets your business requirements. For example, if a critical table is accidentally deleted from the database, what action should you take to recover it? Does your architecture provide the ability to recover in the time specified in a service level agreement (SLA)?
- **Timely error detection**: If a component in your architecture fails, then fast detection is essential to recover from the unexpected failure. While you may be able to recover quickly from an outage, if it takes an additional 90 minutes to discover the problem, then you may not meet your SLA. Monitoring the health of your environment requires reliable software to view it quickly and the ability to notify the database administrator of a problem.
- **Continuous operation**: Providing the ability for continuous access to your data is essential when very little or no downtime is acceptable to perform maintenance activities. Activities, such as moving a table to another location in the database or even adding CPUs to your hardware, should be transparent to the end user in a high availability architecture.

More specifically, a high availability architecture should have the following traits:

- Tolerate failures such that processing continues with minimal or no interruption
- Be transparent to—or tolerant of—system, data, or application changes
- Provide built-in preventative measures
- Provide proactive monitoring and fast detection of failures
- Provide fast recoverability
- Automate detection and recovery operations
- Protect the data so that there is minimal or no data loss
- Implement the operational best practices to manage your environment
- Achieve the goals set in SLAs (for example, recovery time (RTO) and recovery point (RPO)) for the lowest possible total cost of ownership.

1.3 Importance of Availability

The importance of high availability varies among applications. However, the need to deliver increasing levels of availability continues to accelerate as enterprises re-engineer their solutions to gain competitive advantage. Most often, these new solutions rely on immediate access to critical business data. When data is not available, the operation can cease to function. Downtime can lead to lost productivity, lost revenue, damaged customer relationships, bad publicity, and lawsuits.

It is not always easy to place a direct cost on downtime. Angry customers, idle employees, and bad publicity are all costly, but not directly measured in currency. On the other hand, lost revenue and legal penalties incurred because SLA objectives are not met can easily be quantified. The cost of downtime can quickly grow in industries that are dependent on their solutions to provide service.

Other factors to consider in the cost of downtime are the maximum tolerable length of a single unplanned outage, and the maximum frequency of allowable incidents. If the event lasts less than 30 seconds, then it may cause very little impact and may be barely perceptible to end users. As the length of the outage grows, the effect may grow exponentially and result in a negative impact on the business. Alternatively, frequent outages, even if short in duration, may similarly disrupt business operations. When designing a solution, it is important to understand the true cost of downtime to understand how the business can benefit by availability improvements.

Oracle provides a range of high availability solutions that fit every organization regardless of size. Small workgroups and global enterprises alike are able to extend the reach of their critical business applications. With Oracle and the Internet, applications and data are reliably accessible everywhere, at any time.

1.4 Causes of Downtime

One of the challenges in designing a high availability solution is examining and addressing all of the possible causes of downtime. It is important to consider causes of both unplanned and planned downtime when designing a fault tolerant and resilient IT infrastructure. Planned downtime can be just as disruptive to operations, especially in global enterprises that support users in multiple time zones.

Table 1–1 describes unplanned outage types and provides examples of each type.

Table 1–1 Causes of Unplanned Downtime

Type	Description	Examples
Site failure	A site failure outage occurs when an event causes all or a significant portion of an application to stop processing or slow to an unusable service level. A site failure may affect all processing at a data center, or a subset of applications supported by a data center.	▪ Extended site-wide power failure ▪ Site-wide network failure ▪ Natural disaster making a data center inoperable ▪ Terrorist or malicious attack on operations or the site
Computer failure	A computer failure outage occurs when the system running the database becomes unavailable because it has crashed or is no longer accessible.	▪ Database system hardware failure ▪ Operating system failure ▪ Oracle instance failure ▪ Network interface failure
Storage failure	A storage failure outage occurs when the storage holding some or all of the database contents becomes unavailable because it has shut down or is no longer accessible.	▪ Disk drive failure ▪ Disk controller failure ▪ Storage array failure
Human error	A human error outage occurs when unintentional or malicious actions are committed that cause data in the database to become logically corrupt or unusable. The service level impact of a human error outage can vary significantly depending on the amount and critical nature of the affected data.	▪ File deletion (at the file system level). ▪ Dropped database object ▪ Inadvertent data changes ▪ Malicious data changes

Table 1–1 (Cont.) Causes of Unplanned Downtime

Type	Description	Examples
Data corruption	A corrupt block is a block that has been changed so that it differs from what Oracle Database expects to find. Block corruptions fall under two categories: physical and logical block corruptions: ■ In a **physical corruption**, which is also called a media corruption, the database does not recognize the block at all: the checksum is invalid, the block contains all zeros, or the header and footer of the block do not match. ■ In a **logical corruption**, the contents of the block are logically inconsistent. Examples of logical corruption include corruption of a row piece or index entry. Block corruptions can also be divided into interblock corruption and intrablock corruption: ■ In **intrablock corruption**, the corruption occurs in the block itself and can be either a physical or a logical corruption. ■ In an **interblock corruption**, the corruption occurs between blocks and can only be logical corruption. A data corruption outage occurs when a hardware, software or network component causes corrupt data to be read or written. The service level impact of a data corruption outage may vary, from a small portion of the database (down to a single database block) to a large portion of the database (making it essentially unusable).	■ Operating system or storage device driver failure ■ Faulty host bus adapter ■ Disk controller failure ■ Volume manager error causing bad disk read or writes ■ Software defects
Lost Writes	A lost write is another form of data corruption, but it is much more evasive to detect and repair quickly. A data block stray or lost write occurs when: ■ For a **lost write**, an I/O subsystem acknowledges the completion of the block write even though the write I/O did not occur in the persistent storage. On a subsequent block read on the primary database, the I/O subsystem returns the stale version of the data block, which might be used to update other blocks of the database, thereby corrupting it. ■ For a **stray write**, the write I/O completed but it was written somewhere else, and a subsequent read operation returns the stale value. ■ For an Oracle RAC system, a read I/O from one cluster node returns stale data after a write I/O is completed from another node (lost write). For example, this occurs if an NFS file system is mounted in Oracle RAC without disabling attribute caching (for example, without using the `noac` option). In this case, the write I/O from one node is not immediately visible to another node because it is cached.	■ Operating system or storage device driver failure ■ Faulty host bus adapter ■ Disk controller failure ■ Volume manager error ■ Other application software ■ Lack of Network File Systems (NFS) write visibility across a cluster
Hang or slow down	Hang or slow down occurs when the database or the application is unable to process transactions because of a resource or lock contention. Perceived hang can be caused by lack of system resources.	■ Database or application deadlocks ■ Runaway processes that consume system resources ■ Log on or system faults ■ Combination of application peaks with lack of system or database resources ■ Archived redo log destination or flash recovery area destination become full

Table 1–2 describes planned outage types and provides examples of each types.

Table 1–2 Causes of Planned Downtime

Type	Description	Examples
System and database changes	Planned system changes occur when performing routine and periodic maintenance operations and new deployments. Planned system changes include any scheduled changes to the operating environment that occur outside the organizational data structure in the database. The service level impact of a planned system change varies significantly depending on the nature and scope of the planned outage, the testing and validation efforts made before implementing the change, and the technologies and features in place to minimize the impact.	▪ Adding or removing processors to or from an SMP server ▪ Adding or removing nodes to or from a cluster ▪ Adding or removing disks drives or storage arrays ▪ Changing configuration parameters ▪ Upgrading or patching system hardware and software ▪ Upgrading or patching Oracle software ▪ Upgrading or patching application software ▪ System platform migration ▪ Database relocation ▪ Moving from 32 bits to 64 bits ▪ Migrating to cluster architecture ▪ Migrating to new storage
Data changes	Planned data changes occur when there are changes to the logical structure or physical organization of Oracle Database objects. The primary objective of these changes is to improve performance or manageability.	▪ Table definition changes ▪ Adding table partitioning ▪ Creating and rebuilding indexes
Application Changes	Planned application changes may include data changes and schema and programmatic changes. The primary objective of these changes is to improve performance, manageability, and functionality.	▪ Application upgrades

Oracle offers high availability solutions to help avoid both unplanned and planned downtime, and recover from failures. Chapter 2 discusses each of these high availability solutions in detail.

1.5 What Does This Book Contain?

Choosing and implementing the architecture that best fits your availability requirements can be a daunting task. This architecture must:

- Encompass redundancy across all components
- Provide protection and tolerance from computer failures, storage failures, human errors, data corruption, lost writes, system hangs or slowdown, and site disasters
- Recover from outages as quickly and transparently as possible
- Provide solutions to eliminate or reduce planned downtime
- Provide consistent high performance
- Be easy to deploy, manage, and scale
- Achieve SLA's at the lowest possible total cost of ownership

To help you select the most suitable architecture for your organization, this book describes several high availability architectures and provides guidelines for choosing

the one that best meets your requirements. Knowledge of the Oracle Database server, Oracle RAC, and Oracle Data Guard terminology is required to understand the configuration and implementation details.

Chief technology officers and information technology architects can benefit from reading the following chapters:

- Chapter 3, "Determining Your High Availability Requirements"
- Chapter 4, "High Availability Architectures and Solutions"

Database administrators and network administrators can find useful information in the following chapters:

- Chapter 2, "Oracle Database High Availability Features and Products"
- Chapter 4, "High Availability Architectures and Solutions"

> **See Also:** Oracle High Availability Best Practice recommendations in the:
>
> - *Oracle Database High Availability Best Practices*
> - The MAA white papers that can be downloaded from
>
>   `http://www.otn.oracle.com/goto/maa`

2

Oracle Database High Availability Features and Products

Oracle Database offers an integrated suite of high availability solutions that increase availability and eliminate or minimize both planned and unplanned downtime. These solutions help enterprises maintain business continuity 24 hours a day, seven days a week. However, the Oracle high availability solutions go beyond reducing downtime by providing solutions to increase system utilization on the primary and secondary systems and to help improve overall performance, scalability, and manageability.

The chapter contains the following sections that outline the key impacts of the Oracle high availability features on businesses and applications:

- Oracle High Availability Features and Solutions for Unplanned Downtime
- Oracle High Availability Features and Solutions for Planned Downtime
- Optimizing Grid Computing and Disaster Recovery Solutions
- Optimizing Manageability

2.1 Oracle High Availability Features and Solutions for Unplanned Downtime

Oracle provides the following features for high availability:

- Fast-Start Fault Recovery
- Oracle Real Application Clusters and Oracle Clusterware
- Oracle Data Guard
- Oracle Streams
- Transportable Technologies
- Oracle Flashback Technology
- Automatic Storage Management
- Recovery Manager
- Oracle Secure Backup
- Data Recovery Advisor
- Flash Recovery Area
- Oracle Security Features
- LogMiner

- Hardware Assisted Resilient Data (HARD) Initiative
- Data Block Corruption Prevention and Detection Parameters

Also, the "Oracle High Availability Solutions and Recovery for Unplanned Downtime" section on page 2-22 provides a summary of the key high availability solutions that address different types of unplanned downtime along with the recovery time for each solution.

See Also:

- The "High Availability" chapter in *Oracle Database Concepts*. for an overview of the high availability features
- The "Availability" section in the *Oracle Database New Features Guide*.for a list of all the new high availability features introduced in Oracle Database 11*g* Release 1 (11.1)

2.1.1 Fast-Start Fault Recovery

Oracle provides fast and predictable recovery from system faults and database failures. The Fast-Start Fault Recovery technology included in Oracle Database automatically bounds database recovery time at startup by using self-tuned checkpoint processing. This makes recovery time fast and predictable, and improves the ability to meet service-level objectives. The Oracle Fast-Start Fault Recovery feature can reduce recovery time on a heavily laden database from tens of minutes to a few seconds.

Fast-Start Fault Recovery features include:

- Predictable, bounded recovery from instance, database, and computer failures
- Database checkpointing that is self-tuning to maintain a desired recovery time objective

See Also: *Oracle Database Backup and Recovery User's Guide*

2.1.2 Oracle Real Application Clusters and Oracle Clusterware

Oracle Real Application Clusters (Oracle RAC) and Oracle Clusterware allow the Oracle Database to run any packaged or custom application across a set of clustered servers. This capability provides the highest levels of availability and the most flexible scalability. If a clustered server fails, Oracle Database continues running on the surviving servers. When more processing power is needed, you can add another server without interrupting access to data.

Oracle RAC enables multiple instances that are linked by an interconnect to share access to an Oracle database. In an Oracle RAC environment, the Oracle Database runs on two or more systems in a cluster while concurrently accessing a single shared database. The result is a single database system that spans multiple hardware systems and enabling Oracle RAC to provide high availability and redundancy during failures in the cluster. Oracle RAC accommodates all system types, from read-only data warehouse (DSS) systems to update-intensive online transaction processing (OLTP) systems.

Oracle Clusterware is software that, when installed on servers running the same operating system, enables the servers to be bound together to operate as if they are one server and manages the availability of user applications and Oracle databases. Oracle Clusterware also provides all of the features required for cluster management,

including node membership, group services, global resource management, and high availability functions:

- For high availability, you can place Oracle databases (single-instance or Oracle RAC databases), and user applications (Oracle and non Oracle) under the management and protection of Oracle Clusterware so that the databases and applications restart when a process fails or so that a failover to another node occurs after a node failure.
- For cluster management, Oracle Clusterware presents multiple independent servers as if they are a single-system image or one virtual server. This single virtual server is preserved across the cluster for all management operations, enabling administrators to perform installations, configurations, backups, upgrades, and monitoring functions once. Then, Oracle Clusterware automatically distributes the execution of these management functions to the appropriate nodes in the cluster.

Oracle Clusterware is a requirement for using Oracle RAC and it is the only clusterware that you need for most platforms on which Oracle RAC operates. Although the Oracle Database continues to support select third-party clusterware products on specified platforms, using Oracle Clusterware provides the benefit of dispensing with proprietary vendor clusterware and using an integrated software stack from Oracle that provides disk management with Oracle ASM to data management with the Oracle Database and Oracle RAC. In addition, Oracle Database features, such as Oracle Services, use the underlying Oracle Clusterware mechanisms to provide their capabilities.

Oracle Clusterware requires two clusterware components: a voting disk to record node membership information and the Oracle Cluster Registry (OCR) to record cluster configuration information. The voting disk and the OCR must reside on shared storage. The Oracle Clusterware requires that each node be connected to a private network over a private interconnect.

2.1.2.1 Benefits of Using Oracle Real Application Clusters and Oracle Clusterware

Together, Oracle RAC and Oracle Clusterware provide the following benefits:

- Ability to tolerate and quickly recover from computer and instance failures
- Ability to apply Oracle Clusterware upgrades, patch sets, and interim patches in a rolling fashion

  For example, upgrading Oracle Clusterware from Oracle 10*g* to Oracle 11*g*, patching Oracle Clusterware from Oracle 10.2.0.3 to 10.2.0.4, and patching Oracle Clusterware from Oracle 10.2.0.2 Bundle 1 to Oracle 10.2.0.2 Bundle 2
- Rolling upgrades for system and hardware changes
- Rolling patch upgrades for some interim patches
- Service relocation and, when you perform additional Fast Application Notification (FAN) and client configuration, distribution of FAN events so applications can react immediately to achieve fast, automatic, and intelligent connection and failover
- Fast, automatic detection of connection failures and removal of terminated connections for any Java application using Oracle Universal Connection Pool (UCP), Fast Connection Failover, and FAN events
- Work request balancing using Oracle UCP Runtime Connection Load Balancing
- Runtime connection load balancing with Oracle UCP, OCI, and ODP.NET

- Load balancing advisory
- Flexibility to increase processing capacity using commodity hardware without downtime or changes to the application
- Comprehensive manageability integrating database and cluster features
- Scalability across database instances

2.1.2.2 Benefits of Using Oracle Clusterware

Oracle Clusterware provides the following benefits

- Automatically restarts failed Oracle processes
- Automatically manages and fails over Oracle Virtual IP (VIP) on another node in the cluster on node failures
- Automatically restarts resources from failed nodes on surviving nodes
- For Oracle RAC databases, all Oracle processes are under the control of Oracle Clusterware by default; for Oracle single-instance databases, you can configure the Oracle processes into a resource group that is under the control of Oracle Clusterware
- For Oracle and non Oracle applications, Oracle Clusterware provides an application programming interface (API) that enables you to control other Oracle processes with Oracle Clusterware, such as restart or react to failures and certain rules
- Manages node membership and prevents split brain syndrome in which two or more instances attempt to control the database
- Provides the ability to perform rolling release upgrades of Oracle Clusterware, with no downtime for applications

> **See Also:** *Oracle Real Application Clusters Administration and Deployment Guide* and *Oracle Clusterware Administration and Deployment Guide*

2.1.3 Oracle Data Guard

Oracle Data Guard ensures high availability, data protection, and disaster recovery for enterprise data. Oracle Data Guard provides a comprehensive set of services that create, maintain, manage, and monitor one or more standby databases to enable Oracle databases to survive disasters and data corruptions. Oracle Data Guard maintains standby databases as transactionally consistent copies of the primary (production) database. Then, if the primary database becomes unavailable because of a planned or an unplanned outage, Oracle Data Guard can switch any standby database to the primary role, minimizing the downtime associated with the outage. Oracle Data Guard can be used with traditional backup, restoration, and cluster techniques to provide a high level of data protection and data availability.

With Oracle Data Guard, administrators can optionally improve primary database performance by offloading resource-intensive backup and reporting operations to standby systems.

An Oracle Data Guard configuration consists of one primary database and one or more standby databases. Using a backup copy of the primary database, you can create up to nine standby databases and incorporate them in a Oracle Data Guard configuration. Once created, Oracle Data Guard automatically maintains each standby

database by transmitting redo data from the primary database and then applying the redo to the standby database.

2.1.3.1 Types of Standby Databases

Similar to a primary database, a standby database can be either a single-instance Oracle database or an Oracle RAC database.

A standby database can be a physical standby database, a snapshot standby database, or a logical standby database, and a Oracle Data Guard configuration can include any combination of these types of standby databases:

- Physical standby database

  A physical standby database provides a physically identical copy of the primary database, with data files that are identical to the primary database. The database schema, including indexes, are the same. A physical standby database is kept synchronized with the primary database, though Redo Apply, which recovers the redo data received from the primary database and applies the redo to the physical standby database.

  You can use a physical standby database for business purposes other than disaster recovery. Starting in Oracle Database 11*g* Release 1, the physical standby database can be opened for read-only access while redo data is being applied to the standby database. This mode, referred to as the **Oracle Active Data Guard option**[1], allows users to access an up-to-date physical standby database for queries at any time. See Section 2.3.4.1, "Oracle Active Data Guard Option for Physical Standby Databases" on page 2-43 for more information.

  Also, you can convert a physical standby database to:

  – A logical standby temporarily, called a transient logical standby database, to perform a rolling upgrade.

    `http://www.otn.oracle.com/goto/maa`

  – A snapshot standby database temporarily, to be used as a clone or a test database.

    See Section 4.1.5.2, "Overview of Multiple Standby Database Architectures" on page 4-16, for more information.

    See Also:

    - See Section 4.1.5.2, "Overview of Multiple Standby Database Architectures" on page 4-16
    - The MAA white paper: "Database Rolling Upgrade Using Transient Logical Standby: Oracle Data Guard 11*g*" at `http://www.otn.oracle.com/goto/maa`.

- Snapshot standby database

  A snapshot standby database is an updatable standby database that you create from a physical standby database. A snapshot standby database receives and archives redo data received from the primary database, but the snapshot standby database does not apply redo data from the primary database while the standby is open for read/write access. Thus, the snapshot standby typically diverges from

[1] The Oracle Active Data Guard option is sometimes referred to as "real-time query standby"

the primary database over time. Moreover, local updates to the snapshot standby database cause additional divergence.

Redo data from the primary database is not applied until you convert the snapshot standby database back into a physical standby database, and after all local updates to the snapshot standby database are discarded. With a single command, you can revert a snapshot standby back to a physical standby database, at which time the changes made to the snapshot standby state are discarded, and Redo Apply automatically resynchronizes the physical standby database with the primary database using the redo data that was archived.

- Logical standby database

  A logical standby database contains the same logical information as the primary database, although the physical organization and structure of the data can be different. The logical standby database is kept synchronized with the primary database though SQL Apply, which transforms the redo data received from the primary database into SQL statements and then executes the SQL statements on the standby database.

  A key benefit of a logical standby database is that you can create significant auxiliary structures to optimize the reporting workload, including structures that could have a prohibitive effect on the transactional response time of the primary database. A logical standby database:

  – Can have its data physically reorganized into a different storage type with different partitioning having many different indexes, and having on-demand refresh materialized views created and maintained.

  – Can be used to drive the creation of data cubes and other OLAP data views.

  – Can be used for other business purposes in addition to satisfying disaster recovery requirements, allowing users to access a logical standby database for queries and reporting purposes at any time.

  – Can be used to upgrade Oracle Database software and patch sets with almost no downtime.

  Thus, you can use a logical standby database concurrently for data protection, reporting, and database upgrades.

2.1.3.2 Benefits of Using Oracle Data Guard and Standby Databases

Oracle Data Guard provides the following benefits:

- Maintenance of real-time, transactionally consistent database copies to provide protection against unplanned downtime and disaster.
- Data protection and fast repair against computer failures, human errors, data corruption, lost writes, and site failures.
- Automatic failover with flexible data protection levels to support all network configurations and business requirements.
- Faster redo application, redo transport, and role transitions with various enhancements.
- Reduction of planned downtime for system changes, some platform migrations, hardware and system upgrades, and Oracle patch set and database upgrades (see also Table 2–1, " Outage Types and Oracle High Availability Solutions for Unplanned Downtime").

- Multiple levels of data protection and performance to balance data availability against system performance requirements.
- Support for both physical standby databases (including the Active Data Guard option) and logical standby databases to provide more efficient use of system resources by diverting more querying and reporting functions from the primary database to standby databases (with the logical standby databases providing greater flexibility for any activity that requires access to a standby database that is open for read/write access). See also "Benefits of Physical Standby Databases" and "Benefits of Logical Standby Databases" on page 2-8, and Section 2.3.4.1, "Oracle Active Data Guard Option for Physical Standby Databases" on page 2-43.
- Support for snapshot standby database for reporting or testing (cloning) purposes and automatic resynchronization with the primary database once reporting or testing has completed. See also "Benefits of Snapshot Standby Databases" on page 2-7.
- Managed and automatic role transition and application notification to minimize planned and unplanned downtime.
- Automatic or automated resynchronization of a failed primary database following a failover.
- Management of all systems as a single configuration for simplified administration.
- Increased flexibility for Data Guard configurations where the primary and standby systems may have different CPU architectures, operating systems (for example, Windows and Linux), operating system binaries (32-bit and 64-bit), and Oracle database binaries (32-bit and 64-bit); this is subject to restrictions that are defined in support note 413484.1.

Benefits of Physical Standby Databases

- Guarantees a physical, block-for-block copy of the primary database
- Can be open for read-only queries while Redo Apply is active for real-time reporting (requires the Oracle Active Data Guard option that is described in Section 2.3.4.1, "Oracle Active Data Guard Option for Physical Standby Databases" on page 2-43)
- At role transition, offers assurance that the standby database is an exact replica of the old primary database
- Can be used to offload backups from primary database
- Very high performance, completely transparent to workload profile
- Has no data type restrictions
- Useful for minimizing downtime for many planned maintenance events

Benefits of Snapshot Standby Databases

- Inherits all the attributes of a physical standby database
- Can be open for read/write access and can process transactions independent of the primary database
- Protects the primary database the entire time it is open for read-write I/O
- Allows you to issue a single command to convert a Snapshot Standby back to a synchronized physical standby database
- Provides an ideal test system, especially when combined with Oracle Real Application Testing

Benefits of Logical Standby Databases

- Provides a logical, transaction-for-transaction copy of the primary database
- Allows creation of additional objects, modification of objects
- Provides the ability to skip apply on certain objects
- Supports real-time reporting
- Is open for read/write I/O (the data in tables that is maintained by SQL Apply cannot be changed)
- Performance varies depending on the workload
- Minimizes downtime for software upgrades

2.1.4 Oracle Streams

Oracle Streams is a very flexible and powerful database feature to implement fine-grained replication, multimaster replication, many-to-one replication, data transformation, hub and spoke replication, and message queuing.

Comparing Streams and Data Guard

Oracle Streams is designed for information sharing. It enables highly customized replication strategies to satisfy the many varied uses of data replicated to a target database. These same capabilities also make Oracle Streams a useful technology for addressing high availability and disaster recovery requirements and for minimizing planned downtime during upgrades to new database releases and patch sets.

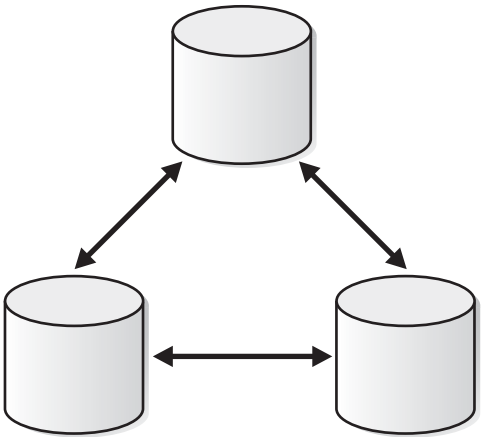
Oracle Data Guard is designed for simple, one-way replication of an entire database expressly for maintaining a synchronized copy that can assume the primary role in the event of a failure. Oracle Data Guard Redo Apply (physical standby) best exemplifies this notion of *simplicity*, as a disaster recovery solution that is both datatype and application agnostic, and able to scale to very high levels of performance. While Oracle Data Guard also provides capabilities that enable a standby database to offload from the primary database the overhead of performing backups, queries, and reports, these capabilities are ancillary to the primary mission of Oracle Data Guard, and are provided to increase your investments in high availability and disaster recovery. To get additional value from an Oracle Data Guard configuration, you can use Oracle Data Guard SQL Apply (logical standby database) to minimize planned downtime during upgrades to new database releases and patch sets.

Streams Messaging and Information Flow

Oracle Streams enables information sharing. Using Oracle Streams, each unit of shared information is called a message, and you can share these messages in a stream. The stream can propagate information within a database or from one database to another.

For example, Figure 2–1 shows an Oracle Streams multimaster configuration where all sites are directly connected to all other sites participating in the replication environment. The multimaster configuration enables data to be replicated between all locations in a near realtime manner.

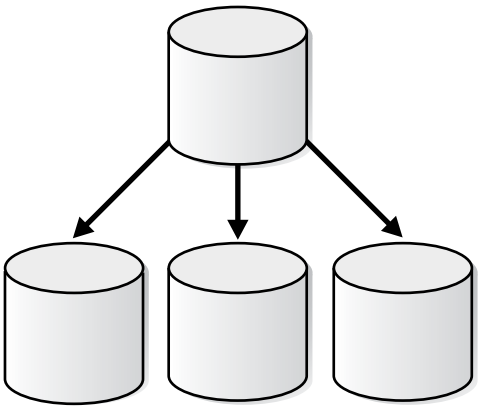
Figure 2–1 Oracle Streams Multimaster Configuration



Another example is the Oracle Streams 1-N, or *hub-and-spoke configuration* in which changes made at the primary or hub location are propagated to the remote or spoke locations in a near real-time manner.

Although it is possible to configure a hub-and-spoke configuration for bidirectional replication, you may prefer to restrict updates to a single location, the hub, as shown in Figure 2–2. In query intensive environments, you can still balance the load between multiple locations, with fast local access, while updates are restricted to the hub. By offloading reporting to the spoke locations, you improve performance at the hub, or primary OLTP location. This type of configuration is easier to implement than multimaster replication because it is not necessary to establish connectivity between all locations in the replication environment and it is not necessary to implement a conflict resolution strategy.

Figure 2–2 Information Dissemination with Oracle Streams (1-N configuration)

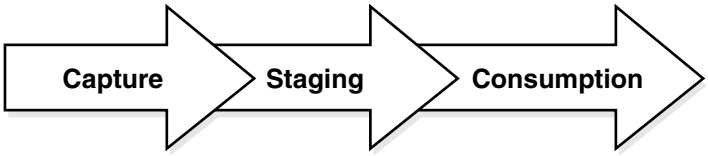


The stream routes specific information to specific destinations. The result is a feature that provides greater functionality and flexibility than traditional solutions for capturing and managing messages, and sharing the messages with other databases and applications. Oracle Streams provides the capabilities needed to build and operate distributed enterprises and applications, data warehouses, and high availability solutions. You can use all of the capabilities of Oracle Streams at the same time. If your business requirements change, then you can implement a new capability of Oracle Streams without sacrificing existing capabilities.

As with any Oracle Streams configuration, there are three phases: capture, stage (propagate), and consume (apply). Using Oracle Streams, you control what information is put into a stream, how the stream flows or is routed from database to database, what happens to messages in the stream as they flow into each database, and how the stream terminates. By configuring specific capabilities of Oracle Streams, you can address specific requirements. Based on your specifications, Oracle Streams can capture, stage, and manage messages in the database automatically, including, but not limited to, data manipulation language (DML) changes and data definition language (DDL) changes. You can also put user-defined messages into a stream, and Streams can propagate the information to other databases or applications automatically. When messages reach a destination, Streams can consume them based on your specifications.

The following figure shows the Oracle Streams information flow.

Figure 2–3 Oracle Streams Information Flow

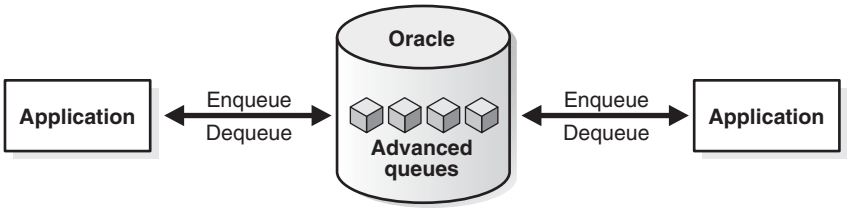


With Oracle Streams, you can create a local or remote copy of a production database. In the event of human error or a catastrophe, you can use the copy to resume processing. You can use Oracle Streams to configure flexible high availability environments.

You can use the features of Oracle Streams to achieve little or no database downtime during database upgrade and maintenance operations. Maintenance operations include migrating a database to a different platform, migrating a database to a different character set, modifying database schema objects to support upgrades to user-created applications, and applying an Oracle software patch.

Figure 2–4 shows an application that explicitly enqueues and dequeues messages through Oracle Streams Advanced Queuing as a method of sharing information with business partners or customers with different messaging systems. Once enqueued, messages can be transformed and propagated as desired, before being dequeued to a business partner's application that is a nondatabase oriented messaging systems.

Figure 2–4 Oracle Streams Message Queuing



Benefits of Using Oracle Streams

Oracle Streams provides the following benefits:

- Data protection by maintaining a full or partial remote copy of the database
- Achieves little or no downtime during database upgrade or maintenance operations such as migrating a database to a different platform or character set, modifying database objects to support upgrades to applications, and applying an Oracle software patch

- Data replication by capturing DML and DDL changes made to database objects and replicating these changes to one or more other databases; bidirectional replication environment, in which exactly two databases share the replicated database objects and data, is possible.
- Event management and notification by enqueuing messages or capturing events, propagating the messages and events through queues, and dequeuing and applying or acting upon the message or event (as shown in Figure 2–4)
- Supports heterogeneous platforms across databases in the configuration
- Allows character sets to differ between replicas
- Permits fine-grained control of data sharing

Note: Although Oracle Streams requires some initial implementation investment, it is well worth the effort because of the high flexibility you get with Streams.

See Also: *Oracle Streams Concepts and Administration*

2.1.5 Oracle Flashback Technology

Flashback technology provides a set of features to switch between views of the data as it existed at different points in time. Using Flashback features you can query past versions of schema objects and historical data. You can also perform change analysis and self-service repair to recover from logical corruption while the database is online.

Flashback technology provides a SQL interface to quickly analyze and repair human errors. Flashback technology provides fine-grained analysis and repair for localized damage such as deleting the wrong customer order. Flashback technology also enables correction of more widespread damage, yet does it quickly to avoid long downtime. Flashback technology is unique to Oracle Database and supports recovery at all levels including row, transaction, table, tablespace, and database.

Most of the flashback features use undo data while other features, such as Flashback Database and Block Media Recovery, use flashback logs:

- Undo tablespace—A dedicated tablespace that stores only undo information when the database is run in automatic undo management mode.
- Flashback data archive—An archive that is stored in a tablespace and contains transactional changes to every record in a table for the duration of the record's lifetime. The archived data can be retained for much longer duration than the retention period offered by an undo tablespace.
- Flashback logs—Oracle-generated logs used to perform flashback operations. The database can only write flashback logs to the flash recovery area. Flashback logs are written sequentially and are not archived. They cannot be backed up to disk.

The following list describes the Flashback features:

- **Oracle Flashback Query**

  Oracle Flashback Query provides the ability to view the data as it existed in the past by using the Automatic Undo Management system to obtain metadata and historical data for transactions. Undo data is persistent and survives a database malfunction or shutdown. The unique features of Flashback Query not only provide the ability to query previous versions of tables, they also provide a powerful mechanism to recover from erroneous operations.

Uses of Flashback Query include:

- Recovering lost data or undoing incorrect, committed changes. For example, rows that have been deleted or updated can be immediately repaired even after they have been committed.
- Comparing current data with the corresponding data at some time in the past. For example, using a daily report that shows the changes in data from yesterday, it is possible to compare individual rows of table data, or find intersections or unions of sets of rows.
- Checking the state of transactional data at a particular time, such as verifying the account balance on a certain day.
- Simplifying application design by removing the need to store certain types of temporal data. Using a Flashback Query, it is possible to retrieve past data directly from the database.
- Applying packaged applications, such as report generation tools, to past data.
- Providing self-service error correction for an application, enabling users to undo and correct their errors.

See Also: *Oracle Database Advanced Application Developer's Guide*

- **Oracle Flashback Versions Query**

  Oracle Flashback Versions Query is an extension to SQL that you can use to retrieve the versions of rows in a given table that existed in a specific time interval. Oracle Flashback Versions Query returns a row for each version of the row that existed in the specified time interval. For any given table, a new row version is created each time the `COMMIT` statement is executed.

  Flashback Versions Query is a powerful tool for the DBA to run analysis to determine the sources of problems. Additionally, application developers can use Flashback Versions Query to build customized applications for auditing purposes.

  See Also: *Oracle Database Advanced Application Developer's Guide*

- **Oracle Flashback Transaction**

  Oracle Flashback Transaction is a new feature in Oracle Database 11*g* Release 1 that can easily back out a transaction and its dependent transactions. The `DBMS_FLASHBACK.TRANSACTION_BACKOUT()` procedure rolls back a transaction and its dependent transactions while the database remains online. This recovery operation uses undo data to create and execute the compensating transactions that return the affected data to its original state. You can query the `DBA_FLASHBACK_TRANSACTION_STATE` view to see the current state of a transaction with respect to whether the transaction has been backed out using dependency rules or forced out by either:

  - Backing out nonconflicting rows
  - Applying undo SQL

  Oracle Flashback Transaction increases availability during logical recovery by easily and quickly backing out a specific transaction or set of transactions and their dependent transactions, with one command while the database remains online.

  See Also: *Oracle Database Advanced Application Developer's Guide*

- **Oracle Flashback Transaction Query**

  Oracle Flashback Transaction Query provides a mechanism to view all changes made to the database at the transaction level. When used in conjunction with Flashback Versions Query, it offers a fast and efficient means to recover from a human or application error. Flashback Transaction Query increases the ability to perform online diagnosis of problems in the database by returning the database user that changed the row, and performs analysis and audits on transactions.

  See Also: *Oracle Database Advanced Application Developer's Guide*

- **Oracle Flashback Table**

  Oracle Flashback Table recovers a table to a previous point in time. It provides a fast, online solution for recovering a table or set of tables that has been modified by a human or application error. In most cases, Flashback Table alleviates the need for administrators to perform more complicated point-in-time recovery operations. Even after a flashing back a table, the data in the original table is not lost; it can later be reverted back to the original state.

  See Also: *Oracle Database Backup and Recovery User's Guide*

- **Oracle Flashback Drop**

  Dropping objects by accident is a problem for database users and database administrators alike. While there is no easy way to recover dropped tables, indexes, constraints, or triggers, Oracle Flashback Drop provides a safety net when dropping objects. When you drop a table, Oracle automatically places it into the Recycle Bin. The Recycle Bin is a virtual container where all dropped objects reside. You can continue to query data in a dropped table.

  See Also: *Oracle Database Backup and Recovery User's Guide*

- **Oracle Flashback Restore Points**

  When an Oracle Flashback recovery operation is performed on the database, the DBA must determine the point in time—identified by the System Change Number (SCN) or timestamp—to which the data can later be flashed back. Oracle Flashback restore points are labels you can define that can be substituted for the SCN or transaction time used in Flashback Database, Flashback Table, and Recovery Manager (RMAN) operations. Furthermore, a database can be flashed back through a previous database recovery and open resetlogs by using guaranteed restore points. Guaranteed restore points allow major database changes—such as database batch jobs, upgrade, or patch—to be quickly undone by ensuring that the undo required to rewind the database is retained.

  Using the Oracle Flashback Restore Points feature provides the following benefits:

  – Provides the ability to quickly restore to a consistent state, to a point in time that was before a planned operation that has gone awry (for example, a failed batch job, an Oracle software upgrade, or an application upgrade.
  – Allows the snapshot standby to be resynchronized with the production database.
  – Allows for a quick mechanism to restore a test or cloned database back to its original state.

  See Also: *Oracle Database Backup and Recovery User's Guide*

- **Oracle Flashback Database**

  Oracle Flashback Database provides a more efficient alternative to database point-in-time recovery. With Oracle Flashback Database, current data files can be reverted to their contents at a past time. The result is much like restoring data from data file backups and executing point-in-time database recovery. However, Flashback Database skips the data file restoration and most of the application of redo data.

  Enabling Oracle Flashback Database provides the following benefits:

  - Eliminates the time to restore a backup when fixing human error that has a database-wide impact.
  - Because human errors can be quickly undone, it allows standby databases to use real-time apply to synchronize with the primary database.
  - Allows quick standby database reinstantiation after a database failover.

  See Also:

  - *Oracle Database Backup and Recovery User's Guide*
  - *Oracle Database SQL Language Reference*

- **Block Recovery Using Flashback Logs**

  Starting with Oracle Database release 11.1, block recovery can optionally retrieve a more recent copy of a data block from the flashback logs to reduce recovery time. Furthermore, a corrupted block encountered during instance recovery does not result in instance recovery failing. The block is automatically marked as corrupt and added to the RMAN corruption list in the `V$DATABASE_BLOCK_CORRUPTION` table. You can subsequently issue the RMAN `RECOVER BLOCK` command to fix the associated block.

  See Also: *Oracle Database Backup and Recovery User's Guide*

- **Flashback Data Archive**

  An archive that is stored in a tablespace and contains transactional changes to every record in a table for the duration of the record's lifetime. The archived data can be retained for a much longer duration than the retention period offered by an undo tablespace.

  See Also: *Oracle Database Advanced Application Developer's Guide*

2.1.6 Automatic Storage Management

Automatic Storage Management (ASM) provides a vertically integrated file system and volume manager directly in the Oracle kernel, resulting in:

- Significantly less work to provision database storage
- Higher level of availability
- Elimination of the expense, installation, and maintenance of specialized storage products
- Unique capabilities for database applications

For optimal performance, ASM spreads files across all available storage. To protect against data loss, ASM extends the concept of SAME (stripe and mirror everything)

and adds more flexibility in that it can mirror at the database file level rather than the entire disk level.

More importantly, ASM simplifies the processes of setting up mirroring, adding disks, and removing disks. Instead of managing hundreds and possibly thousands of files (as in a large data warehouse), DBAs using ASM create and administer a larger-grained object called a disk group. The disk group identifies the set of disks that are managed as a logical unit. Automation of file naming and placement of the underlying database files save administrators time and ensure adherence to standard best practices.

The ASM native mirroring mechanism (2-way or 3-way) is an option that protects against storage failures. With ASM mirroring, you can provide an additional level of data protection with the use of failure groups. A failure group is a set of disks sharing a common resource (disk controller or an entire disk array) whose failure can be tolerated. Once defined, an ASM failure group intelligently places redundant copies of the data in separate failure groups. This ensures that the data is available and transparently protected against the failure of any component in the storage subsystem.

ASM provides the following benefits:

- Provides the ability to mirror and stripe across drives and storage arrays
- Automatically re-mirrors from a failed drive to remaining drives
- Automatically rebalances stored data when disks are added or removed while the database remains online
- Allows for operational simplicity in managing database storage
- Provides local read capability, which gives better performance in an extended cluster
- Supports very large databases
- Supports ASM rolling upgrades
- Supports finer granularity in tuning and security
- ASM Fast Mirror Resync, which provides fast repair after a temporary disk failure

> **See Also:** *Oracle Automatic Storage Management Administrator's Guide*

2.1.7 Recovery Manager

Recovery Manager (RMAN) is an Oracle utility to manage database backup and, more importantly, the recovery of the database. RMAN eliminates operational complexity while providing superior performance and availability of the database.

RMAN determines the most efficient method of executing the requested backup, restoration, or recovery operation and then submits these operations to the Oracle Database server for processing. RMAN and the server automatically identify modifications to the structure of the database and dynamically adjust the required operation to adapt to the changes.

RMAN provides the following benefits:

- Automatic channel failover on backup and restore operations
- Automatic failover to a previous backup when the restore operation discovers a missing or corrupt backup
- Automatic creation of new database and temporary files during recovery

- Automatic recovery through a previous point-in-time recovery—recovery through resetlogs
- Block media recovery enables the data file to remain online while fixing the block corruption
- Fast incremental backups using block change tracking
- Fast backup and restore operations with intrafile and interfile parallelism
- Enhanced security with Virtual Private Catalog
- Lower space consumption when creating a database over the network by eliminating staging areas
- Merge incremental backups into image copies in the background providing up-to-date recoverability
- Optimized backup and restore of required files only
- Retention policy ensures that relevant backups are retained
- Ability to resume backup and restore of previously failed operations
- Automatic backup of the control file and the server parameter file ensuring that backup metadata is available in times of database structural changes and media failure and disasters
- Online backup does not require the database to be placed into hot backup mode

See Also: *Oracle Database Backup and Recovery User's Guide*

2.1.8 Oracle Secure Backup

Oracle Secure Backup is a centralized tape backup management solution providing performant, heterogeneous data protection in distributed UNIX, Linux, Windows, and Network Attached Storage (NAS) environments. By protecting file system and Oracle database data, Oracle Secure Backup provides a complete tape backup solution for your IT environment.

Oracle Secure Backup is tightly integrated with RMAN to provide the media management layer for RMAN, supporting releases since Oracle9*i*. With optimized integration points, Oracle Secure Backup and RMAN provide the fastest and most efficient tape backup capability for the Oracle database.

You can backup distributed servers to local and remote tape devices from a central Oracle Secure Backup administrative server using backup policies, calendar-based scheduling for *lights out* operations, or on-demand backup for immediate requirements. With its highly scalable client/server architecture, Oracle Secure Backup provides local and remote data protection, leveraging SSL for secure intradomain communication and two-way server authentication.

The following list describes the key benefits of Oracle Secure Backup:

- Optimized tape backup for the Oracle database by backing up only the currently used blocks and increasing backup performance by 10% to 25%.
- Policy-based management allows backup administrators to exercise precise control over the backup domain.
- Dynamic drive sharing for increased tape resource use.
- Heterogeneous storage area network (SAN) support allowing NAS, UNIX, Windows, and Linux to share tape drives and media.

- File system backup at the file, directory, file system or raw partition level with full, incremental and offsite backup scheduling.
- Integrated with Oracle Enterprise Manager, providing an intuitive, familiar interface.
- Backup encryption to tape.
- Broad tape-device support for new and legacy tape devices in SAN and SCSI environments.
- Network Data Management Protocol (NDMP) support for highly efficient backup of NAS filers.
- Scalable, low-cost licensing model reduces IT costs and operational considerations.

See Also: *Oracle Secure Backup Administrator's Guide*

2.1.9 Data Recovery Advisor

Data Recovery Advisor is a new feature in Oracle Database 11*g* that automatically diagnoses persistent (on disk) data failures, presents appropriate repair options, and runs repair operations at your request.

Note: The initial release of Data Recovery Advisor does not support Oracle RAC. In addition, while you can use Data Recovery Advisor when managing a primary database in a Data Guard configuration, you cannot use Data Recovery Advisor to troubleshoot a physical standby database. Data Recovery Advisor only takes the presence of a standby database into account when recommending repair strategies if you are using Enterprise Manager 11*g* Grid Control.

Data Recovery Advisor includes the following functionality:

- Failure Diagnosis

  The first symptoms of database failure are usually error messages, alarms, trace files and dumps, and failed health checks. Assessing these symptoms can be complicated, error-prone and time-consuming. Data Recovery Advisor automatically diagnoses data failures and informs you about them.

- Failure Impact Assessment

  After a failure is diagnosed, you must understand its extent and assess its impact on applications before devising a repair strategy. Data Recovery Advisor automatically assesses the impact of a failure and displays it in a easily understood format.

- Repair Generation

  Typically, Data Recovery Advisor presents several repair options, offering trade-offs in recovery time and potential data loss. If there are multiple failures present, you must also determine the best sequence of repair steps. In some situations it can be advantageous to consolidate repairs. Data Recovery Advisor does all this for you, automatically determining the best repair options.

- Repair Feasibility Checks

  Before presenting repair options, Data Recovery Advisor validates them with respect to the specific environment and availability of media components required to complete the proposed repair. The feasibility check is fast and validates if the

required backups are available. The actual contents of these backups will be validated during repair.

- Repair Automation

  If you accept the suggested repair option, Data Recovery Advisor automatically performs the repairs, verifies that the repair was successful, and closes the appropriate failures.

- Validation of Data Consistency and Database Recoverability

  Data Recovery Advisor can validate the consistency of your data, and backups and redo stream, whenever you choose.

- Early Detection of Corruption

  Through Health Monitor, you can schedule periodic runs of Data Recovery Advisor diagnostic checks to detect, analyze, and repair data failures before a database process executing a transaction discovers the corruption and signals an error. Early warnings can limit the damage caused by corruption.

- Integration of Data Validation and Repair

  Data Recovery Advisor is a single tool for data validation and repair.

See Also: "Diagnosing and Repairing Failures with the Data Recovery Advisor" in *Oracle Database Backup and Recovery User's Guide*

2.1.10 Flash Recovery Area

The flash recovery area is a unified storage location for all recovery-related files and activities in Oracle Database. After this feature is enabled, all RMAN backups, archived redo logs, control file autobackups, and data file copies are automatically written to a specified file system or automatic storage management disk group, and the management of this disk space is handled by RMAN and the database server.

Performing a backup to disk is faster because using the flash recovery area eliminates the bottleneck of writing to tape. More importantly, if database media recovery is required, then data file backups are readily available. Restoration and recovery time is reduced because you do not need to find a tape and a free tape device to restore the needed data files and archived redo logs.

The flash recovery area provides the following benefits:

- Unified storage location of related recovery files
- Management of the disk space allocated for recovery files, which simplifies database administration tasks
- Fast, reliable disk-based backup and restoration
- Ability to backup and restore the entire flash recovery area
- Ability to tolerate failures to the flash recovery area

See Also: *Oracle Database Backup and Recovery User's Guide*

2.1.11 Oracle Security Features

The best protection against human errors is to prevent their occurrence. The best way to prevent human errors is to restrict user access to only those data and services truly needed to perform business functions. Oracle provides a wide range of security tools to control access to application data by authenticating database users and then

enabling administrators to grant them only those privileges required to perform their duties.

In addition, the Oracle Database security model provides the ability to restrict data access at a row level using Virtual Private Database, thereby further isolating database users from data that they do not need to access.

Oracle security features include the following benefits:

- Authentication control to validate the identities of entities using networks, databases, and applications. Network sessions between databases, such as redo transport sessions, are also authenticated.
- Authorization control to provide limits to access and actions linked by database user identities and roles.
- Access control to objects, providing protection regardless of the entity seeking to access or alter them.
- Auditing control to monitor and gather data about specific database activities, investigate suspicious activity, deter users (or others) from inappropriate activities, and detect problems with authorization or access control implementation.
- Security policy management using profiles.
- Encryption of data residing in the database and backups, or transferred to and from databases.

> **See Also:** *Oracle Database Security Guide* and *Oracle Data Guard Concepts and Administration*

2.1.12 LogMiner

Oracle log files contain useful information about the activities and history of the Oracle database. Log files contain all data necessary to perform database recovery, and also record all changes made to the data and metadata in the database.

LogMiner is a fully relational tool that allows redo log files to be read, analyzed, and interpreted using SQL. Using LogMiner, you can analyze log files to:

- Track or audit changes to data
- Provide supplemental information for tuning and capacity planning
- Retrieve critical information for debugging complex applications
- Recover deleted data
- Provide additional browser-based simplification to help troubleshoot and resolve logical failures

LogMiner features include:

- Pinpointing when a logical corruption to the database—such as errors made at the application level—may have occurred
- Determining the necessary actions to perform fine-grained recovery at the transaction level
- Providing performance tuning and capacity planning through trend analysis
- Performing post auditing

> **See Also:** *Oracle Database Utilities*

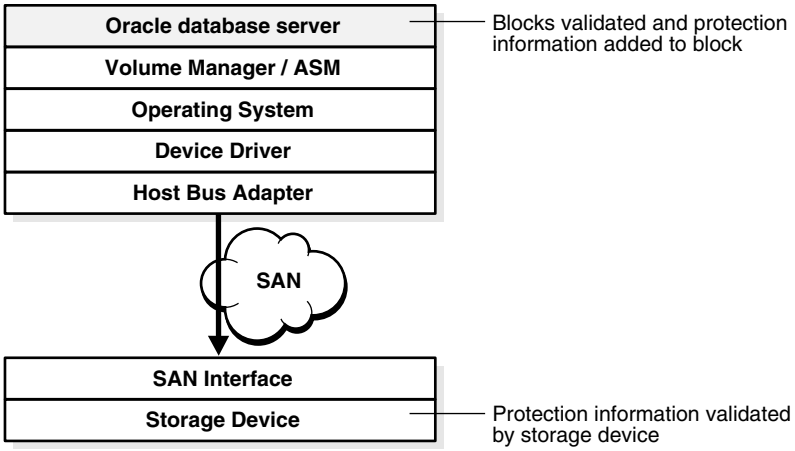
2.1.13 Hardware Assisted Resilient Data (HARD) Initiative

The Hardware Assisted Resilient Data (HARD) Initiative is an initiative between Oracle and hardware vendors to prevent data corruptions from being written out to disk. Data corruption is very rare, but when it happens, it can have a catastrophic effect on a database, and therefore a business.

Under the HARD Initiative, Oracle works with selected system and storage vendors to build operating system and storage components that can detect corruption early and prevent corrupted data from being written to disk. The key approach is block checking where the storage subsystem validates the Oracle block contents.

Any data files and log files located on HARD-compliant storage is protected. You must also enable the HARD validation feature on the storage, using the vendor-provided interface. When Oracle writes data to the storage, the storage system validates the data. If the data appears to be corrupted, then the write is either rejected with an error, or it is accepted with an error logged by the storage in the internal logs.

Figure 2–5 Oracle Data Validation



Storage vendors may choose to implement some or all of the checks in their implementation. Also, each vendor's implementation is unique and their control interfaces may have different features.

> **See Also:** The HARD initiative page for the latest vendor and implementation information at http://www.oracle.com/technology/deploy/availability/htdocs/HARD.html

2.1.14 Data Block Corruption Prevention and Detection Parameters

Before Oracle Database 11*g*, block corruptions detected by RMAN were recorded in `V$DATABASE_BLOCK_CORRUPTION`. In Oracle Database 11*g*, several database components and utilities in addition to RMAN can detect a corrupt block and record it in that view. Oracle Database automatically updates this view when block corruptions are detected or repaired (for example, using block media recovery or data file recovery). The benefit is that the time it takes to discover block corruptions is shortened.

In addition, you can use the `DB_ULTRA_SAFE` initialization parameter to automatically configure the appropriate data protection block checking level in the database. The performance impact may vary depending on the application and available system resources, but the effect can vary from 1% to 10%.

The DB_ULTRA_SAFE initialization parameter:

- Controls the setting of other related initialization parameters, including DB_BLOCK_CHECKING, DB_BLOCK_CHECKSUM, and DB_LOST_WRITE_PROTECT
- Controls other data protection behavior in the Oracle Database, such as requiring ASM to perform sequential mirror writes

By making it possible to detect data corruptions in a timely manner, these features provide critical high availability benefits for the Oracle database.

> **See Also:** *Oracle Database Reference* for more information about these views and initialization parameters

2.1.15 Oracle High Availability Solutions and Recovery for Unplanned Downtime

Oracle provides high availability solutions to prevent, tolerate and reduce downtime for all types of unplanned failures.

Table 2–1 describes the various Oracle high availability solutions for unplanned downtime along with the recovery time that can be attained with each solution. The table shows how the features discussed in Section 2.1.1 through Section 2.1.14 can be used to address various causes of unplanned downtime. Also, see Table 4–4 on page 4-33 for a summary of the attainable recovery times for all types of unplanned downtime for each Oracle high availability architecture.

Table 2–1 Outage Types and Oracle High Availability Solutions for Unplanned Downtime

Outage Scope	Oracle Solution	Benefits
Site Failures	Oracle Data Guard	▪ Fast-start failover and fast application notification with integrated Oracle clients.
Site Failures	Oracle Streams	▪ Online replica database resumes processing.
Site Failures	Recovery Manager	▪ Fully managed database recovery and integration with Oracle Secure Backup.
Computer Failures	Oracle Real Application Clusters and Oracle Clusterware	▪ Automatic recovery of failed nodes and instances. ▪ Fast application notification with integrated Oracle client failover.
Computer Failures	Fast-Start Fault Recovery	▪ Tunable and predictable cache recovery from computer failures.
Computer Failures	Oracle Data Guard	▪ Fast-start failover and fast application notification with integrated Oracle clients.
Computer Failures	Oracle Streams	▪ Online replica database resume processing.
Storage Failures	Automatic Storage Management	▪ Mirroring and online automatic rebalance places redundant copies of the data in separate failure groups.
Storage Failures	Oracle Data Guard	▪ Fast-start failover and fast application notification with integrated Oracle clients.
Storage Failures	Recovery Manager with Flash Recovery Area	▪ Fully managed database recovery and managed disk-based backups.
Storage Failures	Oracle Streams	▪ Online replica database resumes processing.
Data Corruption	Hardware Assisted Resilient Data (HARD) Initiative	▪ Corruption prevention in a storage array.

Table 2–1 (Cont.) Outage Types and Oracle High Availability Solutions for Unplanned Downtime

Outage Scope	Oracle Solution	Benefits
Data Corruption	Data Block Corruption Prevention and Detection Parameters Database initialization settings such as `DB_ULTRA_SAFE`, `DB_BLOCK_CHECKING`, `DB_BLOCK_CHECKSUM`	▪ Different levels of block corruption prevention and detection at the database level.
Data Corruption	Data Recovery Advisor and Recovery Manager with Flash Recovery Area	▪ Data Recovery Advisor automatically detects data corruptions and advises you about the best recovery plan. ▪ RMAN online block-media recovery time in Oracle Database 11*g* is faster, because RMAN can now use flashback logs to restore a more current copy of the data block for recovery.
Data Corruption	Oracle Data Guard	▪ Fast-start failover and fast application notification with integrated Oracle clients.
Data Corruption	Oracle Streams	▪ Online replica database resumes processing.
Human Errors	Oracle Security Features	▪ Restrict access as prevention.
Human Errors	Oracle Flashback Technology	▪ Fine-grained and database-wide rewind capability.
Human Errors	LogMiner	▪ Redo log analysis.

Table 2–1 (Cont.) Outage Types and Oracle High Availability Solutions for Unplanned Downtime

Outage Scope	Oracle Solution	Benefits
Lost writes	Oracle Data Guard, Recovery Manager, and the DB_LOST_WRITE_PROTECT initialization parameter Also, setting DB_ULTRA_SAFE to DATA_ONLY or DATA_AND_INDEX automatically enables DB_LOST_WRITE_PROTECT	▪ DB_LOST_WRITE_PROTECT initialization parameter provides lost write detection. ▪ If a lost write that occurred on the primary database is detected either by the physical standby database or during media recovery of the primary database, recovery is stopped to preserve the consistency of the database. However, failing over to the standby database using Data Guard will result in some data loss. ▪ If a lost write is detected on the standby database, you can restore the affected file and restart Redo Apply if the lost write is isolated and the hardware problem is corrected. **Note:** Lost writes can corrupt the entire database, which in many cases may required that you rebuild the affected database after resolving the hardware issue.
Lost writes	Hardware Assisted Resilient Data (HARD) Initiative	▪ Detection and prevention of stray or misdirected writes to another data file. Customers should check with their HARD-compatible storage vendor to learn whether the vendor has implemented this additional protection. For the most comprehensive lost write protection, use Oracle Data Guard. ▪ HARD does ***not*** detect a lost write in the following cases: *If any layer of software or hardware (host driver, volume manager, host bus adapter, storage array firmware) just acknowledges the write but did not issue it. *If the write was mistakenly written to a nondatabase file (for example, the write I/O was misdirected to the swap file). ▪ For the most comprehensive lost write protection, use Oracle Data Guard and set either the DB_ULTRA_SAFE parameter (to DATA_ONLY or DATA_AND_INDEX) or set the DB_LOST_WRITE_PROTECT parameter (to TYPICAL or FULL) on both the primary and standby databases.
Hangs or slow down	Oracle Database and Oracle Enterprise Manager	▪ Oracle Database attempts to resolve hangs automatically. ▪ Oracle Enterprise Manager or a customized application heartbeat can be configured to detect application or response time slowdown and react to these SLA breaches. For example, you can configure the Enterprise Manager Beacon to monitor and detect application response times. Then, after a certain threshold, Enterprise Manager can call the Oracle Data Guard DBMS_DG.INITIATE_FS_FAILOVER PL/SQL procedure to initiate a failover. See the section about "Application Initiated Fast-Start Failover" in *Oracle Data Guard Broker*.

2.2 Oracle High Availability Features and Solutions for Planned Downtime

Planned downtime can be just as disruptive to operations as unplanned downtime. This holds especially true for global enterprises that need to support users in multiple time zones, or for those that need to provide Internet access to customers 24 hours a day, seven days a week.

In the past, planned downtime became necessary when performing periodic maintenance or when migrating to new deployments. Periodic maintenance—such as patching or reconfiguring the system—may be necessary to update the database, application, operating system, middleware, or network. New deployments include major upgrades or new rollouts of the hardware, database, application, operating system, middleware, or network.

Oracle provides the following high availability solutions to eliminate or reduce planned downtime for system and database changes, data changes, and application changes:

- For system and database changes, see Dynamic Resource Provisioning on page 2-25
- For data changes, see Online Reorganization and Redefinition on page 2-36
- For application changes, see Online Application Maintenance and Upgrades on page 2-39

2.2.1 Dynamic Resource Provisioning

This section describes dynamic resource provisioning under the following topics:

- Dynamic Reconfiguration of the Database
- Autotuning Memory Management
- Automated Distribution of Data files, Control Files, and Log Files

2.2.1.1 Dynamic Reconfiguration of the Database

Oracle continues to broaden support for dynamic reconfiguration of the database, enabling it to adapt to changes in hardware demands without any service interruptions. Oracle Database dynamically accommodates various changes to hardware and database configurations:

- Add and remove processors from an SMP server
- Add and remove nodes and instances in an Oracle RAC environment
- Dynamically grow and shrink its shared memory allocation and automatically tune memory online using Automatic Shared Memory Management
- Add and remove database disks online without disturbing database activities using Automatic Storage Management (ASM)
- Add and remove storage arrays online without disturbing database activities using ASM
- Automatically rebalance the I/O load across the database storage using ASM
- Move data files online when adding or dropping disks using ASM, which automatically rebalances database storage whenever the storage configuration is changed

- Change almost all initialization parameters without shutting down the instance by using the SQL*Plus `ALTER SESSION` statement to change the value of a parameter during a session, or the `ALTER SYSTEM` statement to change the value of a parameter in all sessions of an instance for the duration of the instance

These capabilities provide no-cost system changes and capacity on-demand provisioning, both of which are fundamental requirements of enterprise Grid computing.

2.2.1.2 Autotuning Memory Management

Beginning with Oracle Database 11*g* two memory management initialization parameters, `MEMORY_TARGET` and `MEMORY_MAX_TARGET` enable automatic management of the system global area (SGA), program global area (PGA), and other memory required to run Oracle Database.

> **Note:** `MEMORY_MAX_TARGET` is the value up to which `MEMORY_TARGET` can grow dynamically. If these initialization parameters are left at their default values (`0`), then Oracle Database does not autotune memory. If one parameter is set to a nonzero value and other is not set, then Oracle Database internally sets both parameters to the nonzero value.

Oracle Database uses a noncentralized policy to free and acquire memory in each subcomponent of the SGA and the PGA. Oracle Database autotunes memory by prompting the operating system to transfer granules of memory from less needy to more needy components. The granularity of the memory transfer is dependent on the current free memory and the amount of memory the operating system requires to maintain a basic level of service.

> **Note:** Automatic memory management with the `MEMORY_TARGET` and `MEMORY_MAX_TARGET` initialization parameters is supported on Linux, Windows, Solaris, HP-UX, and AIX. See the *Oracle Database Concepts* and *Oracle Database Administrator's Guide* for more information about all supported platforms.

2.2.1.3 Automated Distribution of Data files, Control Files, and Log Files

ASM automates and simplifies the layout of data files, control files, and log files. Database files are automatically distributed across all available disks. Database storage is rebalanced whenever the storage configuration changes, including adding and removing disks or storage arrays. ASM provides redundancy through the mirroring of database files, and provides optimal performance by automatically striping database files across available disks.

> **See Also:** *Oracle Database Concepts* and *Oracle Automatic Storage Management Administrator's Guide* for more information about ASM

2.2.2 Oracle High Availability Solutions and Recovery Times for Planned Downtime

Oracle provides high availability solutions to prevent, tolerate and reduce downtime for all types of planned maintenance. Table 2–2 describes the various Oracle high availability solutions for planned downtime along with the outage time that can be attained with each solution and their known considerations. In all cases, Oracle recommends you perform extensive testing before performing any rolling upgrade.

See Also: Table 4–5 on page 4-33 for a summary of the attainable recovery times for all types of planned downtime for each Oracle high availability architecture.

Table 2–2 Oracle High Availability Solutions for Planned Downtime

Maintenance Type	Oracle Recommended Solution	Solution Description	Outage Time
Operating system and hardware upgrades	Oracle Real Application Clusters and Oracle Clusterware	Section 2.2.2.1	No downtime
Oracle interim patches	Oracle Real Application Clusters (Oracle RAC)	Section 2.2.2.3	No downtime[1]
Online patches for debug and interim patches where the scope of the upgrade is small	Online Patching	Section 2.2.2.4	No downtime
Oracle Clusterware upgrades and patches	Cluster Ready Services (CRS)	Section 2.2.2.5	No downtime
ASM upgrades	Automatic Storage Management	Section 2.2.2.6	No downtime
Storage migration[2]	Automatic Storage Management	Section 2.2.2.7	No downtime
Migrating to ASM or migrating a single-instance database to Oracle RAC	Oracle Data Guard	Section 2.2.2.2	Seconds to minutes
Patch set and database upgrades	Oracle Data Guard using SQL Apply and logical standby databases	Section 2.2.2.8	Seconds to minutes
Platform Migration Across Windows and Linux Platforms	Oracle Data Guard	Section 2.2.2.8	Seconds to minutes
Platform Migration across same endian format platforms	Transportable Database	Section 2.2.2.9	Minutes to hours
Platform migration across different endian format platforms	Transportable Tablespace	Section 2.2.2.10	Minutes to hours
Application upgrades	Online Application Maintenance and Upgrades	Section 2.2.5	

[1] Patches that cannot be applied by performing a rolling upgrade can be applied with the `MINIMIZE_DOWNTIME` option of the OPatch utility to reduce the availability impact of the patch application.

[2] An example is migration from traditional storage to low-cost storage

See Also:

- *Oracle Data Guard Concepts and Administration* for more information about using Data Guard with SQL Apply to upgrade an Oracle database
- *Oracle Database Concepts* and *Oracle Database Administrator's Guide* for more information about transportable tablespace
- The MAA white papers about rolling upgrade best practices at

  `http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm`

2.2.2.1 Avoiding Downtime During Operating System and Hardware Upgrades

Using Oracle RAC is the recommended solution for avoiding downtime during system and hardware upgrades.

If you cannot perform the upgrade using Oracle RAC, then the recommended solutions is to use Oracle Data Guard and physical standby databases. See Section 2.2.2.2 for more information.

Oracle RAC Solution Description

Perform the following steps:

1. Stop the application service.

   This implicitly redirects connections off of the target instance when using FAN.

2. Shut down target instance or instances with the `IMMEDIATE` option.

3. Shut down and disable Oracle Clusterware.

   Disabling Oracle Clusterware prevents it from starting automatically.

4. Perform maintenance.

5. Enable and start Oracle Clusterware.

   This step implicitly starts the database instances.

6. Start the application service.

   This step implicitly redirects connections back on to the target instance when using FAN.

7. Repeat all steps on the next node.

Additional Considerations

Verify the following:

- Ensure the planned maintenance can be done in a rolling fashion from an operating system perspective.
- Ensure the database and clusterware versions are certified with the new system and hardware changes.

> **See Also:** Your operating system-specific Oracle Real Application Clusters Installation Guide

2.2.2.2 Using Oracle Data Guard for System and Cluster Upgrades and Migrations

Oracle Data Guard and physical standby databases are the recommended solution for performing system and cluster upgrades that are not upgradeable using Oracle RAC rolling upgrades. Oracle Data Guard is also recommended for migrations to ASM, Oracle RAC, 64-bit systems, Windows to Linux or Linux to Windows, or same processor architecture platforms. For example:

- Use Oracle Data Guard for system upgrades that cannot be upgraded using Oracle RAC rolling upgrades due to system restrictions.
- Use Oracle Data Guard when migrating to ASM, from a noncluster environment to Oracle RAC, to a different platform with the same endian format or to a different platform with the same processor architecture.

In general, you first upgrade the physical standby database and then perform a Data Guard switchover to the physical standby database, as follows:

1. Upgrade the system or change the physical standby database system to your target environment.

For example, you can convert the standby database from a single-instance database to an Oracle RAC database by using ASM, without any impact on the primary database. Then, restart the standby database, ensure that it matches your target environment, and wait for Redo Apply to finish applying all redo data to the standby database.

2. Perform a Data Guard switchover—optimally the switchover should take only seconds to minutes.
3. Shut down the original primary database (now the standby database).
4. Upgrade or make system changes to the original primary database.
5. Restart the upgraded database as a standby database and allow recovery to automatically synchronize the databases.
6. Optionally, perform a Data Guard switchover to return the standby database to the primary database role.

Additional Considerations

- For fastest switchover, configure the standby database to use real-time apply and, if possible, ensure the databases are synchronized before the switchover operation.
- Use this approach if Oracle RAC rolling upgrade or online patching is not possible. See *Oracle Data Guard Concepts and Administration* for more information.
- The conversion from 32 to 64-bit is automatic if you are applying an Oracle Database patch set or doing an Oracle Database upgrade at the same time. If you are upgrading only the operating system, you may need to perform additional post-upgrade steps that are described in support note 414043.1. Also, see the *Oracle Database Upgrade Guide* for more information about upgrades.

2.2.2.3 Oracle Interim Database Patches

Use Oracle RAC to avoid downtime when applying Oracle interim database patches. You can apply approximately 90% of the new patches using Oracle RAC.

If you cannot apply patches using Oracle RAC, then use Oracle Data Guard and physical standby databases. See Section 2.2.2.2 for more information.

Solution Description

Oracle interim (one-off) patches to database software are usually applied to implement known fixes for software problems, or to apply diagnostic patches to gather information about a problem. Plan to apply patches during a schedule maintenance outage.

Oracle provides the capability to do rolling patch upgrades with Oracle RAC with little or no database downtime using the `opatch` command-line utility.

An Oracle RAC rolling upgrade enables all but one of the instances of the Oracle RAC installation to be available during the scheduled outage, further reducing the impact on the application downtime required for scheduled outages. The Oracle opatch utility enables you to apply the patch successively to the different instances in an Oracle RAC installation.

Additional Considerations

Performing a rolling upgrade is possible only for patches that are certified for rolling upgrades. Typically, patches that can be installed in a rolling upgrade include:

- Patches that do not affect the contents of the database, such as the data dictionary

- Patches not related to Oracle RAC internode communication
- Patches related to client-side tools such as SQL*Plus, Oracle utilities, development libraries, and Oracle Net
- Patches that do not change shared database resources, such as data file headers, control files, and common header definitions of kernel modules

Do not use Oracle RAC to perform rolling upgrades of patch sets.

> **See Also:** Your operating system-specific Oracle Real Application Clusters Installation Guide

2.2.2.4 Online Patching

Using Online Patching is the recommended solution for avoiding downtime when an online patch is available.

Solution Description

Online patches are a special type of interim patch that you can apply while the instance remains online.

Oracle provides the capability to perform online patching with any Oracle database using the opatch command-line utility.

Additional Considerations

- Oracle provides online patches when the changed code is small in scope and complexity, such as with diagnostic patches or small bug fixes.
- Oracle provides online patches when the patch does not change shared memory structures in the System Global Area (SGA), or other critical internal code structures.
- Applying an online patch increases memory consumption on the system because each Oracle process uses more memory from the Program Global Area (PGA) during the patch application. Take memory requirements into consideration before you begin applying an online patch. Each online patch is unique and the memory requirements are patch specific. As is always the case, the best practice is to apply the patch on your test system first. Doing so also enables you to assess the effect of the online patch on your production system and estimate any additional memory usage.

> **See Also:** *Oracle Universal Installer and OPatch User's Guide for Windows and UNIX* for information about online patching and OPatch, and see *Oracle Database Upgrade Guide* for an overview of rolling upgrades and rolling patches

2.2.2.5 Upgrading Oracle Clusterware

Performing rolling upgrades of the Oracle Clusterware using Cluster Ready Services (CRS) software is the recommended solution for avoiding downtime when upgrading Oracle Clusterware.

Solution Description

You can perform all upgrades to Oracle Clusterware in a rolling fashion.

> **See Also:** Your operating system-specific Oracle Clusterware installation guide

2.2.2.6 Upgrading Automatic Storage Management (ASM)

Performing rolling upgrades is the recommended solution for upgrading ASM.

Solution Description

You can perform all upgrades starting with Oracle Database 11*g* (and later releases) in a rolling fashion.

> **See Also:** *Oracle Automatic Storage Management Administrator's Guide*

2.2.2.7 Storage Migration

Using ASM is the recommended solution for performing storage migrations.

Solution Description

ASM enables you to add all disks in one storage array and subsequently drop all disks from another array. ASM automatically rebalances and migrates data to the new storage while the database remains operational.

Additional Considerations

Before removing the source storage array, ensure that the rebalancing is complete.

> **See Also:** The chapter about performing ASM Data Migration in the *Oracle Database Backup and Recovery User's Guide*

2.2.2.8 Patch Set and Database Upgrades

Oracle Data Guard using SQL Apply is the recommended solution for performing patch set and database upgrades with minimal downtime. Section 2.2.2.8.1 describes this solution. If the source database is using data types not supported by SQL Apply, you can use Extended Datatype Support (EDS) to accommodate several more advanced data types.

If the source database is using a software version not supported by SQL Apply rolling upgrade (earlier than Oracle Database release 10.1.0.3) and using EDS cannot sufficiently resolve SQL Apply data type conflicts, then consider using Database Upgrade Assistant (DBUA)[2] or transportable tablespace. DBUA provides a graphical user interface (GUI) utility that guides you through the upgrade process and is the simplest and recommended method of upgrading a database. However, if the time it takes DBUA to upgrade a database does not fit in the defined maintenance window, then consider using transportable tablespaces to perform a database upgrade in less than one hour.

Use transportable tablespaces if you cannot use SQL Apply but the maintenance window requires downtime to be less than an hour in duration, and the database being upgraded has a small number of simple schemas and data files that do not need to be transferred as part of the transport process (such as when the data files will be used in place). Section 2.2.2.8.2 describes the transportable tablespace solution.

Finally, Oracle Streams is the solution that provides the most flexibility when performing database upgrades and additional data type support. Section 2.2.2.8.3 describes this solution.

[2] DBUA incurs downtime. The amount of downtime is dependent on a number of factors. See *Oracle Database High Availability Best Practices* for additional considerations when choosing DBUA as an upgrade option. See *Oracle Database Upgrade Guide* for instructions on using DBUA to upgrade Oracle Database software.

See Also: *Oracle Database High Availability Best Practices* for more information and for help choosing the database upgrade method appropriate for your configuration

2.2.2.8.1 Solution Description for Database Upgrades Using Data Guard and SQL Apply

Follow these steps to leverage Data Guard using SQL Apply to upgrade an Oracle database:

1. Upgrade logical standby database to the new release and evaluate the change.
2. Ensure that SQL Apply has applied all redo data to the logical standby database.
3. Disconnect applications.
4. Perform Data Guard switchover.
5. Reconnect applications to the new primary database.
6. Shut down the original primary database (now the logical standby database).
7. Execute database software upgrade steps on the new standby database.
8. Restart the standby database and allow recovery to synchronize.
9. Optionally perform a Data Guard Switchover to return to the original database.

Additional Considerations

- SQL Apply rolling upgrades are only supported for Oracle Database release 10.1.0.3 and higher. For complete information, see the chapter about using SQL Apply to upgrade the Oracle Database in *Oracle Data Guard Concepts and Administration*.
- SQL Apply has some data type restrictions (see *Oracle Data Guard Concepts and Administration* for a list of the restrictions). If there are data type restrictions, consider implementing Extended Datatype Support (EDS).

  EDS enables SQL Apply to replicate changes to tables that contain some data types not natively supported from one database to another. Beginning with Oracle Database 10*g* Release 2 (10.2.0.4) Patch Set 3, SQL Apply supports the ability for triggers to fire on the logical standby database, which provides the basis of EDS. For an overview of EDS, see the MAA white paper "Extended Datatype Support" available at

  `http://www.oracle.com/technology/deploy/availability/pdf/maa_edtsoverview.pdf`

  For examples using EDS to support data types that are not natively supported by SQL Apply, see support note 559353.1.
- Oracle Data Guard is the best approach if performing an Oracle RAC rolling upgrade is not possible and there are no data type restrictions.

See Also: The MAA white paper: "Rolling Database Upgrades Using Data Guard SQL Apply" available at `http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm`

2.2.2.8.2 Solution Description for Database Upgrades Using Transportable Tablespaces

If you cannot use Data Guard SQL Apply because of data type conflicts and testing shows that upgrading with DBUA cannot meet uptime requirements, then consider using transportable tablespace to upgrade your database.

The following high-level steps leverage the transportable tablespace feature to upgrade an Oracle database:

1. Install the Oracle Database software on the target system and perform initial steps on the source database to prepare for the transport process.
2. Prepare the source and target database:
   a. Gather information from the source database.
   b. Create the target database with Database Configuration Assistant (DBCA).
   c. Prepare the target database for Data Pump usage and to accept the tablespaces being transported.
3. Perform the transport:
   a. Ready the source database for transport by disconnecting users and restricting access to source database, make all user tablespaces `READ ONLY`, and capture sequence starting values from the source database.
   b. Stop Redo Apply and shut down the standby database.
   c. Transport the user tablespaces.
4. Verify that the target database is complete and functional, and then backup the target database.

> **See Also:** The MAA white paper "Database Upgrade Using Transportable Tablespaces" available at http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm

Additional Considerations

- The transportable tablespaces feature is an option for performing database upgrade in less than one hour for databases that have simple schemas and where the data files do not need to be transferred as part of the transport process (such as when the data files will be used in place). See the MAA white paper "Database Upgrade using Transportable Tablespaces" available on the MAA Web site at

  http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm

- Using transportable tablespaces reduces database upgrade time by moving all user tablespaces from a database running an earlier software release to an empty target database running a current software release. With transportable tablespaces, tablespace data files are plugged into the database by copying the data files to the target database, then importing the object metadata into the target database.

2.2.2.8.3 Solution Description for Database Upgrades Using Streams Streams is similar in function to Data Guard SQL Apply but provides added flexibility if your database includes data types that SQL Apply does not support. Like SQL Apply, Streams can take advantage of Extended Datatype Support (EDS) to replicate changes to tables that contain some data types not natively supported from one database to another.

The following high-level steps describe how to perform a database upgrade:

1. Before you begin the upgrade process, see *Oracle Streams Concepts and Administration* for information about how to perform a database upgrade on a database that has user-defined types.
2. Create a duplicate database. (Ideally the replica will start out as a physical standby database that is up to date.)

3. Activate and upgrade the database to the later version.
4. Enable Oracle Streams replication.
5. During the upgrade of the replica, the source database continues ahead. Once the replica is caught up, perform a switchover.

> **See Also:** *Oracle Streams Concepts and Administration* for complete information about online database upgrade with Oracle Streams

2.2.2.9 Platform Migration Across Same Endian Format Platforms

Consider that following approaches when performing platform migrations across same endian format platforms:

- Oracle Data Guard (physical standby database) is the recommended solution for performing platform migration across Linux and Windows platforms. Section 2.2.2.2 on page 2-28 describes this solution.
- If cross-platform physical standby database is not available for the platform combination to be migrated, then use transportable database. Section 2.2.2.9.1, "Solution Description for Platform Migration Using Transportable Database" on page 2-34 describes this solution.
- If transportable database cannot perform the migration quickly enough, then use Oracle Streams. Section 2.2.2.9.2, "Solution Description for Platform Migration Using Oracle Streams" on page 2-35 describes this solution.

2.2.2.9.1 Solution Description for Platform Migration Using Transportable Database

Transportable database should be used for platform migration only when cross-platform physical standby database or logical standby database is not supported for the platform combination in question[3].

For example, if you want to move from Windows x86-64 to Linux x86-64, it is best to use cross-platform standby database instead of transportable database. There is less downtime (simply the time it takes to switchover) and it is possible to run the standby database on the new platform for a period of time to ensure that everything is working as planned.

The high-level steps (with target system conversion) are as follows:

1. Place the source database in read/only mode
2. Run RMAN `CONVERT DATABASE` command
3. Move files to the target system
4. Run RMAN generated script to convert data files with `UNDO` to target platform format
5. Run RMAN generated script to complete the migration

When using transportable database, the downtime required for a platform migration is determined by the time needed to:

- Place the source database in read-only mode
- Convert data files that contain `UNDO` to the new platform format (data files without `UNDO` do not require conversion)

[3] Beginning with Oracle Database 11*g*, the primary and standby systems in a Data Guard configuration can have different CPU architectures, operating systems (for example, Windows and Linux), operating system binaries (32-bit and 64-bit), and Oracle database binaries (32-bit and 64-bit). For the latest capabilities and restrictions, see support note 413484.1.

- Transfer all data files from the source system to the target system

  You can significantly minimize this time by using a storage infrastructure that can make the data files available to the target system without the need to physically move the files.

- Invalidate and recompile all PL/SQL using SQL scripts `utlirp.sql` and `utlrp.sql`

  See Also: The "Platform Migration using Transportable Database" white paper available at `http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm`

2.2.2.9.2 Solution Description for Platform Migration Using Oracle Streams Oracle Streams enables updates on the multiple masters and provides support for heterogeneous platforms with different database releases. Therefore, Oracle Streams may provide the fastest approach for database upgrades and platform migration.

Oracle Streams has data type limitations and restrictions, such as for advanced queue and object types. But in some cases you can work around the limitations by creating shadow tables on the source database. You can create a trigger on tables with unsupported data types to capture and propagate changes to tables with supported data types. Those changes are replicated by Streams to the target database. You can customize the apply mechanism to apply the changes to the original tables in the target database.

Oracle Streams implementations require additional administrative effort for testing, setup, and configuration because Streams is designed to be a more flexible architecture.

The following high-level steps describe how to perform a platform migration with Oracle Streams:

1. Set up the Streams environment on the source database.
2. Instantiate the replica database (target database) using the new target version or on the target platform.
3. Set up the Streams environment on the target database.
4. Enable Streams to propagate all changes made on the source database to the target database to completely synchronize the target database with the source.
5. Connect users to target database and shutdown source database.
6. Remove the Streams configuration.

See Also: *Oracle Streams Concepts and Administration*

2.2.2.10 Platform Migration Across Different Endian Format Platforms

Consider the following approaches when performing platform migrations on different endian format platforms:

- Transportable tablespace is the recommended solution for performing platform migration across different endian format platforms and reduces downtime significantly. See the "Solution Description for Transportable Tablespace" section on page 2-36 for more details.
- Oracle Data Pump is the simplest of all the approaches. See the *Oracle Database Utilities* for complete information about using Oracle Data Pump.

- For planned downtime of potentially seconds, consider using Oracle Streams as described in Section 2.2.2.9.2, "Solution Description for Platform Migration Using Oracle Streams" on page 2-35.

Solution Description for Transportable Tablespace

Migrating a database to a new platform using a different endian format with transportable tablespaces requires the following high level steps:

1. Create a new, empty database on the target platform.
2. Import objects required for transport operations from the source database into the target database.
3. Export transportable metadata for all user tablespaces from the source database.
4. Transfer data files for user tablespaces to the target system.
5. Use RMAN to convert the data files to the target system's endian format.
6. Import transportable metadata for all user tablespaces into the target database.
7. Import the remaining database objects and metadata (that were not moved by the transport operation) from the source database into the target database.

If the target database is being moved to a new location (for example, to a new data center) during the migration, then create a physical standby database from the original primary database co-located with the target database. After a Data Guard switchover, transport the tablespaces from the source to the target without incurring the file transfer time as part of the downtime.

Additional Considerations

Transportable tablespace has limitations and restrictions in regard to character sets, opaque types, and system tablespace objects. Unlike previous solutions, the steps are not automated.

Perform a platform migration using transportable tablespaces if all of the following are true:

- The source and target platforms have different endian formats.
- The time required to perform a full Data Pump Export and Import does not fit in the maintenance window.

> **See Also:** The "Oracle Database 10*g* Release 2 Best Practices: Platform Migration using Transportable Tablespaces" MAA white paper available at `http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm`

2.2.3 Online Reorganization and Redefinition

One way to enhance availability and manageability is to allow user access to the database during a data reorganization operation. The Online Reorganization and Redefinition feature in Oracle Database offers administrators significant flexibility to modify the physical attributes of a table and transform both data and table structure while allowing user access to the database. This capability improves data availability, query performance, response time, and disk space usage. All of these are important in a mission-critical environment and make the application upgrade process easier, safer, and faster.

This online architecture provides the following benefits:

- Online table reorganization and redefinition:
  - Change any physical attribute of the table online, including moving the table to a new location, partitioning the table, and converting the table from one organization (such as heap-organized) to another (such as index-organized).
  - Change many logical attributes such as column names, types, and sizes. Columns can be added, deleted, or merged. However, you cannot modify the primary key of the table.
- Online index operations:
  - Create indexes online and analyze them simultaneously. You can also use online repair of the physical guess component of logical ROWIDs (used in secondary indexes and in the mapping table for index-organized tables).
  - Reorganize an index-organized table and secondary indexes online to eliminate the reorganization maintenance window. Secondary indexes support efficient use of block hints (physical guesses). You can also perform online repair of invalid physical guesses of logical ROWIDs stored in secondary indexes on an index-organized table.
  - Reorganize an index-organized table or table partition without rebuilding its secondary indexes, resulting in a short reorganization maintenance window.
- Online move of a partitioned table
- Online reorganization support for advanced queues, clustered tables, materialized views, and abstract data types (objects)
- Fast `ADD COLUMN` operations with default value (does not need to update all rows to default value)
- Invisible Indexes speed application migration and testing:
  - Speeds up migration with explicit hints, then drops when finished
  - Prevents premature use of newly created indexes
  - Tests effects of `DROP INDEX`, making the index *visible* if needed, thus there is no need for an index rebuild
- Online index builds with *no* pause to DML (no exclusive DML locks are required)
- No recompilation of dependent objects when online redefinition does not logically affect objects (for example, when columns are added to tables, or when procedures are added to packages)
- Easier table DDL operations online (there is an option to wait for active DML operations instead of aborting)
- Support for redefinition of tables that have materialized views or materialized view logs

The ability to modify table physical attributes and transform both data and table structure has been available since Oracle8*i*. Table 2–3 provides a comprehensive table of data reorganization capabilities.

Table 2–3 New Data Reorganization Capabilities by Release

Action	Oracle 9*i*	Oracle Database 10*g* Release 1	Oracle Database 10*g* Release 2	Oracle Database 11*g*
Online Reorganization using the package `DBMS_REDEFINITION`	Modify table storage parameters Move the table to a different tablespace Add support for parallel queries Add or drop partitioning support Re-create the table to avoid fragmentation Change from a table to an Index-Organized Table, or vice-versa Add or drop a column Transform a column using a function	Clones grants, constraints, and triggers Convert a `LONG` to a `LOB` Reorganize using a unique key Specify columns to order table by	Reorganize a single partition Advanced queue and clustered tables Table containing an ADT Retain and clone statistics Clone check and not null constraints Copies dependent objects for nested tables	Table with materialized view logs or materialized views No recompilation of dependent objects when redefinition does not logically affect objects

Reclaiming Unused Space	Not applicable	Use the `SHRINK SPACE` clause on the following statements:
		`ALTER TABLE`
		`ALTER INDEX`
		`ALTER MATERIALIZED VIEW`
		`ALTER MATERIALIZED VIEW LOG`
			Not applicable	Not applicable

Index Create Online	`CREATE INDEX emp.ename.idx ON emp(ename) ONLINE;` ■ Parallel operations supported ■ Partitions supported ■ All index types except cluster	Not applicable	Not applicable	DML lock-free online index creation, allowing transparent creation with no dependency on workload
Index Coalesce Online	`ALTER INDEX emp.ename_idx COALESCE;` ■ Parallel operations supported ■ Partitions supported ■ All index types	Not applicable	Not applicable	Not applicable
Index-Organized Table Move Online	`ALTER TABLE emp MOVE ONLINE;` ■ Parallel operations not supported ■ Partitions supported ■ Index-Organized Table only	Not applicable	Not applicable	Not applicable

See Also: *Oracle Database Administrator's Guide*

2.2.4 Transportable Technologies

Transportable technologies provides transportable database and transportable tablespace:

- **Transportable database** moves an entire database (user data and the Oracle dictionary) to a new platform with the same endian format. Transportable database permits a minimal downtime migration to a new platform by avoiding the time-consuming method of unloading all user data from the source database and loading it into the target database.
- **Transportable tablespaces** moves a subset of one database into another, even among platforms that differ in endian format:
  - You can use the cross-platform capability of transportable tablespaces to migrate all user data in a database to a new platform with a different endian format. Leveraging transportable tablespaces in this manner permits a minimal downtime migration to a new platform by avoiding the time-consuming method of unloading all user data from the source database and loading it into the target database.
  - You can use transportable tablespaces to reduce downtime for database upgrades in circumstances where the database has simple schemas and when the data files do not have to be copied during the transport process (for example, when the data files are used in place.

    See Also: *Oracle Database Administrator's Guide* for details about how to move or copy tablespaces to another database, including details about transporting tablespaces across platforms

2.2.5 Online Application Maintenance and Upgrades

The following sections describe features that can significantly reduce (or eliminate) the application downtime required to make changes to an application's database objects.

2.2.5.1 Oracle Streams for Rolling Upgrades

Consider using Oracle Streams for fast rolling upgrades. However, note that while Oracle Streams upgrades can achieve little or no database down time, your ability to configure this solution will require some operational investment. See Section 2.1.4, "Oracle Streams" on page 2-8 and Oracle Streams Concepts and Administration for more information.

2.2.5.2 DDL with the WAIT Option

Data definition language (DDL) commands require exclusive locks on internal structures. If DDL commands are issued, these locks may not be available causing the statement to immediately fail even though the DDL could have possibly succeeded subseconds later. Specifying DDL with the `WAIT` option (the new default) resolves this issue. You specify the wait time instance-wide (in the initialization parameter file) and modify the wait time on a session level.

Specifying DDL commands with the `WAIT` option provides more flexibility to define grace periods for such commands to succeed instead of raising an error right away, thus requiring additional application logic to handle such errors.

See Also: *Oracle Database Administrator's Guide*

2.2.5.3 ENABLE, DISABLE and FOLLOWS Clauses for CREATE TRIGGER

The states (`ENABLE` and `DISABLE`) and ordering (`FOLLOWS`) are triggers to control the firing of triggers. These additional states allow greater administrative control for triggers. You can use the `CREATE TRIGGER` statement in a disabled state to validate successful compilation before enabling. In addition, the trigger order can be controlled with the `FOLLOWS` clause.

See Also: *Oracle Database Advanced Application Developer's Guide*

2.2.5.4 Enhanced ADD COLUMN Functionality

Default values of columns are maintained in the data dictionary for columns specified as `NOT NULL`.

Adding new columns with `DEFAULT` values and `NOT NULL` constraint no longer requires the default value to be stored in all existing records. This enhancement not only enables a schema modification in sub seconds and independent of the existing data volume, but it also consumes no space.

See Also: *Oracle Database Administrator's Guide*

2.2.5.5 Finer Grained Dependencies

In releases before Oracle Database 11*g*, metadata would record mutual dependencies between objects with the granularity of the whole object. For example, PL/SQL unit P depends on PL/SQL unit Q, or view V depends on table T. In cases such as these, the dependent objects were sometimes invalidated when there was no logical requirement to do so. For example, if view V depends only on columns C1, C2, and C3 in table T and a new column, C99, is added, the validity of view V is not logically affected. Nevertheless, in earlier releases, V was invalidated by the addition of column C99.

Beginning with Oracle Database 11*g*, dependency metadata is recorded at a finer level of granularity so that the addition of C99 does not invalidate view V. Similarly, if procedure P depends only on elements E1 and E2 in package PKG, then if element E99 is added to PKG, procedure P is not invalidated. (In Oracle Database 10*g*, this change to PKG would invalidate procedure P.)

By reducing the consequential invalidation of dependent objects in response to changes in the objects they depend upon, application availability is increased. The benefit occurs both in the development environment and when a live application is parsed or upgraded. The benefit occurs when an Oracle Database patch set is applied because changes to schema objects must be compatible and, therefore does not cause consequential invalidations.

See Also: *Oracle Database Advanced Application Developer's Guide*

2.2.5.6 Invisible Indexes

An invisible index provides an alternative to making an index unusable or even to dropping the index. An invisible index is maintained for any DML operation but is not used by the optimizer unless you explicitly specify the index with a hint.

Applications often have to be modified without being able to bring the complete application offline. Invisible indexes enable you to leverage temporary index structures for certain operations or modules of an application without affecting the overall application. Furthermore, you can use invisible indexes to test the removal of an index without dropping it right away, thus enabling a grace period for testing in production environments.

See Also: *Oracle Database Administrator's Guide*

2.2.5.7 Dependent PL/SQL Recompilation After Online Table Redefinition

This feature minimizes the need to recompile dependent PL/SQL packages after an online table redefinition. If the redefinition does not logically affect the PL/SQL packages, recompilation is not needed. This optimization is turned on by default.

This feature reduces the time and effort to manually recompile dependent PL/SQL after an online table redefinition. This also includes views, synonyms, and other table dependent objects (with the exception of triggers) that are not logically affected by the redefinition.

2.3 Optimizing Grid Computing and Disaster Recovery Solutions

With Grid Computing and standby database capabilities, you can leverage and scale your existing system infrastructure. For the primary database, this implies that all hardware resources are leveraged for performance and scalability. For secondary or disaster recovery systems, you can use system and database resources with the Active Data Guard option to serve a production purpose while in the standby database role. With Oracle Database 11*g*, Oracle Data Guard can be an integral part of your IT operations and application business.

This section covers the following topics:

- Grid Computing
- Database Server Grid
- Database Storage Grid
- Disaster Recovery Solutions with Better Standby Database Usage

2.3.1 Grid Computing

Grid computing is a computing architecture that effectively pools large numbers of servers and storage into a flexible, on-demand computing resource for all enterprise computing needs.

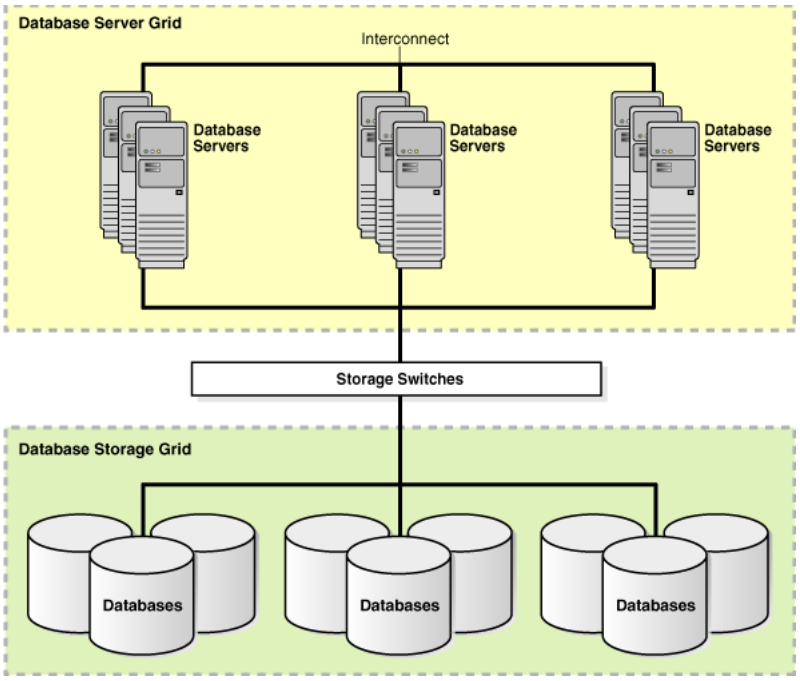
Oracle Database captures the cost advantages of Grid enterprise computing without sacrificing performance, scalability, security, manageability, functionality, or system availability.

- A **Database Server Grid** is a collection of commodity servers connected together to run on one or more databases.
- A **Database Storage Grid** is a collection of low-cost modular storage arrays combined together and accessed by the computers in the Database Server Grid.

With the Database Server and Storage Grids, you can build standby database and testing Hubs to leverage a pool of system resources. The system resources can be dynamically allocated and deallocated depending on various priorities. For example, if the production database fails over to one of the standby databases in the standby hub, it will acquire more system and storage resources while the testing resources may be temporarily starved. With Grid technologies, you can enable high level of utilization and low TCO without sacrificing business requirements.

Figure 2–6 illustrates the Database Server Grid and Database Storage Grid in a Grid enterprise computing environment.

Figure 2–6 Grid Computing Environment



2.3.2 Database Server Grid

The availability of low-cost and reliable blade servers, small multiprocessor servers, and inexpensive open-source operating systems such as Linux, have made it possible to build a Database Server Grid that is highly available, scalable, flexible, and manageable.

Oracle RAC is the technology that enables a Database Server Grid by providing a single database that spans multiple low-cost servers yet appears to the application as a single, unified database system. Oracle RAC provides flexibility to dynamically provision resources and services in the Grid as computing needs change, and to add systems to the Grid as capacity demands increase. In addition, Oracle RAC provides protection from system failures by automatically recovering the processing of a failed node by any of the surviving systems running the database, and facilitating the reconnection of clients and redistribution of load affected by the failed system.

2.3.3 Database Storage Grid

The availability of low-cost ATA disk-based storage arrays and low-cost storage networks has made it possible to use a Database Storage Grid with Oracle Database at very low cost. A database administrator can use the ASM interface to specify the disks in the Database Storage Grid that ASM should manage across all server and storage platforms. ASM partitions the disk space and evenly distributes the data storage throughout the entire storage array. Additionally, ASM automatically redistributes the data storage as storage arrays are added or removed from the Database Storage Grid.

2.3.4 Disaster Recovery Solutions with Better Standby Database Usage

Beginning in Oracle Database 11*g*, standby databases can be used for dynamic IT and application requirements in addition to providing disaster recovery. The Active Data Guard option in Oracle Data Guard enables you to use physical standby databases for

other useful work during normal operations, in addition to providing a disaster-recovery solution.

The following sections describe the Oracle Data Guard features that help you to leverage physical standby databases for additional business purposes:

- Oracle Active Data Guard Option for Physical Standby Databases
- Web Scale Using Standby Reader Farms

2.3.4.1 Oracle Active Data Guard Option for Physical Standby Databases

Redo Apply (physical standby database) is a popular solution for disaster recovery due to its relative simplicity, high performance, and superior level of data protection. Beginning with Oracle Database 11*g*, you can open a physical standby database for read-only access while Redo Apply is active. Thus, Active Data Guard enables you to run queries and reports against an up-to-date physical standby database without compromising data protection or extending recovery time in the event a failover is required. Thus, every physical standby database can support productive uses even while in standby role.

To enable the Active Data Guard option, open the database in read-only mode and then issue the `ALTER DATABASE RECOVER MANAGED STANDBY` statement. Note that the `COMPATIBLE` parameter must be set to 11.0.0 on both the primary and physical standby databases. Using this feature is totally transparent to applications.

The Oracle Active Data Guard option provides an ultimate high availability solution because it:

- Supports Oracle RAC on the primary and standby databases

  Active Data Guard works on both single-instance and Oracle RAC physical standby databases. Although Redo Apply can be running on only one Oracle RAC instance, all of the other instances can run in read-only mode.

- Returns transactionally consistent results that are very close to being up to date with the primary database

  Depending on any delay settings or apply rates, the standby database can be lagging seconds behind the primary database. The queries are always transactionally consistent and represent a consistent view of the last committed transaction at that time.

- Allows fast switchovers or failovers because the redo generated by the primary database while the standby database was opened read-only has already been applied to the standby database, making it immediately available to assume the primary database role

- Enables you to use fast-start failover to allow for automatic fast failover in the case the primary database fails

---

Note: Transactions that attempt to modify a physical standby database running with Active Data Guard enabled will fail with an error.

See Also: *Oracle Data Guard Concepts and Administration* for complete information about using Active Data Guard

2.3.4.2 Web Scale Using Standby Reader Farms

Beginning with Oracle Database 11*g*, you can use both physical standby databases (using the Active Data Guard option) and logical standby databases to deploy a *reader farm*. An example of such a configuration is provided Figure 2–7, complete with the use of Data Guard fast-start failover to automatically fail over should the primary database fail. Note that all standby databases in the reader farm automatically recognize the new primary database after a fast-start failover occurs.

Because a Data Guard configuration can support multiple standby databases, some customers have used this capability to boost read performance of the most demanding Web applications beyond what the underlying system and storage architecture can support. This provides a relatively low-cost method of scaling out using a Grid architecture where I/O is the driving factor.

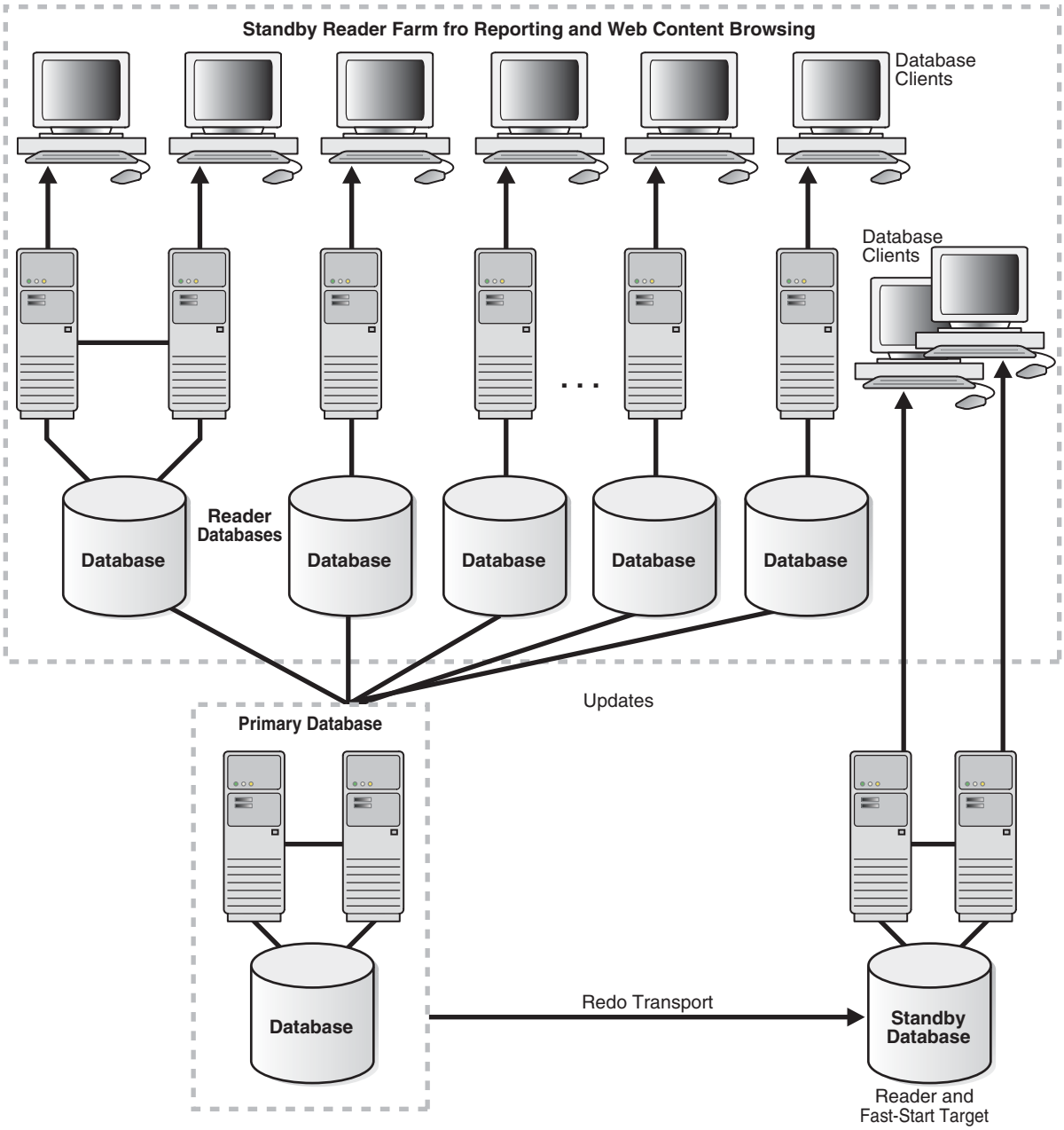
The concept is straightforward—a single primary database that supports read/write transactions, and multiple standby databases that provide read-only access to Web users. Such an approach scales read performance linearly as additional standby databases are added. It is also an effective way to isolate faults, because problems that affect one standby database are isolated from the other standby databases in the configuration.

Creating a reader farm of physical standby databases provides the following benefits:

- Fault isolation
- High performance with physical standby databases and Redo Apply
- Seamless support for all DDL and data types using Redo Apply
- All reader databases are kept up-to-date with changes made to the primary database
- Automatic, zero or minimal data loss failover capability
- Management as a unified configuration through Grid Control
- Scale-out using single writer database and *n* reader databases
- Rolling upgrade capabilities

Figure 2–7 shows a good example of how you can leverage Oracle Data Guard, physical standby databases and Active Data Guard to provide the flexibility necessary to grow your business quickly, while still providing disaster recovery. In the configuration, the primary database transmits redo data to multiple standby database, one of which is also enabled for fast-start failover for automatic, zero or minimal data loss failover.

Figure 2–7 Standby Database Reader Farms



If a fast-start failover is triggered in the Data Guard configuration in Figure 2–7, then:

- Automatic failover occurs to the designated standby database
- All standby databases accept data from new primary database
- You can perform a switchover at a convenient time in the future to return all databases to their original roles

2.4 Optimizing Manageability

Complex environments demand coordinated configuration changes, system upgrades and new application roll-outs. Manageability in Oracle Database 11*g* has improved

dramatically to automate and simplify operations in high availability architectures, and represents a major milestone in the drive toward self-managing Oracle databases.

This section contains these topics:

- Intelligent Infrastructure
- Change Assurance
- Oracle Enterprise Manager Grid Control

2.4.1 Intelligent Infrastructure

Oracle Database has a sophisticated self-management infrastructure that allows the database to learn about itself and use this information to adapt to workload variations or to automatically remedy any potential problem. The self-management infrastructure includes the following:

- Automatic Workload Repository

  The Automatic Workload Repository (AWR) is a built-in repository that contains performance statistics used by Oracle Database for problem detection and self-tuning purposes. At regular intervals, Oracle Database makes a snapshot of vital statistics and workload information and stores them in the AWR. The data contained in the snapshots is then analyzed by the Automatic Database Diagnostic Monitor (ADDM). See the *Oracle Database Performance Tuning Guide* for information about the AWR.

- Automatic Maintenance Tasks

  By analyzing the information stored in the AWR, the database can identify the need to perform routine maintenance tasks. The automated maintenance tasks infrastructure (known as "AutoTask") enables Oracle Database to automatically schedule such operations. AutoTask schedules automatic maintenance tasks to run in a set of Oracle Scheduler windows known as maintenance windows. Maintenance windows are those windows that are members of the Oracle Scheduler window group `MAINTENANCE_WINDOW_GROUP`. See the *Oracle Database Administrator's Guide* and the *Oracle Database 2 Day DBA* for more information.

- Fault diagnosability infrastructure

  Oracle Database includes an advanced fault diagnosability infrastructure for preventing, detecting, diagnosing, and resolving problems. The problems that are targeted are critical errors such as those caused by database code bugs, metadata corruption, and customer data corruption. This includes:

  – The automatic diagnostic repository (ADR), which is a file-based repository for database diagnostic data such as traces, the alert log, health monitor reports, and more. It has a unified directory structure across multiple instances and multiple products.

  – The incident packaging services that a database administrator can use to automatically and easily gather all diagnostic data (traces, health check reports, SQL test cases, and more) pertaining to a critical error and package the data into a zip file suitable for transmission to Oracle Support.

  See the *Oracle Database Administrator's Guide* for more information about these components.

- Server generated alerts

  For problems that cannot be resolved automatically and require administrators to be notified (such as running out of space) the Oracle Database provides

server-generated alerts. Oracle Database can monitor itself and send out alerts to notify you of any problem and provide recommendations on how the reported problem can be resolved. This ensures quick problem resolution and helps prevent potential failures.

- Advisor framework

  Oracle Database includes a number of advisors for different subsystems in the database to automatically determine how the operation of the corresponding subcomponents could be further optimized. The SQL Tuning Advisor and the SQL Access Advisor, for example, provide recommendations for running SQL statements faster. Memory advisors help size the various memory components without resorting to trial-and-error techniques. The Segment Advisor handles space-related issues, such as recommending wasted-space reclamation and analyzing growth trends, while the Undo Advisor guides you in sizing the undo tablespace correctly. See the *Oracle Database 2 Day DBA* for more information about using advisors.

2.4.2 Change Assurance

Oracle Database 11*g* introduces automatic capture and replay of workloads before and after changes so that you can analyze the impact of a database or a SQL change:

- Database Replay

  The Database Replay feature enables you to perform real-world testing by capturing the actual database workload on the production system and replaying it on the test system. It also provides analysis and reporting to highlight potential problems (for example, errors encountered and divergence in performance) and recommend ways to remedy the problems.

- SQL Performance Analyzer

  SQL performance regression is always a concern during system changes such as database upgrades, initialization parameter changes, and adding or dropping indexes. The SQL Performance Analyzer feature alleviates this concern by providing an easy way to assess the impact of a change on the performance of SQL statements by comparing and contrasting their response times before and after the change. SQL Performance Analyzer enables you to capture the SQL workload from the source system, such as the production database, and to replay it on the test system where the change has been applied.

See Also: *Oracle Database Performance Tuning Guide*

2.4.3 Oracle Enterprise Manager Grid Control

By reducing the amount of human intervention required to execute routine and repetitive tasks, services become more stable, reliable, and available. This is particularly important when administrators need to manage very large numbers of systems as efficiently as possible.

Oracle Enterprise Manager Grid Control is an HTML-based interface that provides the administrator with complete monitoring across the entire Oracle technology stack—business applications, application servers, databases, and the E-Business Suite—and non Oracle components. If a component of fast application notification becomes unavailable or experiences performance problems, then Grid Control displays the automatically generated alert so that the administrator can take the appropriate recovery action.

The components of Grid Control include:

- Oracle Management Service (OMS)

  The OMS is now a set of J2EE applications that renders the interface for Grid Control, works with all Management Agents to process monitoring information, and uses the Management Repository as its persistent data store.

- Oracle Management Agents

  These are processes deployed on each monitored host to monitor all targets on the host, communicate that information to OMS, and maintain the host and its targets.

- Oracle Management Repository

  This is a schema in Oracle Database that contains all available information about administrators, targets, and applications managed by Grid Control.

Communication between Grid Control, the OMS, and Oracle Management Agents is done through HTTP. Also, you can enable SSL to allow secure communications between tiers in firewall-protected environments. The Management Agent uploads collected monitoring data to the OMS, which in turn loads the data into the Management Repository. Changes in a target state (such as an availability state change) result in an alert being generated to Grid Control.

Using Grid Control, an administrator can:

- Monitor architecture components and be alerted when a failure occurs
- View overall system status, such as the number of nodes in the database cluster and their current status
- View alerts aggregated across all instances
- Set thresholds for alert generation for each database on a clusterwide basis
- Monitor performance metrics across all instances
- Perform database clusterwide operations such as backup and recovery
- Interconnect monitoring of cluster databases

See Also:

- *Oracle Database High Availability Best Practices* chapter that describes using Grid Control to monitor and maintain a highly available environment across all tiers of the application stack
- *Oracle Enterprise Manager Grid Control Installation and Basic Configuration* and *Oracle Enterprise Manager Concepts* for more information about Oracle Enterprise Manager Grid Control
- The MAA white papers for configuring Enterprise Manager for high availability at

  `http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm`

3

Determining Your High Availability Requirements

This chapter includes the following topics:

3.1 About Determining High Availability Requirements

Any enterprise that is designing and implementing a high availability strategy must begin by performing a thorough analysis of the business drivers that require high availability. Implementing high availability may involve critical tasks such as:

- Retiring legacy systems
- Investment in more capable and robust systems and facilities
- Redesign of the overall IT architecture and operations to adapt to this high availability model
- Redesign of business processes
- Hiring and training of personnel

An analysis of business requirements for high availability combined with an understanding of the level of investment required to implement different high availability solutions enables the development of a high availability architecture that will achieve both business and technical objectives. This chapter provides a simple framework that can be used effectively to evaluate the high availability requirements of a business.

3.2 Analysis Framework for Determining High Availability Requirements

The elements of this analysis framework are:

- Business Impact Analysis
- Cost of Downtime
- Recovery Time Objective (RTO)
- Recovery Point Objective (RPO)
- Manageability Goal
- Total Cost of Ownership (TCO) and Return On Investment (ROI)

3.2.1 Business Impact Analysis

A rigorous business impact analysis identifies the critical business processes in an organization, calculates the quantifiable loss risk for unplanned and planned IT outages affecting each of these business processes, and outlines the effects of these outages. It takes into consideration essential business functions, people and system resources, government regulations, and internal and external business dependencies. This analysis is done using objective and subjective data gathered from interviews with knowledgeable and experienced personnel, reviewing business practice histories, financial reports, IT systems logs, and so on.

The business impact analysis categorizes the business processes based on the severity of the impact of IT-related outages. For example, consider a semiconductor manufacturer with chip fabrication plants located worldwide. Semiconductor manufacturing is an intensely competitive business requiring huge financial investment that is amortized over high production volumes. The human resource applications used by plant administration are unlikely to be considered as mission critical as the applications that control the manufacturing process in the plant. Failure of the applications supporting the fabrication process will affect production levels and have a direct impact on financial results of the company.

In a similar fashion, an internal knowledge management system is likely to be considered mission critical for a management consulting firm because the business of a client-focused company is based on internal research accessibility for its consultants and knowledge workers. The cost of downtime of such a system is extremely high for this business. This leads us to the next element in the high availability requirements framework: *cost of downtime*.

3.2.2 Cost of Downtime

A complete business impact analysis provides the insight needed to quantify the costs of unplanned and planned downtime. Understanding this cost is essential because this helps prioritize your high availability investment and has a direct influence on the high availability technologies chosen to minimize the downtime risk.

Various reports have been published, documenting the costs of downtime across industry verticals. Examples include costs that range from millions of dollars for each hour of brokerage operations and credit card sales, to tens of thousands of dollars for each hour of package shipping services.

These numbers are staggering and the reasons are obvious. The Internet can connect the business directly to millions of customers. Application downtime can disrupt this connection, cutting off a business from its customers. In addition to lost revenue, downtime can have an equally negative effect on other critical and interdependent business issues such as customer relationships, competitive advantages, legal obligations, industry reputation, and shareholder confidence.

3.2.3 Recovery Time Objective (RTO)

The business impact analysis will determine your recovery time objective (RTO). RTO is defined as the maximum amount of time that an IT-based business process can be down before the organization starts suffering unacceptable consequences (financial losses, impact to customer satisfaction, reputation, and so on). RTO indicates the downtime tolerance of a business process or an organization in general.

The RTO requirements are driven by the mission-critical nature of the business. Thus, for a system running a stock exchange, the RTO is zero or very near to zero.

An organization is likely to have varying RTO requirements across its various business processes. Thus, for a high volume e-commerce Web site, for which there is an expectation of rapid response times and for which customer switching costs are very low, the Web-based customer interaction system that drives e-commerce sales is likely to have an RTO close to zero. However, the RTO of the systems that support back-end operations, such as shipping and billing, can be higher. If these back-end systems are down, then the business may resort to manual operations temporarily without a significantly visible impact.

3.2.4 Recovery Point Objective (RPO)

The business impact analysis also determines your recovery point objective (RPO). RPO is the maximum amount of data an IT-based business process may lose before causing detrimental harm to the organization. RPO indicates the data-loss tolerance of a business process or an organization in general. This data loss is often measured in terms of time, for example, 5 hours or 2 days worth of data loss.

A stock exchange where millions of dollars worth of transactions occur every minute cannot afford to lose any data. Thus, its RPO must be zero. Referring to the e-commerce example, the Web-based sales system does not strictly require an RPO of zero, although a low RPO is essential for customer satisfaction. However, its back-end merchandising and inventory update system may have a higher RPO; lost data in this case can be reentered.

3.2.5 Manageability Goal

A *manageability goal* is more subjective than either the RPO or the RTO. It results from an objective evaluation of the skill sets and management resources available in an organization, and the degree to which the organization can successfully manage all elements of a high availability architecture. In a fashion similar to how RPO and RTO measures an organization's tolerance toward downtime or data loss, your manageability goal measures the organization's tolerance to complexity in the IT environment. To the extent that less complexity is a requirement, simpler methods of achieving high availability are preferred over methods that may be more complex to manage, even if the latter could attain more aggressive RTO and RPO objectives. Having a good understanding of manageability goals helps organizations differentiate between what is possible and what is practical to implement.

3.2.6 Total Cost of Ownership (TCO) and Return On Investment (ROI)

Understanding TCO and ROI is essential to selecting an high availability architecture that also achieves the business goals of your organization. TCO includes all costs such as acquisition, implementation, systems, networks, facilities, staff, training, and support, over the useful life of the solution chosen. Likewise, the ROI calculation captures all of the financial benefits that accrue to a given high availability architecture.

For example, consider a high availability architecture in which IT systems/storage at a remote standby site remains idle and has no other business use that can be served by its standby systems. The only return on investment for the standby site is the cost of downtime avoided by its use in a failover scenario. Contrast this with a different high availability architecture that enables IT systems and storage at the standby site to be used productively while in the standby role (for example, for reports or for offloading the primary system of the overhead of end-user queries), that makes the standby system a production system in its own right. The return on investment of such an

architecture includes both the cost of downtime avoided and the financial benefits that accrue to its use as a production system while in the standby database role.

3.3 High Availability Architecture Requirements

Using the high availability analysis framework, a business can:

1. Complete a business impact analysis
2. Identify and categorize the critical business processes that have the high availability requirements
3. Formulate the cost of downtime
4. Establish utilization, RTO, and RPO goals for these various business processes.
5. Understand your goals for manageability, TCO, and ROI

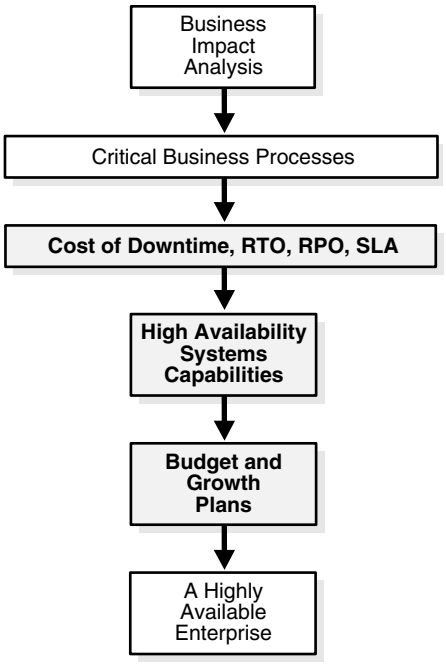
This framework enables the business to define service level agreements (SLAs) in terms of high availability for critical aspects of its business. For example, it can categorize its business processes into several high availability tiers:

- Tier 1 processes have maximum business impact. They have the most stringent high availability requirements, with RTO and RPO close to zero, and the systems supporting it need to be available on a continuous basis. For a business with a high-volume e-commerce presence, this may be the Web-based customer interaction system.
- Tier 2 processes that have slightly relaxed high availability and RTO/RPO requirements. The second tier of an e-commerce business may be their supply chain and merchandising systems. For example, these systems do not need to maintain extremely high degrees of availability and may have nonzero RTO/RPO values. Thus, the high availability systems and technologies chosen to support these two tiers of businesses are likely to be different from those of the tier 1 processes.
- Tier 3 processes may be related to internal development and quality assurance processes. Systems supporting these processes need not have the rigorous high availability requirements of the other tiers.

The next step for the business is to evaluate the capabilities of the various high availability systems and technologies, and choose the ones that meet its SLA requirements, within the guidelines as dictated by business performance issues, budgetary constraints, and anticipated business growth.

Figure 3–1 illustrates this process.

Figure 3–1 Planning and Implementing a Highly Available Enterprise



The following sections provide further details about this methodology:

- High Availability Systems Capabilities
- Business Performance, Budget, and Growth Plans

> **See Also:** "Choosing the Correct High Availability Architecture" on page 4-27

3.3.1 High Availability Systems Capabilities

A broad range of high availability and business continuity solutions exists today. As the sophistication and scope of these systems increase, they make more of the IT infrastructure, such as the data storage, server, network, applications, and facilities, highly available. They also reduce RTO and RPO from days to hours, or even to minutes and seconds. Increased availability often comes with an increased cost, and on some occasions, with an increased impact on systems performance. With Oracle Grid infrastructure, higher availability can equate to lower cost, greater scalability, and more complete utilization of system resources. The high availability approach to satisfying business requirements may differ for a legacy system.

Organizations need to carefully analyze the capabilities of these high availability systems and map their capabilities to the business requirements to ensure they have an optimal combination of high availability solutions to keep their business running. Consider the business with a significant e-commerce presence as an example.

For this business, the IT infrastructure supporting the system that customers encounter, the core e-commerce engine, must be highly available and disaster proof. The business may consider clustering for the Web servers, application servers and the database servers serving this e-commerce engine. With built-in redundancy, clustered solutions eliminate single points of failure. Also, modern clustering solutions are application transparent, provide scalability to accommodate future business growth, and provide load-balancing to handle heavy traffic. Thus, such clustering solutions are ideally suited for mission-critical high-transaction applications.

If unplanned and planned outages occur, the data that supports the high volume e-commerce transactions must be protected adequately and be available with minimal downtime. This data should not only be backed up at regular intervals at the local data centers, but should also be replicated to databases at a remote data center connected over a high-speed, redundant network. This remote data center should be equipped with secondary servers and databases readily available, and be synchronized with the primary servers and databases. This gives the business the capability to switch to these servers at a moment's notice with minimal downtime if there is an outage, instead of waiting for hours and days to rebuild servers and recover data from backed-up tapes. Factors to consider when planning a remote data center include the network bandwidth and latency (distance) between sites, and usage consideration (such as whether the sites are fully or partially staffed). These factors should be used to determine whether remote data centers are feasible and their location in relation to the primary data center.

Maintaining synchronized remote data centers is an example where redundancy is built along the entire system's infrastructure. This may be expensive; however, the mission-critical nature of the systems and the data it protects may warrant this expense. Considering another aspect of the business, the high availability requirements are less stringent for systems that gather clickstream data and perform data mining. The cost of downtime is low, and the RTO and RPO requirements for this system could be a few days, because even if this system is down and some data is lost, that does not have a detrimental effect on the business. While the business may need powerful computers to perform data mining, it does not need to mirror this data on a real-time basis. To obtain data protection, perform regularly scheduled backups, and archive the tapes for offsite storage.

For this e-commerce business, the back-end merchandising and inventory systems are expected to have higher high availability requirements than the data mining systems, and thus they may employ technologies such as local mirroring or local snapshots, in addition to scheduled backups and offsite archiving.

The business should employ a management infrastructure that performs overall systems management, administration and monitoring, and provides an executive dashboard. This management infrastructure should be highly available and fault tolerant.

Finally, the overall IT infrastructure for this e-commerce business should be extremely secure, to protect against malicious external and internal electronic attacks.

3.3.2 Business Performance, Budget, and Growth Plans

High availability solutions must also be based on business performance issues. For example, a business may use a zero-data-loss solution that synchronously mirrors every transaction on the primary database to a remote database. However, considering the speed-of-light limitations and the physical limitations associated with a network, there are round-trip-delays in the network transmission. These delays increase with distance and vary based on network bandwidth, traffic congestion, router latencies, and so on. Thus, this synchronous mirroring, if performed over large WAN distances, may impact the primary site performance. Online buyers may notice these system latencies and be frustrated with long system response times; consequently, they may go somewhere else for their purchases. This is an example where the business must make a trade-off between having a zero data loss solution and maximizing system performance. Conversely, if the business drivers justify the investment to avoid making this tradeoff, a multisite architecture can be implemented that places a synchronous zero data loss standby site in close proximity to the primary site and a second asynchronous standby site located up to thousands of miles away.

High availability solutions must also be based on financial considerations and future growth estimates. It is tempting to build redundancies throughout the IT infrastructure and claim that the infrastructure is completely failure proof. Although higher availability does not always equate higher cost, going to extremes with such solutions may lead to budget overruns or an unmanageable and unscalable combination of solutions that is extremely complex and expensive to integrate and maintain.

A high availability solution that has very impressive performance benchmark results may look good in theory. However, if an investment is made in such a solution without a careful analysis of how the technology capabilities match the business drivers, then a business may end up with a solution that:

- Does not integrate well with the rest of the system infrastructure
- Has annual integration and maintenance costs that easily exceed the up-front implementation costs
- Forces a vendor lock-in

Cost-conscious and business-savvy decision makers must invest only in solutions that are well-integrated, standards-based, easy to implement, maintain and manage, and have a scalable architecture for accommodating future business growth.

4

High Availability Architectures and Solutions

The Maximum Availability Architecture (MAA) is Oracle's best practices blueprint. It is based on proven Oracle high availability technologies and recommendations. The goal of the MAA is to remove the complexity in designing the optimal high availability architecture by providing configuration recommendations and tuning tips to get the most out of your architecture and Oracle features.

This chapter describes the various high availability architectures in an Oracle environment and helps you to choose the correct architecture for your organization.

It includes the following sections:

- Oracle Database High Availability Architectures
- Choosing the Correct High Availability Architecture
- Integrating Application Server High Availability
- Integrating High Availability for All Applications

4.1 Oracle Database High Availability Architectures

The following sections provide an overview of the Oracle Database high availability architectures:

- Oracle Database
- Oracle Database with Oracle Clusterware (Cold Failover Cluster)
- Oracle Database with Oracle Real Application Clusters (Oracle RAC)
- Oracle Database with Oracle RAC on Extended Clusters
- Oracle Database with Data Guard
- Oracle Database with Oracle Clusterware and Data Guard
- Oracle Database with Oracle RAC and Data Guard
- Oracle Database with Streams

All of these architectures must leverage the MAA best practices.

See the "Choosing the Correct High Availability Architecture" section on page 4-27 for a comparison of the different architectures and highlights of the benefits and considerations.

Once you have chosen an architecture, you can them implement it using the operational and configuration best practices described in the MAA white papers and the *Oracle Database High Availability Best Practices*. These best practices are required to maximize the full benefits of each architecture. See Chapter 5, "MAA and High

Availability Best Practices" for more information about the best practices documentation.

4.1.1 Oracle Database

Oracle Database is a single-instance, noncluster database. Although this architecture does not have the node or database redundancy, there are numerous high availability features that can be used in this architecture and any subsequent database architectures. These features make the standalone database on a single computer attractive and available for certain failures and planned maintenance activities.

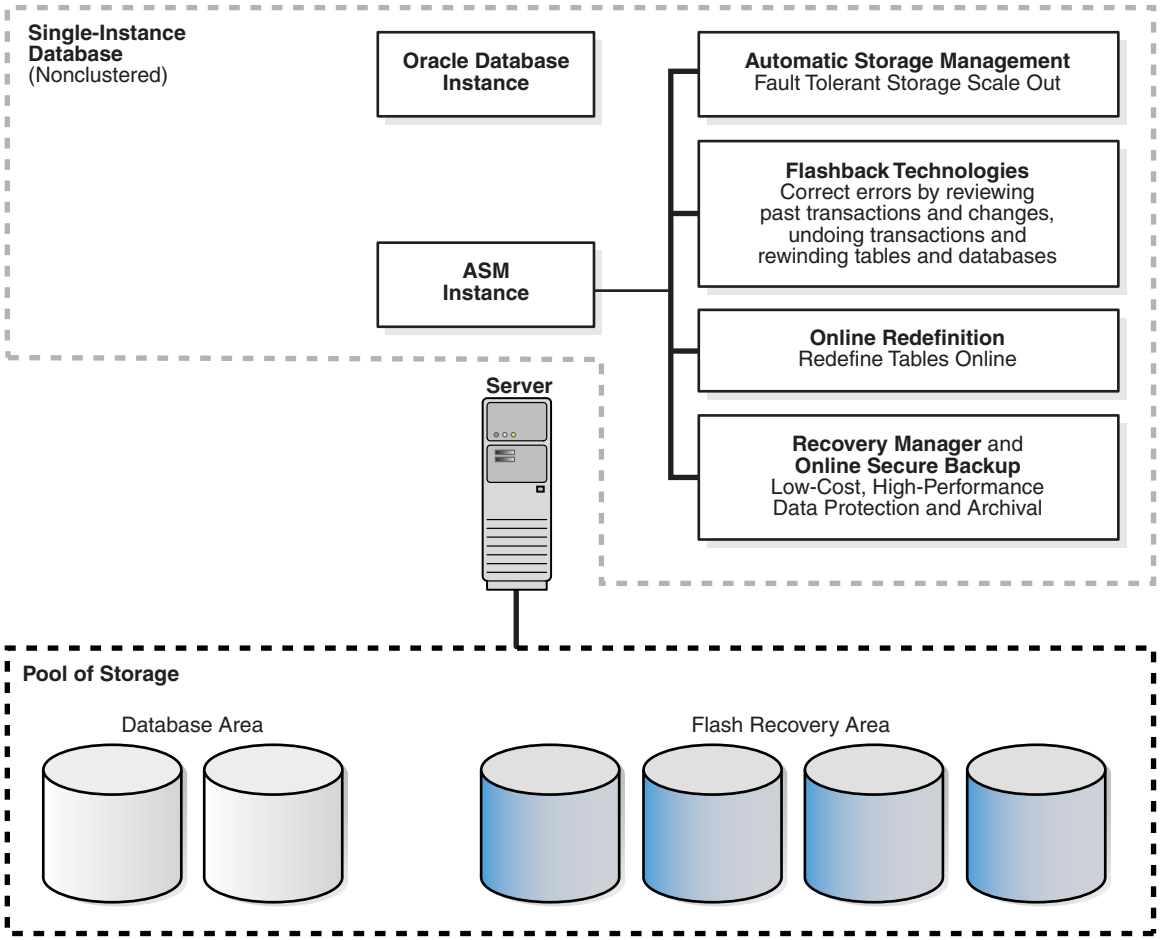
Oracle recommends that you leverage the following Oracle features for this architecture. This is the base foundation for subsequent high availability architectures.

- Fast-Start Fault Recovery bounds and optimizes instance and database recovery times.
- Automatic Storage Management (ASM) tolerates storage failures and optimizes storage performance and usage.
- Oracle Flashback Technology optimizes logical failure repair. Oracle recommends that you use automatic undo management with sufficient space to attain your desired undo retention guarantee, enable Flashback Database and allocate sufficient space and I/O bandwidth in the flash recovery area.
- Recovery Manager (RMAN) optimizes local repair of data failures. Oracle recommends that you create and store the local backups in the flash recovery area.
- Flash Recovery Area manages local recovery related files.
- Online Reorganization and Redefinition allows for dynamic data changes.
- Oracle Security Features prevent unauthorized access and changes.
- Data Recovery Advisor provides intelligent advise and repair of different data failures
- Data Block Corruption Prevention and Detection Parameters detects and prevents some corruptions and lost writes.
- Dynamic Resource Provisioning allows for dynamic system changes.
- Online Patching allows for dynamic database patches of typical diagnostic patches
- Oracle Secure Backup provides a centralized tape backup management solution.

Figure 4–1 shows a basic, single-node Oracle Database that includes an ASM instance.[1] This architecture takes advantage of several high availability features, including Flashback Database, Online Redefinition, Recovery Manager, and Oracle Secure Backup.

[1] Single-instance databases can use clustered ASM (Storage GRID) or nonclustered ASM.

Figure 4–1 Single-Node, Nonclustered Oracle Database with an ASM Instance



4.1.2 Oracle Database with Oracle Clusterware (Cold Failover Cluster)

Oracle Clusterware is software that manages the availability of user applications and Oracle databases. The servers on which you want to run Oracle Clusterware must be running the same operating system.

Many high availability architectures today use clusters alone to provide some rudimentary node redundancy and automatic node failover. However, when you use Oracle Clusterware, there is no need or advantage to using third-party clusterware.

Oracle Clusterware provides a number of benefits over third-party clusterware:

- Oracle Clusterware enables you to use an entire software solution from Oracle, avoiding the cost and complexity of maintaining additional cluster software.

  By reducing the number of combinations of software necessary to coordinate and support, you can increase the manageability and availability of your system software.

- Oracle Clusterware provides seamless integration with, and migration to, Oracle Real Application Clusters (Oracle RAC) and Oracle Data Guard.

  Section 4.1.7 on page 4-22 describes how you can achieve the highest level of availability with Oracle RAC and Oracle Data Guard

- Oracle Clusterware includes all of the features required for cluster management, including node membership, group services, global resource management, and

high availability functions such as managing third-party applications, event management, and Oracle notification services that enable Oracle clients to reconnect to the new primary database after a failure.

- Oracle Clusterware uses a private network and a voting disk to detect and resolve *split brain*[2] scenarios.

With Oracle Clusterware you can provide a *cold failover cluster* to protect an Oracle instance from a system or server failure. The basic function of a cold failover cluster is to monitor a database instance running on a server, and if a failure is detected, to restart the instance on a spare server in the cluster. Network addresses are failed over to the backup node. Clients on the network experience a period of lockout while the failover takes place and are then served by the other database instance once the instance has started. Also, you can use the Oracle Clusterware ability to relocate applications and application resources (using the `CRS_RELOCATE` command) as a way to move the workload to another node so you can perform planned system maintenance on the production server.

The cold failover cluster solution with Oracle Clusterware provides these additional advantages over a basic database architecture:

- Automatic recovery of node and instance failures in minutes
- Automatic notification and reconnection of Oracle integrated clients[3]
- Ability to customize the failure detection mechanism

  For example, you can use your favorite application query in the database check action. Providing application-specific failure detection means Oracle Clusterware can fail over not only during the obvious cases such as when the instance is down, but also in the cases when, for example, an application query is not meeting a particular service level.
- High availability functionality to manage third-party applications
- Rolling release upgrades of Oracle Clusterware

The operation of an Oracle Clusterware cold failover cluster is depicted in Figure 4–2 and Figure 4–3. These figures show how you can use the Oracle Clusterware framework to make both the Oracle database and your custom applications highly available.

Figure 4–2 shows a configuration that uses Oracle Clusterware to extend the basic Oracle Database architecture and provide cold failover cluster. In the figure, the configuration is operating in normal mode in which Node 1 is the active instance connected to the Oracle Database that is servicing applications and users. Node 2 is connected to Node 1 and to the Oracle Database, but it is currently standby mode.

[2] Network splits, commonly referred to as split brains, occur when nodes on one side of the cluster cannot see the nodes on the other side of the cluster.

[3] Oracle Clusterware sends the service events and FAN-integrated clients automatically react to those events.

Figure 4–2 Oracle Database with Oracle Clusterware (Before Cold Failover Cluster)

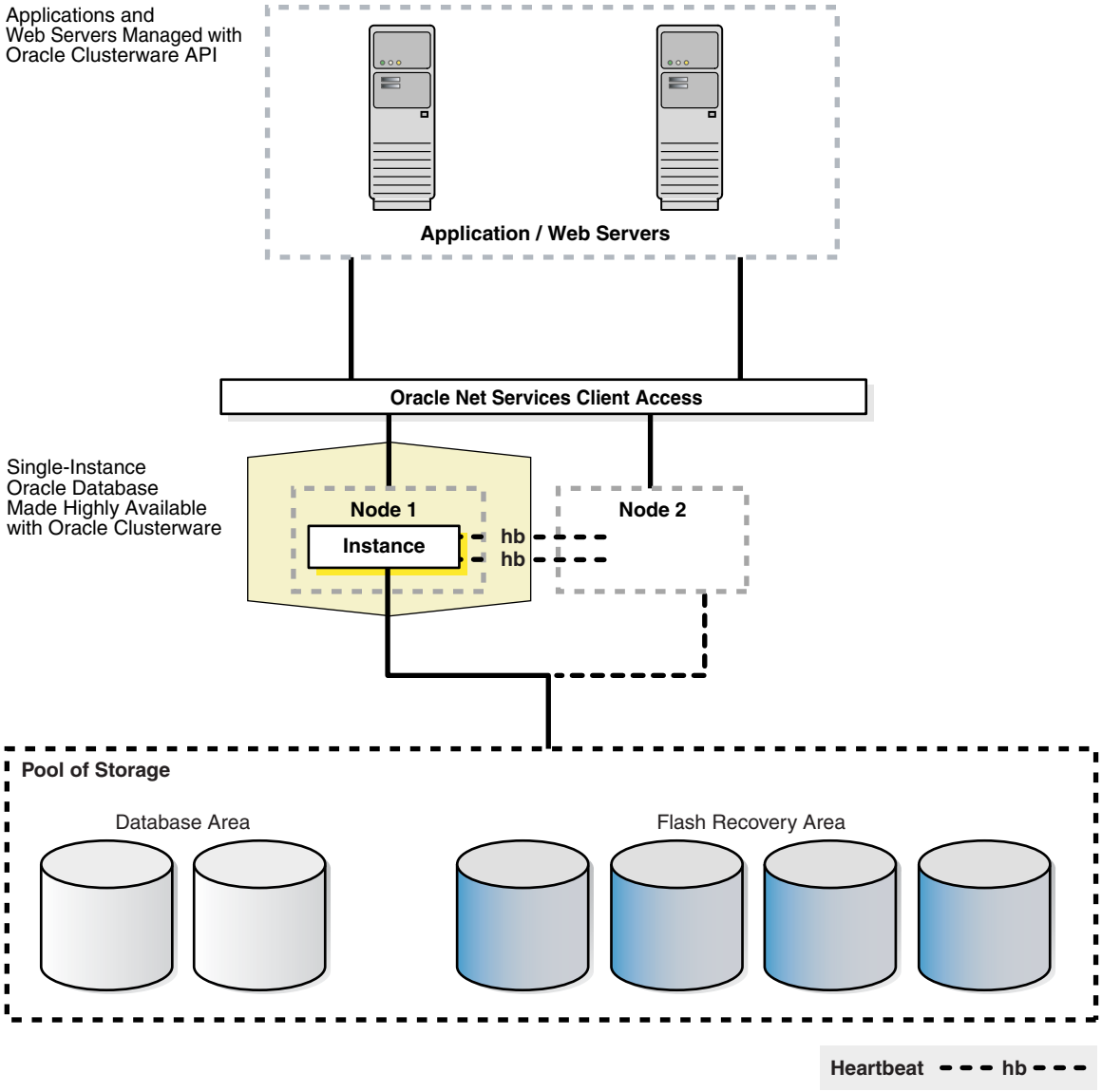
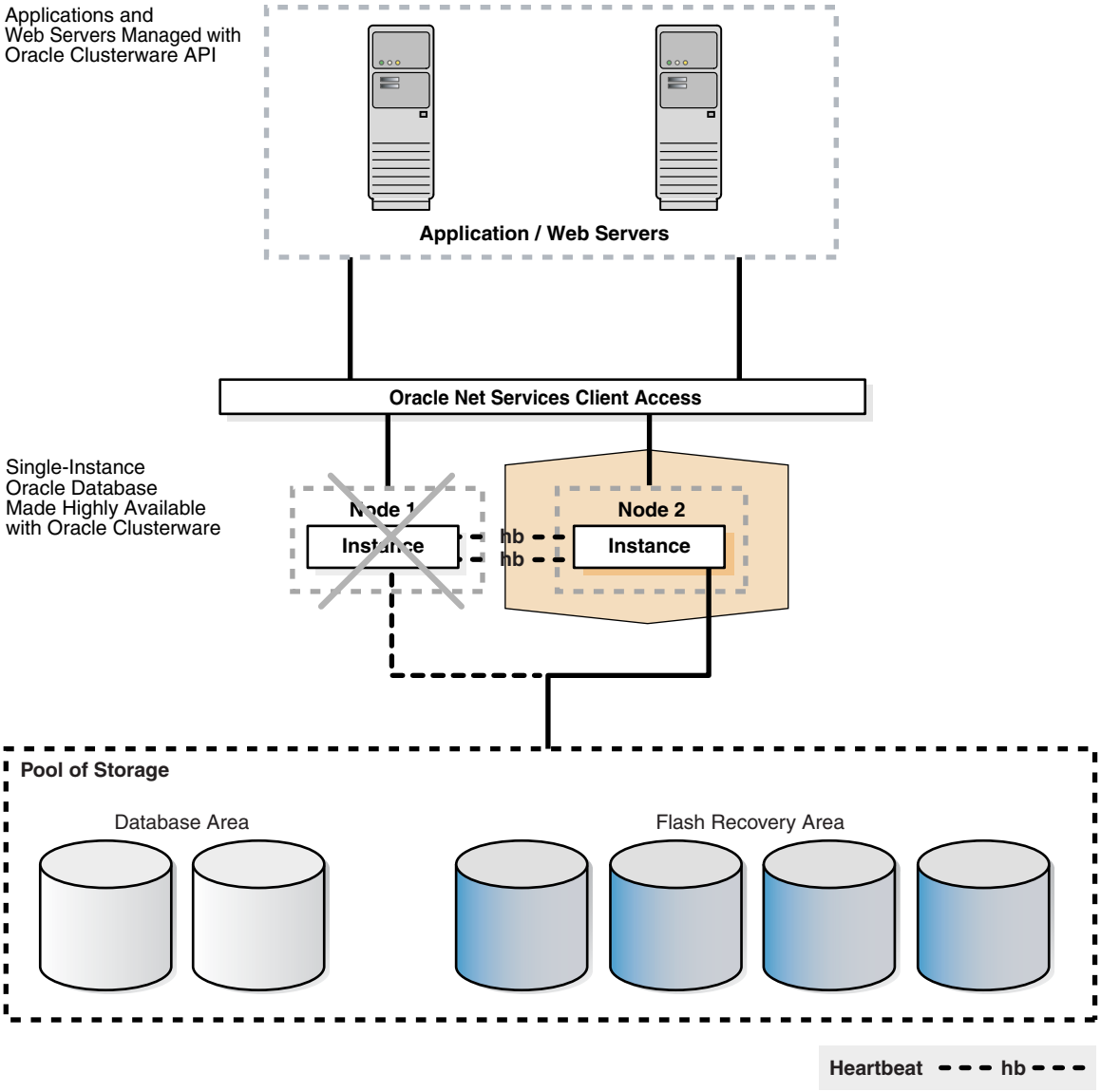


Figure 4–3 shows the Oracle Clusterware configuration after a cold failover cluster has occurred. In the figure, Node 2 is now the active instance connected to the Oracle Database and servicing applications and users. Node 1 is connected to Node 2 and to the Oracle Database but Node 1 is currently idle, in standby mode.

To provide this transparent failover capability, Oracle Clusterware requires a virtual IP address for each node in the cluster. With Oracle Clusterware you also define an *application* virtual IP address so users can access the application independently of the node in the cluster where the application is running. You can define multiple application VIPs, with generally one application VIP defined for each application running. The application VIP is tied to the application by making it dependent on the application resource defined by Cluster Ready Services (CRS).

Figure 4–3 Oracle Database with Oracle Clusterware (After Cold Failover Cluster)



Note: Neither Oracle Enterprise Manager or Oracle Universal Installer (OUI) provide configuration support for Oracle Clusterware. To configure an Oracle Clusterware environment, follow the step-by-step instructions in your platform-specific Oracle Clusterware installation guide.

4.1.3 Oracle Database with Oracle Real Application Clusters (Oracle RAC)

An architecture that combines the Oracle Database with Oracle RAC is inherently a highly available system. Unlike a traditional monolithic database server that is expensive and is not flexible to changing capacity and resource demands, Oracle RAC combines the processing power of multiple interconnected computers to provide system redundancy, scalability, and high availability.

The clusters that are typical of Oracle RAC environments can provide continuous service for both planned and unplanned outages. Oracle RAC builds higher levels of

availability on top of the standard Oracle features. All single instance high availability features, such as the Flashback technologies and online reorganization, also apply to Oracle RAC. Applications scale in an Oracle RAC environment to meet increasing data processing demands without changing the application code. In addition, allowing maintenance operations to occur on a subset of components in the cluster while the application continues to run on the rest of the cluster can reduce planned downtime.

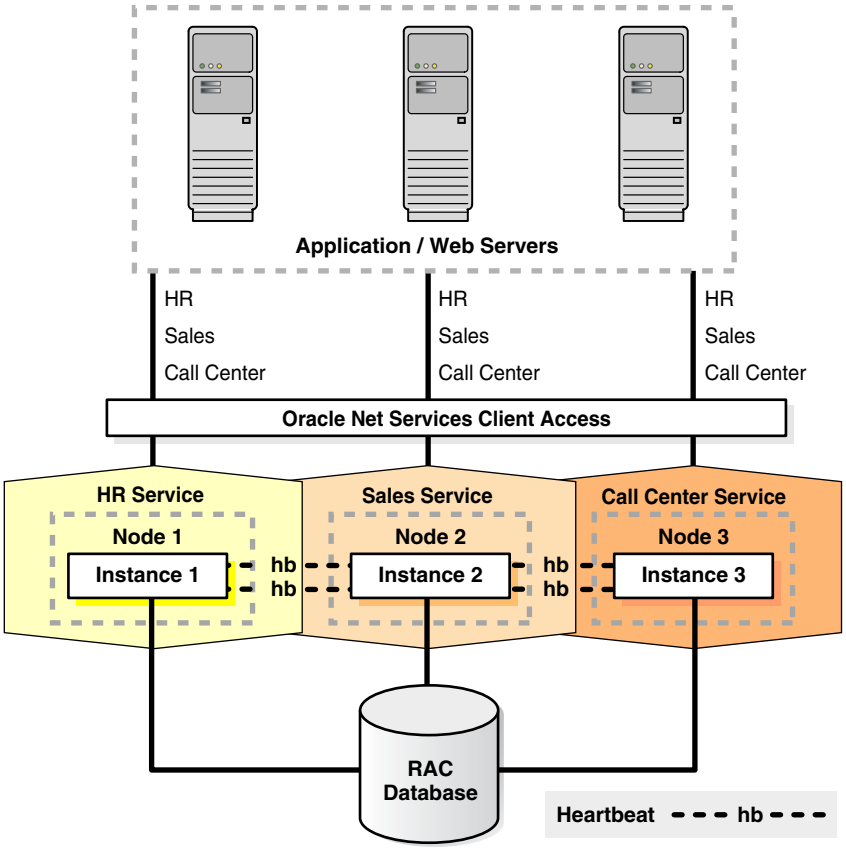
Oracle RAC exploits the redundancy that is provided by clustering to deliver availability with *n* - 1 node failures in an *n*-node cluster. Unlike the cold cluster model where one node is completely idle, all instances and nodes can be active to scale your application.

The Oracle Database with Oracle RAC architecture provides the following benefits over a traditional monolithic database server and the cold failover cluster model:

- Scalability across database instances
- Flexibility to increase processing capacity using commodity hardware without downtime or changes to the application
- Ability to tolerate and quickly recover from computer and instance failures (measured in seconds)
- Rolling upgrades for system and hardware changes
- Rolling patch upgrades for some interim patches
- Fast, automatic, and intelligent connection and service relocation and failover
- Load balancing advisory and runtime connection load balancing
- Comprehensive manageability integrating database and cluster features

Figure 4–4 shows the Oracle Database with Oracle RAC architecture.

Figure 4–4 Oracle Database with Oracle RAC Architecture



4.1.4 Oracle Database with Oracle RAC on Extended Clusters

The Oracle Database with Oracle RAC architecture is designed primarily as a scalability and availability solution that resides in a single data center. It is possible, under certain circumstances, to build and deploy an Oracle RAC system where the nodes in the cluster are separated by greater distances. This architecture is referred to as an *extended cluster*. [4]

An Oracle RAC extended cluster is an architecture that provides extremely fast recovery from a site failure and allows for all nodes, at all sites, to actively process transactions as part of single database cluster. For example, for a business that has a corporate campus, the extended Oracle RAC configuration could consist of individual Oracle RAC nodes being located in separate buildings. Oracle RAC on an extended cluster provides greater availability than a local Oracle RAC cluster, but an extended cluster may not completely fulfill the disaster recovery requirements of your organization.

When the two data centers are located relatively close to each other, extended clusters can provide great protection for some disasters, but not all. You should determine if both sites are likely to be affected by the same disaster. For example, if the extended cluster configuration is set up properly, it can protect against disasters such as a local power outage, an airplane crash, or server room flooding. However, an extended cluster cannot protect against comprehensive disasters such as earthquakes, hurricanes, and regional floods that affect a greater geographical area. (For complete

[4] Extended clusters may be referred to as stretch clusters, campus clusters, metro clusters, or geo clusters.

disaster recovery, use the architecture described in Section 4–9, "Oracle Database with Oracle RAC and Data Guard - MAA".)

The advantages to using Oracle RAC on extended clusters include:

- Ability to fully use all of the system resources without jeopardizing the overall failover times for instance and node failures
- Extremely rapid recovery if one site should fail
- All of the Oracle RAC benefits listed in Section 4.1.3, "Oracle Database with Oracle Real Application Clusters (Oracle RAC)" on page 4-8

> **Note:** While an extended cluster architecture can be effective and has been successfully implemented, you should implement it only in the environments (involving the distance, latency, and degree of protection) recommended in this discussion.

Figure 4–5 shows an Oracle RAC extended cluster for a configuration that has multiple active instances on six nodes at two different locations: three nodes at Site A and three at Site B. The public and private interconnects, and the Storage Area Network (SAN) are all on separate dedicated channels, with each one configured redundantly. For availability reasons, the Oracle Database is a single database that is mirrored at both of the sites. Also, to prevent a full cluster outage if either site fails, the configuration includes a third voting disk on an inexpensive, low-end standard Network File System (NFS) mounted device.

Figure 4–5 Oracle RAC On an Extended Cluster

See Also:

- *Oracle Database High Availability Best Practices* for information about configuring Oracle Database 11*g* with Oracle RAC on extended clusters
- The white paper about extended clusters on the Oracle Real Application Clusters Web site at `http://www.oracle.com/technology/products/database/clustering/`
- The white paper about using standard NFS to support a third voting disk on a stretch cluster configuration that is available on the Oracle RAC Web site at `http://www.oracle.com/technology/products/database/clustering/index.html`

4.1.5 Oracle Database with Data Guard

Oracle Data Guard is a high availability and disaster-recovery solution that provides very fast automatic failover (referred to as fast-start failover) in the case of database failures, node failures, corruption, and media failures. Furthermore, the standby databases can be used for read-only access and subsequently for reader farms, for reporting purposes, and for testing and development purposes.

While traditional solutions (such as backup and recovery from tape, storage based remote mirroring, and database log shipping) can deliver some level of high availability, Data Guard provides the most comprehensive high availability and disaster recovery solution for Oracle databases.

Data Guard provides a number of advantages over traditional solutions, including the following:

- Fast, automatic or automated failover for data corruptions, lost writes, and database and site failures
- Protection against data corruptions and lost writes on the primary database
- Reduced downtime with Data Guard rolling upgrade capabilities
- Ability to offload primary database activities, such as backups, queries or reporting without sacrificing RTO and RPO
- Site failures do not require instance restart, storage remastering, or application reconnections
- Transparent to applications
- Effective network utilization

In addition, for data resident in Oracle databases, Oracle Data Guard, with its built in zero data loss capability, is more efficient, less expensive and better optimized for data protection and disaster recovery than traditional remote mirroring solutions. Oracle Data Guard provides a compelling set of technical and business reasons that justify its adoption as the disaster recovery and data protection technology of choice, over traditional remote mirroring solutions.

The following list summarizes the advantages of using Oracle Data Guard compared to using remote mirroring solutions:

- **Better Network Efficiency**—With Oracle Data Guard, only the redo data needs to be sent to the remote site. However, if a remote mirroring solution is used for data protection, typically you must mirror the database files, the online redo logs, the archived redo logs and the control file. If the flash recovery area is on the source volume that is remotely mirrored, then you must also remotely mirror the flashback logs. Thus, compared to Data Guard, a remote mirroring solution must transmit each change many more times to the remote site.
- **Better Performance**—Data Guard only transmits writes to the redo logs of the primary database, whereas remote mirroring solutions must transmit these writes and every write I/O to data files, additional members of online log file groups, archived redo log files, and control files. Data Guard is designed so that it does not affect the Oracle database writer (DBWR) process that writes to data files, because anything that slows down DBWR process affects database performance. However, remote mirroring solutions affect DBWR process performance because they subject all DBWR process write I/Os to network and disk I/O induced delays inherent to synchronous, zero-data-loss configurations. Compared to mirroring, Data Guard provides better performance and is more efficient, Data Guard always verifies the state of the standby database and validates the data before applying redo, and Data Guard enables you to use the standby database for updates while it continues to protect the primary database.
- **Better suited for WANs**—Remote mirroring solutions based on storage systems often have a distance limitation due to the underlying communication technology (Fibre Channel, ESCON) used by the storage systems. In a typical example, the maximum distance between these two boxes connected in a point-to-point fashion and running synchronously can be only 10 km. Using specialized devices this

distance can be extended to 66 km. However, when the standby data center is more than 66 km apart, you must use a series of repeaters and converters from third-party vendors. These devices convert ESCON/Fibre Channel to the appropriate IP, ATM or SONET networks.

- **Better resilience and data protection**—Oracle Data Guard ensures much better data protection and data resilience than remote mirroring solutions, because corruptions introduced on the production database probably can be mirrored by remote mirroring solutions to the standby site, but corruptions are eliminated by Data Guard. For example, if a stray write occurs to a disk, or there is a corruption in the file system, or the Host Bus Adaptor corrupts a block as it is written to disk, then a remote mirroring solution may propagate this corruption to the DR site. Because Data Guard only propagates the redo data in the logs, and the log file consistency is checked before it is applied, all such external corruptions are eliminated by Data Guard.
- **Higher Flexibility**—Data Guard is implemented on top of pure commodity hardware. It only requires a standard TCP/IP-based network link between the two computers. There is no fancy or expensive hardware required. It also allows the storage to be laid out in a different fashion from the primary. For example, you can put the files on different disks, volumes, file systems, and so on.
- **Better Functionality**—Data Guard, with its full suite of data protection features (Redo Apply for physical standby databases and SQL Apply for logical standby databases, multiple protection modes, push-button automated switchover and failover capabilities, automatic gap detection and resolution, GUI-driven management and monitoring framework, cascaded redo log destinations), is a much more comprehensive and effective solution optimized for data protection and disaster recovery than remote mirroring solutions.
- **Higher ROI**—Businesses have to ensure that they are getting as much value as possible from their IT investments, and no IT infrastructure is sitting idle. Data Guard is designed to allow businesses get something useful out of their expensive investment in a disaster-recovery site. Typically, this is not possible with remote mirroring solutions.

The recommended high availability and disaster-recovery architectures that leverage Oracle Data Guard are described in the following sections:

- Overview of Single Standby Database Architectures
- Overview of Multiple Standby Database Architectures

4.1.5.1 Overview of Single Standby Database Architectures

A single standby database architecture consists of the following key traits and recommendations:

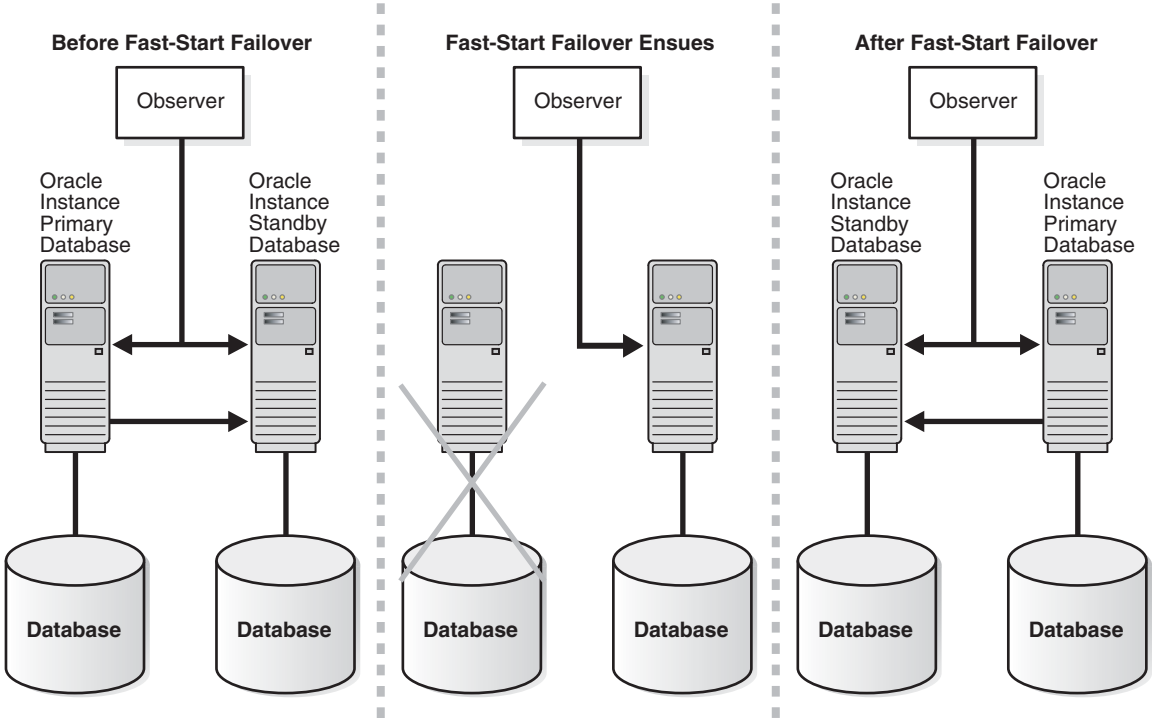
- Primary database resides in Site A.
- Standby database resides in Site B. If zero data loss is required with minimum performance impact on the primary database, the best practice is to locate the secondary site within 200 miles from the primary database. Note, however, that the synchronous redo transport does not impose any physical distance limitation.
- Fast-start failover is recommended to provide automatic failover without user intervention and bounded recovery time. If the primary database uses the asynchronous redo transport, configure your maximum data loss tolerance or the Data Guard broker's `FastStartFailoverLagLimit` property to meet your business requirements. The observer (thin client watchdog) resides in the

application tier and monitors the availability of the primary database. See *Oracle Data Guard Broker* for a detailed description of the observer.

- Use a physical standby database if read-only access is sufficient.
- Evaluate logical standby databases if additional indexes are required for reporting purposes and if your application only uses data types supported by logical standby database and SQL Apply.

Figure 4–6 shows the relationships between the primary database, target standby database, and the observer before, during, and after a fast-start failover occurs.

Figure 4–6 Relationship of Primary and Standby Databases and the Observer During Fast-Start Failover



The following list describes examples of Data Guard configurations using single standby databases:

- A national energy company uses a standby database located in a separate facility 10 miles away from its primary data center. Outages or data loss that could impact customer service and safety are avoided by using Data Guard synchronous transport and automatic failover (fast-start failover).
- An infrastructure services provider to the telecommunication industry utilizes a single standby database located over 400 miles away from the primary configured for synchronous redo transport, enabling zero data loss failover for maximum data protection and high availability.
- A telecommunications provider uses asynchronous redo transport to synchronize a primary database on the west cost of the United Sates, with a standby database on the east coast, over 2,200 miles away. This scenario enables the provider to use existing data centers that are geographically isolated, offering a unique level of high availability.
- A global manufacturing company used Data Guard to replace storage-based remote mirroring and maintain a standby database at its recovery site 50 miles

away from the primary site. Data Guard provides more comprehensive data protection and its more efficient network utilization means there is plenty of headroom to grow without incurring the additional expense of upgrading their network.

4.1.5.2 Overview of Multiple Standby Database Architectures

This architecture is identical to the single-standby database architecture that was described in Section 4.1.5.1, except that there are multiple standby databases in the same Data Guard configuration. The following list describes some implementations for a multiple standby database architecture:

- Continuous and transparent disaster or high availability protection in case of an outage at the primary database or the targeted standby database
- Reader farms or look up databases
- Reporting databases
- Regional reporting or reader databases for better response time
- Synchronous transport transmits to a more *local* standby database, and asynchronous transport transmits to a more *remote* standby database to provide optimum levels of performance and data protection
- Testing and development clones using snapshot standby databases
- Rolling upgrades

Note that it is possible to convert a physical standby database to a logical standby database or to a snapshot standby database, or you can create additional logical standby databases or snapshot standby databases:

- **Transient logical standby databases** can be used to minimize downtime for database upgrades. Using transient logical standby databases is helpful in Data Guard architectures where there are no logical standby databases.

  In a multiple standby database environment, you can create a transient logical standby database temporarily (for planned maintenance) and then convert it back to the physical standby database role. For example, you can use transient logical standby databases to minimize downtime for database upgrades, when required. There is no need to create a separate logical standby database to perform upgrades. The high-level steps for rolling upgrades with a transient logical standby database are as follows:

  1. Start performing a rolling database upgrade with the physical standby database.
  2. Temporarily convert the physical standby database to a logical standby database to perform the upgrade. (Note that data type restrictions are limited for the short window of time required to perform an upgrade.)
  3. Revert the logical standby database back to the physical standby database role.

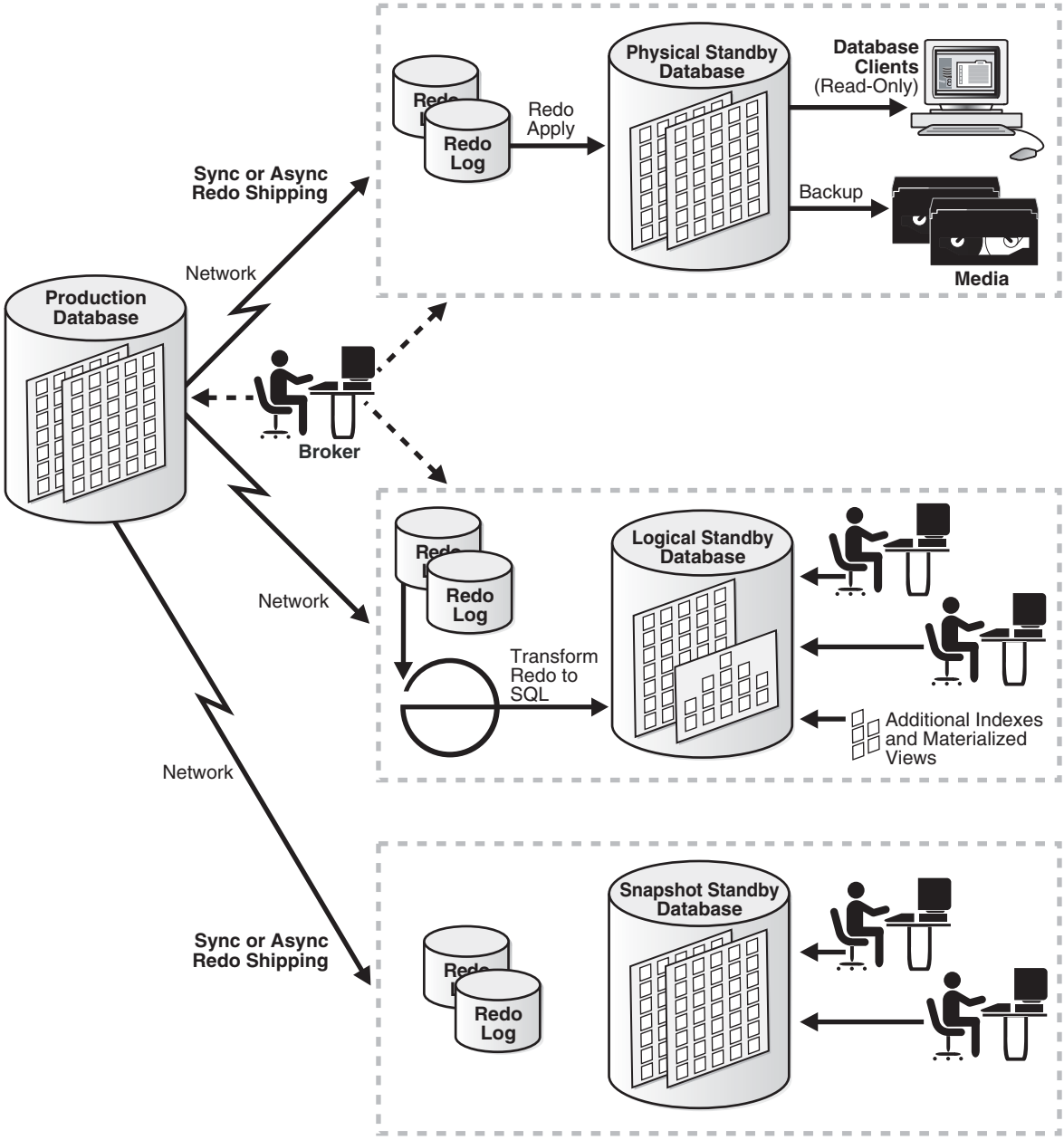
  See Also: *Oracle Data Guard Concepts and Administration* or the *Oracle Database High Availability Best Practices* for step-by-step instructions about performing a rolling upgrade with a transient logical standby database

- **Snapshot standby databases** can be used as a clone or a test database to test new functionality and new releases. The snapshot standby database continues to receive and queue redo data so data protection and RPO are not sacrificed.

Snapshot standby databases diverge from the primary database over time because redo data from the primary database is not applied when it is received. Redo Apply does not apply the redo data until you convert the snapshot standby database back into a physical standby database, and all local updates that were made to the snapshot standby database are discarded. Although the local updates to the snapshot standby database cause additional divergence, the data in the primary database is fully protected by means of the redo logs that are located at the standby site.

Figure 4–7 shows the production database at the primary site and multiple standby databases at secondary sites. Also, see Figure 2–7, "Standby Database Reader Farms" on page 2-45 for another example of a multiple standby database environment.

Figure 4–7 Oracle Database with Data Guard Architecture on Primary and Multiple Standby Sites



See Also:

- *Oracle Data Guard Concepts and Administration* for more information about the various types of standby databases and to find out what datatypes are supported by logical standby databases
- *Oracle Database High Availability Best Practices* for configuration best practices
- The white papers about Oracle Data Guard and standby databases at

  http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm

The following list describes examples of Data Guard configurations using multiple standby databases:

- A world-recognized financial institution uses two remote physical standby databases for continuous data protection after failover. If the primary system should fail, the first standby database becomes the new primary. The second standby database automatically receives data from the new primary, insuring that data is protected at all times.
- A nationally recognized insurance provider in the U.S. maintains two standby databases in the same Data Guard configuration, one physical and one logical standby database. Their strategy further mitigates risk by maintaining multiple standby databases, each implemented using a different architectures - Redo Apply and SQL Apply.
- A world-recognized e-commerce site utilizes multiple standby databases—a mix of both physical and logical databases - both for disaster recovery purposes and to scale-out read performance by provisioning multiple logical standby databases using SQL Apply.
- A global provider of information services to legal and financial institutions uses multiple standby databases in the same Data Guard configuration to minimize downtime during major database upgrades and platform migrations.

Also, for large data centers where there is a need to support many applications with Data Guard requirements, you can build a Data Guard hub to reduce the total cost of ownership.

With the Database Server and Storage Grid, you can build standby database and testing Hubs that leverage a pool of system resources. The system resources can be dynamically allocated and deallocated depending on various priorities. For example, if the primary database fails over to one of the standby databases in the standby hub, the new primary database acquires more system and storage resources while the testing resources may be temporarily starved. With the Oracle Grid technologies, you can enable a high level of utilization and low TCO, without sacrificing business requirements.

A Data Guard hub can consists of:

- Several standby databases in an Oracle RAC environment residing in a cluster of servers, called a grid server
- Leveraging the storage grid

The premise of the standby hub is that it provides higher utilization with lower cost. The probability of failing over all the databases at the same time is unlikely. Thus,

when there is a failover, you can prioritize the system resources to production activity and allocate new system resources in a grid for the standby database functions. At the time of role transition, more storage and system resources can be allocated toward that application.

For example, a Data Guard hub could include multiple databases and applications that are supported in a Grid server and storage architecture. This configuration consists of a central resource supporting 10 applications and databases in the grid compared to managing 10 separate system or storage units in a nongrid infrastructure.

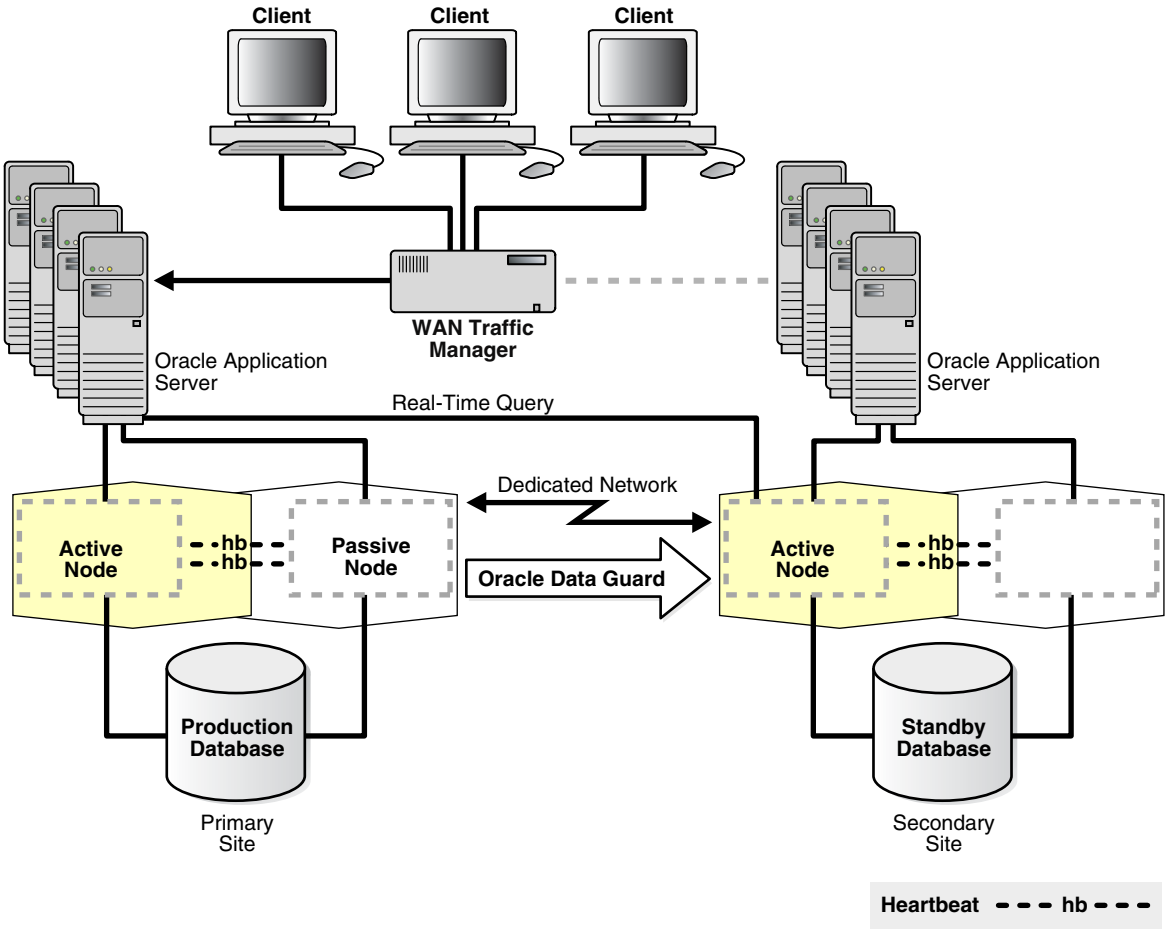
Another possible configuration might be a testing hub consisting of snapshot standby databases. With the snapshot standby database hub, you can leverage the combined storage and server resources of a Grid instead of building and managing individual servers for each application.

4.1.6 Oracle Database with Oracle Clusterware and Data Guard

If your business does not require the scalability and additional high availability benefits provided by Oracle RAC, but you still need all the benefits of Oracle Data Guard and cold failover cluster, then this architecture is a good compromise. With Oracle Database 11*g*, Oracle Clusterware cold failover cluster combined with Oracle Data Guard makes a tightly integrated solution in which failover to the secondary node in the cold failover cluster is transparent and does not require you to reconfigure the Data Guard environment or perform additional steps.

Figure 4–8 shows an Oracle Clusterware and Oracle Data Guard architecture that consists of a primary and a secondary site. Both the primary and secondary sites contain Oracle application servers, two database instances, and an Oracle Database.

Figure 4–8 Oracle Clusterware (Cold Failover Cluster) and Oracle Data Guard



In Figure 4–8:

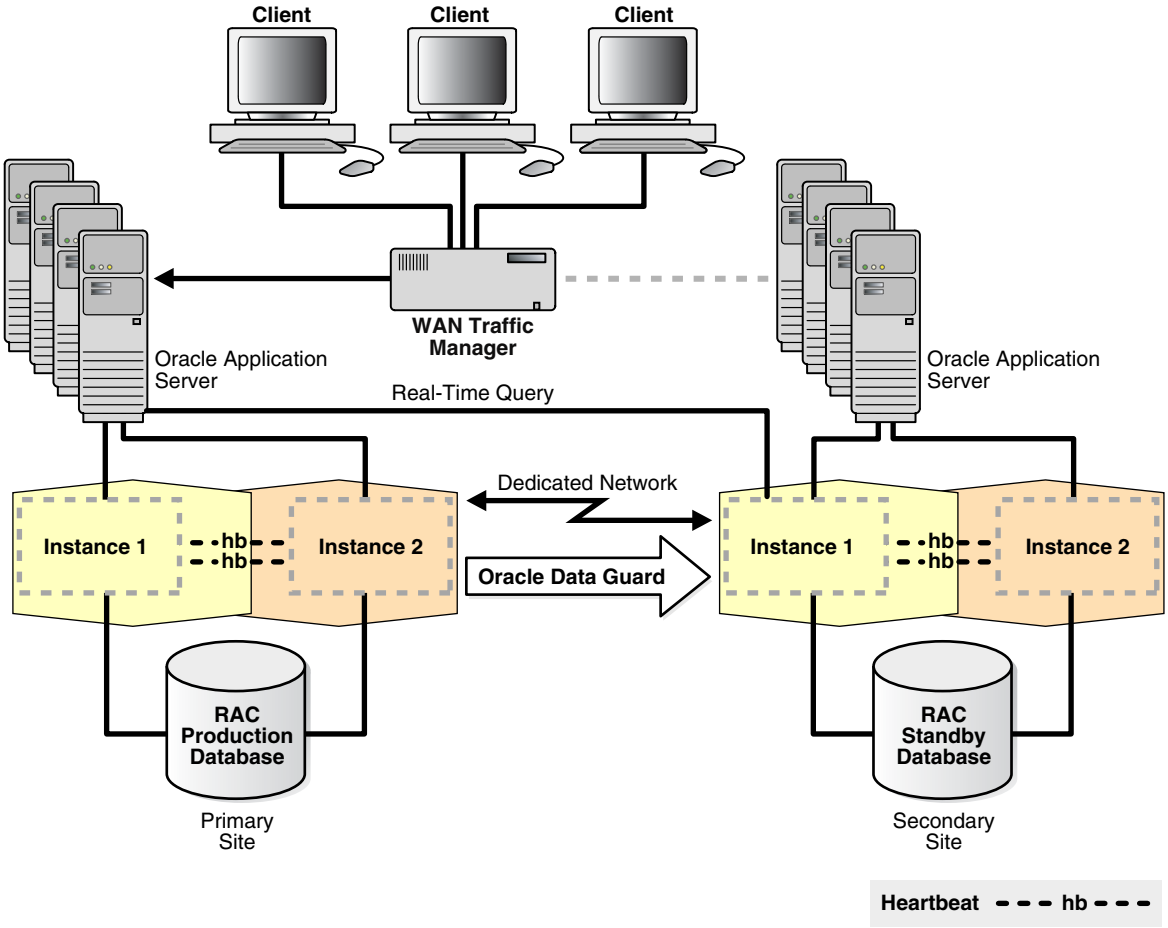
- The application servers on the secondary site are connected to the WAN traffic manager by a dotted line to indicate that they are not actively processing client requests at this time. The application server on the secondary site can be active and processing client requests such as queries if the standby database is a physical standby database with the Active Data Guard option enabled, or if it is a logical standby database.
- Oracle Data Guard transmits redo data from the primary database to the secondary site to keep the databases synchronized.
- Oracle Clusterware manages the availability of both the user applications and Oracle databases.
- Oracle Clusterware provides tolerance of node failures, while Data Guard provides additional protection against data corruptions, lost writes, and database and site failures. (See Oracle Database with Data Guard on page 4-12 for a complete description.)
- Although cold failover cluster is not shown in Figure 4–8, you can configure it by adding a passive node on the secondary site.

4.1.7 Oracle Database with Oracle RAC and Data Guard

You can achieve the highest level of availability when using Oracle RAC and Oracle Data Guard without application changes. These Oracle features provide the most comprehensive architecture for reducing downtime for scheduled outages and preventing, detecting, and recovering from unscheduled outages. This architecture combines the benefits of both Oracle RAC and Data Guard and it is the recommended architecture for Maximum Availability Architecture (MAA).

To protect against site failures, the MAA recommends Oracle RAC and Data Guard reside on separate systems (clusters) and data centers. Figure 4–9 shows the recommended MAA configuration, with Oracle Database, Oracle RAC, and Data Guard. Configuring symmetric sites is recommended to ensure that each site can accommodate the performance and scalability requirements of the application after any role transition. Furthermore, operational practices across role transitions is simplified when the sites are symmetric.

Figure 4–9 Oracle Database with Oracle RAC and Data Guard - MAA



4.1.8 Oracle Database with Streams

Similar to using Oracle Data Guard in SQL Apply mode, Oracle Streams can capture database changes, propagate them to destinations, and apply the changes at these destinations. Streams is optimized for replicating data. Streams can capture changes at a source database, and the captured changes can be propagated asynchronously to replica databases. A logical copy configured and maintained using Streams is called a

replica, not a logical standby database, because it provides many capabilities that are beyond the scope of the normal definition of a standby database.

You might choose to use Streams to configure and maintain a logical copy of your production database. Although using Streams might require additional work, it offers increased flexibility that might be required to meet specific business requirements.

Oracle Database with Streams provides granularity and control over what is replicated and how it is replicated. It supports bi-directional replication, data transformations, subsetting, custom apply functions, and heterogeneous platforms. It also gives users complete control over the routing of change records from the primary database to a replica database. Streams can capture data changes at the primary database or downstream at a replica database, thus enabling users to build hub and spoke network configurations that can support hundreds of replica databases.

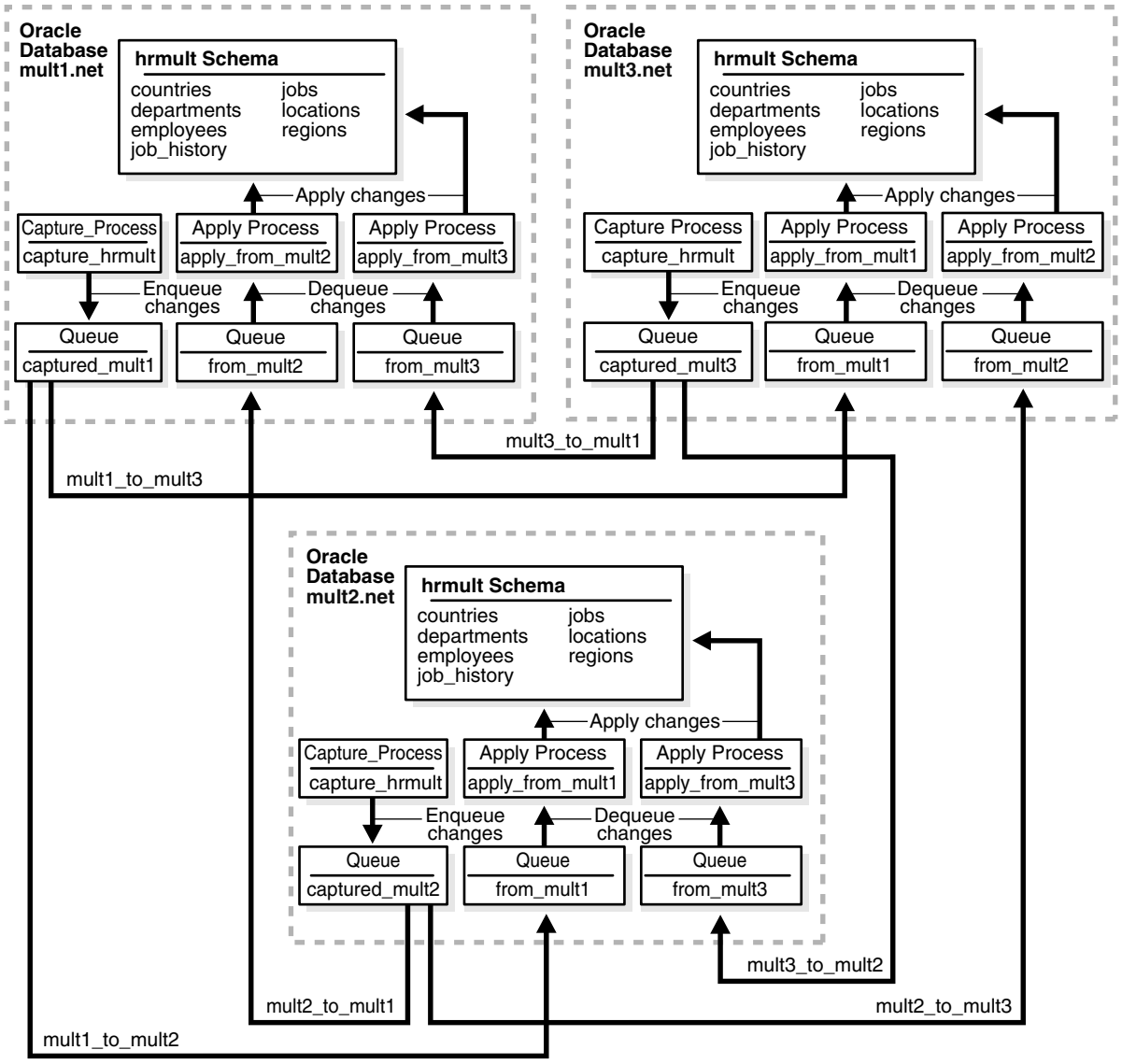
Consider using Oracle Database with Streams if one or more of the following conditions are true:

- Updates are required on both sites or databases, and the changes need to be propagated bidirectionally
- Site configurations are on heterogeneous platforms
- Different character sets are required between the primary database and its replicas
- Fine control of information and data sharing are required
- More investment and expertise to build and maintain an integrated high availability solution is available

Figure 4–10 shows a sample Oracle Database using Streams to replicate data for a schema among three Oracle databases. DML and DDL changes made to tables in the `hr` schema are captured at all databases in the environment and propagated to each of the other databases in the environment.

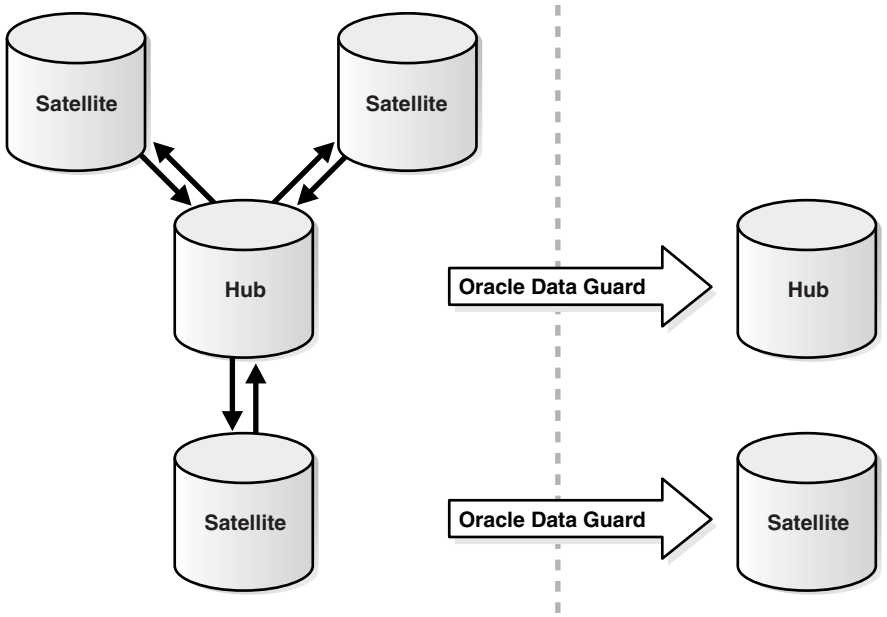
> **See Also:** *Oracle Streams Replication Administrator's Guide* for complete information about constructing multiple-source replication environments using Streams

Figure 4–10 Oracle Database with Streams Architecture That Shares Data From Multiple Databases



You can configure Streams with Data Guard to provide protection for the individual databases in the configuration. Figure 4–11 shows a hub and spoke network configuration in which Oracle Data Guard is providing additional data protection for the hub and one of the satellites.

Figure 4–11 Oracle Streams Hub and Spoke Network Configuration



4.2 Choosing the Correct High Availability Architecture

This section summarizes the advantages of the different high availability architectures and provides guidelines for you to choose the correct high availability architecture for your business.

Chapter 3, "Determining Your High Availability Requirements" describes how the high availability requirements for the business plus its allotted budget determine the appropriate architecture. The key factors include:

- Recovery time objective (RTO) and recovery point objective (RPO) for unplanned outages and planned maintenance
- Manageability Overhead (MO)
- Total Cost of Ownership (TCO) and Return On Investment (ROI)

For example, Table 4–1 provides some insight into the probability of different outages during unplanned and planned activities. The data is derived from actual user experiences and from Oracle service requests.

Table 4–1 Frequency of Outages

Activity	Outage
Media or disk failures	High
Application patches	High
Application failures	High
Logical or user failures that manipulate logical data (DMLs and DDLs)	High
Data corruptions and lost writes	Medium
Computer failures	Medium
Database patches	Medium

Table 4–1 (Cont.) Frequency of Outages

Activity	Outage
Hardware patches and upgrades	Low
Operating system patches and upgrades	Low
Database or application upgrades	Low
Database failures	Low
Platform migrations	Very low
Site failures	Very low

Table 4–2 recommends architectures based on your business requirements for RTO, RPO, MO, scalability, and other factors.

Table 4–2 High Availability Architecture Recommendations

Consider Using ..	Business or Application Impact ...
Oracle Database with Oracle Clusterware (Cold Failover Cluster)	■ Maximum RTO for instance or node failure is in minutes ■ MO is low ■ ROI is low ■ Rolling upgrade and patch capabilities for Oracle Clusterware with zero database downtime.
Oracle Database with Oracle Real Application Clusters (Oracle RAC)	■ Maximum RTO for instance or node failure is zero for the database[1] ■ MO is medium ■ ROI is high ■ Rolling upgrade for system, clusterware, operating system and some Oracle interim patches ■ Database scalability beyond one instance or node
Oracle Database with Oracle RAC on Extended Clusters	■ All of the business benefits of Oracle Real Application Clusters ■ MO is high[2] ■ ROI is medium ■ Additional protection from data center failure with special considerations that are documented in "Oracle Database with Oracle RAC on Extended Clusters" on page 4-9 ■ Highest level of availability for server or computer room failure ■ High availability benefits and workload balancing outweigh performance concerns ■ Willing to make additional provisions for remote data protection to protect against database, data, and cluster failures and corruptions

Table 4–2 (Cont.) High Availability Architecture Recommendations

Consider Using ..	Business or Application Impact ...
Oracle Database with Data Guard	■ Maximum RTO for instance or node failure is in seconds to minutes ■ Maximum RTO for data corruptions, database, or site failures is in seconds to minutes ■ MO is low ■ ROI is high ■ Rolling upgrade for system, clusterware, database, and operating system ■ Offload read-only, reporting, testing and backup activities to the standby database For physical standby databases, this solution: ■ Supports very high primary database throughput ■ Provides the simplicity of a physical replica ■ Provides maximum protection from physical corruptions ■ Provides read-only access to synchronized standby and fast incremental backups to offload production For logical standby databases, this solution: ■ Provides the simplest form of one-way logical replication ■ Allows for structural changes to the standby database, such as changes to local tables, adding schemas, indexes, and materialized views ■ Offloads production by providing read-only access to a synchronized standby database and allows read/write access to local tables that are not being modified by the primary database
Oracle Database with Oracle Clusterware and Data Guard	■ All of the business benefits of Oracle Clusterware (Cold Failover Cluster) and Oracle Data Guard ■ MO is low ■ ROI is medium
Oracle Database with Oracle RAC and Data Guard	■ All of the business benefits of Oracle RAC and Oracle Data Guard ■ MO is medium ■ ROI is high

Table 4–2 (Cont.) High Availability Architecture Recommendations

Consider Using ..	Business or Application Impact ...
Oracle Database with Streams	■ Maximum RTO for instance or node failure is in seconds to minutes ■ Maximum RTO for data corruption, cluster, database, or site failures is in seconds to minutes ■ MO is high[2] ■ ROI is high ■ Rolling upgrade for system, clusterware, operating system, database and application ■ Support for bidirectional replication and update anything and anywhere ■ Support for heterogeneous platforms, versions and character sets ■ Support for fine grained, n-way multimaster, hub & spoke, or many-to-one replication architectures ■ Flexible propagation and management of data, transactions, and events ■ With Oracle RAC integration, database scalability is possible

[1] Database is still available, but a portion of the application connected to the failed system is temporarily affected.

[2] Architectures for which the MO is "High" might require additional time and expertise to build and maintain, but offer increased flexibility and capabilities required to meet specific business requirements.

Table 4–3 identifies the additional capabilities provided by the architectures that build on the Oracle Database and attempts to label each architecture with its greatest strengths.

Table 4–3 Additional Capabilities of High Level Oracle High Availability Architectures

Oracle High Availability Architecture	Key Characteristics and Additional Capabilities
Oracle Database (Base Architecture) The foundation for all high availability architectures	▪ Fast-Start Fault Recovery bounds and optimizes instance and database recovery times to minutes. ▪ Automatic Storage Management tolerates storage failures and optimizes storage performance and utilization. ▪ Oracle Flashback Technology optimizes logical failure repair. ▪ Recovery Manager optimizes local repair of data failures using local backups. ▪ Oracle Secure Backup provides a centralized tape backup management solution. ▪ Flash Recovery Area manages local recovery related files automatically. ▪ Online Reorganization and Redefinition allows for dynamic data changes. ▪ Oracle Security Features prevents unauthorized access and changes. ▪ Data Block Corruption Prevention and Detection Parameters detects and prevents some corruptions and lost writes. ▪ Dynamic Resource Provisioning allows for dynamic system changes. ▪ Online Patching allows for dynamic database patches of typically diagnostic patches. ▪ Data Recovery Advisor diagnoses persistent (on disk) data failures, presents appropriate repair options, and runs repair operations at your request. Support is for single-instance databases only.
Oracle Database with Oracle Clusterware (Cold Failover Cluster)	▪ All of the benefits of Oracle Database ▪ Automatic and fast failover for computer failure ▪ Minimum rolling upgrade capabilities for system, clusterware, and operating system[1]
Oracle Database with Oracle Real Application Clusters (Oracle RAC) High availability, scalability, and foundation of server database grids	▪ All of the benefits of Oracle Database ▪ Scalability beyond a single system ▪ Automatic recovery of failed nodes and instances ▪ Fast application notification (FAN) with integrated Oracle client failover ▪ Rolling upgrade for system, clusterware, operating system and some Oracle interim patches[1]
Oracle Database with Oracle RAC on Extended Clusters Database Grid with site failure protection	▪ All of the benefits of Oracle Database ▪ Protection from site failure

Table 4–3 (Cont.) Additional Capabilities of High Level Oracle High Availability Architectures

Oracle High Availability Architecture	Key Characteristics and Additional Capabilities
Oracle Database with Data Guard Simplest high availability, data protection, and disaster-recovery solution	▪ All of the benefits of Oracle Database ▪ Automatic and fast failover for computer failure, storage failure, data corruption, for configured ORA- errors or conditions and database failures ▪ Protection from site failure ▪ Rolling upgrade for system, clusterware, database, and operating system[2] ▪ Offload backups to the standby database ▪ Offload read and reporting workload to the standby database ▪ Only comprehensive lost write protection
Oracle Database with Oracle Clusterware and Data Guard High availability solution with added data and disaster recovery protection.	▪ The sum of benefits of Oracle Clusterware with Data Guard
Oracle Database with Oracle RAC and Data Guard Best high availability, data protection and disaster-recovery solution with scalability built in	▪ The sum of benefits of Oracle RAC with Data Guard
Oracle Database with Streams[3] Bidirectional replication and information management	▪ Replica database (or databases) are available for read/write use ▪ Provides heterogeneous platform support ▪ Fast failover for computer failure and storage failure ▪ Protection from site failure ▪ Minimizes downtime for computer or site maintenance and database and application upgrades

[1] Rolling upgrades with Oracle Clusterware and Oracle RAC incur zero downtime.

[2] Rolling upgrades with Oracle Data Guard incur minimal downtime.

[3] The initial investment to build a robust solution is well worth the long-term flexibility and capabilities that Streams delivers to meet specific business requirements.

Table 4–4 shows the recovery time including detection and client failover time of an integrated Oracle client, whenever relevant. You should adopt the MAA best practices to achieve the optimal recovery time and configuration. Oracle High Availability Best Practice recommendations can be found in the *Oracle Database High Availability Best Practices* and in the white papers that can be downloaded from:

`http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm`

Table 4–4 Attainable Recovery Times for Unplanned Outages

Outage Scope	Oracle Database	Cold Cluster	Oracle RAC and Oracle RAC on Extended Clusters	Data Guard	Oracle RAC and Data Guard	Streams
Site failure	Hours to days	Hours to days	No downtime[4] if the outage affects one building Hours to days if the outage affects building	Seconds to a minute[1]	Seconds to a minute[1]	No downtime[2]
Computer failure	Minutes to hours[3]	Minutes	No downtime[4]	Seconds to a minute	No downtime[4]	No downtime[4]
Storage failure	No downtime[5]	No downtime[5]	No downtime[3]	No downtime[3]	No downtime[3]	No downtime[3]
Human error	< 30 minutes[6]	< 30 minutes[6]	< 30 minutes[4]	< 30 minutes[4]	< 30 minutes[4]	< 30 minutes[4]
Data corruption	Potentially hours[7]	Potentially hours[7]	Potentially hours[7]	Seconds to a minute	Seconds to a minute	Seconds to a minute

1 Recovery time indicated applies to database and existing connection failover. Network connection changes and other site-specific failover activities may lengthen overall recovery time.

2 The portion of any application connected to the failed system is temporarily affected. You can configure the failed application connections to fail over to the replica.

3 Recovery time consists largely of the time it takes to restore the failed system.

4 Database is still available, but a portion of the application connected to the failed system is temporarily affected.

5 Storage failures are prevented by using ASM with mirroring and its automatic rebalance capability.

6 Recovery time for human errors depend primarily on detection time. If it takes seconds to detect a malicious DML or DLL transaction, it typically only requires seconds to flashback the appropriate transactions. Longer detection time usually leads to longer recovery time required to repair the appropriate transactions. An exception is undropping a table, which is literally instantaneous regardless of detection time.

7 Recovery time depends on the age of the backup used for recovery and the number of log changes scanned to make the corrupt data consistent with the database.

Table 4–5 compares the attainable recovery times of each Oracle high availability architecture for all types of planned downtime.

Table 4–5 Attainable Recovery Times for Planned Outages

System Change or Data Change	Outage Type	Oracle Database	Oracle RAC	Data Guard	MAA	Streams
System change - Dynamic Resource Provisioning	--	No downtime	No downtime	No downtime	No downtime	No downtime
System change - Rolling Upgrade	System level upgrade	Minutes to hours	No downtime	Seconds to five minutes	No downtime	No downtime
System change - Rolling Upgrade	Cluster or site wide upgrade	Minutes to hours	Minutes to hours	Seconds to five minutes	Seconds to five minutes	No downtime[1]

Table 4–5 (Cont.) Attainable Recovery Times for Planned Outages

System Change or Data Change	Outage Type	Oracle Database	Oracle RAC	Data Guard	MAA	Streams
System change - Rolling Upgrade	Storage migration	No downtime[2]	No downtime[2]	No downtime[2]	No downtime[2]	No downtime[2]
System change - Rolling Upgrade	Database one-off patch	Minutes to an hour	No downtime[3]	Seconds to five minutes	No downtime[3]	No downtime
System change - Rolling Upgrade	Database patch set and version upgrade	Minutes to hours	Minutes to hours	Seconds to five minutes	Seconds to five minutes	No downtime[1]
System change - Rolling Upgrade	Platform migration	Minutes to hours	Minutes to hours	Minutes to hours	Minutes to hours	No downtime[1]
Data change	Online Reorganization and Redefinition	No downtime	No downtime	No downtime[4]	No downtime[4]	No downtime[4]

[1] Applications (or a portion of an application) connected to the system that is being maintained may be temporarily affected.

[2] ASM automatically rebalances stored data when disks are added or removed while the database remains online. For storage migration, you are required to leverage both storage arrays by ASM temporarily.

[3] For qualified one-off patches only.

[4] Tables can be reorganized online using the `DBMS_REDEFINITION` package. However, the online changes are not supported by SQL Apply or data capture, and therefore the effects of this subprogram are not visible on the logical standby database or replica database. For more information, see *Oracle Data Guard Concepts and Administration* or *Oracle Streams Replication Administrator's Guide*.

4.3 Integrating Application Server High Availability

The Oracle Application Server provides flexible and automated high availability solutions for Oracle Application Server to ensure that applications that you deploy on Oracle Application Server meet the required availability to achieve your business goals. The solutions introduced in this book are described in detail in the *Oracle Application Server High Availability Guide*.

This section contains the following topics:

- Oracle Application Server High Availability Architectures
- Redundant Architectures
- High Availability Services in Oracle Application Server

4.3.1 Oracle Application Server High Availability Architectures

Oracle Application Server provides high availability and disaster recovery solutions for maximum protection against any kind of failure with flexible installation, deployment, and security options. The redundancy of Oracle Application Server local high availability and disaster recovery originates from its redundant high availability architectures.

At a high level, Oracle Application Server local high availability architectures include several active-active and active-passive architectures for the OracleAS middle-tier and

the OracleAS Infrastructure. Although both types of solutions provide high availability, active-active solutions generally offer higher scalability and faster failover, although, they tend to be more expensive as well. With either the active-active or the active-passive category, multiple solutions exist that differ in ease of installation, cost, scalability, and security.

Building on top of the local high availability solutions is the Oracle Application Server disaster recovery solution. This unique solution combines the proven Oracle Data Guard technology in Oracle Database with advanced disaster recovery technologies in the application realm to create a comprehensive disaster recovery solution for the entire application system. Disaster recovery solutions typically set up two homogeneous sites, one active and one passive. Each site is a self-contained system. The active site is generally called the production site, and the passive site is called the standby site. During normal operation, the production site services requests; in the event of a site failover or switchover, the standby site takes over the production role and all requests are routed to that site. To maintain the standby site for failover, not only must the standby site contain homogeneous installations and applications, data and configurations must also be synchronized constantly from the production site to the standby site. Oracle Application Server instances can be installed in either site as long as they do not interfere with the instances in the disaster recovery setup. Configurations and data must be synchronized regularly between the two sites to maintain homogeneity.

4.3.2 Redundant Architectures

Oracle Application Server provides redundancy by offering support for multiple instances supporting the same workload. These redundant configurations provide increased availability either through a distributed workload, through a failover setup, or both.

From the entry point to an Oracle Application Server system (content cache) to the back end layer (data sources), all the tiers that are crossed by a request can be configured in a redundant manner with Oracle Application Server. The configuration can be an active-active configuration using OracleAS Cluster or an active-passive configuration using OracleAS Cold Failover Cluster.

4.3.3 High Availability Services in Oracle Application Server

Oracle Application Server provides different features and topologies to support high availability across the its stack. This includes solutions that extend across both the OracleAS middle-tier and the OracleAS Infrastructure tier.

The *Oracle Application Server High Availability Guide* describes the following high availability services in Oracle Application Server in detail:

- Process death detection and automatic restart
- Configuration management
- State replication
- Server load balancing and failover
- Backup and recovery
- Disaster recovery

4.4 Integrating High Availability for All Applications

A highly available and resilient application requires that every component of the application must be highly available or tolerate failures and changes. For example, a highly available application must analyze every component that affects the application including the network topology, application server, application flow and design, systems, and the database configuration and architecture. This book has focused primarily on the database high availability solutions.

See the high availability solutions and recommendations for Oracle Application Server, Enterprise Manager and Applications on the MAA Web site at:

http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm

5

MAA and High Availability Best Practices

Choosing and implementing the architecture that best fits the availability requirements of a business can be a daunting task. This architecture must encompass appropriate redundancy, provide adequate protection from all types of outages, ensure consistent high performance and robust security, while being easy to deploy, manage, and scale. Needless to mention, this architecture should be driven by well-understood business requirements.

To build, implement and maintain such an architecture, a business needs high availability best practices that involve both technical and operational aspects of its IT systems and business processes. Such a set of best practices removes the complexity of designing a high availability architecture, maximizes availability while using minimum system resources, reduces the implementation and maintenance costs of the high availability systems in place, and makes it easy to duplicate the high availability architecture in other areas of the business.

An enterprise with a well-articulated set of high availability best practices that encompass high availability analysis frameworks, business drivers and system capabilities, enjoys an improved operational resilience and enhanced business agility.

Once you have chosen a high availability architecture as described in Chapter 4, you can implement it using the MAA best practices documentation. See the MAA and high availability white papers located on the MAA OTN Web site and in the *Oracle Database High Availability Best Practices* documentation for a description of the best practices:

- MAA and high availability best practices white papers

  Oracle offers various white papers that provide technical details on its various high availability technologies, along with best practice recommendations on configuring and using such technologies. Oracle high availability white papers can be downloaded from

  `http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm`

- *Oracle Database High Availability Best Practices*

  This book provides detailed best practice recommendations and information. It can help you to configure a new high availability environment, or migrate an existing configuration to create a redundant, reliable system without sacrificing simplicity and performance.

  The book's step-by-step method of describing the best practices removes the complexity of designing a high availability architecture, maximizes availability while using minimum system resources, reduces the implementation and maintenance costs of the high availability systems in place, and makes it easy to duplicate the high availability architecture in other areas of the business.

Index

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W