

Oracle® AI Database

Autonomous Health Framework User's Guide



26ai
G43733-01
October 2025

ORACLE®

Copyright © 2016, 2025, Oracle and/or its affiliates.

Primary Authors: Nirmal Kumar, Janet Stern

Contributing Authors: Aparna Kamath, Douglas Williams, Mark Bauer, Richard Strohm, Subhash Chandra

Contributors: Ankita Khandelwal, Arpit Shukla, Carol Colrain, Daniel Semler, Gareth Chapman, Girdhari Ghantiyala, Girish Adiga, Jesus Guillermo Munoz Nunez, Macharapu Prasanth, Mark Scardina, Pallavi Kamath, Robert Caldwell, Sahil Kumar, Troy Anthony, Vern Wagman, Walter Battistella

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	i
Documentation Accessibility	i
Related Documentation	i
Conventions	ii

1 Introduction to Oracle Autonomous Health Framework

1.1 Oracle Autonomous Health Framework Problem and Solution Space	1
1.1.1 Availability Issues	1
1.1.2 Performance Issues	2
1.2 Components of Autonomous Health Framework	3
1.2.1 Introduction to Oracle Autonomous Health Framework Configuration Audit Tools	4
1.2.2 Introduction to Cluster Health Monitor	4
1.2.3 Introduction to Oracle Trace File Analyzer	5
1.2.4 Introduction to Oracle Cluster Health Advisor	5
1.2.5 Introduction to Blocker Resolver	6
1.2.5.1 Using the Cluster Resource Activity Log to Monitor Cluster Resource Failures	7

Part I Analyzing the Cluster Configuration

2 Proactively Detecting and Diagnosing Performance Issues for Oracle RAC

2.2 Removing Grid Infrastructure Management Repository	2
2.1 Oracle Cluster Health Advisor Architecture	3
2.3 Monitoring the Oracle Real Application Clusters (Oracle RAC) Environment with Oracle Cluster Health Advisor	4
2.4 Using Cluster Health Advisor for Health Diagnosis	4
2.5 Calibrating an Oracle Cluster Health Advisor Model for a Cluster Deployment	7
2.6 Viewing the Details for an Oracle Cluster Health Advisor Model	10
2.7 Managing the Oracle Cluster Health Advisor Repository	10

2.8	Viewing the Status of Cluster Health Advisor	11
2.9	Enhanced Cluster Health Advisor Support for Oracle Pluggable Databases	12

Part II Automatically Monitoring the Cluster

3 Collecting Operating System Resources Metrics

3.1	Understanding Cluster Health Monitor Services	2
3.2	Collecting Cluster Health Monitor Data	2
3.3	Operating System Metrics Collected by Cluster Health Monitor	2
3.4	Detecting Component Failures and Self-healing Autonomously	10

4 Monitoring System Metrics for Cluster Nodes

4.1	Monitoring Oracle Clusterware with Oracle Enterprise Manager	1
4.2	Monitoring Oracle Clusterware with Cluster Health Monitor	2

Part III Automatic Problem Solving

5 Resolving Database and Database Instance Delays

5.1	Blocker Resolver Architecture	1
5.2	Optional Configuration for Blocker Resolver	2
5.3	Blocker Resolver Diagnostics and Logging	3

Part IV Appendixes

A OCLUMON Command Reference

A.1	oclumon analyze	A-1
A.2	oclumon dumpnodeview	A-4
A.3	oclumon chmdiag	A-9
A.4	oclumon localrepo	A-9
A.4.1	oclumon localrepo getconfig	A-9
A.4.2	oclumon localrepo setconfig	A-10
A.5	oclumon version	A-11
A.6	oclumon debug	A-11

B Querying Cluster Resource Activity Log

B.1	crsctl query calog	B-1
-----	--------------------	-----

C chactl Command Reference

C.1	chactl monitor	C-2
C.2	chactl unmonitor	C-3
C.3	chactl status	C-4
C.4	chactl config	C-5
C.5	chactl calibrate	C-6
C.6	chactl query diagnosis	C-7
C.7	chactl query model	C-10
C.8	chactl query repository	C-10
C.9	chactl query calibration	C-11
C.10	chactl remove model	C-14
C.11	chactl rename model	C-14
C.12	chactl export model	C-15
C.13	chactl import model	C-15
C.14	chactl set maxretention	C-15
C.15	chactl resize repository	C-16

D Behavior Changes, Deprecated and Desupported Features

D.1	Oracle Database Quality of Service (QoS) Management is Deprecated and Desupported in Release 21c	D-1
D.2	Autonomous Health Framework Scope (AHF Scope) Deprecated	D-1

Preface

Oracle Autonomous Health Framework User's Guide explains how to use the Oracle Autonomous Health Framework diagnostic components.

The diagnostic components include Oracle ORAchk, Oracle EXAchk, Cluster Health Monitor, Oracle Trace File Analyzer Collector, Oracle Cluster Health Advisor, and Blocker Resolver.

Oracle Autonomous Health Framework User's Guide also explains how to install and configure Oracle Trace File Analyzer Collector.

This Preface contains these topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documentation](#)
- [Conventions](#)

Audience

Database administrators can use this guide to understand how to use the Oracle Autonomous Health Framework diagnostic components. This guide assumes that you are familiar with Oracle Database concepts.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documentation

For more information, see the following Oracle resources:

Related Topics

- *Oracle Automatic Storage Management Administrator's Guide*
- *Oracle AI Database 2 Day DBA*
- *Oracle AI Database Concepts*

- *Oracle AI Database Examples Installation Guide*
- *Oracle AI Database Licensing Information User Manual*
- *Oracle AI Database Release Notes*
- *Oracle AI Database Upgrade Guide*
- *Oracle Grid Infrastructure Installation and Upgrade Guide*
- *Oracle Real Application Clusters Installation Guide for Linux and UNIX*
- *Oracle Real Application Clusters Installation Guide for Microsoft Windows*

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.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction to Oracle Autonomous Health Framework

Oracle Autonomous Health Framework is a collection of components that analyzes the diagnostic data collected, and proactively identifies issues before they affect the health of your clusters or your Oracle Real Application Clusters (Oracle RAC) databases.

Most of the Oracle Autonomous Health Framework components are already available in Oracle Database 12c release 1 (12.1).

- [Oracle Autonomous Health Framework Problem and Solution Space](#)
Oracle Autonomous Health Framework (AHF) maximizes availability and performance by enforcing best practices, capturing data at first failure, monitoring the whole system (server, database, I/O, and network) to proactively discover issues and notify the user and provide timely bug resolution by suggesting fixes automatically after failure.
- [Components of Autonomous Health Framework](#)
This section describes the diagnostic components that are part of Oracle Autonomous Health Framework.

1.1 Oracle Autonomous Health Framework Problem and Solution Space

Oracle Autonomous Health Framework (AHF) maximizes availability and performance by enforcing best practices, capturing data at first failure, monitoring the whole system (server, database, I/O, and network) to proactively discover issues and notify the user and provide timely bug resolution by suggesting fixes automatically after failure.

System administrators can use most of the components in Oracle Autonomous Health Framework interactively during installation, patching, and upgrading. Database administrators can use Oracle Autonomous Health Framework to diagnose operational runtime issues and mitigate the impact of these issues.

- [Availability Issues](#)
Availability issues are runtime issues that threaten the availability of software stack.
- [Performance Issues](#)
Performance issues are runtime issues that threaten the performance of the system.

1.1.1 Availability Issues

Availability issues are runtime issues that threaten the availability of software stack.

Availability issues can result from either software issues (Oracle Database, Oracle Grid Infrastructure, operating system) or the underlying hardware resources (CPU, Memory, Network, Storage).

The components within Oracle Autonomous Health Framework address the following availability issues:

Examples of Server Availability Issues

Server availability issues can cause a server to be evicted from the cluster and shut down all the database instances that are running on the server.

Examples of such issues are:

- **Issue:** Network congestion on the private interconnect can cause time-critical internode or storage I/O to have excessive latency or dropped packets. This type of failure typically builds up and can be detected early, and corrected or relieved.

Solution: If a change in the server configuration causes this issue, then Cluster Verification Utility (CVU) detects it if the issue persists for more than an hour. However, Oracle Cluster Health Advisor detects the issue within minutes and presents corrective actions.

- **Issue:** Network failures on the private interconnect caused by a pulled cable or failed network interface card (NIC) can immediately result in evicted nodes.

Solution: Although these types of network failures cannot be detected early, the cause can be narrowed down by using Cluster Health Monitor and Oracle Trace File Analyzer to pinpoint the time of the failure and the network interfaces involved.

Examples of Database Availability Issues

Database availability issues can cause an Oracle database or one of the instances of the database to become unresponsive and thus unavailable to users.

Examples of such issues are:

- **Issue:** Runaway queries or delays can deny critical database resources such as locks, latches, or CPU to other sessions. Denial of critical database resources results in database or an instance of a database being non-responsive to applications.

Solution: Blocker Resolver detects and automatically resolves these types of delays. Also, Oracle Cluster Health Advisor detects, identifies, and notifies the database administrator of such delays and provides an appropriate corrective action.

- **Issue:** Denial-of-service (DoS) attacks, vulnerabilities, or simply software bugs can cause a database or a database instance to be unresponsive.

Solution: Proactive recommendations of known issues and their resolutions provided by Oracle Orachk can prevent such occurrences. If these issues are not prevented, then automatic collection of logs by Oracle Trace File Analyzer, in addition to data collected by Cluster Health Monitor, can speed up the correction of these issues.

- **Issue:** Configuration changes can cause database outages that are difficult to troubleshoot. For example, incorrect permissions on the `oracle.bin` file can prevent session processes from being created.

Solution: Use Cluster Verification Utility and Oracle Orachk to speed up identification and correction of these types of issues. You can generate a diff report using Oracle Orachk to see a baseline comparison of two reports and a list of differences. You can also view configuration reports created by Cluster Verification Utility to verify whether your system meets the criteria for an Oracle installation.

1.1.2 Performance Issues

Performance issues are runtime issues that threaten the performance of the system.

Performance issues can result from either software issues (bugs, configuration problems, data contention, and so on) or client issues (demand, query types, connection management, and so on).

Server and database performance issues are intertwined and difficult to separate. It is easier to categorize them by their origin: database server or client.

Examples of Database Server Performance Issues

- **Issue:** Deviations from best practices in configuration can cause database server performance issues.
Solution: Oracle Orachk detects configuration issues when Oracle Orachk runs periodically and notifies the database administrator of the appropriate corrective settings.
- **Issue:** A session can cause other sessions to slow down waiting for the blocking session to release its resource or complete its work.
Solution: Blocker Resolver detects these chains of sessions and automatically terminates the root holder session to relieve the bottleneck.
- **Issue:** Unresolved known issues or unpatched bugs can cause database server performance issues.
Solution: These issues can be detected through the automatic Oracle Orachk reports and flagged with associated patches or workarounds. Oracle Orachk is regularly enhanced to include new critical issues, either in existing products or in new product areas.

Examples of Performance Issues Caused by Database Client

- **Issue:** Misconfigured parameters such as SGA and PGA allocation, number of sessions or processes, CPU counts, and so on, can cause database performance degradation.
Solution: Oracle Orachk and Oracle Cluster Health Advisor detect the settings and consequences respectively and notify you automatically with recommended corrective actions.

1.2 Components of Autonomous Health Framework

This section describes the diagnostic components that are part of Oracle Autonomous Health Framework.

- [Introduction to Oracle Autonomous Health Framework Configuration Audit Tools](#)
Oracle ORAchk and Oracle EXAchk provide a lightweight and non-intrusive health check framework for the Oracle stack of software and hardware components.
- [Introduction to Cluster Health Monitor](#)
Cluster Health Monitor is a component of Oracle Grid Infrastructure, which continuously monitors and stores Oracle Clusterware and operating system resources metrics.
- [Introduction to Oracle Trace File Analyzer](#)
Oracle Trace File Analyzer is a utility for targeted diagnostic collection that simplifies diagnostic data collection for Oracle Clusterware, Oracle Grid Infrastructure, and Oracle Real Application Clusters (Oracle RAC) systems, in addition to single instance, non-clustered databases.
- [Introduction to Oracle Cluster Health Advisor](#)
Oracle Cluster Health Advisor continuously monitors cluster nodes and Oracle RAC databases for performance and availability issue precursors to provide early warning of problems before they become critical.

- [Introduction to Blocker Resolver](#)
Blocker Resolver is an Oracle Real Application Clusters (Oracle RAC) environment feature that autonomously resolves delays and keeps the resources available.

1.2.1 Introduction to Oracle Autonomous Health Framework Configuration Audit Tools

Oracle ORAchk and Oracle EXAchk provide a lightweight and non-intrusive health check framework for the Oracle stack of software and hardware components.

Oracle ORAchk and Oracle EXAchk:

- Automates risk identification and proactive notification before your business is impacted
- Runs health checks based on critical and reoccurring problems
- Presents high-level reports about your system health risks and vulnerabilities to known issues
- Enables you to drill-down specific problems and understand their resolutions
- Enables you to schedule recurring health checks at regular intervals
- Sends email notifications and diff reports while running in daemon mode
- Integrates the findings into Oracle Health Check Collections Manager and other tools of your choice
- Runs in your environment with no need to send anything to Oracle

You have access to Oracle ORAchk and Oracle EXAchk as a value add-on to your existing support contract. There is no additional fee or license required to run Oracle ORAchk and Oracle EXAchk.

Use Oracle EXAchk for Oracle Engineered Systems except for Oracle Database Appliance. For all other systems, use Oracle ORAchk.

Run health checks for Oracle products using the command-line options.

For more information, see *Oracle Autonomous Health Framework Checks and Diagnostics User's Guide*.

Related Topics

- [Oracle Autonomous Health Framework Checks and Diagnostics User's Guide](#)

1.2.2 Introduction to Cluster Health Monitor

Cluster Health Monitor is a component of Oracle Grid Infrastructure, which continuously monitors and stores Oracle Clusterware and operating system resources metrics.

Enabled by default, Cluster Health Monitor:

- Assists node eviction analysis
- Logs all process data locally
- Enables you to define pinned processes
- Listens to CSS and GIPC events
- Categorizes processes by type
- Supports plug-in collectors such as `traceroute`, `netstat`, `ping`, and so on

- Provides CSV output for ease of analysis

Cluster Health Monitor serves as a data feed for other Oracle Autonomous Health Framework components such as Oracle Cluster Health Advisor.

Related Topics

- [Collecting Operating System Resources Metrics](#)
CHM is a high-performance, lightweight daemon that collects, analyzes, aggregates, and stores a large set of operating system metrics to help you diagnose and troubleshoot system issues.

1.2.3 Introduction to Oracle Trace File Analyzer

Oracle Trace File Analyzer is a utility for targeted diagnostic collection that simplifies diagnostic data collection for Oracle Clusterware, Oracle Grid Infrastructure, and Oracle Real Application Clusters (Oracle RAC) systems, in addition to single instance, non-clustered databases.

Enabled by default, Oracle Trace File Analyzer:

- Provides comprehensive first failure diagnostics collection
- Efficiently collects, packages, and transfers diagnostic data to Oracle Support
- Reduces round trips between customers and Oracle

Oracle Trace File Analyzer reduces the time required to obtain the correct diagnostic data, which eventually saves your business money.

For more information, see *Oracle Autonomous Health Framework Checks and Diagnostics User's Guide*.

New Attention Log for Efficient Critical Issue Resolution

Diagnosability of database issues is enhanced through a new attention log, as well as classification of information written to database trace files. The new attention log is written in a structured format (XML or JSON) that is much easier to process or interpret and only contains information that requires attention from an administrator. The contents of trace files now contains information that enables much easier classification of trace messages, such as for security and sensitivity.

Enhanced diagnosability features simplify database administration and improve data security.

For more information, see Attention Log

Related Topics

- [Oracle Autonomous Health Framework Checks and Diagnostics User's Guide](#)

1.2.4 Introduction to Oracle Cluster Health Advisor

Oracle Cluster Health Advisor continuously monitors cluster nodes and Oracle RAC databases for performance and availability issue precursors to provide early warning of problems before they become critical.

Oracle Cluster Health Advisor does the following:

- Detects node and database performance problems
- Provides early-warning alerts and corrective action
- Supports on-site calibration to improve sensitivity

In Oracle Database 12c release 2 (12.2.0.1), Oracle Cluster Health Advisor supports the monitoring of two critical subsystems of Oracle Real Application Clusters (Oracle RAC): the database instance and the host system. Oracle Cluster Health Advisor determines and tracks the health status of the monitored system. It periodically samples a wide variety of key measurements from the monitored system.

Over a hundred database and cluster node problems have been modeled, and the specific operating system and Oracle Database metrics that indicate the development or existence of these problems have been identified. This information is used to construct a trained, calibrated model that is based on a normal operational period of the target system.

Oracle Cluster Health Advisor runs an analysis multiple times a minute. Oracle Cluster Health Advisor estimates an expected value of an observed input based on the default model. Oracle Cluster Health Advisor then performs anomaly detection for each input based on the difference between observed and expected values. If sufficient inputs associated with a specific problem are abnormal, then Oracle Cluster Health Advisor raises a warning and generates an immediate targeted diagnosis and corrective action.

Oracle Cluster Health Advisor models are conservative to prevent false warning notifications. However, the default configuration may not be sensitive enough for critical production systems. Therefore, Oracle Cluster Health Advisor provides an onsite model calibration capability to use actual production workload data to form the basis of its default setting and increase the accuracy and sensitivity of node and database models.

You can also use Oracle Cluster Health Advisor to diagnose and triage past problems. Specify the past dates through the command-line interface CHACTL, AHF Insights, or AHF Scope.

1.2.5 Introduction to Blocker Resolver

Blocker Resolver is an Oracle Real Application Clusters (Oracle RAC) environment feature that autonomously resolves delays and keeps the resources available.

Enabled by default, Blocker Resolver:

- Reliably detects database delays and deadlocks
- Autonomously resolves database delays and deadlocks
- Logs all detections and resolutions
- Provides SQL interface to configure sensitivity (Normal/High) and trace file sizes

A database delays when a session blocks a chain of one or more sessions. The blocking session holds a resource such as a lock or latch that prevents the blocked sessions from progressing. The chain of sessions has a root or a final blocker session, which blocks all the other sessions in the chain. Blocker Resolver resolves these issues autonomously by detecting and resolving the delays.

- [Using the Cluster Resource Activity Log to Monitor Cluster Resource Failures](#)
The cluster resource activity log provides precise and specific information about a resource failure, separate from diagnostic logs.

Related Topics

- [Resolving Database and Database Instance Delays](#)
Blocker Resolver preserves the database performance by resolving delays and keeping the resources available.

1.2.5.1 Using the Cluster Resource Activity Log to Monitor Cluster Resource Failures

The cluster resource activity log provides precise and specific information about a resource failure, separate from diagnostic logs.

If an Oracle Clusterware-managed resource fails, then Oracle Clusterware logs messages about the failure in the **cluster resource activity log**. Failures can occur as a result of a problem with a resource, a hosting node, or the network. The cluster resource activity log provides a unified view of the cause of resource failure.

Writes to the cluster resource activity log are tagged with an activity ID and any related data gets the same parent activity ID, and is nested under the parent data. For example, if Oracle Clusterware is running and you run the `crsctl stop clusterware -all` command, then all activities get activity IDs, and related activities are tagged with the same parent activity ID. On each node, the command creates sub-IDs under the parent IDs, and tags each of the respective activities with their corresponding activity ID. Further, each resource on the individual nodes creates sub-IDs based on the parent ID, creating a hierarchy of activity IDs. The hierarchy of activity IDs enables you to analyze the data to find specific activities.

For example, you may have many resources with complicated dependencies among each other, and with a database service. On Friday, you see that all of the resources are running on one node but when you return on Monday, every resource is on a different node, and you want to know why. Using the `crsctl query calog` command, you can query the cluster resource activity log for all activities involving those resources and the database service. The output provides a complete flow and you can query each sub-ID within the parent service failover ID, and see, specifically, what happened and why.

You can query any number of fields in the cluster resource activity log using filters. For example, you can query all the activities written by specific operating system users such as `root`. The output produced by the `crsctl query calog` command can be displayed in either a tabular format or in XML format.

The cluster resource activity log is an adjunct to current Oracle Clusterware logging and alert log messages.

Note

Oracle Clusterware does not write messages that contain security-related information, such as log-in credentials, to the cluster activity log.

Use the following commands to manage and view the contents of the cluster resource activity log:

Part I

Analyzing the Cluster Configuration

You can use tools in the Autonomous Health Framework to analyze your cluster configuration.

- [Proactively Detecting and Diagnosing Performance Issues for Oracle RAC](#)
Oracle Cluster Health Advisor provides system and database administrators with early warning of pending performance issues, and root causes and corrective actions for Oracle RAC databases and cluster nodes. Use Oracle Cluster Health Advisor to increase availability and performance management.

2

Proactively Detecting and Diagnosing Performance Issues for Oracle RAC

Oracle Cluster Health Advisor provides system and database administrators with early warning of pending performance issues, and root causes and corrective actions for Oracle RAC databases and cluster nodes. Use Oracle Cluster Health Advisor to increase availability and performance management.

Oracle Cluster Health Advisor estimates an expected value of an observed input based on the default model, which is a trained calibrated model based on a normal operational period of the target system. Oracle Cluster Health Advisor then performs anomaly detection for each input based on the difference between observed and expected values. If sufficient inputs associated with a specific problem are abnormal, then Oracle Cluster Health Advisor raises a warning and generates an immediate targeted diagnosis and corrective action.

Oracle Cluster Health Advisor also sends warning messages to Enterprise Manager Cloud Control using the Oracle Clusterware event notification protocol.

The ability of Oracle Cluster Health Advisor to detect performance and availability issues on Oracle Exadata systems has been improved in this release.

With the Oracle Cluster Health Advisor support for Oracle Solaris, you can now get early detection and prevention of performance and availability issues in your Oracle RAC database deployments.

For more information on *Installing Grid Infrastructure Management Repository*, see Oracle® *Grid Infrastructure Grid Infrastructure Installation and Upgrade Guide 20c for Linux*.

- [Oracle Cluster Health Advisor Architecture](#)
Oracle Cluster Health Advisor runs as a highly available cluster resource, ochad, on each node in the cluster.
- [Removing Grid Infrastructure Management Repository](#)
GIMR is desupported in Oracle AI Database 26ai. If GIMR is configured in your existing Oracle Grid Infrastructure installation, then remove the GIMR.
- [Monitoring the Oracle Real Application Clusters \(Oracle RAC\) Environment with Oracle Cluster Health Advisor](#)
Oracle Cluster Health Advisor is automatically provisioned on each node by default when Oracle Grid Infrastructure is installed for Oracle Real Application Clusters (Oracle RAC) or Oracle RAC One Node database.
- [Using Cluster Health Advisor for Health Diagnosis](#)
Oracle Cluster Health Advisor raises and clears problems autonomously.
- [Calibrating an Oracle Cluster Health Advisor Model for a Cluster Deployment](#)
As shipped with default node and database models, Oracle Cluster Health Advisor is designed not to generate false warning notifications.
- [Viewing the Details for an Oracle Cluster Health Advisor Model](#)
Use the `chactl query model` command to view the model details.

- [Managing the Oracle Cluster Health Advisor Repository](#)
Oracle Cluster Health Advisor repository stores the historical records of cluster host problems, database problems, and associated metric evidence, along with models.
- [Viewing the Status of Cluster Health Advisor](#)
SRVCTL commands are the tools that offer total control on managing the life cycle of Oracle Cluster Health Advisor as a highly available service.
- [Enhanced Cluster Health Advisor Support for Oracle Pluggable Databases](#)
The Cluster Health Advisor (CHA) diagnostic capabilities have been extended to support 4K PDBs, up from 256 in Oracle AI Database 26ai.

Related Topics

- [Introduction to Oracle Cluster Health Advisor](#)
Oracle Cluster Health Advisor continuously monitors cluster nodes and Oracle RAC databases for performance and availability issue precursors to provide early warning of problems before they become critical.
- Installing Grid Infrastructure Management Repository

2.2 Removing Grid Infrastructure Management Repository

GIMR is desupported in Oracle AI Database 26ai. If GIMR is configured in your existing Oracle Grid Infrastructure installation, then remove the GIMR.

1. Confirm if Grid Infrastructure Management Repository (GIMR) is configured in the current release.

```
srvctl config mgmtldb
```

Note

If GIMR is not configured, then do not follow this procedure.

2. Confirm if Oracle Fleet Patching and Provisioning (Oracle FPP) is configured in central server mode in the current release.

```
srvctl config rhpsrv
```

Note

If Oracle FPP is configured on your cluster, then you are recommended to use the Oracle FPP Self-Upgrade feature for smooth migration of the metadata from GIMR to the new metadata repository. Refer to Oracle Fleet Patching and Provisioning Self Upgrade for more information about how to use the Oracle FPP Self-Upgrade feature.

3. As the `grid` user, log in to any cluster node and create a new directory owned by `grid` to store the GIMR deletion script.

```
mkdir -p $ORACLE_HOME/gimrdel  
chown grid:oinstall $ORACLE_HOME/gimrdel
```

4. Download `scriptgimr.zip` from the My Oracle Support Note 2972418.1 to the `$ORACLE_HOME/gimrdel` directory.
5. Extract the `reposScript.sh` script from the `scriptgimr.zip` and ensure that the `grid` user has read and execute permissions on the `reposScript.sh` script.

```
unzip -q $ORACLE_HOME/gimrdel/scriptgimr.zip
```

6. Optional: Query and export the CHA user models.

```
Grid_home/bin/chactl query model  
Grid_home/bin/chactl export model -name model_name -file model_name.svm
```

7. If Oracle FPP was configured in central mode, then export the Oracle FPP Metadata to re-configure Oracle FPP after upgrading to Oracle Grid Infrastructure 26ai.

```
Grid_home/crs/install/reposScript.sh -  
export_dir=dir_to_export_Oracle_FPP_metadata
```

8. Run the `reposScript.sh` script, in delete mode, from the `/gimrdel` directory.

```
$ORACLE_HOME/gimrdel/reposScript.sh -mode="Delete"
```

Note

Oracle FPP stops working if you delete the GIMR, but do not upgrade to Oracle Grid Infrastructure 26ai and re-configure Oracle FPP.

Related Topics

- [My Oracle Support Note 2972418.1](#)

2.1 Oracle Cluster Health Advisor Architecture

Oracle Cluster Health Advisor runs as a highly available cluster resource, `ochad`, on each node in the cluster.

Each Oracle Cluster Health Advisor daemon (`ochad`) monitors the operating system on the cluster node and optionally, each Oracle Real Application Clusters (Oracle RAC) database instance on the node.

The `ochad` daemon receives operating system metric data from the Cluster Health Monitor and gets Oracle RAC database instance metrics from a memory-mapped file. The daemon does not require a connection to each database instance. This data, along with the selected model, is used in the Health Prognostics Engine of Oracle Cluster Health Advisor for both the node and each monitored database instance in order to analyze their health multiple times a minute.

2.3 Monitoring the Oracle Real Application Clusters (Oracle RAC) Environment with Oracle Cluster Health Advisor

Oracle Cluster Health Advisor is automatically provisioned on each node by default when Oracle Grid Infrastructure is installed for Oracle Real Application Clusters (Oracle RAC) or Oracle RAC One Node database.

Oracle Cluster Health Advisor does not require any additional configuration.

When Oracle Cluster Health Advisor detects an Oracle Real Application Clusters (Oracle RAC) or Oracle RAC One Node database instance as running, Oracle Cluster Health Advisor autonomously starts monitoring the cluster nodes. Use CHACTL while logged in as the Grid user to turn on monitoring of the database.

To monitor the Oracle Real Application Clusters (Oracle RAC) environment:

1. To monitor a database, run the following command:

```
$ chactl monitor database -db db_unique_name
```

Oracle Cluster Health Advisor monitors all instances of the Oracle Real Application Clusters (Oracle RAC) or Oracle RAC One Node database using the default model. Oracle Cluster Health Advisor cannot monitor single-instance Oracle databases, even if the single-instance Oracle databases share the same cluster as Oracle Real Application Clusters (Oracle RAC) databases.

Each database instance is monitored independently both across Oracle Real Application Clusters (Oracle RAC) database nodes and when more than one database run on a single node.

2. To stop monitoring a database, run the following command:

```
$ chactl unmonitor database -db db_unique_name
```

Oracle Cluster Health Advisor stops monitoring all instances of the specified database. However, Oracle Cluster Health Advisor does not delete any data or problems until it is aged out beyond the retention period.

3. To check monitoring status of all cluster nodes and databases, run the following command:

```
$ chactl status
```

Use the `-verbose` option to see more details, such as the models used for the nodes and each database.

2.4 Using Cluster Health Advisor for Health Diagnosis

Oracle Cluster Health Advisor raises and clears problems autonomously.

The Oracle Grid Infrastructure user can query the stored information using CHACTL.

To query the diagnostic data:

1. To query currently open problems, run the following command:

```
chactl query diagnosis -db db_unique_name -start time -end time
```

In the syntax example, *db_unique_name* is the name of your database instance. You also specify the start time and end time for which you want to retrieve data. Specify date and time in the YYYY-MM-DD HH24:MI:SS format.

2. Use the `-htmlfile file_name` option to save the output in HTML format.

Example 2-1 Cluster Health Advisor Output Examples in Text and HTML Format

This example shows the default text output for the `chactl query diagnosis` command for a database named *oltpacdb*.

```
$ chactl query diagnosis -db oltpacdb -start "2016-02-01 02:52:50" -end
"2016-02-01 03:19:15"
2016-02-01 01:47:10.0 Database oltpacdb DB Control File IO Performance
(oltpacdb_1) [detected]
2016-02-01 01:47:10.0 Database oltpacdb DB Control File IO Performance
(oltpacdb_2) [detected]
2016-02-01 02:52:15.0 Database oltpacdb DB CPU Utilization (oltpacdb_2)
[detected]
2016-02-01 02:52:50.0 Database oltpacdb DB CPU Utilization (oltpacdb_1)
[detected]
2016-02-01 02:59:35.0 Database oltpacdb DB Log File Switch (oltpacdb_1)
[detected]
2016-02-01 02:59:45.0 Database oltpacdb DB Log File Switch (oltpacdb_2)
[detected]
```

Problem: DB Control File IO Performance

Description: CHA has detected that reads or writes to the control files are slower than expected.

Cause: The Cluster Health Advisor (CHA) detected that reads or writes to the control files were slow because of an increase in disk IO.

The slow control file reads and writes may have an impact on checkpoint and Log Writer (LGWR) performance.

Action: Separate the control files from other database files and move them to faster disks or Solid State Devices.

Problem: DB CPU Utilization

Description: CHA detected larger than expected CPU utilization for this database.

Cause: The Cluster Health Advisor (CHA) detected an increase in database CPU utilization because of an increase in the database workload.

Action: Identify the CPU intensive queries by using the Automatic Diagnostic and Defect Manager (ADDM) and follow the recommendations given there. Limit the number of CPU intensive queries or

relocate sessions to less busy machines. Add CPUs if the CPU capacity is insufficient to support

the load without a performance degradation or effects on other databases.

Problem: DB Log File Switch

Description: CHA detected that database sessions are waiting longer than expected for log switch completions.

Cause: The Cluster Health Advisor (CHA) detected high contention during log switches

because the redo log files were small and the redo logs switched frequently.

Action: Increase the size of the redo logs.

The timestamp displays date and time when the problem was detected on a specific host or database.

① Note

The same problem can occur on different hosts and at different times, yet the diagnosis shows complete details of the problem and its potential impact. Each problem also shows targeted corrective or preventive actions.

Here is an example of what the output looks like in the HTML format.

```
$ chactl query diagnosis -start "2016-07-03 20:50:00" -end "2016-07-04 03:50:00" -htmlfile ~/chaprob.html
```

Figure 2-1 Cluster Health Advisor Diagnosis HTML Output

Timestamp	Target Information	Event Name	Detected/Cleared
2016-07-03 01:49:30.0	Host rwsbi07	Host CPU Utilization	detected
2016-07-03 01:49:50.0	Host rwsbi06	Host CPU Utilization	detected
2016-07-03 05:54:55.0	Host rwsbi06	Host Memory Consumption	detected
2016-07-04 03:40:00.0	Host rwsbi07	Host CPU Utilization	cleared
2016-07-04 03:40:05.0	Host rwsbi06	Host CPU Utilization	cleared
2016-07-04 03:40:05.0	Host rwsbi06	Host Memory Consumption	cleared

Problem	Description	Cause	Action
Host CPU Utilization	CHA detected larger than expected CPU utilization on this node. The available CPU resource may not be sufficient to support application failover or relocation of databases to this node.	The Cluster Health Advisor (CHA) detected an unexpected increase in CPU utilization by databases or applications on this node.	Identify CPU intensive processes and databases by reviewing Cluster Health Monitoring (CHM) data. Relocate databases to less busy machines, or limit the number of connections to databases on this node. Add nodes if more resources are required.
Host Memory Consumption	CHA detected that more memory than expected is consumed on this server. The memory is not allocated by sessions of this database.	The Cluster Health Advisor (CHA) detected an increase in memory consumption by other databases or by applications not connected to a database on this node.	Identify the top memory consumers by using the Cluster Health Monitor (CHM).

Related Topics

- [chactl query diagnosis](#)

Use the `chactl query diagnosis` command to return problems and diagnosis, and suggested corrective actions associated with the problem for specific cluster nodes or Oracle Real Application Clusters (Oracle RAC) databases.

2.5 Calibrating an Oracle Cluster Health Advisor Model for a Cluster Deployment

As shipped with default node and database models, Oracle Cluster Health Advisor is designed not to generate false warning notifications.

You can increase the sensitivity and accuracy of the Oracle Cluster Health Advisor models for a specific workload using the `chactl calibrate` command.

Oracle recommends that a minimum of 6 hours of data be available and that both the cluster and databases use the same time range for calibration.

The `chactl calibrate` command analyzes a user-specified time interval that includes all workload phases operating normally. This data is collected while Oracle Cluster Health Advisor is monitoring the cluster and all the databases for which you want to calibrate.

1. To check if sufficient data is available, run the `query calibration` command.

Note

The `query calibration` command is supported only with GIMR. GIMR is optionally supported in Oracle Database 19c. However, it's desupported in Oracle AI Database 26ai.

If 720 or more records are available, then Oracle Cluster Health Advisor successfully performs the calibration. The calibration function may not consider some data records to be normally occurring for the workload profile being used. In this case, filter the data by using the `KPISET` parameters in both the `query calibration` command and the `calibrate` command.

For example:

```
$ chactl query calibration -db oltpacdb -timeranges  
'start=2016-07-26 01:00:00,end=2016-07-26 02:00:00,start=2016-07-26  
03:00:00,end=2016-07-26 04:00:00'  
-kpi set 'name=CPUPERCENT min=20 max=40, name=IOTHROUGHPUT min=500  
max=9000' -interval 2
```

2. Start the calibration and store the model under a user-specified name for the specified date and time range.

For example:

```
$ chactl calibrate cluster -model weekday -timeranges 'start=2016-07-03  
20:50:00,end=2016-07-04 15:00:00'
```

3. Use the new model to monitor the cluster as follows:

For example:

```
$ chactl monitor cluster -model weekday
```

Example 2-2 Output for the chactl query calibrate command

Database name : oltpacdb
 Start time : 2016-07-26 01:03:10
 End time : 2016-07-26 01:57:25
 Total Samples : 120
 Percentage of filtered data : 8.32%
 The number of data samples may not be sufficient for calibration.

1) Disk read (ASM) (Mbyte/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
4.96	0.20	8.98	0.06	25.68
<25	<50	<75	<100	>=100
97.50%	2.50%	0.00%	0.00%	0.00%

2) Disk write (ASM) (Mbyte/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
27.73	9.72	31.75	4.16	109.39
<50	<100	<150	<200	>=200
73.33%	22.50%	4.17%	0.00%	0.00%

3) Disk throughput (ASM) (IO/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
2407.50	1500.00	1978.55	700.00	7800.00
<5000	<10000	<15000	<20000	>=20000
83.33%	16.67%	0.00%	0.00%	0.00%

4) CPU utilization (total) (%)

MEAN	MEDIAN	STDDEV	MIN	MAX
21.99	21.75	1.36	20.00	26.80
<20	<40	<60	<80	>=80
0.00%	100.00%	0.00%	0.00%	0.00%

5) Database time per user call (usec/call)

MEAN	MEDIAN	STDDEV	MIN	MAX		
267.39	264.87	32.05	205.80	484.57		
<10000000	<20000000	<30000000	<40000000	<50000000	<60000000	<70000000
>=70000000						
100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Database name : oltpacdb
 Start time : 2016-07-26 03:00:00
 End time : 2016-07-26 03:53:30
 Total Samples : 342
 Percentage of filtered data : 23.72%
 The number of data samples may not be sufficient for calibration.

1) Disk read (ASM) (Mbyte/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
12.18	0.28	16.07	0.05	60.98
<25	<50	<75	<100	>=100
64.33%	34.50%	1.17%	0.00%	0.00%

2) Disk write (ASM) (Mbyte/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
57.57	51.14	34.12	16.10	135.29
<50	<100	<150	<200	>=200
49.12%	38.30%	12.57%	0.00%	0.00%

3) Disk throughput (ASM) (IO/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
5048.83	4300.00	1730.17	2700.00	9000.00
<5000	<10000	<15000	<20000	>=20000
63.74%	36.26%	0.00%	0.00%	0.00%

4) CPU utilization (total) (%)

MEAN	MEDIAN	STDDEV	MIN	MAX
23.10	22.80	1.88	20.00	31.40
<20	<40	<60	<80	>=80
0.00%	100.00%	0.00%	0.00%	0.00%

5) Database time per user call (usec/call)

MEAN	MEDIAN	STDDEV	MIN	MAX			
744.39	256.47	2892.71	211.45	45438.35			
<10000000	<20000000	<30000000	<40000000	<50000000	<60000000	<70000000	
>=70000000							
100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	

Related Topics

- [chactl calibrate](#)
Use the `chactl calibrate` command to create a new model that has greater sensitivity and accuracy.
- [chactl query calibration](#)
Use the `chactl query calibration` command to view detailed information about the calibration data of a specific target.
- [chactl Command Reference](#)
The Oracle Cluster Health Advisor commands enable the Oracle Grid Infrastructure user to administer basic monitoring functionality on the targets.

2.6 Viewing the Details for an Oracle Cluster Health Advisor Model

Use the `chactl query model` command to view the model details.

- You can review the details of an Oracle Cluster Health Advisor model at any time using the `chactl query model` command.

For example:

```
$ chactl query model -name weekday
Model: weekday
Target Type: CLUSTERWARE
Version: OS12.2_V14_0.9.8
OS Calibrated on: Linux amd64
Calibration Target Name: MYCLUSTER
Calibration Date: 2016-07-05 01:13:49
Calibration Time Ranges: start=2016-07-03 20:50:00,end=2016-07-04 15:00:00
Calibration KPIs: not specified
```

You can also rename, import, export, and delete the models.

2.7 Managing the Oracle Cluster Health Advisor Repository

Oracle Cluster Health Advisor repository stores the historical records of cluster host problems, database problems, and associated metric evidence, along with models.

Note

Applicable only if GIMR is configured. GIMR is optionally supported in Oracle Database 19c. However, it's desupported in Oracle AI Database 26ai.

The Oracle Cluster Health Advisor repository is used to diagnose and triage periodic problems. By default, the repository is sized to retain data for 16 targets (nodes and database instances) for 72 hours. If the number of targets increase, then the retention time is automatically decreased. Oracle Cluster Health Advisor generates warning messages when the retention time goes below 72 hours, and stops monitoring and generates a critical alert when the retention time goes below 24 hours.

Use `CHACTL` commands to manage the repository and set the maximum retention time.

- To retrieve the repository details, use the following command:

```
$ chactl query repository
```

For example, running the command mentioned earlier shows the following output:

```
specified max retention time(hrs) : 72
available retention time(hrs)      : 212
available number of entities       : 2
```

```
allocated number of entities      : 0
total repository size(gb)        : 2.00
allocated repository size(gb)     : 0.07
```

2. To set the maximum retention time in hours, based on the current number of targets being monitored, use the following command:

```
$ chactl set maxretention -time number_of_hours
```

For example:

```
$ chactl set maxretention -time 80
max retention successfully set to 80 hours
```

Note

The `maxretention` setting limits the oldest data retained in the repository, but is not guaranteed to be maintained if the number of monitored targets increase. In this case, if the combination of monitored targets and number of hours are not sufficient, then increase the size of the Oracle Cluster Health Advisor repository.

3. To increase the size of the Oracle Cluster Health Advisor repository, use the `chactl resize repository` command.

For example, to resize the repository to support 32 targets using the currently set maximum retention time, you would use the following command:

```
$ chactl resize repository -entities 32
repository successfully resized for 32 targets
```

2.8 Viewing the Status of Cluster Health Advisor

SRVCTL commands are the tools that offer total control on managing the life cycle of Oracle Cluster Health Advisor as a highly available service.

Use SRVCTL commands to check the status and configuration of Oracle Cluster Health Advisor service on any active hub or leaf nodes of the Oracle RAC cluster.

Note

A target is monitored only if it is running and the Oracle Cluster Health Advisor service is also running on the host node where the target exists.

1. To check the status of Oracle Cluster Health Advisor service on all nodes in the Oracle RAC cluster:

```
srvctl status cha [-help]
```

For example:

```
# srvctl status cha
Cluster Health Advisor is running on nodes racNode1, racNode2.
Cluster Health Advisor is not running on nodes racNode3, racNode4.
```

2. To check if Oracle Cluster Health Advisor service is enabled or disabled on all nodes in the Oracle RAC cluster:

```
srvctl config cha [-help]
```

For example:

```
# srvctl config cha
Cluster Health Advisor is enabled on nodes racNode1, racNode2.
Cluster Health Advisor is not enabled on nodes racNode3, racNode4.
```

2.9 Enhanced Cluster Health Advisor Support for Oracle Pluggable Databases

The Cluster Health Advisor (CHA) diagnostic capabilities have been extended to support 4K PDBs, up from 256 in Oracle AI Database 26ai.

Going forward, this is crucial for Oracle Autonomous Database deployments. CHA's problem detection and root cause analysis will be improved by considering DB events such as reconfiguration. This improves detection, analysis, and targeted preventative actions for problems such as instance evictions.

Part II

Automatically Monitoring the Cluster

You can use components of Autonomous Health Framework to monitor your cluster on a regular basis.

- [Collecting Operating System Resources Metrics](#)
CHM is a high-performance, lightweight daemon that collects, analyzes, aggregates, and stores a large set of operating system metrics to help you diagnose and troubleshoot system issues.
- [Monitoring System Metrics for Cluster Nodes](#)
This chapter explains the methods to monitor Oracle Clusterware.

3

Collecting Operating System Resources Metrics

CHM is a high-performance, lightweight daemon that collects, analyzes, aggregates, and stores a large set of operating system metrics to help you diagnose and troubleshoot system issues.

Supported Platforms

Linux, Microsoft Windows, Solaris, AIX, IBM Z Series, and ARM

Why CHM is unique

CHM	Typical OS Collector
Last man standing - daemon runs memory locked, RT scheduling class ensuring consistent data collection under system load.	Inconsistent data dropouts due to scheduling delays under system load.
High fidelity data sampling rate, 5 seconds. Very low resource usage profile at 5-second sampling rates.	Running multiple utilities creates additional overhead on the system being monitored, and worsens with higher sampling rates.
High Availability daemon, collated data collections across multiple resource categories. Highly optimized collector (data read directly from the operating system, same source as utilities).	Set of scripts/command-line utilities, for example, <code>top</code> , <code>ps</code> , <code>vmstat</code> , <code>iostat</code> , and so on re-directing their output to one or more files for every collection sample.
Collected data is collated into a system snapshot overview (Nodeview) on every sample, Nodeview also contains additional summarization and analysis of the collected data across multiple resource categories.	System snapshot overviews across different resource categories are very tedious to collate.
Significant inline analysis and summarization during data collection and collation into the Nodeview greatly reduces tedious, manual, time-consuming analysis to drive meaningful insights.	The analysis is time-consuming and processing-intensive as the output of various utilities across multiple files needs to be collated, parsed, interpreted, and then analyzed for meaningful insights.
Performs Clusterware-aware specific metrics collection (Process Aggregates, ASM/OCR/VD disk tagging, Private/Public NIC tagging). Also provides an extensive toolset for in-depth data analysis and visualization.	None

- [Understanding Cluster Health Monitor Services](#)
Cluster Health Monitor uses system monitor (`osysmond`) service to collect operating system metrics.
- [Collecting Cluster Health Monitor Data](#)
Collect Cluster Health Monitor data from any node in the cluster.
- [Operating System Metrics Collected by Cluster Health Monitor](#)
Review the metrics collected by CHM.

- [Detecting Component Failures and Self-healing Autonomously](#)
Improved ability to detect component failures and self-heal autonomously improves business continuity.

Related Topics

- [Introduction to Cluster Health Monitor](#)
Cluster Health Monitor is a component of Oracle Grid Infrastructure, which continuously monitors and stores Oracle Clusterware and operating system resources metrics.

3.1 Understanding Cluster Health Monitor Services

Cluster Health Monitor uses system monitor (`osysmond`) service to collect operating system metrics.

About the System Monitor Service

The system monitor service (`osysmond`) is a real-time monitoring and operating system metric collection service that runs on each cluster node. The system monitor service is managed as a High Availability Services (HAS) resource.

`osysmond` persists the collected operating system metrics under a directory in `ORACLE_BASE`.

Metric Repository is auto-managed on the local filesystem. You can change the location and size of the repository.

- Nodeview samples are continuously written to the repository (JSON record)
- Historical data is auto-archived into hourly zip files
- Archived files are automatically purged once the default retention limit is reached (default: 200 MB)

3.2 Collecting Cluster Health Monitor Data

Collect Cluster Health Monitor data from any node in the cluster.

Oracle recommends that you run the `tfactl diagcollect` command to collect diagnostic data when an Oracle Clusterware error occurs.

3.3 Operating System Metrics Collected by Cluster Health Monitor

Review the metrics collected by CHM.

Overview of Metrics

CHM groups the operating system data collected into a **Nodeview**. A **Nodeview** is a grouping of metric sets where each metric set contains detailed metrics of a unique system resource.

Brief description of metric sets are as follows:

- **CPU metric set:** Metrics for top 127 CPUs sorted by usage percentage
- **Device metric set:** Metrics for 127 devices that include ASM/VD/OCR along with those having a high average wait time
- **Process metric set:** Metrics for 127 processes

- Top 25 CPU consumers (idle processes not reported)
- Top 25 Memory consumers (RSS < 1% of total RAM not reported)
- Top 25 I/O consumers
- Top 25 File Descriptors consumers (helps to identify top inode consumers)
- Process Aggregation: Metrics summarized by foreground and background processes for all Oracle Database and Oracle ASM instances
- **Network metric set:** Metrics for 16 NICS that include public and private interconnects
- **NFS metric set:** Metrics for 32 NFS ordered by round trip time
- **Protocol metric set:** Metrics for protocol groups TCP, UDP, and IP
- **Filesystem metric set:** Metrics for filesystem utilization
- **Critical resources metric set:** Metrics for critical system resource utilization
 - CPU Metrics: system-wide CPU utilization statistics
 - Memory Metrics: system-wide memory statistics
 - Device Metrics: system-wide device statistics distinct from individual device metric set
 - NFS Metrics: Total NFS devices collected every 30 seconds
 - Process Metrics: system-wide unique process metrics

CPU Metric Set

Contains metrics from all CPU cores ordered by usage percentage.

Table 3-1 CPU Metric Set

Metric Name (units)	Description
system [%]	Percentage of CPU utilization occurred while running at the system level (kernel).
user [%]	Percentage of CPU utilization occurred while running at the user level (application).
usage [%]	Total utilization (system[%] + user[%]).
nice [%]	Percentage of CPU utilization occurred while running at the user level with nice priority.
ioWait [%]	Percentage of time that the CPU was idle during which the system had an outstanding disk I/O request.
steal [%]	Percentage of time spent in involuntary wait by the virtual CPU while the hypervisor was servicing another virtual processor.

Device Metric Set

Contains metrics from all disk devices/partitions ordered by their service time in milliseconds.

Table 3-2 Device Metric Set

Metric Name (units)	Description
ioR [KB/s]	Amount of data read from the device.
ioW [KB/s]	Amount of data written to the device.

Table 3-2 (Cont.) Device Metric Set

Metric Name (units)	Description
numIOs [#s]	Average disk I/O operations.
qLen [#]	Number of I/O queued requests, that is, in a wait state.
aWait [msec]	Average wait time per I/O.
svcTm [msec]	Average service time per I/O request.
util [%]	Percent utilization of the device (same as '%util' metric from the <code>iostat -x</code> command. Represents the percentage of time device was active).

Process Metric Set

Contains multiple categories of summarized metric data computed across all system processes.

Table 3-3 Process Metric Set

Metric Name (units)	Description
pid	Process ID.
pri	Process priority (raw value from the operating system).
psr	The processor that process is currently assigned to or running on.
pPid	Parent process ID.
nice	Nice value of the process.
state	State of the process. For example, R->Running, S->Interruptible sleep, and so on.
class	Scheduling class of the process. For example, RR->RobinRound, FF->First in First out, B->Batch scheduling, and so on.
fd [#]	Number of file descriptors opened by this process, which is updated every 30 seconds.
name	Name of the process.
cpu [%]	Process CPU utilization across cores. For example, 50% => 50% of single core, 400% => 100% usage of 4 cores.
thrds [#]	Number of threads created by this process.
vmem [KB]	Process virtual memory usage (KB).
shMem [KB]	Process shared memory usage (KB).
rss [KB]	Process memory-resident set size (KB).
ioR [KB/s]	I/O read in kilobytes per second.
ioW [KB/s]	I/O write in kilobytes per second.
ioT [KB/s]	I/O total in kilobytes per second.
cswch [#s]	Context switch per second. Collected only for a few critical Oracle Database processes.
nvcswh [#s]	Non-voluntary context switch per second. Collected only for a few critical Oracle Database processes.

Table 3-3 (Cont.) Process Metric Set

Metric Name (units)	Description
cumulativeCpu [ms]	Amount of CPU used so far by the process in microseconds.

NIC Metric Set

Contains metrics from all network interfaces ordered by their total rate in kilobytes per second.

Table 3-4 NIC Metric Set

Metric Name (units)	Description
name	Name of the interface.
tag	Tag for the interface, for example, public , private , and so on.
mtu [B]	Size of the maximum transmission unit in bytes supported for the interface.
rx [Kbps]	Average network receive rate.
tx [Kbps]	Average network send rate.
total [Kbps]	Average network transmission rate (rx[Kb/s] + tx[Kb/s]).
rxPkt [#s]	Average incoming packet rate.
txPkt [#s]	Average outgoing packet rate.
pkt [#s]	Average rate of packet transmission (rxPkt[#s] + txPkt[#s]).
rxDscrd [#s]	Average rate of dropped/discarded incoming packets.
txDscrd [#s]	Average rate of dropped/discarded outgoing packets.
rxUnicast [#s]	Average rate of unicast packets received.
rxNonUnicast [#s]	Average rate of multicast packets received.
dscrd [#s]	Average rate of total discarded packets (rxDscrd + txDscrd).
rxErr [#s]	Average error rate for incoming packets.
txErr [#s]	Average error rate for outgoing packets.
Err [#s]	Average error rate of total transmission (rxErr[#s] + txErr[#s]).

NFS Metric Set

Contains top 32 NFS ordered by round trip time. This metric set is collected once every 30 seconds.

Table 3-5 NFS Metric Set

Metric Name (units)	Description
op [#s]	Number of read/write operations issued to a filesystem per second.

Table 3-5 (Cont.) NFS Metric Set

Metric Name (units)	Description
bytes [#/sec]	Number of bytes read/write per second from a filesystem.
rtt [s]	This is the duration from the time that the client's kernel sends the RPC request until the time it receives the reply.
exe [s]	This is the duration from that NFS client does the RPC request to its kernel until the RPC request is completed, this includes the RTT time above.
retrains [%]	This is the retransmission's frequency in percentage.

Protocol Metric Set

Contains specific metrics for protocol groups TCP, UDP, and IP. Metric values are cumulative since the system starts.

Table 3-6 TCP Metric Set

Metric Name (units)	Description
failedConnErr [#]	Number of times that TCP connections have made a direct transition to the CLOSED state from either the SYN-SENT state or the SYN-RCVD state, plus the number of times that TCP connections have made a direct transition to the LISTEN state from the SYN-RCVD state.
estResetErr [#]	Number of times that TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state.
segRetransErr [#]	Total number of TCP segments retransmitted.
rxSeg [#]	Total number of TCP segments received on TCP layer.
txSeg [#]	Total number of TCP segments sent from TCP layer.

Table 3-7 UDP Metric Set

Metric Name (units)	Description
unkPortErr [#]	Total number of received datagrams for which there was no application at the destination port.
rxErr [#]	Number of received datagrams that could not be delivered for reasons other than the lack of an application at the destination port.
rxPkt [#]	Total number of packets received.
txPkt [#]	Total number of packets sent.

Table 3-8 IP Metric Set

Metric Name (units)	Description
ipHdrErr [#]	Number of input datagrams discarded due to errors in their IPv4 headers.
addrErr [#]	Number of input datagrams discarded because the IPv4 address in their IPv4 header's destination field was not a valid address to be received at this entity.
unkProtoErr [#]	Number of locally-addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.
reasFailErr [#]	Number of failures detected by the IPv4 reassembly algorithm.
fragFailErr [#]	Number of IPv4 discarded datagrams due to fragmentation failures.
rxPkt [#]	Total number of packets received on IP layer.
txPkt [#]	Total number of packets sent from IP layer.

Filesystem Metric Set

Contains metrics for filesystem utilization. Collected only for **GRID_HOME** filesystem.

Table 3-9 Filesystem Metric Set

Metric Name (units)	Description
mount	Mount point.
type	Filesystem type, for example, ext4 .
tag	Filesystem tag, for example, GRID_HOME .
total [KB]	Total amount of space (KB).
used [KB]	Amount of used space (KB).
avbl [KB]	Amount of available space (KB).
used [%]	Percentage of used space.
ifree [%]	Percentage of free file nodes.

System Metric Set

Contains a summarized metric set of critical system resource utilization.

Table 3-10 CPU Metrics

Metric Name (units)	Description
pCpus [#]	Number of physical processing units in the system.
Cores [#]	Number of cores for all CPUs in the system.
vCpus [#]	Number of logical processing units in the system.
cpuHt	CPU Hyperthreading enabled (Y) or disabled (N).
osName	Name of the operating system.
chipName	Name of the chip of the processing unit.
system [%]	Percentage of CPUs utilization that occurred while running at the system level (kernel).

Table 3-10 (Cont.) CPU Metrics

Metric Name (units)	Description
user [%]	Percentage of CPUs utilization that occurred while running at the user level (application).
usage [%]	Total CPU utilization (system[%] + user[%]).
nice [%]	Percentage of CPUs utilization occurred while running at the user level with NICE priority.
ioWait [%]	Percentage of time that the CPUs were idle during which the system had an outstanding disk I/O request.
Steal [%]	Percentage of time spent in involuntary wait by the virtual CPUs while the hypervisor was servicing another virtual processor.
cpuQ [#]	Number of processes waiting in the run queue within the current sample interval.
loadAvg1	Average system load calculated over time of one minute.
loadAvg5	Average system load calculated over time of five minutes.
loadAvg15	Average system load calculated over time of 15 minutes. High load averages imply that a system is overloaded; many processes are waiting for CPU time.
Intr [#s]	Number of interrupts occurred per second in the system.
ctxSwitch [#s]	Number of context switches that occurred per second in the system.

Table 3-11 Memory Metrics

Metric Name (units)	Description
totalMem [KB]	Amount of total usable RAM (KB).
freeMem [KB]	Amount of free RAM (KB).
avblMem [KB]	Amount of memory available to start a new process without swapping.
shMem [KB]	Memory used (mostly) by tmpfs .
swapTotal [KB]	Total amount of physical swap memory (KB).
swapFree [KB]	Amount of swap memory free (KB).
swpIn [KB/s]	Average swap in rate within the current sample interval (KB/sec).
swpOut [KB/s]	Average swap-out rate within the current sample interval (KB/sec).
pgIn [#s]	Average page in rate within the current sample interval (pages/sec).
pgOut [#s]	Average page out rate within the current sample interval (pages/sec).
slabReclaim [KB]	The part of the slab that might be reclaimed such as caches.
buffer [KB]	Memory used by kernel buffers.
Cache [KB]	Memory used by the page cache and slabs.

Table 3-11 (Cont.) Memory Metrics

Metric Name (units)	Description
bufferAndCache [KB]	Total size of buffer and cache (buffer[KB] + Cache[KB]).
hugePageTotal [#]	Total number of huge pages present in the system for the current sample interval.
hugePageFree [KB]	Total number of free huge pages in the system for the current sample interval.
hugePageSize [KB]	Size of one huge page in KB, depends on the operating system version. Typically the same for all samples for a particular host.

Table 3-12 Device Metrics

Metric Name (units)	Description
disks [#]	Number of disks configured in the system.
ioR [KB/s]	Aggregate read rate across all devices.
ioW [KB/s]	Aggregate write rate across all devices.
numIOs [#s]	Aggregate I/O operation rate across all devices.

Table 3-13 NFS Metrics

Metric Name (units)	Description
nfs [#]	Total NFS devices.

Table 3-14 Process Metrics

Metric Name (units)	Description
fds [#]	Number of open file structs in system.
procs [#]	Number of processes.
rtProcs [#]	Number of real-time processes.
procsInDState	Number of processes in uninterruptible sleep.
sysFdLimit [#]	System limit on a number of file structs.
procsOnCpu [#]	Number of processes currently running on CPU.
procsBlocked [#]	Number of processes waiting for some event/resource becomes available, such as for the completion of an I/O operation.

Process Aggregates Metric Set

Contains aggregated metrics for all processes by process groups.

Table 3-15 Process Aggregates Metric Set

Metric Name (units)	Description
DBBG	User Oracle Database background process group.
DBFG	User Oracle Database foreground process group.

Table 3-15 (Cont.) Process Aggregates Metric Set

Metric Name (units)	Description
MDBBG	MGMTDB background processes group.
MDBFG	MGMTDB foreground processes group.
ASMBG	ASM background processes group.
ASMFG	ASM foreground processes group.
IOXBG	IOS background processes group.
IOXFG	IOS foreground processes group.
APXBG	APX background processes group.
APXFG	APX foreground processes group.
CLUST	Clusterware processes group.
OTHER	Default group.

For each group, the below metrics are aggregated to report a group summary.

Metric Name (units)	Description
processes [#]	Total number of processes in the group.
cpu [%]	Aggregated CPU utilization.
rss [KB]	Aggregated resident set size.
shMem [KB]	Aggregated shared memory usage.
thrds [#]	Aggregated thread count.
fds [#]	Aggregated open file-descriptor.
cpuWeight [%]	Contribution of the group in overall CPU utilization of the machine.

3.4 Detecting Component Failures and Self-healing Autonomously

Improved ability to detect component failures and self-heal autonomously improves business continuity.

Cluster Health Monitor introduces a new diagnostic feature that identifies critical component events that indicate pending or actual failures and provides recommendations for corrective action. These actions may sometimes be performed autonomously. Such events and actions are then captured and admins are notified through components such as Oracle Trace File Analyzer.

Terms Associated with Diagnosability

CHMDiag: CHMDiag is a python daemon managed by `osysmond` that listens for events and takes actions. Upon receiving various events/actions, CHMDiag validates them for correctness, does flow control, and schedules the actions for runs. CHMDiag monitors each action to its completion, and kills an action if it takes longer than pre-configured time specific to that action.

This JSON file describes all events/actions and their respective attributes. All events/actions have uniquely identifiable IDs. This file also contains various configurable properties for various actions/events. CHMDiag loads this file during its startup.

CRFE API: CRFE API is used by all C clients to send events to CHMDiag. This API is used by internal clients like components (RDBMS/CSS/GIPC) to publish events/actions.

This API also provides support for both synchronous and asynchronous publication of events. Asynchronous publication of events is done through a background thread which will be shared by all CRFE API clients within a process.

CHMDIAG_BASE: This directory resides in `ORACLEB_BASE/hostname/crf/chmddiag`. This directory path contains following directories, which are populated or managed by CHMDiag.

- **ActionsResults:** Contains all results for all of the invoked actions with a subdirectory for each action.
- **EventsLog:** Contains a log of all the events/actions received by CHMDiag and the location of their respective action results. These log files are also auto-rotated after reaching a fixed size.
- **CHMDiagLog:** Contains CHMDiag daemon logs. Log files are auto-rotated and once they reach a specific size. Logs should have sufficient debug information to diagnose any problems that CHMDiag could run into.
- **Config:** Contains a run sub-directory for CHMDiag process pid file management.

New commands to query, collect, and describe CHMDiag events/actions sent by various components:

- **oclumon chmddiag description:** Use the `oclumon chmddiag description` command to get a detailed description of all the supported events and actions.
- **oclumon chmddiag query:** Use the `oclumon chmddiag query` command to query CHMDiag events/actions sent by various components and generate an HTML or a text report.
- **oclumon chmddiag collect:** Use the `oclumon chmddiag collect` command to collect all events/actions data generated by CHMDiag into the specified output directory location.

Related Topics

-
-
-

4

Monitoring System Metrics for Cluster Nodes

This chapter explains the methods to monitor Oracle Clusterware.

Oracle recommends that you use Oracle Enterprise Manager to monitor everyday operations of Oracle Clusterware.

Cluster Health Monitor monitors the complete technology stack, including the operating system, ensuring smooth cluster operations. Both the components are enabled, by default, for any Oracle cluster. Oracle strongly recommends that you use both the components. Also, monitor Oracle Clusterware-managed resources using the Clusterware resource activity log.

- [Monitoring Oracle Clusterware with Oracle Enterprise Manager](#)
Use Oracle Enterprise Manager to monitor the Oracle Clusterware environment.
- [Monitoring Oracle Clusterware with Cluster Health Monitor](#)
You can use the OCLUMON command-line tool to interact with Cluster Health Monitor.

4.1 Monitoring Oracle Clusterware with Oracle Enterprise Manager

Use Oracle Enterprise Manager to monitor the Oracle Clusterware environment.

When you log in to Oracle Enterprise Manager using a client browser, the **Cluster Database Home** page appears where you can monitor the status of both Oracle Database and Oracle Clusterware environments. Oracle Clusterware monitoring includes the following details:

- Notifications if there are any VIP relocations
- Status of the Oracle Clusterware on each node of the cluster using information obtained through the Cluster Verification Utility (CVU)
- Notifications if node applications (`nodeapps`) start or stop
- Notification of issues in the Oracle Clusterware alert log for the Oracle Cluster Registry, voting file issues (if any), and node evictions

The **Cluster Database Home** page is similar to a single-instance Database Home page. However, on the Cluster Database Home page, Oracle Enterprise Manager displays the system state and availability. The system state and availability includes a summary about alert messages and job activity, and links to all the database and Oracle Automatic Storage Management (Oracle ASM) instances. For example, track problems with services on the cluster including when a service is not running on all the preferred instances or when a service response time threshold is not being met.

Use the Oracle Enterprise Manager **Interconnects** page to monitor the Oracle Clusterware environment. The Interconnects page displays the following details:

- Public and private interfaces on the cluster
- Overall throughput on the private interconnect
- Individual throughput on each of the network interfaces
- Error rates (if any)

- Load contributed by database instances on the interconnect
- Notifications if a database instance is using public interface due to misconfiguration
- Throughput contributed by individual instances on the interconnect

All the information listed earlier is also available as collections that have a historic view. The historic view is useful with cluster cache coherency, such as when diagnosing problems related to cluster wait events. Access the Interconnects page by clicking the **Interconnect** tab on the Cluster Database home page.

Also, the Oracle Enterprise Manager **Cluster Database Performance** page provides a quick glimpse of the performance statistics for a database. Statistics are rolled up across all the instances in the cluster database in charts. Using the links next to the charts, you can get more specific information and perform any of the following tasks:

- Identify the causes of performance issues
- Decide whether resources must be added or redistributed
- Tune your SQL plan and schema for better optimization
- Resolve performance issues

The charts on the Cluster Database Performance page include the following:

- **Chart for Cluster Host Load Average:** The **Cluster Host Load Average** chart in the Cluster Database Performance page shows potential problems that are outside the database. The chart shows maximum, average, and minimum load values for available nodes in the cluster for the previous hour.
- **Chart for Global Cache Block Access Latency:** Each cluster database instance has its own buffer cache in its System Global Area (SGA). Using Cache Fusion, Oracle RAC environments logically combine buffer cache of each instance to enable the database instances to process data as if the data resided on a logically combined, single cache.
- **Chart for Average Active Sessions:** The **Average Active Sessions** chart in the Cluster Database Performance page shows potential problems inside the database. Categories, called wait classes, show how much of the database is using a resource, such as CPU or disk I/O. Comparing CPU time to wait time helps to determine how much of the response time is consumed with useful work rather than waiting for resources that are potentially held by other processes.
- **Chart for Database Throughput:** The **Database Throughput** charts summarize any resource contention that appears in the Average Active Sessions chart, and also show how much work the database is performing on behalf of the users or applications. The **Per Second** view shows the number of transactions compared to the number of logons, and the amount of physical reads compared to the redo size for each second. The **Per Transaction** view shows the amount of physical reads compared to the redo size for each transaction. Logons is the number of users that are logged on to the database.

In addition, the **Top Activity** drop-down menu on the **Cluster Database Performance** page enables you to see the activity by wait events, services, and instances. In addition, you can see the details about SQL/sessions by going to a prior point in time by moving the slider on the chart.

4.2 Monitoring Oracle Clusterware with Cluster Health Monitor

You can use the OCLUMON command-line tool to interact with Cluster Health Monitor.

OCLUMON is included with Cluster Health Monitor. You can use it to query the Cluster Health Monitor repository to display node-specific metrics for a specified time period. You can also use OCLUMON to perform miscellaneous administrative tasks, such as the following:

- Changing the debug levels with the `oclumon debug` command
- Querying the version of Cluster Health Monitor with the `oclumon version` command
- Viewing the collected information in the form of a node view using the `oclumon dumpnodeview` command
- Changing the metrics datafile size using the `ocloumon manage` command

Related Topics

- [OCLUMON Command Reference](#)
Use the command-line tool to query the Cluster Health Monitor repository to display node-specific metrics for a specific time period.

Part III

Automatic Problem Solving

Some situations can be automatically resolved with tools in the Autonomous Health Framework.

- [Resolving Database and Database Instance Delays](#)
Blocker Resolver preserves the database performance by resolving delays and keeping the resources available.

5

Resolving Database and Database Instance Delays

Blocker Resolver preserves the database performance by resolving delays and keeping the resources available.

- [Blocker Resolver Architecture](#)
Blocker Resolver autonomously runs as a `DIA0` task within the database.
- [Optional Configuration for Blocker Resolver](#)
You can adjust the sensitivity, and control the size and number of the log files used by Blocker Resolver.
- [Blocker Resolver Diagnostics and Logging](#)
Blocker Resolver autonomously resolves delays and continuously logs the resolutions in the database alert logs and the diagnostics in the trace files.

Related Topics

- [Introduction to Blocker Resolver](#)
Blocker Resolver is an Oracle Real Application Clusters (Oracle RAC) environment feature that autonomously resolves delays and keeps the resources available.

5.1 Blocker Resolver Architecture

Blocker Resolver autonomously runs as a `DIA0` task within the database.

Blocker Resolver works in the following three phases:

- **Detect:** In this phase, Blocker Resolver collects the data on all the nodes and detects the sessions that are waiting for the resources held by another session.
- **Analyze:** In this phase, Blocker Resolver analyzes the sessions detected in the **Detect** phase to determine if the sessions are part of a potential delay. If the sessions are suspected as delayed, Blocker Resolver then waits for a certain threshold time period to ensure that the sessions are delayed.
- **Verify:** In this phase, after the threshold time period is up, Blocker Resolver verifies that the sessions are delayed and selects a session that's causing the delay.

After selecting the session that's causing the delay, Blocker Resolver applies resolution methods on that session. If the chain of sessions or the delay resolves automatically, then Blocker Resolver does not apply delay resolution methods. However, if the delay does not resolve by itself, then Blocker Resolver resolves the delay by terminating the session that's causing the delay. If terminating the session fails, then Blocker Resolver terminates the process of the session. This entire process is autonomous and does not block resources for a long period and does not affect the performance.

For example, if a high rank session is included in the chain of delayed sessions, then Blocker Resolver expedites the termination of the session that's causing the delay. Termination of the session that's causing the delay prevents the high rank session from waiting too long and helps to maintain performance objective of the high rank session.

5.2 Optional Configuration for Blocker Resolver

You can adjust the sensitivity, and control the size and number of the log files used by Blocker Resolver.

Note

The `DBMS_HANG_MANAGER` package is deprecated in Oracle AI Database 26ai. Use `DBMS_BLOCKER_RESOLVER` instead. The `DBMS_HANG_MANAGER` package provides a method of changing some configuration parameters and constraints to address session issues. This package is being replaced with `DBMS_BLOCKER_RESOLVER`. `DBMS_HANG_MANAGER` can be removed in a future release.

Sensitivity

If Blocker Resolver detects a delay, then Blocker Resolver waits for a certain threshold time period to ensure that the sessions are delayed. Change threshold time period by using `DBMS_BLOCKER_RESOLVER` to set the `sensitivity` parameter to either `Normal` or `High`. If the `sensitivity` parameter is set to `Normal`, then Blocker Resolver waits for the default time period. However, if the sensitivity is set to `High`, then the time period is reduced by 50%.

By default, the `sensitivity` parameter is set to `Normal`. To set Blocker Resolver sensitivity, run the following commands in SQL*Plus as SYS user:

- To set the `sensitivity` parameter to `Normal`:

```
exec dbms_blocker_resolver.set(dbms_blocker_resolver.sensitivity,  
dbms_blocker_resolver.sensitivity_normal);
```

- To set the `sensitivity` parameter to `High`:

```
exec dbms_blocker_resolver.set(dbms_blocker_resolver.sensitivity,  
dbms_blocker_resolver.sensitivity_high);
```

Size of the Trace Log File

The Blocker Resolver logs detailed diagnostics of the delays in the trace files with `_base_` in the file name. Change the size of the trace files in bytes with the `base_file_size_limit` parameter. Run the following command in SQL*Plus, for example, to set the trace file size limit to 100 MB:

```
exec dbms_blocker_resolver.set(dbms_blocker_resolver.base_file_size_limit,  
104857600);
```

Number of Trace Log Files

The base Blocker Resolver trace files are part of a trace file set. Change the number of trace files in trace file set with the `base_file_set_count` parameter. Run the following command in SQL*Plus, for example, to set the number of trace files in trace file set to 6:

```
exec dbms_blocker_resolver.set(dbms_blocker_resolver.base_file_set_count,6);
```

By default, `base_file_set_count` parameter is set to 5.

5.3 Blocker Resolver Diagnostics and Logging

Blocker Resolver autonomously resolves delays and continuously logs the resolutions in the database alert logs and the diagnostics in the trace files.

Blocker Resolver logs the resolutions in the database alert logs as Automatic Diagnostic Repository (ADR) incidents with incident code `ORA-32701`.

You also get detailed diagnostics about the delay detection in the trace files. Trace files and alert logs have file names starting with `database_instance_dia0_`.

- The trace files are stored in the `$ ADR_BASE/diag/rdbms/database name/database_instance/incident/incdir_XXXXXX` directory
- The alert logs are stored in the `$ ADR_BASE/diag/rdbms/database name/database_instance/trace` directory

Example 5-1 Blocker Resolver Trace File for a Local Instance

This example shows an example of the output you see for Blocker Resolver for the local database instance

```
Trace Log File .../oracle/log/diag/rdbms/hm1/hm1/incident/incdir_111/
hm1_dia0_11111_i1111.trc
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
...
*** 2016-07-16T12:39:02.715475-07:00
HM: Hang Statistics - only statistics with non-zero values are listed

      current number of active sessions 3
      current number of hung sessions 1
instance health (in terms of hung sessions) 66.67%
      number of cluster-wide active sessions 9
      number of cluster-wide hung sessions 5
      cluster health (in terms of hung sessions) 44.45%

*** 2016-07-16T12:39:02.715681-07:00
Resolvable Hangs in the System
```

Hang ID	Hang Type	Hang Status	Root Inst Num	Root Sess	Chain #hung Sess	Total #hung Sess	Hang Conf	Hang Span	Hang Resolution Action
1	HANG	RSLNPEND	3	44	3	5	HIGH	GLOBAL	Terminate Process

```
Hang Resolution Reason: Although hangs of this root type are typically
self-resolving, the previously ignored hang was automatically resolved.
```

Example 5-2 Error Message in the Alert Log Indicating a Delayed Session

This example shows an example of a Blocker Resolver alert log on the primary instance

```
2016-07-16T12:39:02.616573-07:00
Errors in file .../oracle/log/diag/rdbms/hm1/hm1/trace/hm1_dia0_i1111.trc
(incident=1111):
ORA-32701: Possible hangs up to hang ID=1 detected
Incident details in: .../oracle/log/diag/rdbms/hm1/hm1/incident/incdir_1111/
```

```
hml_dia0_11111_i1111.trc
2016-07-16T12:39:02.674061-07:00
DIA0 requesting termination of session sid:44 with serial # 23456
(ospid:34569) on instance 3
    due to a GLOBAL, HIGH confidence hang with ID=1.
    Hang Resolution Reason: Although hangs of this root type are typically
    self-resolving, the previously ignored hang was automatically resolved.
DIA0: Examine the alert log on instance 3 for session termination status of
hang with ID=1.
```

Example 5-3 Error Message in the Alert Log Showing a Session Delay Resolved by Blocker Resolver

This example shows an example of a Blocker Resolver alert log on the local instance for resolved delays

```
2016-07-16T12:39:02.707822-07:00
Errors in file .../oracle/log/diag/rdbms/hml/hml1/trace/hml1_dia0_11111.trc
(incident=169):
ORA-32701: Possible hangs up to hang ID=1 detected
Incident details in: .../oracle/log/diag/rdbms/hml/hml1/incident/incdir_169/
hml1_dia0_30676_i169.trc
2016-07-16T12:39:05.086593-07:00
DIA0 terminating blocker (ospid: 30872 sid: 44 ser#: 23456) of hang with ID =
1
    requested by master DIA0 process on instance 1
    Hang Resolution Reason: Although hangs of this root type are typically
    self-resolving, the previously ignored hang was automatically resolved.
    by terminating session sid:44 with serial # 23456 (ospid:34569)
...
DIA0 successfully terminated session sid:44 with serial # 23456 (ospid:34569)
with status 0.
```

Part IV

Appendixes

- [OCLUMON Command Reference](#)
Use the command-line tool to query the Cluster Health Monitor repository to display node-specific metrics for a specific time period.
- [Querying Cluster Resource Activity Log](#)
Oracle Clusterware stores logs about resource state changes in the cluster resource activity log.
- [chactl Command Reference](#)
The Oracle Cluster Health Advisor commands enable the Oracle Grid Infrastructure user to administer basic monitoring functionality on the targets.
- [Behavior Changes, Deprecated and Desupported Features](#)
Review information about changes, deprecations, and desupports.

A

OCLUMON Command Reference

Use the command-line tool to query the Cluster Health Monitor repository to display node-specific metrics for a specific time period.

Use OCLUMON to perform miscellaneous administrative tasks, such as changing the debug levels, querying the version of Cluster Health Monitor, and changing the metrics database size.

- [oclumon analyze](#)
Use the `oclumon analyze` command to analyze CHM metrics.
- [oclumon dumpnodeview](#)
Use the `oclumon dumpnodeview` command to view log information from the system monitor service in the form of a node view.
- [oclumon chmdiag](#)
Use the `oclumon chmdiag` to get a detailed description of all the supported events and actions, query CHMDiag events/actions sent by various components and generate an HTML or a text report, and to collect all events/actions data generated by CHMDiag into the specified output directory location.
- [oclumon localrepo](#)
Use the `oclumon localrepo` command to view and change the configuration of local repository.
- [oclumon version](#)
Use the `oclumon version` command to obtain the version of Cluster Health Monitor that you are using.
- [oclumon debug](#)
Use the `oclumon debug` command to set the log level for the Cluster Health Monitor services.

A.1 ocumon analyze

Use the `oclumon analyze` command to analyze CHM metrics.

Syntax

```
oclumon analyze [-h] [-i CHM_METRICS_DIR] -o OUT_DIR [-l LOG_DIR] [--log_level {DEBUG,INFO,WARNING,ERROR}] [-s START_TIME] [-e END_TIME] [-f FORMAT] [--version]
```

Parameters

Table A-1 ocumon analyze Command Parameters

Parameter	Description
<code>-i CHM_METRICS_DIR</code> <code>--chm_metrics_dir CHM_METRICS_DIR</code>	Specify the directory containing CHM metrics.

Table A-1 (Cont.) ocumon analyze Command Parameters

Parameter	Description
-o <i>OUT_DIR</i> --out_dir <i>OUT_DIR</i>	Specify the output directory for the results.
-l <i>LOG_DIR</i> --log_dir <i>LOG_DIR</i>	Specify the log directory.
--log_level {DEBUG,INFO,WARNING,ERROR}	Specify the log level.
-s <i>START_TIME</i> --start_time <i>START_TIME</i>	Specify the start time for analysis in YYYY-MM-DDTHH:MM:SS format.
-e <i>END_TIME</i> --end_time <i>END_TIME</i>	Specify the end time for analysis in YYYY-MM-DDTHH:MM:SS format.
-f <i>FORMAT</i> --format <i>FORMAT</i>	Specify a comma-delimited report format (text,html). Defaults to text format if not specified. Can either text or html or both
--version	Displays the program's version number and exits.

Example A-1 ocumon analyze Examples

To generate text analysis report for the entire CHM repository:

```
ocumon analyze -o /<output-dir>
```

To generate text analysis report from 2024-03-14T05:00:00 to 2024-03-14T05:15:00 duration:

```
ocumon analyze -o /<output-dir> -s 2024-03-14T05:00:00 -e 2024-03-14T05:15:00
```

To generate an HTML analysis report for the entire CHM repository:

```
ocumon analyze -o /<output-dir> -f html
```

To generate the analysis report from an archived CHM dataset:

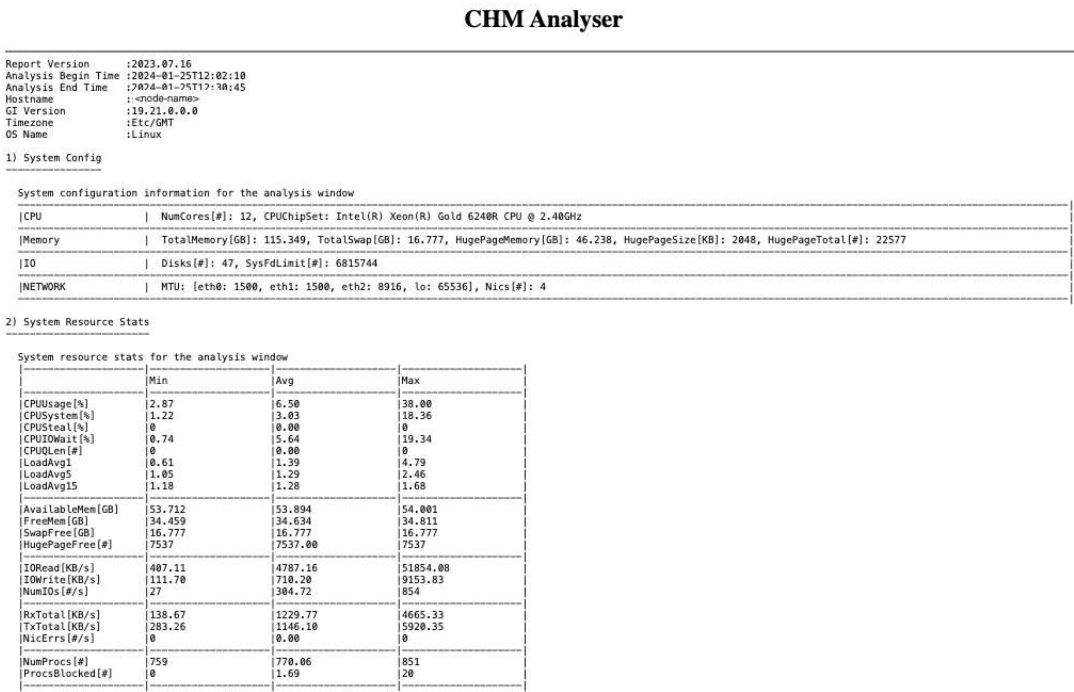
```
ocumon analyze -i /<chm-data-dir> -o /<output-dir>
```

Example A-2 Sample CHM Analysis Report

CHM analysis report contains following sections:

- **Header section:** Contains info about the node, analysis time period, system configuration and system resource stats.

Figure A-1 System Configuration and System resource stats



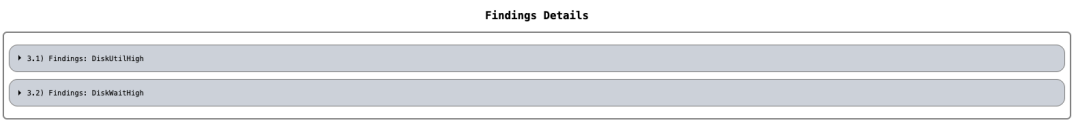
- Observed findings and findings summary timeline section: Contains the list of observed problems, along with a summary timeline of the problems.

Figure A-2 Problematic findings and summary timeline



- Findings details section: Contains detailed contextual information for each of the problems observed above.

Figure A-3 Problematic findings - details



A.2 ocumon dumpnodeview

Use the `ocumon dumpnodeview` command to view log information from the system monitor service in the form of a node view.

Syntax

```
ocumon dumpnodeview [[([(-system | -protocols | -v)] |
    [(-cpu | -process | -procagg | -device | -nic | -filesystem | -thread | -
buddyinfo | -buddyinfoDiffOnly | -nfs)
    [-detail] [-all] [-sort <metric_name>] [-filter <string>] [-head
<rows_count>] [-i <seconds>]])
    [([(-s <start_time> -e <end_time>] | -last <duration>))]] |
    [-inputDataDir <absolute_path> -logDir <absolute_path>]
    [-completeOnly] [-h]]
```

Parameters

Table A-2 ocumon dumpnodeview Command Parameters

Parameter	Description
-system	Dumps system metrics. For example: ocumon dumpnodeview -system
-cpu	Dumps CPU metrics. For example: ocumon dumpnodeview -cpu
-process	Dumps process metrics. For example: ocumon dumpnodeview -process
-procagg	Dumps process aggregate metrics. For example: ocumon dumpnodeview -procagg
-device	Dumps disk metrics. For example: ocumon dumpnodeview -device

Table A-2 (Cont.) ocumon dumpnodeview Command Parameters

Parameter	Description
-nic	Dumps network interface metrics. For example: ocumon dumpnodeview -nic
-filesystem	Dumps filesystem metrics. For example: ocumon dumpnodeview -filesystem
-thread	Dumps thread metrics for pinned processes. For example: ocumon dumpnodeview -thread
-buddyinfo	Dumps Buddyinfo metrics. For example: ocumon dumpnodeview -buddyinfo
-buddyinfoDiffOnly	Dumps Buddyinfo metrics. Use only with -s and -e or -last options. For example: ocumon dumpnodeview -buddyinfoDiffOnly -s "2019-07-10 03:40:25" -e "2019-07-10 03:45:25"
-nfs	Dumps NFS metrics. For example: ocumon dumpnodeview -nfs
-protocols	Dumps network protocol metrics, cumulative values from system start. For example: ocumon dumpnodeview -protocols
-v	Displays verbose node view output. For example: ocumon dumpnodeview -v
-h, --help	Displays the command-line help and exits.

Table A-3 oclumon dumpnodeview Command Flags

Flag	Description
-detail	<p>Use this option to dump detailed metrics. Applicable to the -process and -nic options. For example:</p> <pre>oclumon dumpnodeview -process -detail</pre> <p>.</p>
-all	<p>Use this option to dump the node views of all entries. Applicable to the -process option. For example:</p> <pre>oclumon dumpnodeview -process -all</pre> <p>.</p>
-pinned_only	<p>Use this option to dump the node views of all pinned processes. Applicable to the -process option. For example:</p> <pre>oclumon dumpnodeview -process -pinned_only</pre>
-head <i>rows_count</i>	<p>Use this option to dump the node view of the specified number of metrics rows in the result. Applicable to the -process option. Default is set to 5. For example:</p> <pre>oclumon dumpnodeview -process -head 7</pre> <p>.</p>
-sort <i>metric_name</i>	<p>Use this option to sort based on the specified metric name, supported with the -process, -device, -nic, -cpu, -procagg, -filesystem, -nfs options. For example:</p> <pre>oclumon dumpnodeview -device -sort "ioR"</pre> <p>.</p>

Table A-3 (Cont.) oclumon dumpnodeview Command Flags

Flag	Description
<code>-i seconds</code>	<p>Display data separated by the specified interval in seconds. Must be a multiple of 5. Applicable to continuous mode query.</p> <p>For example:</p> <pre>oclumon dumpnodeview -device -i 5</pre>
<code>-filter string</code>	<p>Use this option to search for a filter string in the Name column of the respective metric.</p> <p>For example, <code>-process -filter "ora"</code> will display the process metrics, which contain "ora" substring in their name.</p> <p>Supported with the <code>-process</code>, <code>-device</code>, <code>-nic</code>, <code>-cpu</code>, <code>-procagg</code>, <code>-thread</code>, and <code>-filesystem</code> options.</p> <p>Supported comparison operators are: <code>=</code>, <code><=</code>, <code>>=</code>, <code><</code>, and <code>></code></p> <p>For example:</p> <pre>oclumon dumpnodeview -process - filter "ora"</pre> <p>.</p>
<code>-show_all_sample_with_filter</code>	<p>All samples where filter doesn't matches will also show in the output. Can be used only with the <code>-filter</code> option.</p> <p>For example:</p> <pre>oclumon dumpnodeview -filter filter_criteria - show_all_sample_with_filter</pre>
<code>-completeOnly</code>	<p>Shows only the complete samples every 30 seconds.</p> <p>For example:</p> <pre>oclumon dumpnodeview -completeOnly</pre>

Table A-4 oclumon dumpnodeview Command Log File Directories

Directory	Description
<code>-inputDataDir <i>absolute_dir_path</i></code>	Specifies absolute path of the directory that contains JSON logs files. For example: <pre>oclumon dumpnodeview -cpu - inputDataDir <i>absolute_path</i></pre>
<code>-logDir <i>absolute_log_dir_path</i></code>	Specifies absolute path of the directory, which will contain the script run logs. For example: <pre>oclumon dumpnodeview -cpu - inputDataDir <i>absolute_path</i> -logDir <i>absolute_log_dir_path</i></pre>

Table A-5 oclumon dumpnodeview Command Historical Query Options

Flag	Description
<code>-s <i>start_time</i></code> <code>-e <i>end_time</i></code>	Use the <code>-s</code> option to specify a time stamp from which to start a range of queries and use the <code>-e</code> option to specify a time stamp to end the range of queries. Specify time in the YYYY-MM-DD HH24:MM:SS format surrounded by double quotation marks (" "). Specify these two options together to obtain a range. For example: <pre>oclumon dumpnodeview -cpu -s "2019-07-10 03:40:25" -e "2019-07-10 03:45:25"</pre>
<code>-last <i>duration</i></code>	Use this option to specify a time, given in HH24:MM:SS format surrounded by double quotation marks (""), to retrieve the last metrics. specifying "2d1h20m20s" will dump metrics for last 2 days 1 hour 20 minutes and 20 seconds For example: <pre>oclumon dumpnodeview -nic -last "2d1h20m20s"</pre>

A.3 ocumon chmdiag

Use the `ocumon chmdiag` to get a detailed description of all the supported events and actions, query CHMDiag events/actions sent by various components and generate an HTML or a text report, and to collect all events/actions data generated by CHMDiag into the specified output directory location.

A.4 ocumon localrepo

Use the `ocumon localrepo` command to view and change the configuration of local repository.

- [ocumon localrepo getconfig](#)
Use the `ocumon localrepo getconfig` to get the configuration of repositories for all the nodes.
- [ocumon localrepo setconfig](#)
Use the `ocumon localrepo setconfig` command to change the configuration of local repository.

A.4.1 ocumon localrepo getconfig

Use the `ocumon localrepo getconfig` to get the configuration of repositories for all the nodes.

Syntax

```
ocumon localrepo getconfig [-reposize] [-repopath] [-retentiontime] [-local  
| -n <node1> ...]
```

Parameters

Parameter	Description
-reposize	Gets the repository size in MB.
-repopath	Gets the repository path.
-retentiontime	Gets an estimation of local repository retention in time units based on the historical data of the currently configured repository size.
-local	Gets the configuration only for the local node.
-n	Gets the configuration for a desired list of nodes.

Example A-3 To view full configuration of repositories for all nodes

```
ocumon localrepo getconfig
Node: <node-name1>
Repository size: 500 MB
Repository path: $ORACLE_HOME/crsdata/<node-name1>/crf/db/json
Repository retention time: 246 Hours

Node: <node-name2>
Repository size: 500 MB
```

```
Repository path: $ORACLE_HOME/crsdata/<node-name2>/crf/db/json  
Repository retention time: 240 Hours
```

Example A-4 To view only the repository path and size of repositories in all nodes

```
oclumon localrepo getconfig -reposize -repopath  
Node: <node-name1>  
Repository size: 500 MB  
Repository path: $ORACLE_HOME/crsdata/<node-name1>/crf/db/json  
  
Node: <node-name2>  
Repository size: 500 MB  
Repository path: $ORACLE_HOME/crsdata/<node-name2>/crf/db/json
```

Example A-5 To view full configuration of the repository for the local node

```
oclumon localrepo getconfig -local  
Node: <node-name>  
Repository size: 500 MB  
Repository path: $ORACLE_HOME/crsdata/<node-name>/crf/db/json  
Repository retention time: 246 Hours
```

Example A-6 To view full configuration for the repositories on specific nodes <node-name1> and <node-name2>

```
oclumon localrepo getconfig -n <node-name1> <node-name2>  
Node: <node-name1>  
Repository size: 500 MB  
Repository path: $ORACLE_HOME/crsdata/<node-name1>/crf/db/json  
Repository retention time: 246 Hours  
  
Node: <node-name2>  
Repository size: 500 MB  
Repository path: $ORACLE_HOME/crsdata/<node-name2>/crf/db/json  
Repository retention time: 240 Hours
```

A.4.2 ocumon localrepo setconfig

Use the `ocumon localrepo setconfig` command to change the configuration of local repository.

Syntax

```
ocumon localrepo setconfig -reposize <size>
```

Where, `setconfig` sets the repository size in MB for all the nodes.

Example A-7 To set the repository size for all the nodes

```
ocumon localrepo setconfig -reposize 200  
json dump log event is sent successfully on <node-name1>  
json dump log event is sent successfully on <node-name2>
```

A.5 oclumon version

Use the `oclumon version` command to obtain the version of Cluster Health Monitor that you are using.

Syntax

```
oclumon version
```

Example A-8 oclumon version

This command produces output similar to the following:

```
Cluster Health Monitor (OS), Release 20.0.0.0.0
Version : 20.3.0.0.0
```

A.6 oclumon debug

Use the `oclumon debug` command to set the log level for the Cluster Health Monitor services.

Syntax

```
oclumon debug [log daemon module:log_level] [version]
```

Parameters

Table A-6 oclumon debug Command Parameters

Parameter	Description
<code>log daemon module:log_level</code>	<p>Use this option change the log level of daemons and daemon modules.</p> <p>Supported daemons are:</p> <ul style="list-style-type: none"><code>osysmond</code><code>client</code><code>all</code> <p>Supported daemon modules are:</p> <ul style="list-style-type: none"><code>osysmond</code>: CRFMOND, CRFM, and allcomp<code>client</code>: OCLUMON, CRFM, and allcomp<code>all</code>: allcomp <p>Supported <code>log_level</code> values are 0, 1, 2, and 3.</p> <p>Where level 0 is lowest default level with minimal logging and level 3 is highest level with maximum logging.</p>
<code>version</code>	Use this option to display the versions of the daemons.

Example A-9 oclumon debug

The following example sets the log level of the system monitor service (osysmond):

```
$ oclumon debug log osysmond CRFMOND:3
```

The following example displays the versions of the daemons:

```
$ oclumon debug version

Cluster Health Monitor (OS), Release 20.0.0.0.0
Version           : 20.3.0.0.0
NODEVIEW Version  : 19.03
Label Date        : 200116
```

B

Querying Cluster Resource Activity Log

Oracle Clusterware stores logs about resource state changes in the cluster resource activity log.

Failures can occur as a result of a problem with a resource, a hosting node, or the network.

The cluster resource activity log provides precise and specific information about a resource failure, separate from diagnostic logs. The cluster resource activity log also provides a unified view of the cause of resource failure.

Use the following commands to view the contents of the cluster resource activity log:

- [crsctl query calog](#)
Query the cluster resource activity logs matching specific criteria.

B.1 crsctl query calog

Query the cluster resource activity logs matching specific criteria.

Syntax

```
crsctl query calog  
[-aftertime "timestamp"]  
[-beforetime "timestamp"]  
[-days "number_of_days"]  
[-duration "time_interval" | -follow]  
[-filter "filter_expression"]  
[-processname "writer_process"]  
[-processid "writer_process_id"]  
[-node "entity_hostname"]  
[-fullfmt | -xmlfmt]
```

Parameters

Table B-1 crsctl query calog Command Parameters

Parameter	Description
-aftertime " <i>timestamp</i> "	<p>Displays the activities logged after a specific time.</p> <p>Specify the timestamp in the YYYY-MM-DD HH24:MI:SS[.FF] [TZH:TZM] or YYYY-MM-DD or YYYY-MM or YYYY or HH24:MI:SS[.FF] [TZH:TZM] format.</p> <p>TZH and TZM stands for time zone hour and minute, and FF stands for microseconds.</p> <p>If you specify [TZH:TZM], then the crsctl command assumes UTC as time zone. If you do not specify [TZH:TZM], then the crsctl command assumes the local time zone of the cluster node from where the crsctl command is run.</p> <p>Use this parameter with -beforetime to query the activities logged at a specific time interval.</p>
-beforetime " <i>timestamp</i> "	<p>Displays the activities logged before a specific time.</p> <p>Specify the timestamp in the YYYY-MM-DD HH24:MI:SS[.FF] [TZH:TZM] or YYYY-MM-DD or YYYY-MM or YYYY or HH24:MI:SS[.FF] [TZH:TZM] format.</p> <p>TZH and TZM stands for time zone hour and minute, and FF stands for microseconds.</p> <p>If you specify [TZH:TZM], then the crsctl command assumes UTC as time zone. If you do not specify [TZH:TZM], then the crsctl command assumes the local time zone of the cluster node from where the crsctl command is run.</p> <p>Use this parameter with -aftertime to query the activities logged at a specific time interval.</p>
-days " <i>number_of_days</i> "	<p>Displays the activities logged in the last number of days specified. The number of days are specified as an integer value.</p>
-duration " <i>time_interval</i> " - follow	<p>Use -duration to specify a time interval that you want to query when you use the -aftertime parameter.</p> <p>Specify the timestamp in the DD HH:MM:SS format.</p> <p>Use -follow to display a continuous stream of activities as they occur.</p>
-filter " <i>filter_expression</i> "	<p>Query any number of fields in the cluster resource activity log using the -filter parameter.</p> <p>To specify multiple filters, use a comma-delimited list of filter expressions surrounded by double quotation marks (" ").</p>
-processname " <i>writer_process</i> "	<p>Displays the activities logged by a specific process identified by name.</p>
-processid " <i>writer_process_id</i> "	<p>Displays the activities logged by a specific process identified by ID.</p>
-node " <i>entity_hostname</i> "	<p>Displays the activities logged by a specific host.</p>
-fullfmt -xmlfmt	<p>To display cluster resource activity log data, choose full or XML format.</p>

Cluster Resource Activity Log Fields

Query any number of fields in the cluster resource activity log using the -filter parameter.

Table B-2 Cluster Resource Activity Log Fields

Field	Description	Use Case
timestamp	The time when the cluster resource activities were logged.	Use this filter to query all the activities logged at a specific time. This is an alternative to - aftertime, -beforetime, and - duration command parameters.
writer_process_id	The ID of the process that is writing to the cluster resource activity log.	Query only the activities spawned by a specific process.
writer_process_name	The name of the process that is writing to the cluster resource activity log.	When you query a specific process, CRSCTL returns all the activities for a specific process.
writer_user	The name of the user who is writing to the cluster resource activity log.	Query all the activities written by a specific user.
writer_group	The name of the group to which a user belongs who is writing to the cluster resource activity log.	Query all the activities written by users belonging to a specific user group.
writer_hostname	The name of the host on which the cluster resource activity log is written.	Query all the activities written by a specific host.
writer_clustername	The name of the cluster on which the cluster resource activity log is written.	Query all the activities written by a specific cluster.
nls_product	The product of the NLS message, for example, CRS, ORA, or srvn.	Query all the activities that have a specific product name.
nls_facility	The facility of the NLS message, for example, CRS or PROC.	Query all the activities that have a specific facility name.
nls_id	The ID of the NLS message, for example 42008.	Query all the activities that have a specific message ID.
nls_field_count	The number of fields in the NLS message.	Query all the activities that correspond to NLS messages with more than, less than, or equal to nls_field_count command parameters.
nls_field1	The first field of the NLS message.	Query all the activities that match the first parameter of an NLS message.
nls_field1_type	The type of the first field in the NLS message.	Query all the activities that match a specific type of the first parameter of an NLS message.
nls_format	The format of the NLS message, for example, Resource '%s' has been modified.	Query all the activities that match a specific format of an NLS message.
nls_message	The entire NLS message that was written to the cluster resource activity log, for example, Resource 'ora.cvu' has been modified.	Query all the activities that match a specific NLS message.

Table B-2 (Cont.) Cluster Resource Activity Log Fields

Field	Description	Use Case
actid	The unique activity ID of every cluster activity log.	Query all the activities that match a specific ID. Also, specify only partial actid and list all activities where the actid is a subset of the activity ID.
is_planned	Confirms if the activity is planned or not. For example, if a user issues the command <code>crsctl stop crs</code> on a node, then the stack stops and resources bounce. Running the <code>crsctl stop crs</code> command generates activities and logged in the calog. Since this is a planned action, the <code>is_planned</code> field is set to true (1). Otherwise, the <code>is_planned</code> field is set to false (0).	Query all the planned or unplanned activities.
onbehalfof_user	The name of the user on behalf of whom the cluster activity log is written.	Query all the activities written on behalf of a specific user.
entity_isoraentity	Confirms if the entity for which the calog activities are being logged is an oracle entity or not. If a resource, such as <code>ora.***</code> , is started or stopped, for example, then all those activities are logged in the cluster resource activity log. Since <code>ora.***</code> is an Oracle entity, the <code>entity_isoraentity</code> field is set to true (1). Otherwise the <code>entity_isoraentity</code> field is set to false (0).	Query all the activities logged by Oracle or non-Oracle entities.

Table B-2 (Cont.) Cluster Resource Activity Log Fields

Field	Description	Use Case
entity_type	<p>The type of the entity, such as <i>server</i>, for which the cluster activity log is written.</p> <p>Entity types that can be used to filter activities</p> <ul style="list-style-type: none"> • <code>resource</code> • <code>resource_type</code> • <code>resource_group</code> • <code>server_category</code> • <code>ohasd</code> - activities generated by <code>ohasd</code> and resources it manages • <code>crsd</code> - activities generated by <code>crsd</code> and resources it manages <p>In addition, GI components can choose to use their own names for entities when they write to activity log.</p>	Query all the activities that match a specific entity.
entity_name	The name of the entity, for example, <i>foo</i> for which the cluster activity log is written.	Query all the cluster activities that match a specific entity name.
entity_hostname	The name of the host, for example, <i>node1</i> , associated with the entity for which the cluster activity log is written.	Query all the cluster activities that match a specific host name.
entity_clustername	The name of the cluster, for example, <i>cluster1</i> associated with the entity for which the cluster activity log is written.	Query all the cluster activities that match a specific cluster name.

Usage Notes

- Combine simple filters into expressions called expression filters using Boolean operators.
- Enclose timestamps and time intervals in double quotation marks ("").
- Enclose the filter expressions in double quotation marks ("").
- Enclose the values that contain parentheses or spaces in single quotation marks (').
- If no matching records are found, then the Oracle Clusterware Control (CRSCTL) utility displays the following message:

```
CRS-40002: No activities match the query.
```

Examples

Examples of filters include:

- `"writer_user==root"`: Limits the display to only root user.
- `"customer_data=='GEN_RESTART@SERVERNAME(rwsbi08)=StartCompleted~'"`: Limits the display to `customer_data` that has the specified value `GEN_RESTART@SERVERNAME(node1)=StartCompleted~`.

To query all the resource activities and display the output in full format:

```
$ crsctl query calog -fullfmt

----ACTIVITY START----
timestamp           : 2016-09-27 17:55:43.152000
writer_process_id    : 6538
writer_process_name  : crsd.bin
writer_user          : root
writer_group         : root
writer_hostname      : nodel
writer_clustername   : cluster1-mbl
customer_data        : CHECK_RESULTS=-408040060~
nls_product          : CRS
nls_facility         : CRS
nls_id               : 2938
nls_field_count      : 1
nls_field1           : ora.cvu
nls_field1_type      : 25
nls_field1_len       : 0
nls_format           : Resource '%s' has been modified.
nls_message          : Resource 'ora.cvu' has been modified.
actid                : 14732093665106538/1816699/1
is_planned           : 1
onbehalfof_user      : grid
onbehalfof_hostname  : nodel
entity_isoraentity   : 1
entity_type          : resource
entity_name          : ora.cvu
entity_hostname      : nodel
entity_clustername   : cluster1-mbl
nls_severity         : INFO
----ACTIVITY END----
```

To query all the resource activities and display the output in XML format:

```
$ crsctl query calog -xmlfmt

<?xml version="1.0" encoding="UTF-8"?>
<activities>
  <activity>
    <timestamp>2016-09-27 17:55:43.152000</timestamp>
    <writer_process_id>6538</writer_process_id>
    <writer_process_name>crsd.bin</writer_process_name>
    <writer_user>root</writer_user>
    <writer_group>root</writer_group>
    <writer_hostname>nodel</writer_hostname>
    <writer_clustername>cluster1-mbl</writer_clustername>
    <customer_data>CHECK_RESULTS=-408040060~</customer_data>
    <nls_product>CRS</nls_product>
    <nls_facility>CRS</nls_facility>
    <nls_id>2938</nls_id>
    <nls_field_count>1</nls_field_count>
    <nls_field1>ora.cvu</nls_field1>
    <nls_field1_type>25</nls_field1_type>
```

```

<nls_field1_len>0</nls_field1_len>
<nls_format>Resource '%s' has been modified.</nls_format>
<nls_message>Resource 'ora.cvu' has been modified.</nls_message>
<actid>14732093665106538/1816699/1</actid>
<is_planned>1</is_planned>
<onbehalfof_user>grid</onbehalfof_user>
<onbehalfof_hostname>node1</onbehalfof_hostname>
<entity_isoraentity>1</entity_isoraentity>
<entity_type>resource</entity_type>
<entity_name>ora.cvu</entity_name>
<entity_hostname>node1</entity_hostname>
<entity_clustername>cluster1-mbl</entity_clustername>
<nls_severity>INFO</nls_severity>
</activity>
</activities>

```

To query resource activities for a two-hour interval after a specific time and display the output in XML format:

```

$ crsctl query calog -aftertime "2016-09-28 17:55:43" -duration "0 02:00:00" -
xmlfmt
<?xml version="1.0" encoding="UTF-8"?>
<activities>
  <activity>
    <timestamp>2016-09-28 17:55:45.992000</timestamp>
    <writer_process_id>6538</writer_process_id>
    <writer_process_name>crsd.bin</writer_process_name>
    <writer_user>root</writer_user>
    <writer_group>root</writer_group>
    <writer_hostname>node1</writer_hostname>
    <writer_clustername>cluster1-mbl</writer_clustername>
    <customer_data>CHECK_RESULTS=1718139884~</customer_data>
    <nls_product>CRS</nls_product>
    <nls_facility>CRS</nls_facility>
    <nls_id>2938</nls_id>
    <nls_field_count>1</nls_field_count>
    <nls_field1>ora.cvu</nls_field1>
    <nls_field1_type>25</nls_field1_type>
    <nls_field1_len>0</nls_field1_len>
    <nls_format>Resource '%s' has been modified.</nls_format>
    <nls_message>Resource 'ora.cvu' has been modified.</nls_message>
    <actid>14732093665106538/1942009/1</actid>
    <is_planned>1</is_planned>
    <onbehalfof_user>grid</onbehalfof_user>
    <onbehalfof_hostname>node1</onbehalfof_hostname>
    <entity_isoraentity>1</entity_isoraentity>
    <entity_type>resource</entity_type>
    <entity_name>ora.cvu</entity_name>
    <entity_hostname>node1</entity_hostname>
    <entity_clustername>cluster1-mbl</entity_clustername>
    <nls_severity>INFO</nls_severity>
  </activity>
</activities>

```

To query resource activities at a specific time:

```
$ crsctl query calog -filter "timestamp=='2016-09-28 17:55:45.992000'"
```

```
2016-09-28 17:55:45.992000 : node1 : INFO : Resource 'ora.cvu' has been
modified. : 14732093665106538/1942009/1 :
```

To query resource activities using filters `writer_user` and `customer_data`:

```
$ crsctl query calog -filter "writer_user==root AND
customer_data=='GEN_RESTART@SERVERNAME(node1)=StartCompleted~'" -fullfmt
```

or

```
$ crsctl query calog -filter "(writer_user==root) AND
(customer_data=='GEN_RESTART@SERVERNAME(node1)=StartCompleted~')" -fullfmt
```

```
----ACTIVITY START----
timestamp           : 2016-09-15 17:42:57.517000
writer_process_id   : 6538
writer_process_name : crsd.bin
writer_user         : root
writer_group        : root
writer_hostname     : node1
writer_clustername  : cluster1-mb1
customer_data       : GEN_RESTART@SERVERNAME(rwsbi08)=StartCompleted~
nls_product         : CRS
nls_facility        : CRS
nls_id              : 2938
nls_field_count     : 1
nls_field1          : ora.testdb.db
nls_field1_type     : 25
nls_field1_len      : 0
nls_format          : Resource '%s' has been modified.
nls_message         : Resource 'ora.devdb.db' has been modified.
actid               : 14732093665106538/659678/1
is_planned          : 1
onbehalfof_user     : oracle
onbehalfof_hostname : node1
entity_isoraentity  : 1
entity_type         : resource
entity_name         : ora.testdb.db
entity_hostname     : node1
entity_clustername  : cluster1-mb1
nls_severity        : INFO
----ACTIVITY END----
```

To query all the calogs that were generated after UTC+08:00 time "2016-11-15 22:53:08":

```
$ crsctl query calog -aftertime "2016-11-15 22:53:08+08:00"
```

To query all the calogs that were generated after UTC-08:00 time "2016-11-15 22:53:08":

```
$ crsctl query calog -aftertime "2016-11-15 22:53:08-08:00"
```

To query all the calogs by specifying the timestamp with microseconds:

```
$ crsctl query calog -aftertime "2016-11-16 01:07:53.063000"
2016-11-16 01:07:53.558000 : node1 : INFO : Resource 'ora.cvu' has been
modified. : 14792791129816600/2580/7 :
2016-11-16 01:07:53.562000 : node2 : INFO : Clean of 'ora.cvu' on 'node2'
succeeded : 14792791129816600/2580/8 :
```

To query all the activities that were written by a specific process by name:

```
$ crsctl query calog -processname crsd.bin

2016-11-16 01:07:53.558000 : node1 : INFO : Resource 'ora.cvu' has been
modified. : 14792791129816600/2580/7 :
2016-11-16 01:07:53.562000 : node2 : INFO : Clean of 'ora.cvu' on 'node2'
succeeded : 14792791129816600/2580/8 :
```

To query all the activities that were written by a specific process by ID:

```
$ crsctl query calog -processid 6538

2016-11-16 01:07:53.558000 : node1 : INFO : Resource 'ora.cvu' has been
modified. : 14792791129816600/2580/7 :
2016-11-16 01:07:53.562000 : node2 : INFO : Clean of 'ora.cvu' on 'node2'
succeeded : 14792791129816600/2580/8 :
```

To query all the activities that were written by a specific node:

```
$ crsctl query calog -node node2
2016-11-16 01:07:53.562000 : node2 : INFO : Clean of 'ora.cvu' on 'node2'
succeeded : 14792791129816600/2580/8 :
```

C

chactl Command Reference

The Oracle Cluster Health Advisor commands enable the Oracle Grid Infrastructure user to administer basic monitoring functionality on the targets.

- [chactl monitor](#)
Use the `chactl monitor` command to start monitoring all the instances of a specific Oracle Real Application Clusters (Oracle RAC) database using the current set model.
- [chactl unmonitor](#)
Use the `chactl unmonitor` command to stop monitoring all the instances of a specific database.
- [chactl status](#)
Use the `chactl status` command to check monitoring status of the running targets.
- [chactl config](#)
Use the `chactl config` command to list all the targets being monitored, along with the current model of each target.
- [chactl calibrate](#)
Use the `chactl calibrate` command to create a new model that has greater sensitivity and accuracy.
- [chactl query diagnosis](#)
Use the `chactl query diagnosis` command to return problems and diagnosis, and suggested corrective actions associated with the problem for specific cluster nodes or Oracle Real Application Clusters (Oracle RAC) databases.
- [chactl query model](#)
Use the `chactl query model` command to list all Oracle Cluster Health Advisor models or to view detailed information about a specific Oracle Cluster Health Advisor model.
- [chactl query repository](#)
Use the `chactl query repository` command to view the maximum retention time, number of targets, and the size of the Oracle Cluster Health Advisor repository.
- [chactl query calibration](#)
Use the `chactl query calibration` command to view detailed information about the calibration data of a specific target.
- [chactl remove model](#)
Use the `chactl remove model` command to delete an Oracle Cluster Health Advisor model along with the calibration data and metadata of the model from the Oracle Cluster Health Advisor repository.
- [chactl rename model](#)
Use the `chactl rename model` command to rename an Oracle Cluster Health Advisor model in the Oracle Cluster Health Advisor repository.
- [chactl export model](#)
Use the `chactl export model` command to export Oracle Cluster Health Advisor models.
- [chactl import model](#)
Use the `chactl import model` command to import Oracle Cluster Health Advisor models.

- [chactl set maxretention](#)
Use the `chactl set maxretention` command to set the maximum retention time for the diagnostic data.
- [chactl resize repository](#)
Use the `chactl resize repository` command to resize the tablespace of the Oracle Cluster Health Advisor repository based on the current retention time and the number of targets.

C.1 chactl monitor

Use the `chactl monitor` command to start monitoring all the instances of a specific Oracle Real Application Clusters (Oracle RAC) database using the current set model.

Oracle Cluster Health Advisor monitors all instances of this database using the same model assigned to the database.

Oracle Cluster Health Advisor uses Oracle-supplied gold model when you start monitoring a target for the first time. Oracle Cluster Health Advisor stores monitoring status of the target in the internal store. Oracle Cluster Health Advisor starts monitoring any new database instance when Oracle Cluster Health Advisor detects or redetects the new instance.

Syntax

```
chactl monitor database -db db_unique_name [-model model_name [-force]][-help]
```

```
chactl monitor cluster [-model model_name [-force]]
```

Parameters

Table C-1 chactl monitor Command Parameters

Parameter	Description
db_unique_name	Specify the name of the database.
model_name	Specify the name of the model.
force	Use the <code>-force</code> option to monitor with the specified model without stopping monitoring the target. Without the <code>-force</code> option, run <code>chactl unmonitor</code> first, and then <code>chactl monitor</code> with the model name.

Examples

- To monitor the *SalesDB* database using the *BlkFridayShopping* default model:

```
$ chactl monitor database -db SalesDB -model BlkFridayShopping
```

- To monitor the *InventoryDB* database using the *Nov2014* model:

```
$ chactl monitor database -db InventoryDB -model Nov2014
```

If you specify the *model_name*, then Oracle Cluster Health Advisor starts monitoring with the specified model and stores the model in the Oracle Cluster Health Advisor internal store.

If you use both the *-model* and *-force* options, then Oracle Cluster Health Advisor stops monitoring and restarts monitoring with the specified model.

- To monitor the *SalesDB* database using the *Dec2014* model:

```
$ chactl monitor database -db SalesDB -model Dec2014
```

- To monitor the *InventoryDB* database using the *Dec2014* model and the *-force* option:

```
$ chactl monitor database -db InventoryDB -model Dec2014 -force
```

Error Messages

Error: no CHA resource is running in the cluster.

Description: Returns when there is no hub or leaf node running the Oracle Cluster Health Advisor service.

Error: the database is not configured.

Description: Returns when the database is not found in either the Oracle Cluster Health Advisor configuration repository or as a CRS resource.

Error: input string "xc#? %" is invalid.

Description: Returns when the command-line cannot be parsed. Also displays the top-level help text.

Error: CHA is already monitoring target <dbname>.

Description: Returns when the database is already monitored.

C.2 chactl unmonitor

Use the `chactl unmonitor` command to stop monitoring all the instances of a specific database.

Syntax

```
chactl unmonitor database -db db_unique_name [-help]
```

Examples

To stop monitoring the *SalesDB* database:

```
$ chactl unmonitor database -db SalesDB
Database SalesDB is not monitored
```


C.3 chactl status

Use the `chactl status` command to check monitoring status of the running targets.

If you do not specify any parameters, then the `chactl status` command returns the status of all running targets.

The monitoring status of an Oracle Cluster Health Advisor target can be either `Monitoring` or `Not Monitoring`. The `chactl status` command shows four types of results and depends on whether you specify a target and `-verbose` option.

The `-verbose` option of the command also displays the monitoring status of targets contained within the specified target and the names of executing models of each printed target. The `chactl status` command displays targets with positive monitoring status only. The `chactl status` command displays negative monitoring status only when the corresponding target is explicitly specified on the command-line.

Syntax

```
chactl status {cluster|database [-db db_unique_name]} [-verbose][-help]
```

Examples

- To display the list of cluster nodes and databases being monitored:

```
#chactl status
Monitoring nodes rac1Node1, rac1Node2
Monitoring databases SalesDB, HRdb
```

Note

A database is displayed with **Monitoring** status, if Oracle Cluster Health Advisor is monitoring one or more of the instances of the database, even if some of the instances of the database are not running.

- To display the status of Oracle Cluster Health Advisor:

```
$ chactl status
Cluster Health Advisor service is offline.
```

No target or the `-verbose` option is specified on the command-line. Oracle Cluster Health Advisor is not running on any node of the cluster.

- To display various Oracle Cluster Health Advisor monitoring states for cluster nodes and databases:

```
$ chactl status database -db SalesDB
Monitoring database SalesDB
```

```
$ chactl status database -db bogusDB
Not Monitoring database bogusDB
```

```
$ chactl status cluster
Monitoring nodes rac1,rac2
Not Monitoring node rac3
```

or

```
$ chactl status cluster
Cluster Health Advisor is offline
```

- To display the detailed Oracle Cluster Health Advisor monitoring status for the entire cluster:

```
$ chactl status -verbose
Monitoring node(s) racNd1, racNd2, racNd3, racNd4 using model MidSparc
```

```
Monitoring database HRdb2, Instances HRdb2I1, HRdb2I2 in server pool
SilverPool using model M6
Monitoring database HRdb, Instances HRdbI4, HRdbI6 in server pool
SilverPool using model M23
Monitoring database testHR, Instances inst3 on node racN7 using model
TestM13
Monitoring database testHR, Instances inst4 on node racN8 using model
TestM14
```

When the target is not specified and the `-verbose` option is specified, the `chactl status` command displays the status of the database instances and names of the models.

C.4 chactl config

Use the `chactl config` command to list all the targets being monitored, along with the current model of each target.

If the specified target is a multitenant container database (CDB) or a cluster, then the `chactl config` command also displays the configuration data status.

Syntax

```
chactl config {cluster|database -db db_unique_name}[-help]
```

Examples

To display the monitor configuration and the specified model of each target:

```
$ chactl config
Databases monitored: prodDB, hrDB

$ chactl config database -db prodDB
Monitor: Enabled
Model: GoldDB

$ chactl config cluster
Monitor: Enabled
Model: DEFAULT_CLUSTER
```

C.5 chactl calibrate

Use the `chactl calibrate` command to create a new model that has greater sensitivity and accuracy.

The user-generated models are effective for Oracle Real Application Clusters (Oracle RAC) monitored systems in your operating environment as the user-generated models use calibration data from the target. Oracle Cluster Health Advisor adds the user-generated model to the list of available models and stores the new model in the Oracle Cluster Health Advisor repository.

If a model with the same name exists, then overwrite the old model with the new one by using the `-force` option.

Key Performance and Workload Indicators

A set of metrics or Key Performance Indicators describe high-level constraints to the training data selected for calibration. This set consists of relevant metrics to describe performance goals and resource utilization bandwidth, for example, response times or CPU utilization.

The Key Performance Indicators are also operating system and database signals which are monitored, estimated, and associated with fault detection logic. Most of these Key Performance Indicators are also either predictors, that is, their state is correlated with the state of other signals, or predicted by other signals. The fact that the Key Performance Indicators correlate with other signals makes them useful as filters for the training or calibration data.

The Key Performance Indicators ranges are used in the `query calibrate` and `calibrate` commands to filter out data points.

The following Key Performance Indicators are supported for database:

- CPUPERCENT - CPU utilization - Percent
- IOREAD - Disk read - Mbyte/sec
- DBTIMEPERCALL - Database time per user call - usec/call
- IOWRITE - Disk write - Mbyte/sec
- IOTHROUGHPUT - Disk throughput - IO/sec

The following Key Performance Indicators are supported for cluster:

- CPUPERCENT - CPU utilization - Percent
- IOREAD - Disk read - Mbyte/sec
- IOWRITE - Disk write - Mbyte/sec
- IOTROUGHPUT - Disk throughput - IO/sec

Syntax

```
chactl calibrate {cluster|database -db db_unique_name} -model model_name
[-force] [-timeranges 'start=time_stamp,end=time_stamp,...']
[-kpiiset 'name=kpi_name min=val max=val,...' ][-help]
```

Specify timestamp in the YYYY-MM-DD HH24:MI:SS format.

Examples

```
chactl calibrate database -db oracle -model weekday
-timeranges 'start=start=2016-09-09 16:00:00,end=2016-09-09 23:00:00'
```

```
chactl calibrate database -db oracle -model weekday
-timeranges 'start=start=2016-09-09 16:00:00,end=2016-09-09 23:00:00'
-kpiiset 'name=CPUPERCENT min=10 max=60'
```

Error Messages

Error: input string "xc#? %" is misconstructured

Description: Confirm if the given model name exists with Warning: *model_name* already exists, please use [-force] message.

Error: *start_time* and/or *end_time* are misconstructured

Description: Input time specifiers are badly constructed.

Error: no sufficient calibration data exists for the specified period, please reselect another period

Description: Evaluator couldn't find enough calibration data.

C.6 chactl query diagnosis

Use the `chactl query diagnosis` command to return problems and diagnosis, and suggested corrective actions associated with the problem for specific cluster nodes or Oracle Real Application Clusters (Oracle RAC) databases.

Syntax

```
chactl query diagnosis [-cluster|-db db_unique_name] [-start time -end time]
[-htmlfile file_name][-help]
```

Specify date and time in the YYYY-MM-DD HH24:MI:SS format.

In the preceding syntax, you must consider the following points:

- If you do not provide any options, then the `chactl query diagnosis` command returns the current state of all monitored nodes and databases. The `chactl query diagnosis` command reports general state of the targets, for example, **ABNORMAL** by showing their diagnostic identifier, for example, `Storage Bandwidth Saturation`. This is a quick way to check for any **ABNORMAL** state in a database or cluster.
- If you provide a time option after the target name, then the `chactl query diagnosis` command returns the state of the specified target restricted to the conditions in the time interval specified. The compressed time series lists the identifiers of the causes for distinct incidents which occurred in the time interval, its start and end time.
- If an incident and cause recur in a specific time interval, then the problem is reported only once. The start time is the start time of the first occurrence of the incident and the end time is the end time of the last occurrence of the incident in the particular time interval.
- If you specify the `-db` option without a database name, then the `chactl query diagnosis` command displays diagnostic information for all databases. However, if a database name is specified, then the `chactl query diagnosis` command displays diagnostic information for all instances of the database that are being monitored.
- If you specify the `-cluster` option without a host name, then the `chactl query diagnosis` command displays diagnostic information for all hosts in that cluster.
- If you do not specify a time interval, then the `chactl query diagnosis` command displays only the current issues for all or the specified targets. The `chactl query diagnosis` command does not display the frequency statistics explicitly. However, you can count the number of normal and abnormal events that occurred in a target in the last 24 hours.
- If no incidents have occurred during the specified time interval, then the `chactl query diagnosis` command returns a text message, for example, `Database/host is operating NORMALLY, or no incidents were found`.
- If the state of a target is **NORMAL**, the command does not report it. The `chactl query diagnosis` command reports only the targets with **ABNORMAL** state for the specified time interval.

Output parameters:

- Incident start Time
- Incident end time (only for the default database and/or host, non-verbose output)
- Target (for example, database, host)
- Problem
 - Description: Detailed description of the problem
 - Cause: Root cause of the problem and contributing factors
- Action: an action that corrects the abnormal state covered in the diagnosis

Reporting Format: The diagnostic information is displayed in a time compressed or time series order, grouped by components.

Examples

To display diagnostic information of a database for a specific time interval:

```
$ chactl query diagnosis -db oltpacdb -start "2016-02-01 02:52:50.0" -end
"2016-02-01 03:19:15.0"
2016-02-01 01:47:10.0 Database oltpacdb DB Control File IO Performance
(oltpacdb_1) [detected]
2016-02-01 01:47:10.0 Database oltpacdb DB Control File IO Performance
```

```
(oltpacdb_2) [detected]
2016-02-01 02:52:15.0 Database oltpacdb DB CPU Utilization (oltpacdb_2)
[detected]
2016-02-01 02:52:50.0 Database oltpacdb DB CPU Utilization (oltpacdb_1)
[detected]
2016-02-01 02:59:35.0 Database oltpacdb DB Log File Switch (oltpacdb_1)
[detected]
2016-02-01 02:59:45.0 Database oltpacdb DB Log File Switch (oltpacdb_2)
[detected]
```

Problem: DB Control File IO Performance

Description: CHA has detected that reads or writes to the control files are slower than expected.

Cause: The Cluster Health Advisor (CHA) detected that reads or writes to the control files were slow because of an increase in disk IO.

The slow control file reads and writes may have an impact on checkpoint and Log Writer (LGWR) performance.

Action: Separate the control files from other database files and move them to faster disks or Solid State Devices.

Problem: DB CPU Utilization

Description: CHA detected larger than expected CPU utilization for this database.

Cause: The Cluster Health Advisor (CHA) detected an increase in database CPU utilization because of an increase in the database workload.

Action: Identify the CPU intensive queries by using the Automatic Diagnostic and Defect Manager (ADDM)

and follow the recommendations given there. Limit the number of CPU intensive queries

or relocate sessions to less busy machines. Add CPUs if the CPU capacity is insufficient to support the load

without a performance degradation or effects on other databases.

Problem: DB Log File Switch

Description: CHA detected that database sessions are waiting longer than expected for log switch completions.

Cause: The Cluster Health Advisor (CHA) detected high contention during log switches

because the redo log files were small and the redo logs switched frequently.

Action: Increase the size of the redo logs.

Error Message

Message: *Target* is operating normally

Description: No incidents are found on the target.

Message: No data was found for active *Target*

Description: No data was found, but the target was operating or active at the time of the query.

Message: Target is not active or was not being monitored.

Description: No data was found because the target was not monitored at the time of the query.

C.7 chactl query model

Use the `chactl query model` command to list all Oracle Cluster Health Advisor models or to view detailed information about a specific Oracle Cluster Health Advisor model.

Syntax

```
chactl query model [-name model_name [-verbose]][-help]
```

Examples

- To list all base Oracle Cluster Health Advisor models:

```
$ chactl query model
Models: MOD1, MOD2, MOD3, MOD4, MOD5, MOD6, MOD7
```

```
$ chactl query model -name weekday
Model: weekday
Target Type: DATABASE
Version: 12.2.0.1_0
OS Calibrated on: Linux amd64
Calibration Target Name: prod
Calibration Date: 2016-09-10 12:59:49
Calibration Time Ranges: start=2016-09-09 16:00:00,end=2016-09-09 23:00:00
Calibration KPIs: not specified
```

- To view detailed information, including calibration metadata, about the specific Oracle Cluster Health Advisor model:

```
$ chactl query model -name MOD5 -verbose
Model: MOD5
CREATION_DATE:      Jan 10,2016 10:10
VALIDATION_STATUS:  Validated
DATA_FROM_TARGET :   inst72, inst75
USED_IN_TARGET :     inst76, inst75, prodDB, evalDB-evalSP
CAL_DATA_FROM_DATE:  Jan 05,2016 10:00
CAL_DATA_TO_DATE:    Jan 07,2016 13:00
CAL_DATA_FROM_TARGETS  inst73, inst75
...
```

C.8 chactl query repository

Use the `chactl query repository` command to view the maximum retention time, number of targets, and the size of the Oracle Cluster Health Advisor repository.

Note

Applicable only if GIMR is configured. GIMR is optionally supported in Oracle Database 19c. However, it's desupported in Oracle AI Database 26ai.

Syntax

```
chactl query repository [-help]
```

Examples

To view information about the Oracle Cluster Health Advisor repository:

```
$ chactl query repository
specified max retention time(hrs) : 72
available retention time(hrs)      : 212
available number of entities       : 2
allocated number of entities       : 0
total repository size(gb)          : 2.00
allocated repository size(gb)      : 0.07
```

C.9 chactl query calibration

Use the `chactl query calibration` command to view detailed information about the calibration data of a specific target.

Syntax**Note**

Applicable only if GIMR is configured. GIMR is optionally supported in Oracle Database 19c. However, it's desupported in Oracle AI Database 26ai.

```
chactl query calibration {-cluster|-db db_unique_name} [-timeranges
'start=time_stamp,end=time_stamp,...'] [-kpiset 'name=kpi_name min=val
max=val,...' ] [-interval val][-help]
```

Specify the interval in hours.

Specify date and time in the YYYY-MM-DD HH24:MI:SS format.

Note

If you do not specify a time interval, then the `chactl query calibration` command displays all the calibration data collected for a specific target.

The following Key Performance Indicators are supported for database:

- CPUPERCENT - CPU utilization - Percent
- IOREAD - Disk read - Mbyte/sec
- DBTIMEPERCALL - Database time per user call - usec/call
- IOWRITE - Disk write - Mbyte/sec
- IOTHROUGHPUT - Disk throughput - IO/sec

The following Key Performance Indicators are supported for cluster:

- CPUPERCENT - CPU utilization - Percent
- IOREAD - Disk read - Mbyte/sec
- IOWRITE - Disk write - Mbyte/sec
- IOTHROUGHPUT - Disk throughput - IO/sec

Examples

To view detailed information about the calibration data of the specified target:

```
$ chactl query calibration -db oltpacdb -timeranges
'start=2016-07-26 01:00:00,end=2016-07-26 02:00:00,start=2016-07-26
03:00:00,end=2016-07-26 04:00:00'
-kpiset 'name=CPUPERCENT min=20 max=40, name=IOTHROUGHPUT min=500 max=9000' -
interval 2
```

```
Database name : oltpacdb
Start time : 2016-07-26 01:03:10
End time : 2016-07-26 01:57:25
Total Samples : 120
Percentage of filtered data : 8.32%
The number of data samples may not be sufficient for calibration.
```

1) Disk read (ASM) (Mbyte/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
4.96	0.20	8.98	0.06	25.68
<25	<50	<75	<100	>=100
97.50%	2.50%	0.00%	0.00%	0.00%

2) Disk write (ASM) (Mbyte/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
27.73	9.72	31.75	4.16	109.39
<50	<100	<150	<200	>=200
73.33%	22.50%	4.17%	0.00%	0.00%

3) Disk throughput (ASM) (IO/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
2407.50	1500.00	1978.55	700.00	7800.00
<5000	<10000	<15000	<20000	>=20000
83.33%	16.67%	0.00%	0.00%	0.00%

4) CPU utilization (total) (%)

MEAN	MEDIAN	STDDEV	MIN	MAX
21.99	21.75	1.36	20.00	26.80

<20	<40	<60	<80	>=80
0.00%	100.00%	0.00%	0.00%	0.00%

5) Database time per user call (usec/call)

MEAN	MEDIAN	STDDEV	MIN	MAX
267.39	264.87	32.05	205.80	484.57

<10000000	<20000000	<30000000	<40000000	<50000000	<60000000	<70000000	>=70000000
100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Database name : oltpacdb

Start time : 2016-07-26 03:00:00

End time : 2016-07-26 03:53:30

Total Samples : 342

Percentage of filtered data : 23.72%

The number of data samples may not be sufficient for calibration.

1) Disk read (ASM) (Mbyte/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
12.18	0.28	16.07	0.05	60.98

<25	<50	<75	<100	>=100
64.33%	34.50%	1.17%	0.00%	0.00%

2) Disk write (ASM) (Mbyte/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
57.57	51.14	34.12	16.10	135.29

<50	<100	<150	<200	>=200
49.12%	38.30%	12.57%	0.00%	0.00%

3) Disk throughput (ASM) (IO/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
5048.83	4300.00	1730.17	2700.00	9000.00

<5000	<10000	<15000	<20000	>=20000
63.74%	36.26%	0.00%	0.00%	0.00%

4) CPU utilization (total) (%)

MEAN	MEDIAN	STDDEV	MIN	MAX
23.10	22.80	1.88	20.00	31.40

<20	<40	<60	<80	>=80
0.00%	100.00%	0.00%	0.00%	0.00%

5) Database time per user call (usec/call)

MEAN	MEDIAN	STDDEV	MIN	MAX			
744.39	256.47	2892.71	211.45	45438.35			
<100000000	<200000000	<300000000	<400000000	<500000000	<600000000	<700000000	
>=700000000							
100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	

C.10 chactl remove model

Use the `chactl remove model` command to delete an Oracle Cluster Health Advisor model along with the calibration data and metadata of the model from the Oracle Cluster Health Advisor repository.

Note

If the model is being used to monitor the targets, then the `chactl remove model` command cannot delete any model.

Syntax

```
chactl remove model -name model_name [-help]
```

Error Message

Error: *model_name* does not exist

Description: The specified Oracle Cluster Health Advisor model does not exist in the Oracle Cluster Health Advisor repository.

C.11 chactl rename model

Use the `chactl rename model` command to rename an Oracle Cluster Health Advisor model in the Oracle Cluster Health Advisor repository.

Assign a descriptive and unique name to the model. Oracle Cluster Health Advisor preserves all the links related to the renamed model.

Syntax

```
chactl rename model -from model_name -to model_name [-help]
```

Error Messages

Error: *model_name* does not exist

Description: The specified model name does not exist in the Oracle Cluster Health Advisor repository.

Error: *dest_name* already exist

Description: The specified model name already exists in the Oracle Cluster Health Advisor repository.

C.12 chactl export model

Use the `chactl export model` command to export Oracle Cluster Health Advisor models.

Syntax

① Note

Applicable only if GIMR is configured. GIMR is optionally supported in Oracle Database 19c. However, it's desupported in Oracle AI Database 26ai.

```
chactl export model -name model_name -file output_file [-help]
```

Example

```
$ chactl export model -name weekday -file /tmp//weekday.mod
```

C.13 chactl import model

Use the `chactl import model` command to import Oracle Cluster Health Advisor models.

Syntax

① Note

Applicable only if GIMR is configured. GIMR is optionally supported in Oracle Database 19c. However, it's desupported in Oracle AI Database 26ai.

```
chactl import model -name model_name -file model_file [-force] [-help]
```

While importing, if there is an existing model with the same name as the model being imported, then use the `-force` option to overwrite.

Example C-1 Example

```
$ chactl import model -name weekday -file /tmp//weekday.mod
```

C.14 chactl set maxretention

Use the `chactl set maxretention` command to set the maximum retention time for the diagnostic data.

The default and minimum retention time is 72 hours. If the Oracle Cluster Health Advisor repository does not have enough space, then the retention time is decreased for all the targets.

Note

Oracle Cluster Health Advisor stops monitoring if the retention time is less than 24 hours.

Syntax

```
chactl set maxretention -time retention_time [-help]
```

Specify the retention time in hours.

Examples

To set the maximum retention time to 80 hours:

```
$ chactl set maxretention -time 80  
max retention successfully set to 80 hours
```

Error Message

Error: Specified time is smaller than the allowed minimum

Description: This message is returned if the input value for maximum retention time is smaller than the minimum value.

C.15 chactl resize repository

Use the `chactl resize repository` command to resize the tablespace of the Oracle Cluster Health Advisor repository based on the current retention time and the number of targets.

Note

- Applicable only if GIMR is configured. GIMR is optionally supported in Oracle Database 19c. However, it's desupported in Oracle AI Database 26ai.
- The `chactl resize repository` command fails if your system does not have enough free disk space or if the tablespace contains data beyond requested resize value.

Syntax

```
chactl resize repository -entities total number of hosts and database  
instances [-force | -eval] [-help]
```

Examples

To set the number of targets in the tablespace to 32:

```
chactl resize repository -entities 32  
repository successfully resized for 32 targets
```


D

Behavior Changes, Deprecated and Desupported Features

Review information about changes, deprecations, and desupports.

- [Oracle Database Quality of Service \(QoS\) Management is Deprecated and Desupported in Release 21c](#)
Starting in Oracle Database release 21c, Oracle Database Quality of Service (QoS) Management is deprecated and desupported.
- [Autonomous Health Framework Scope \(AHF Scope\) Deprecated](#)
The Autonomous Health Framework Scope (AHF Scope) feature of Cluster Health Advisor (CHA) is deprecated in Oracle AI Database 26ai.

D.1 Oracle Database Quality of Service (QoS) Management is Deprecated and Desupported in Release 21c

Starting in Oracle Database release 21c, Oracle Database Quality of Service (QoS) Management is deprecated and desupported.

Oracle Database Quality of Service (QoS) Management automates the workload management for an entire system by adjusting the system configuration based on pre-defined policies to keep applications running at the performance levels needed. Applications and databases are increasingly deployed in systems that provide some of the resource management capabilities of Oracle Database Quality of Service (QoS) Management. At the same time, Oracle's Autonomous Health Framework has been enhanced to adjust and provide recommendations to mitigate events and conditions that impact the health and operational capability of a system and its associated components. For those reasons, Oracle Database Quality of Service (QoS) Management has been deprecated and desupported with Oracle Database 21c.

D.2 Autonomous Health Framework Scope (AHF Scope) Deprecated

The Autonomous Health Framework Scope (AHF Scope) feature of Cluster Health Advisor (CHA) is deprecated in Oracle AI Database 26ai.

Date: April 2025

To enable Oracle to allocate resources to better address customer requirements, Oracle is discontinuing AHF Scope. In its place, Oracle recommends that you use Autonomous Health Framework Insights (AHF Insights).