

Oracle Database 11g: RAC Administration

Student Guide

D50311GC20

Edition 2.0

January 2010

D64916

ORACLE®

Authors

James Womack

James Spiller

Technical Contributors

David Brower

Jean-Francois Verrier

Mark Fuller

Mike Leatherman

Barb Lundhild

S. Matt Taylor

Rick Wessman

Technical Reviewers

Christopher Andrews

Christian Bauwens

Michael Cebulla

Jonathan Creighton

Al Flournoy

Andy Fortunak

Joel Goodman

Michael Hazel

Pete Jones

Jerry Lee

Markus Michalewicz

Peter Sharman

Ranbir Singh

Linda Smalley

Janet Stern

Richard Strohm

Branislav Valny

Doug Williams

Publisher

Sujatha Nagendra

Copyright © 2010, Oracle. All rights reserved.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Contents

I Oracle 11g Release 2 RAC Administration Practice Notes

Oracle 11g R2 RAC Administration Course I-2

Practice Prerequisite I-3

Removing and Reinstalling Grid Infrastructure I-4

11 Real Application Clusters Database Installation

Objectives 11-2

Installing the Oracle Database Software 11-3

Creating the Cluster Database 11-7

Select Database Type 11-8

Database Identification 11-9

Cluster Database Management Options 11-10

Passwords for Database Schema Owners 11-11

Database File Locations 11-12

Recovery Configuration 11-13

Database Content 11-14

Initialization Parameters 11-15

Database Storage Options 11-16

Create the Database 11-17

Monitor Progress 11-18

Postinstallation Tasks 11-19

Check Managed Targets 11-20

Background Processes Specific to Oracle RAC 11-21

Single Instance to RAC Conversion 11-23

Issues for Converting Single Instance Databases to Oracle RAC 11-24

Single-Instance Conversion Using the DBCA 11-25

Conversion Steps 11-26

Single-Instance Conversion Using `rconfig` 11-29

Quiz 11-31

Summary 11-33

Practice 11 Overview 11-34

12 Oracle RAC Administration

Objectives 12-2

Cluster Database Home Page 12-3

Cluster Database Instance Home Page	12-5
Cluster Home Page	12-6
Configuration Section	12-7
Topology Viewer	12-9
Enterprise Manager Alerts and RAC	12-10
Enterprise Manager Metrics and RAC	12-11
Enterprise Manager Alert History and RAC	12-13
Enterprise Manager Blackouts and RAC	12-14
Redo Log Files and RAC	12-15
Automatic Undo Management and RAC	12-16
Starting and Stopping RAC Instances	12-17
Starting and Stopping RAC Instances with SQL*Plus	12-18
Starting and Stopping RAC Instances with <code>srvctl</code>	12-19
Switch Between the Automatic and Manual Policies	12-20
RAC Initialization Parameter Files	12-21
<code>SPFILE</code> Parameter Values and RAC	12-22
EM and <code>SPFILE</code> Parameter Values	12-23
RAC Initialization Parameters	12-25
Parameters That Require Identical Settings	12-27
Parameters That Require Unique Settings	12-28
Quiescing RAC Databases	12-29
Terminating Sessions on a Specific Instance	12-30
How SQL*Plus Commands Affect Instances	12-31
Transparent Data Encryption and Wallets in RAC	12-32
Quiz	12-33
Summary	12-35
Practice 12 Overview	12-36

13 Managing Backup and Recovery for RAC

Objectives	13-2
Protecting Against Media Failure	13-3
Media Recovery in Oracle RAC	13-4
Parallel Recovery in RAC	13-5
Archived Log File Configurations	13-6
RAC and the Fast Recovery Area	13-7
RAC Backup and Recovery Using EM	13-8
Configure RAC Recovery Settings with EM	13-9
Archived Redo File Conventions in RAC	13-10
Configure RAC Backup Settings with EM	13-11
Oracle Recovery Manager	13-12

Configure RMAN Snapshot Control File Location	13-13
Configure Control File and SPFILE Autobackup	13-14
Crosschecking on Multiple RAC Clusters Nodes	13-15
Channel Connections to Cluster Instances	13-16
RMAN Channel Support for the Grid	13-17
RMAN Default Autolocation	13-18
Distribution of Backups	13-19
One Local Drive CFS Backup Scheme	13-20
Multiple Drives CFS Backup Scheme	13-21
Restoring and Recovering	13-22
Quiz	13-23
Summary	13-25
Practice 13 Overview	13-26
14 RAC Database Monitoring and Tuning	
Objectives	14-2
CPU and Wait Time Tuning Dimensions	14-3
RAC-Specific Tuning	14-4
RAC and Instance or Crash Recovery	14-5
Instance Recovery and Database Availability	14-7
Instance Recovery and RAC	14-8
Analyzing Cache Fusion Impact in RAC	14-10
Typical Latencies for RAC Operations	14-11
Wait Events for RAC	14-12
Wait Event Views	14-13
Global Cache Wait Events: Overview	14-14
2-way Block Request: Example	14-16
3-way Block Request: Example	14-17
2-way Grant: Example	14-18
Global Enqueue Waits: Overview	14-19
Session and System Statistics	14-20
Most Common RAC Tuning Tips	14-21
Index Block Contention: Considerations	14-23
Oracle Sequences and Index Contention	14-24
Undo Block Considerations	14-25
High-Water Mark Considerations	14-26
Concurrent Cross-Instance Calls: Considerations	14-27
Monitoring RAC Database and Cluster Performance	14-28
Cluster Database Performance Page	14-29
Determining Cluster Host Load Average	14-30

Determining Global Cache Block Access Latency	14-31
Determining Average Active Sessions	14-32
Determining Database Throughput	14-33
Accessing the Cluster Cache Coherency Page	14-35
Viewing the Cluster Interconnects Page	14-37
Viewing the Database Locks Page	14-39
AWR Snapshots in RAC	14-40
AWR Reports and RAC: Overview	14-41
Active Session History Reports for RAC	14-43
Automatic Database Diagnostic Monitor for RAC	14-45
What Does ADDM Diagnose for RAC?	14-47
EM Support for ADDM for RAC	14-48
Quiz	14-49
Summary	14-51
Practice 14 Overview	14-52

15 Services

Objectives	15-2
Oracle Services	15-3
Services for Policy- and Administrator-Managed Databases	15-4
Default Service Connections	15-5
Create Service with Enterprise Manager	15-6
Create Services with <code>SRVCTL</code>	15-7
Manage Services with Enterprise Manager	15-8
Managing Services with EM	15-9
Manage Services with <code>srvctl</code>	15-10
Use Services with Client Applications	15-11
Services and Connection Load Balancing	15-12
Services and Transparent Application Failover	15-13
Use Services with the Resource Manager	15-14
Services and Resource Manager with EM	15-15
Use Services with the Scheduler	15-16
Services and the Scheduler with EM	15-17
Using Distributed Transactions with RAC	15-19
Distributed Transactions and Services	15-20
Service Thresholds and Alerts	15-22
Services and Thresholds Alerts: Example	15-23
Service Aggregation and Tracing	15-24
Top Services Performance Page	15-25
Service Aggregation Configuration	15-26
Service, Module, and Action Monitoring	15-27

Service Performance Views 15-28

Quiz 15-29

Summary 15-31

Practice 15 Overview 15-32

16 Design for High Availability

Objectives 16-2

Causes of Unplanned Down Time 16-3

Causes of Planned Down Time 16-4

Oracle's Solution to Down Time 16-5

RAC and Data Guard Complementarity 16-6

Maximum Availability Architecture 16-7

RAC and Data Guard Topologies 16-8

RAC and Data Guard Architecture 16-9

Data Guard Broker (DGB) and Oracle Clusterware (OC) Integration 16-11

Fast-Start Failover: Overview 16-12

Data Guard Broker Configuration Files 16-14

Real-Time Query Physical Standby Database 16-15

Hardware Assisted Resilient Data 16-16

Database High Availability: Best Practices 16-17

How Many ASM Disk Groups Per Database? 16-18

Which RAID Configuration for High Availability? 16-19

Should You Use ASM Mirroring Protection? 16-20

What Type of Striping Works Best? 16-21

ASM Striping Only 16-22

Hardware RAID—Striped LUNs 16-23

Hardware RAID—Striped LUNs HA 16-24

Disk I/O Design Summary 16-25

Extended RAC: Overview 16-26

Extended RAC Connectivity 16-27

Extended RAC Disk Mirroring 16-28

Achieving Quorum with Extended RAC 16-29

Additional Data Guard Benefits 16-30

Using a Test Environment 16-31

Quiz 16-32

Summary 16-33

Appendix A: Practices and Solutions

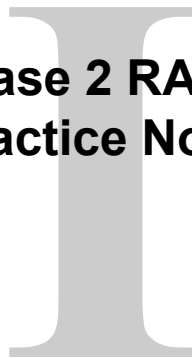
Appendix B: Oracle RAC One Node

- Oracle RAC One Node B-2
- The Omotion Utility B-3
- Oracle RAC One Node and OVM B-4

Appendix C: High Availability of Connections

- Objectives C-2
- Types of Workload Distribution C-3
- Client-Side Load Balancing C-4
- Client-Side, Connect-Time Failover C-5
- Server-Side, Connect-Time Load Balancing C-6
- Fast Application Notification: Overview C-7
- Fast Application Notification: Benefits C-8
- FAN-Supported Event Types C-9
- FAN Event Status C-10
- FAN Event Reasons C-11
- FAN Event Format C-12
- Load Balancing Advisory: FAN Event C-13
- Implementation of Server-Side Callouts C-14
- Server-Side Callout Parse: Example C-15
- Server-Side Callout Filter: Example C-16
- Configuring the Server-Side ONS C-17
- Optionally Configure the Client-Side ONS C-18
- JDBC Fast Connection Failover: Overview C-19
- Using Oracle Streams Advanced Queuing for FAN C-20
- JDBC/ODP.NET FCF Benefits C-21
- Load Balancing Advisory C-22
- JDBC/ODP.NET Runtime Connection Load Balancing: Overview C-23
- Monitor LBA FAN Events C-24
- Transparent Application Failover: Overview C-25
- TAF Basic Configuration Without FAN: Example C-26
- TAF Basic Configuration with FAN: Example C-27
- TAF Preconnect Configuration: Example C-28
- TAF Verification C-29
- FAN Connection Pools and TAF Considerations C-30
- Summary C-31

Oracle 11g Release 2 RAC Administration Practice Notes



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle 11g R2 RAC Administration Course

- This course comprises lessons 11 through 16 of the original five-day Oracle 11g: RAC and Grid Foundation Administration course.
- Script names, lesson and practice numbering remain unchanged in order to:
 - Preserve classroom deployment efforts
 - Minimize testing complexity

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Internal & Oracle Academy Use Only

Practice Prerequisite

To prepare for the practices, you must:

1. Get your cluster and host assignment from your instructor.
2. Log into the first node of your cluster as root, change directory to `/home/oracle/solutions/catchup10` and execute the `catchup.sh` script.

```
# cd /home/oracle/solutions/catchup10
# ./catchup.sh
```

3. This script will take about an hour to complete, so execute the script before the first lecture starts.
4. When finished, you should have Grid Infrastructure installed and configured on all three nodes.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Removing and Reinstalling Grid Infrastructure

If you have completed the Oracle 11g: Grid Foundation Administration course prior to starting this one, you should have a fully functioning three node cluster. If you wish to reset your cluster, perform the following steps:

1. Log into the first node of your cluster as root and change directory to `/home/oracle/solutions/reset` and execute the `reset_master.sh` script. Answer **[Y]** when prompted to overwrite existing files when files are unzipped.
2. Navigate to `/home/oracle/solutions/catchup10` and execute the `catchup.sh` script.

```
# cd /home/oracle/solutions/reset
# ./reset_master.sh
# cd /home/oracle/solutions/catchup10
# ./catchup.sh
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Real Application Clusters Database Installation



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to:

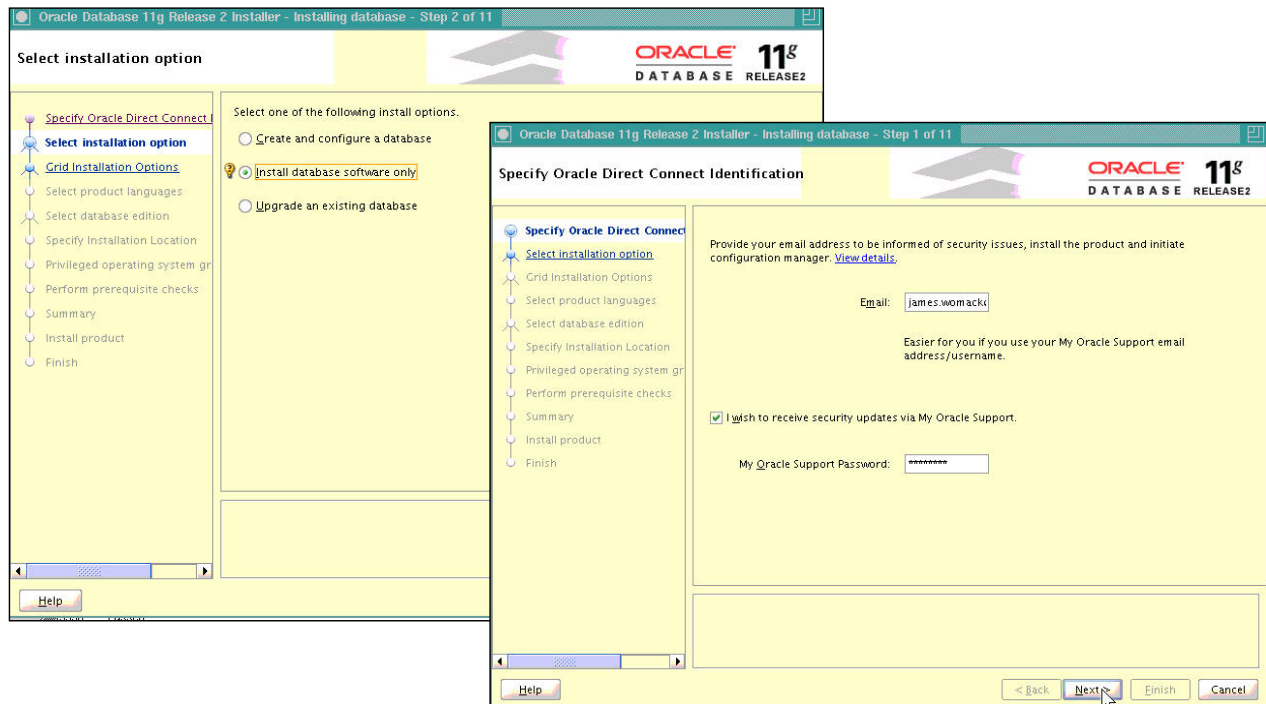
- Install the Oracle database software
- Create a cluster database
- Perform post-database-creation tasks
- Perform a single instance to RAC conversion

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Installing the Oracle Database Software

```
$ /stage/database/Disk1/runInstaller
```



ORACLE

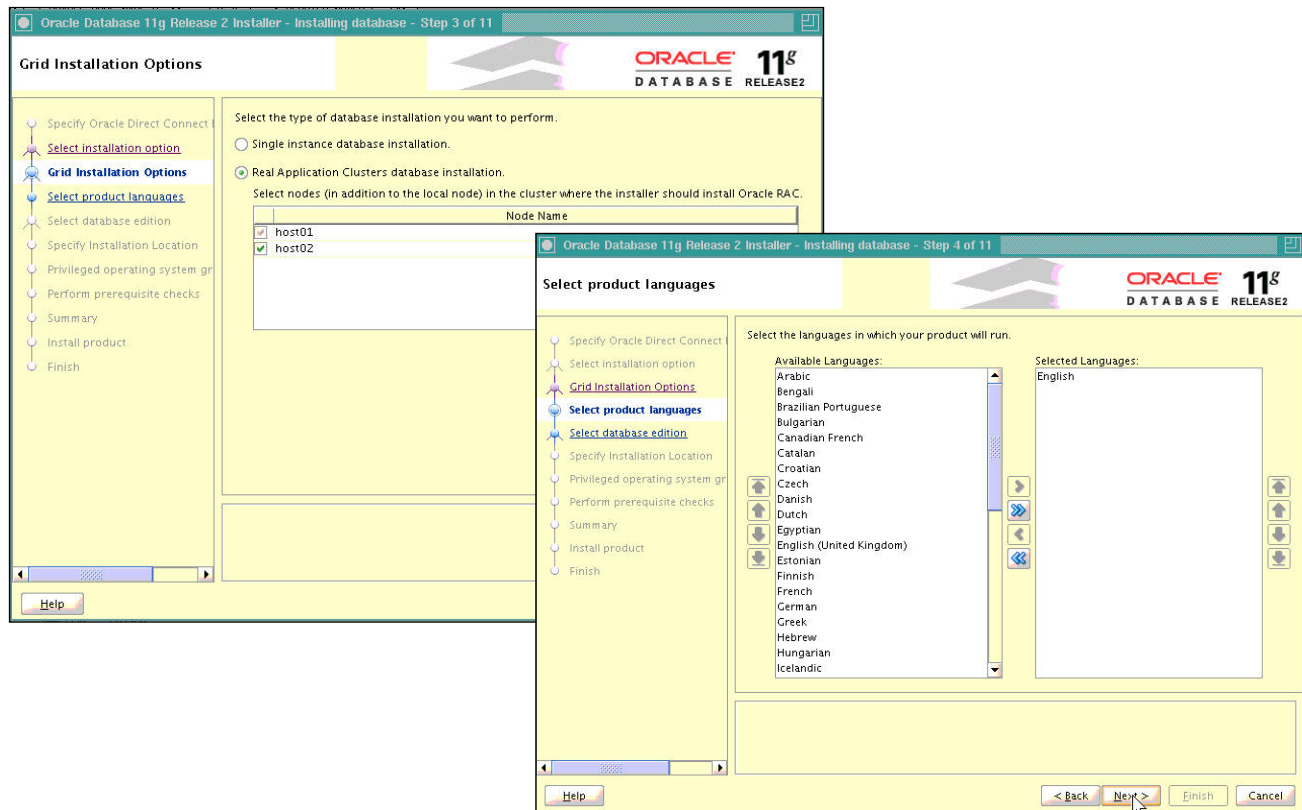
Copyright © 2010, Oracle. All rights reserved.

Installing the Oracle Database Software

The Oracle Universal Installer (OUI) is used to install the Oracle Database 11g Release 2 (11.2) software. Start the OUI by executing the `runInstaller` command from the root directory of the Oracle Database 11g Release 2 CD-ROM or from the software staging location. You can use the Specify Oracle Direct Connect Identification window to specify an email address to receive security updates directly from Oracle Support as they occur. Alternatively, you can elect to opt out of these alerts. If you want to receive them, supply your email address and your Oracle Support password, and click Next.

The “Select installation option” window enables you to create and configure a database, install database software only, or upgrade an existing database. Select the “Install database software only” option and click Next.

Installing the Oracle Database Software



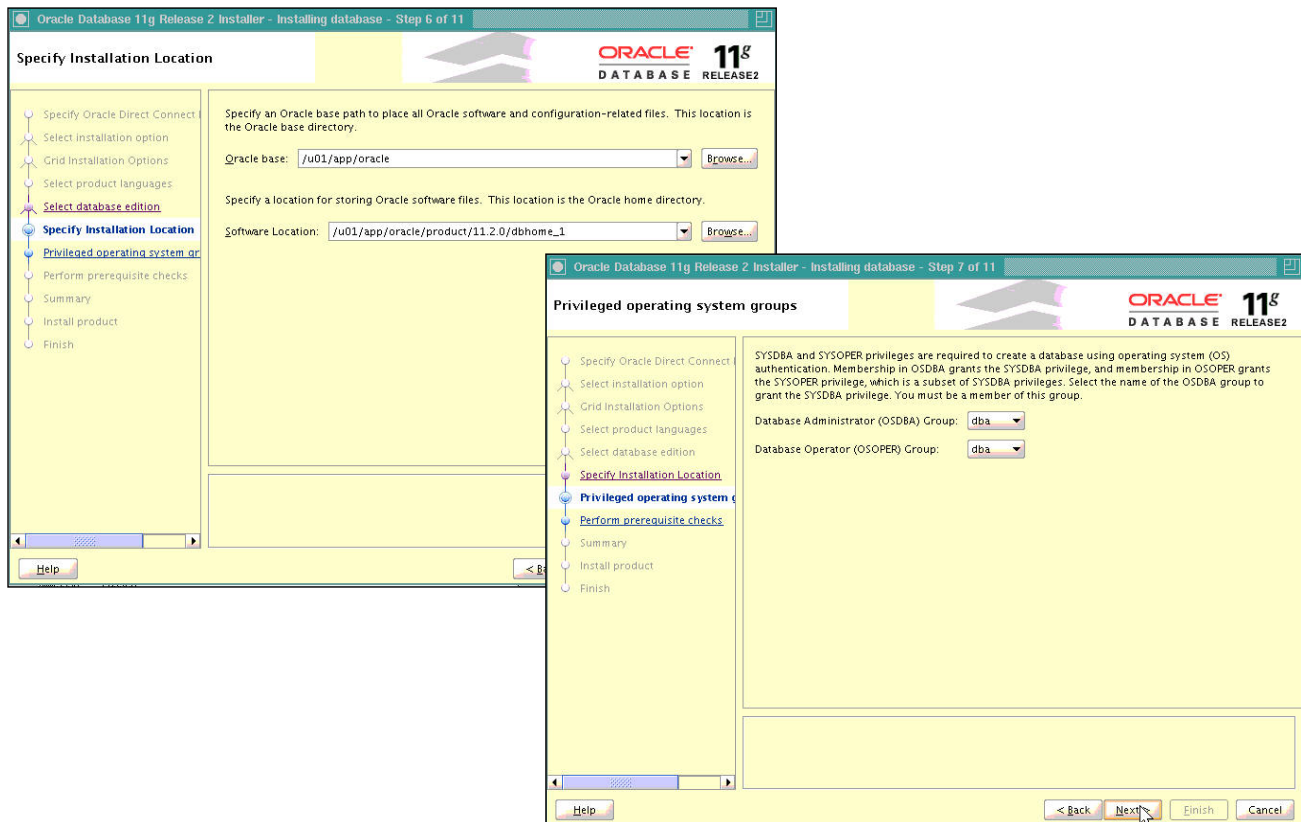
Copyright © 2010, Oracle. All rights reserved.

Installing the Oracle Database Software (continued)

In the Grid Installation Options window, select “Real Application Clusters database installation” and select all nodes in your cluster on which the software should be installed. Click Next to continue. In the “Select product languages” window, select your desired languages from the Available Languages list and click the right arrow to promote the selected languages to the Selected Languages list. Click Next to continue.

In the “Select database edition” window (not shown in the slide above), you select whether to install the Enterprise Edition or the Standard Edition. Select the Enterprise Edition option and click Next to continue.

Installing the Oracle Database Software

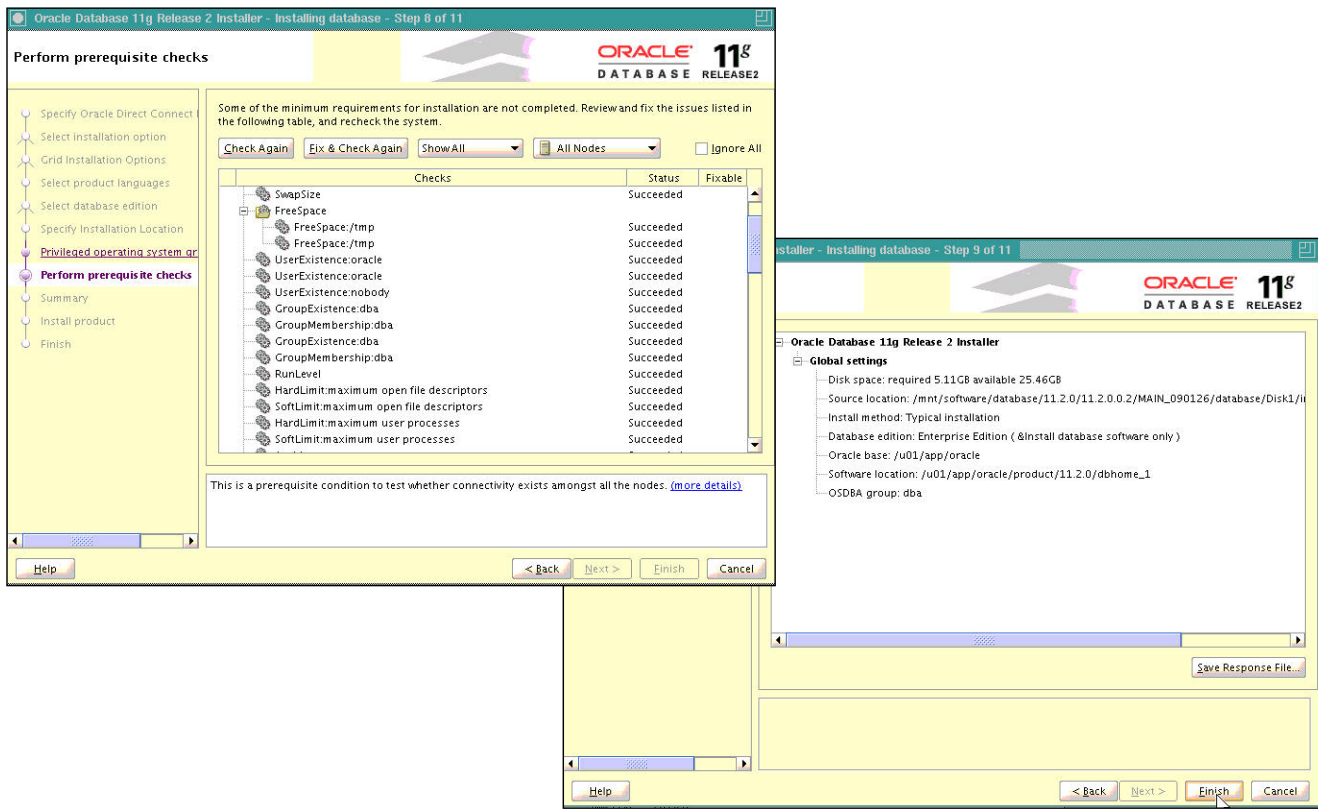


Copyright © 2010, Oracle. All rights reserved.

Installing the Oracle Database Software (continued)

In the Specify Installation Location window, provide a value for ORACLE_BASE if you have not yet already done so. The default ORACLE_BASE location is /u01/app/oracle, provided the RDBMS software is being installed by the oracle account. The Software Location section of the window enables you to specify a value for the ORACLE_HOME location. The default ORACLE_HOME location is /u01/app/oracle/product/11.2.0/dbhome_1. Accept the suggested path or enter your own location. After entering the information, review it for accuracy, and click the **Next** button to continue. In the “Privileged operating system groups” window, select the operating system group that will act as the OSDBA group. Next, select the group that will act as the OSOPER group. Click Next to continue.

Installing the Oracle Database Software



Copyright © 2010, Oracle. All rights reserved.

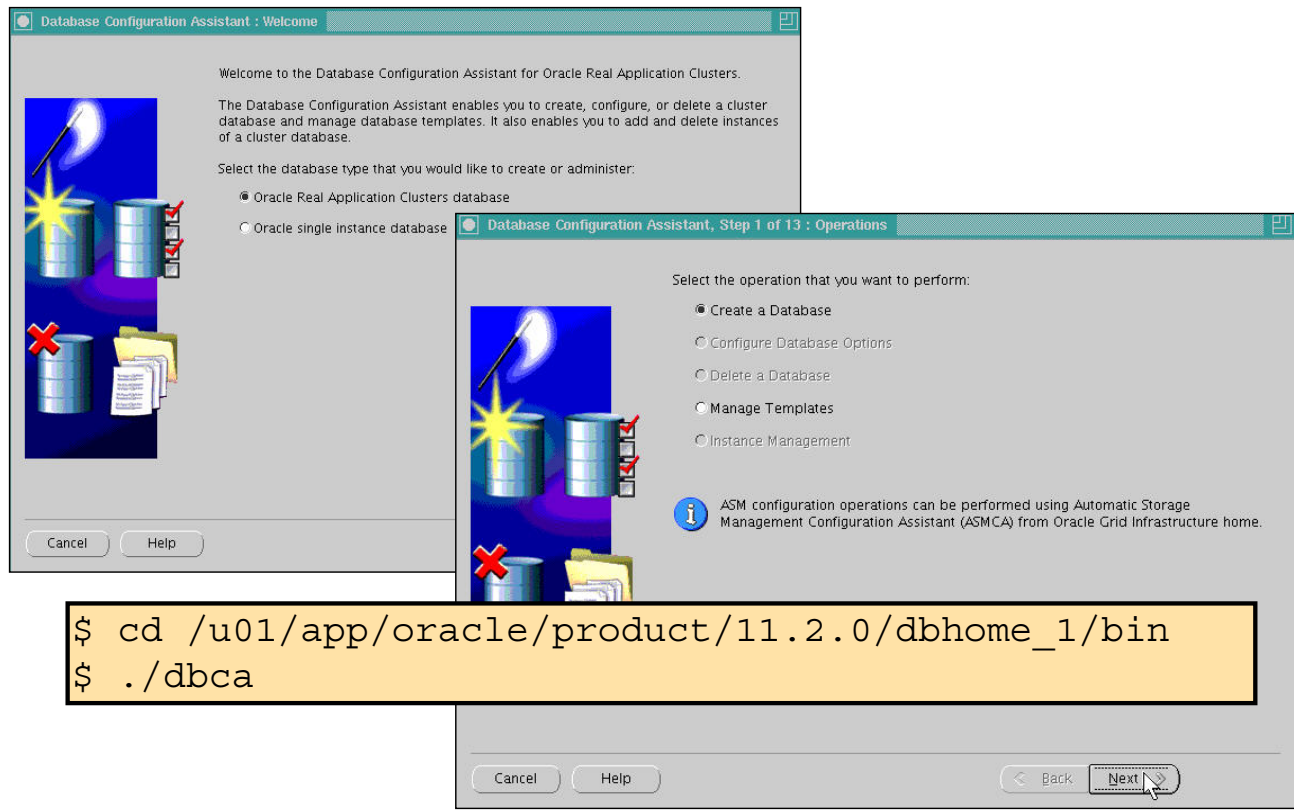
Installing the Oracle Database Software (continued)

The “Perform prerequisite checks” window verifies the operating system requirements that must be met for the installation to be successful. These requirements include:

- Certified operating system check
- Kernel parameters as required by the Oracle Database software
- Required operating system packages and correct revisions

After each successful check, the Status for that check will indicate Succeeded. Any tests that fail are also reported here. If any tests fail, click the “Fix & Check Again” button. The Installer will generate fix-up scripts to correct the system deficiencies if possible. Execute the scripts as directed by the Installer. The tests will be run again after completing the script executions. When all tests have succeeded, click the **Next** button to continue. In the Summary window, review the Global settings and click Finish. At the end of the installation, the OUI will display another window, prompting you to run the `root.sh` scripts on the nodes you chose for the installation. Follow the instructions to run the scripts. When finished, click the OK button to close the Execute Configuration Scripts window and return to the Finish screen. Click Close to complete the installation and close the OUI.

Creating the Cluster Database



```
$ cd /u01/app/oracle/product/11.2.0/dbhome_1/bin
$ ./dbca
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

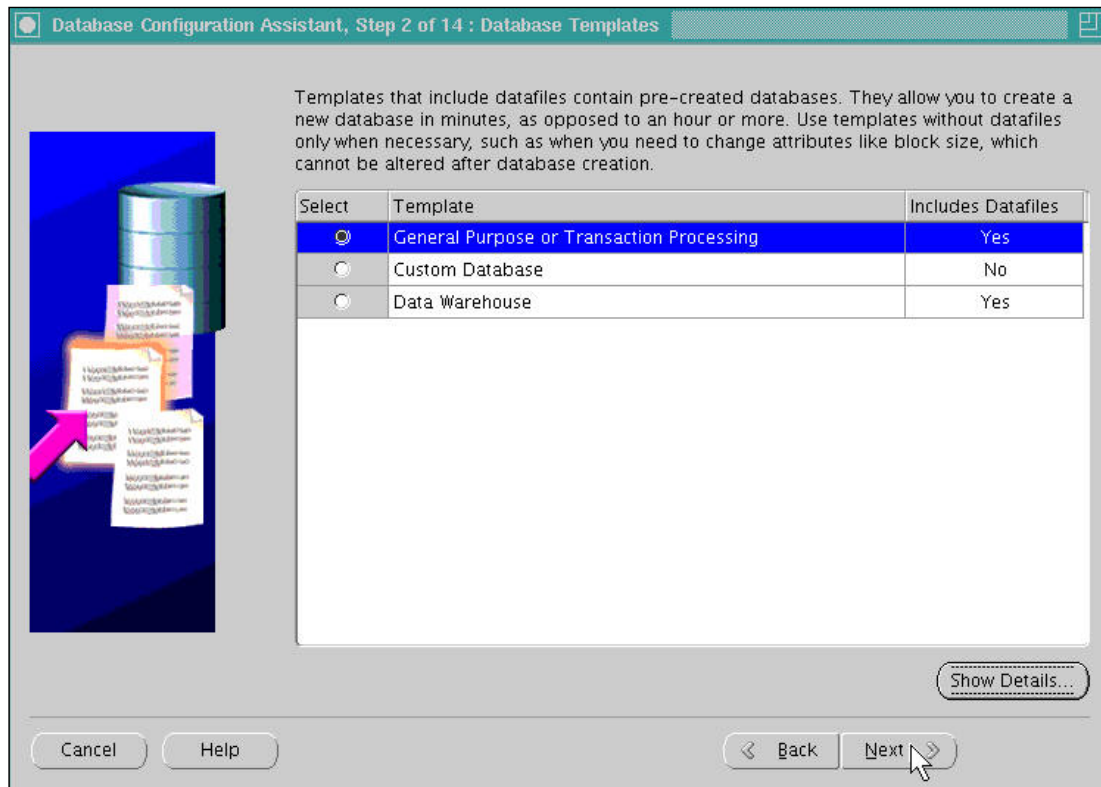
Creating the Cluster Database

To create the cluster database, change directory to `$ORACLE_HOME/bin` on the installing node and execute the database configuration assistant (DBCA) utility as shown below:

```
$ cd /u01/app/oracle/product/11.2.0/dbhome_1/bin
$ ./dbca
```

The Welcome window appears first. You must select the type of database that you want to install. Select the “Oracle Real Application Clusters database” option, and then click Next. The Operations window appears. For a first-time installation, you have only two choices: the first option enables you to create a database and the other option enables you to manage database creation templates. Select the “Create a Database” option, and then click Next to continue.

Select Database Type



ORACLE

Copyright © 2010, Oracle. All rights reserved.

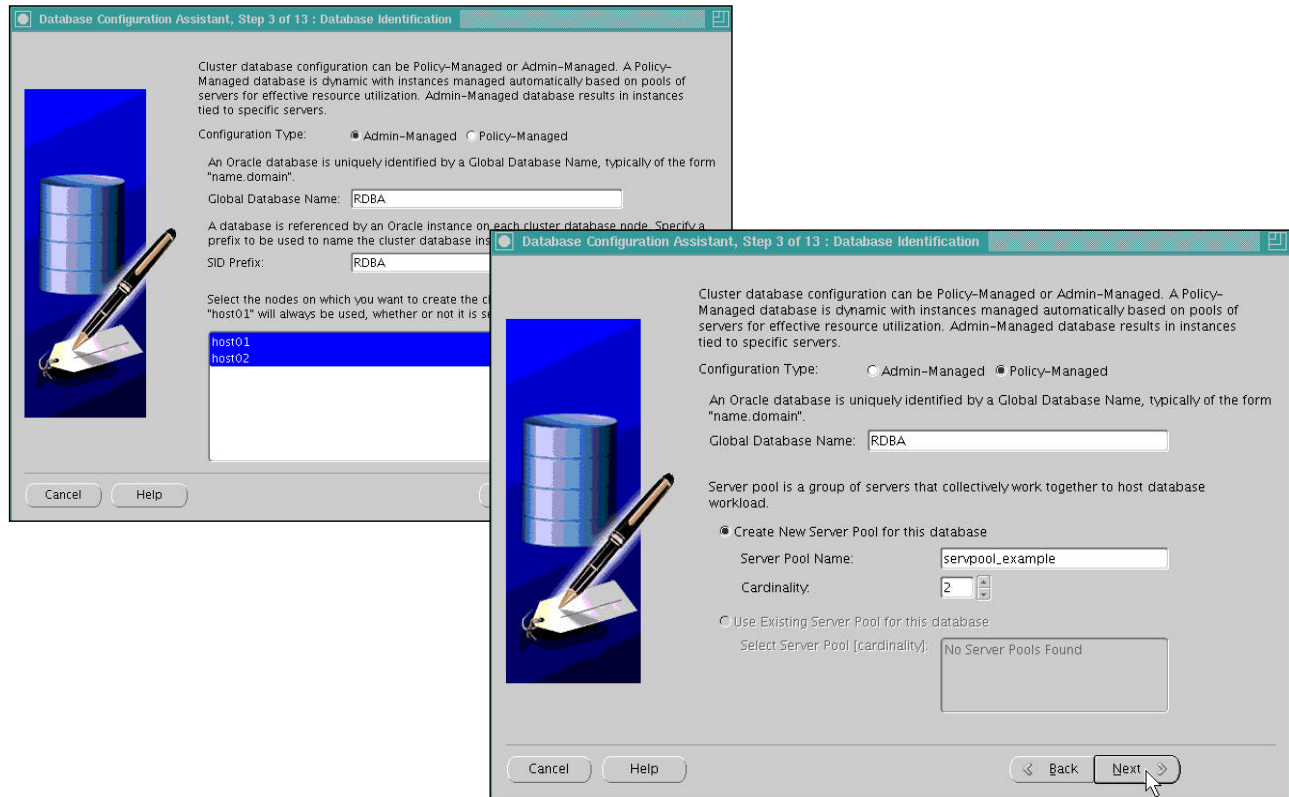
Select Database Type

The Database Templates window appears next. The DBCA tool provides several predefined database types to choose from, depending on your needs. The templates include:

- General Purpose or Transaction Processing
- Custom Database
- Data Warehouse

In the example in the slide above, the “General Purpose or Transaction Processing” option is chosen. Click the Next button to continue.

Database Identification



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Database Identification

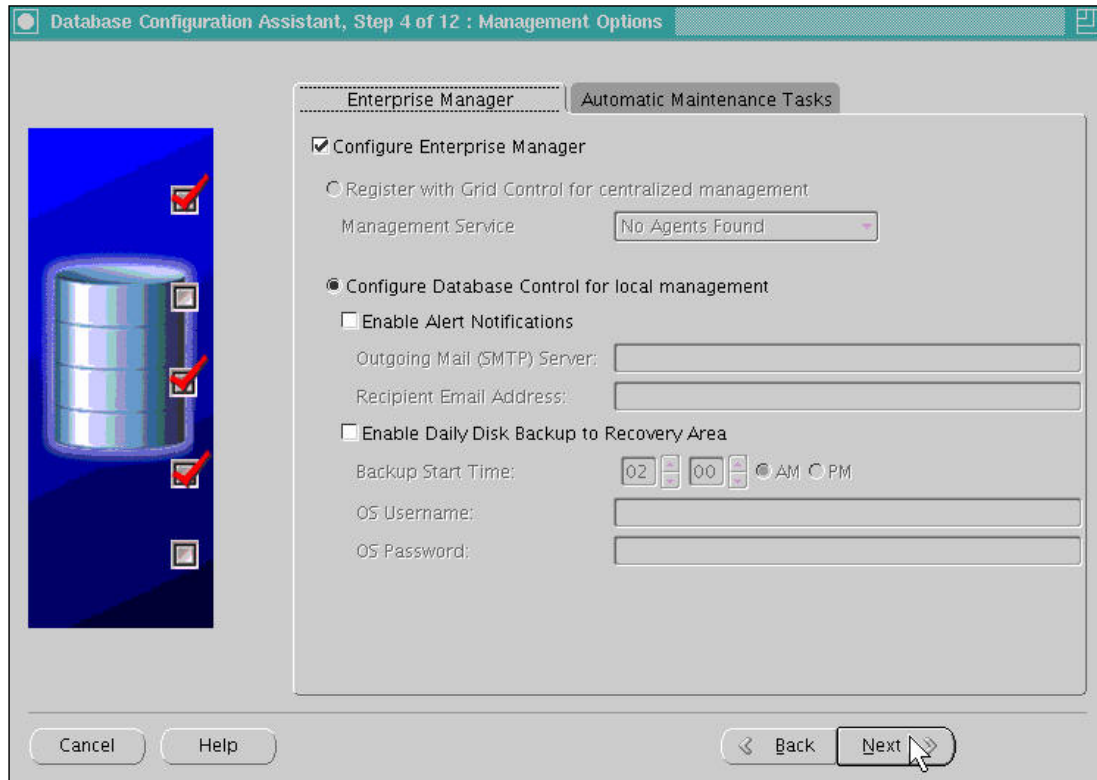
In the Database Identification window, you must choose between an administrator-managed and a policy-managed cluster database.

Administrator-managed RAC databases specify a list of cluster nodes where RAC instances will run. Services may also be specified and associated with preferred and alternative nodes. There is an explicit association between database services, instances, and cluster nodes.

Policy-based management, a new feature in this release, breaks the explicit association between services, instances, and cluster nodes. Policy-based management introduces the concept of server pools, which are logical divisions of a cluster that are dynamically allocated based on relative importance. Database services are associated with server pools and RAC instances are automatically started to satisfy the service to server pool associations. You specify in which server pool the database resource will run and the number of instances needed (cardinality). Oracle Clusterware is responsible for placing the database resource on a server. Server pools are logical divisions of a cluster into pools of servers which are allocated to host databases or other applications. Server pools are managed using `crsctl` and `srvctl` commands. Names must be unique within the resources defined for the cluster.

You must also choose the global database name and the nodes on which to create the cluster database. The global database name can be up to 30 characters in length and must begin with an alphabetical character. When you have finished, click Next to continue.

Cluster Database Management Options



ORACLE

Copyright © 2010, Oracle. All rights reserved.

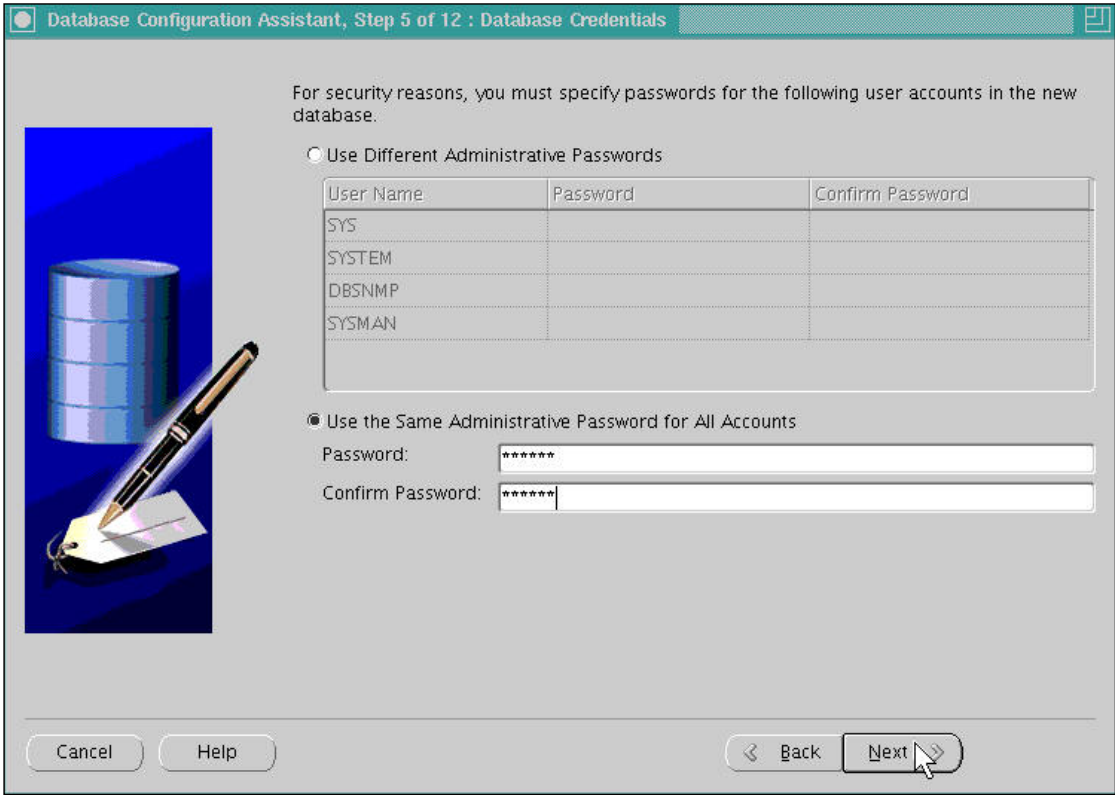
Cluster Database Management Options

The Management Options window is displayed. For small cluster environments, you may choose to manage your cluster with Enterprise Manager Database Control. To do this, select the “Configure Enterprise Manager” check box. If you have Grid Control installed somewhere on your network, you can select the “Use Grid Control for Database Management” option. If you select Enterprise Manager with the Grid Control option and the DBCA discovers agents running on the local node, you can select the preferred agent from a list. Grid Control can simplify database management in large, enterprise deployments.

You can also configure Database Control to send email notifications when alerts occur. If you want to configure this, you must supply a Simple Mail Transfer Protocol (SMTP) or outgoing mail server and an email address. You can also enable daily backups here. You must supply a backup start time as well as operating system user credentials for this option.

If you want to use Grid Control to manage your database, but have not yet installed and configured a Grid Control server, do not click either of the management methods. When you have made your choices, click the Next button to continue.

Passwords for Database Schema Owners



Database Configuration Assistant, Step 5 of 12 : Database Credentials

For security reasons, you must specify passwords for the following user accounts in the new database.

☐ Use Different Administrative Passwords

User Name	Password	Confirm Password
SYS		
SYSTEM		
DBSNMP		
SYSMAN		

☒ Use the Same Administrative Password for All Accounts

Password:

Confirm Password:

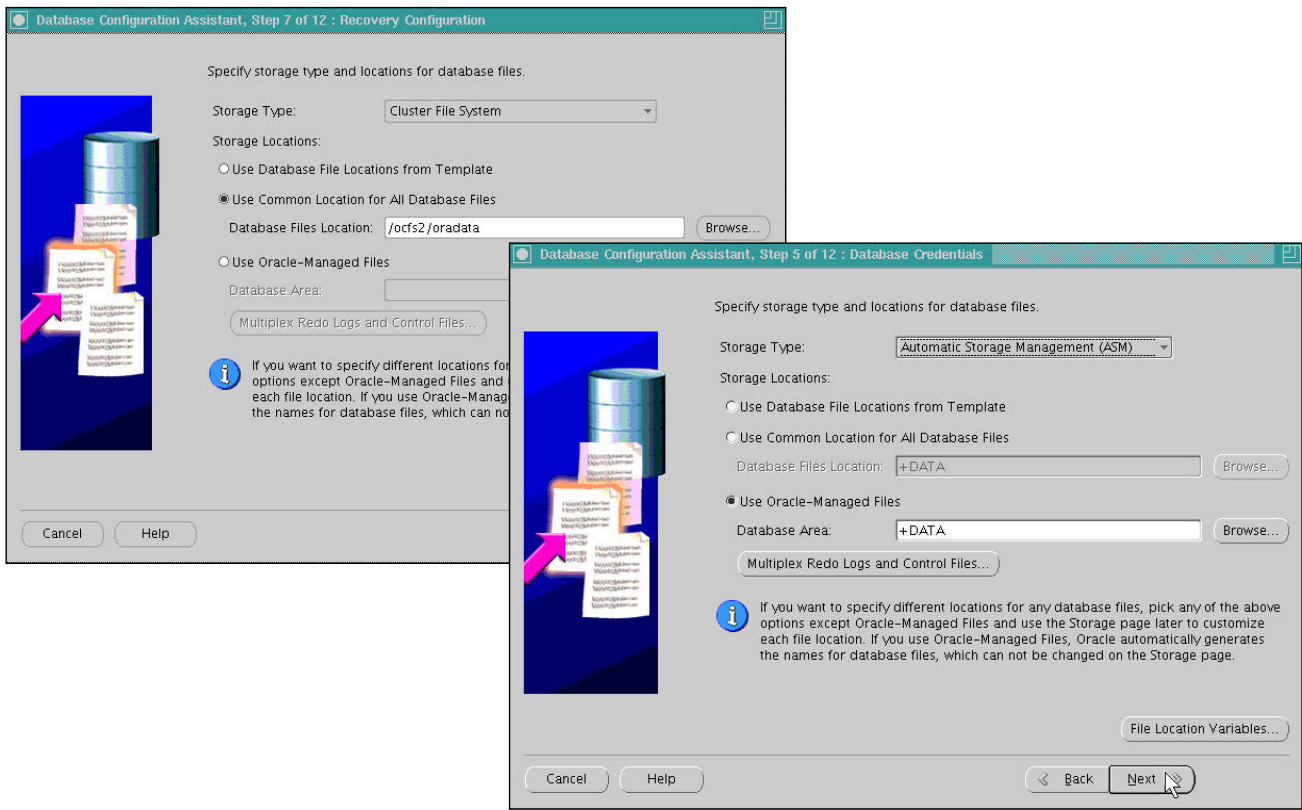
Cancel Help Back Next

Passwords for Database Schema Owners

The Database Credentials window appears next. You must supply passwords for the user accounts created by the DBCA when configuring your database. You can use the same password for all of these privileged accounts by selecting the “Use the Same Administrative Password for All Accounts” option. Enter your password in the Password field, and then enter it again in the Confirm Password field.

Alternatively, you may choose to set different passwords for the privileged users. To do this, select the “Use Different Administrative Passwords” option, enter your password in the Password field, and then enter it again in the Confirm Password field. Repeat this for each user listed in the User Name column. Click the Next button to continue.

Database File Locations



ORACLE

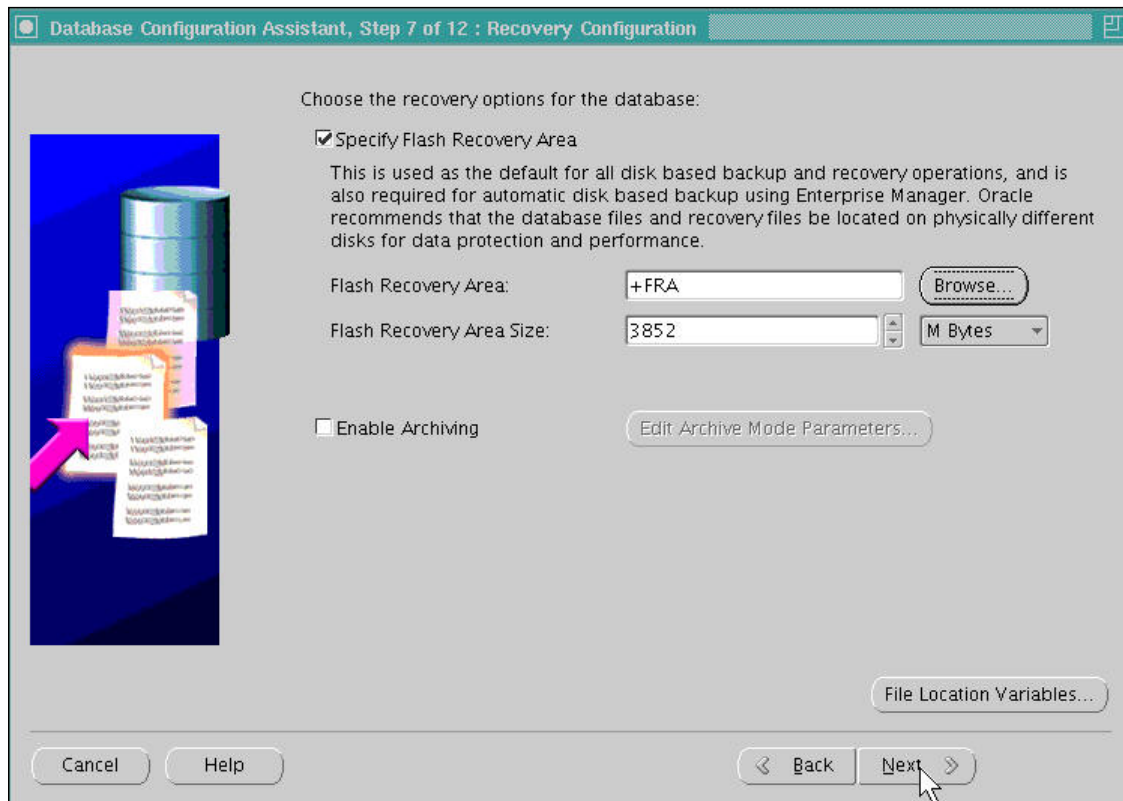
Copyright © 2010, Oracle. All rights reserved.

Database File Locations

In the Database File Locations window, you must indicate where the database files are to be stored. You can choose your storage type from the drop-down list. Detected (and supported) shared storage types are available here. You can choose to use a standard template for file locations, one common location, or Oracle Managed Files (OMF). This cluster database uses ASM and Oracle Managed Files. Therefore, select the Use Oracle-Managed Files option, and enter the disk group name in the Database Area field. Alternatively, you can click the Browse button to indicate the location where the database files are to be created. When you have made your choices, click the Next button to continue.

Note: Step numbers in the title bar can be different, depending on the choices made during the installation.

Recovery Configuration



Database Configuration Assistant, Step 7 of 12 : Recovery Configuration

Choose the recovery options for the database:

☒ Specify Flash Recovery Area

This is used as the default for all disk based backup and recovery operations, and is also required for automatic disk based backup using Enterprise Manager. Oracle recommends that the database files and recovery files be located on physically different disks for data protection and performance.

Flash Recovery Area:

Flash Recovery Area Size:

☐ Enable Archiving

ORACLE

Copyright © 2010, Oracle. All rights reserved.

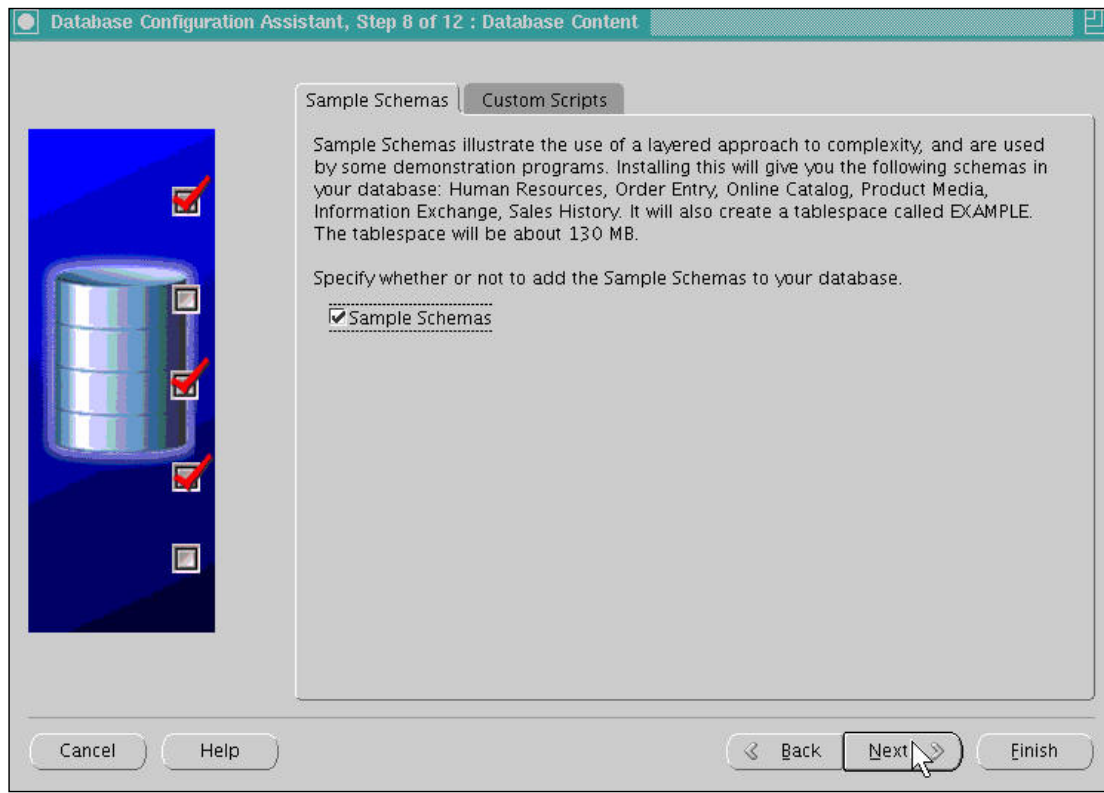
Recovery Configuration

In the Recovery Configuration window, you can select redo log archiving by selecting Enable Archiving. If you are using ASM or cluster file system storages, you can also select the Flash Recovery Area size in the Recovery Configuration window. The size of the area defaults to 2048 megabytes, but you can change this figure if it is not suitable for your requirements. If you are using ASM and a single disk group, the flash recovery area defaults to the ASM Disk Group. If more than one disk group has been created, you can specify it here. If you use a cluster file system, the flash recovery area defaults to `$ORACLE_BASE/flash_recovery_area`. You may also define your own variables for the file locations if you plan to use the Database Storage window to define individual file locations. Select the Enable Archiving check box to enable archiving immediately for the new cluster database.

Note: Flash Recovery Area has been renamed Fast Recovery Area in Oracle Database 11g Release 2 (11.2); however, some of the tools and utilities continue to use the previous name.

When you have completed your entries, click Next, and the Database Content window is displayed.

Database Content



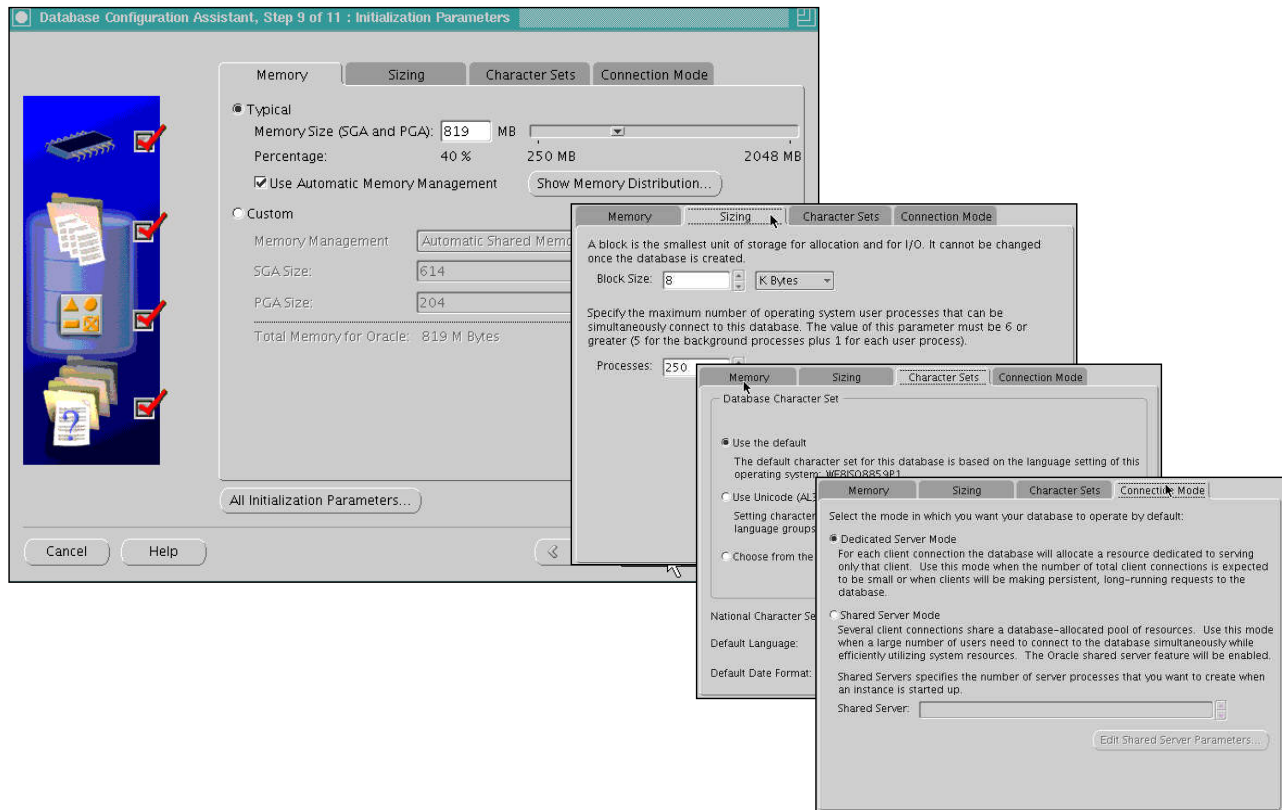
ORACLE

Copyright © 2010, Oracle. All rights reserved.

Database Content

In the Database Content window, you can choose to install the Sample Schemas included with the database distribution. On the Custom Scripts tabbed page, you can choose to run your own scripts as part of the database creation process. When you have finished, click the Next button to continue to the next window.

Initialization Parameters



Copyright © 2010, Oracle. All rights reserved.

Initialization Parameters

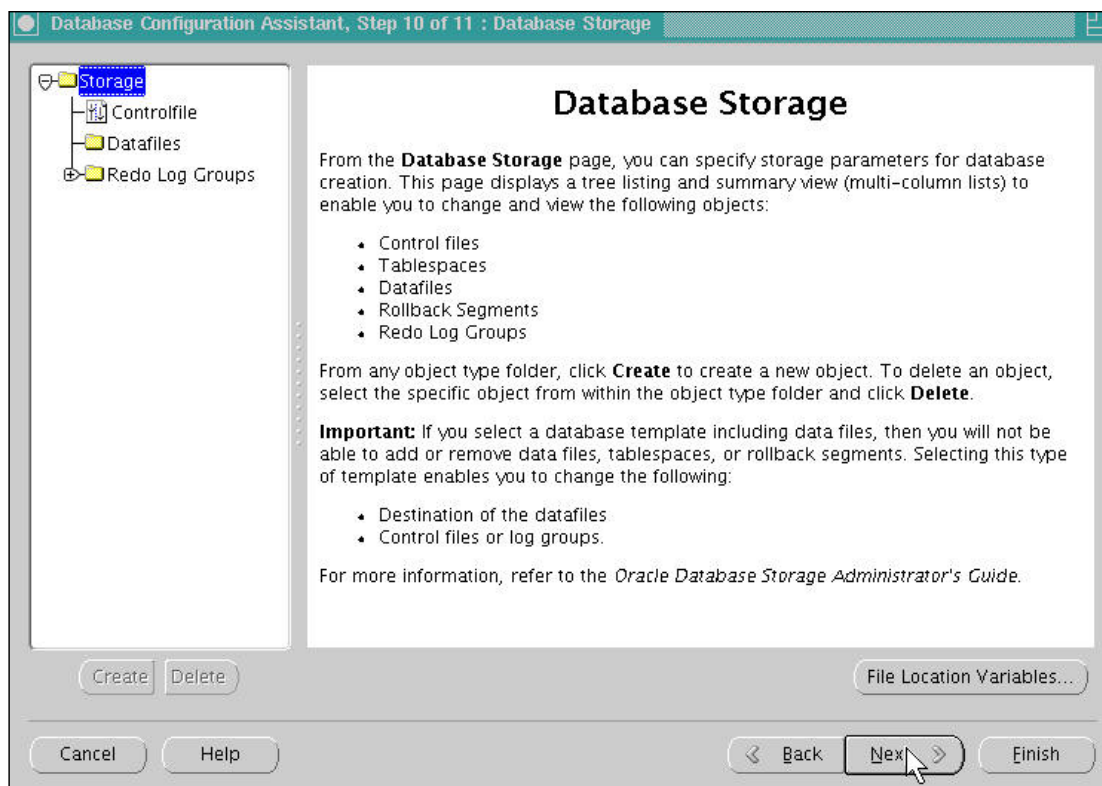
In the Initialization Parameters window, you can set important database parameters. The parameters are grouped under four tabs:

- Memory
- Sizing
- Character Sets
- Connection Mode

On the Memory tabbed page, you can set parameters that deal with memory allocation, including shared pool, buffer cache, Java pool, large pool, and PGA size. Automatic Memory Management is the preferred memory management method and can be selected here. On the Sizing tabbed page, you can adjust the database block size. Note that the default is 8 kilobytes. In addition, you can set the number of processes that can connect simultaneously to the database.

By clicking the Character Sets tab, you can change the database character set. You can also select the default language and the date format. On the Connection Mode tabbed page, you can choose the connection type that clients use to connect to the database. The default type is Dedicated Server Mode. If you want to use Oracle Shared Server, click the Shared Server Mode button. If you want to review the parameters that are not found on the four tabs, click the All Initialization Parameters button. Click the Use Automatic Memory Management button and click the Next button to continue.

Database Storage Options



ORACLE

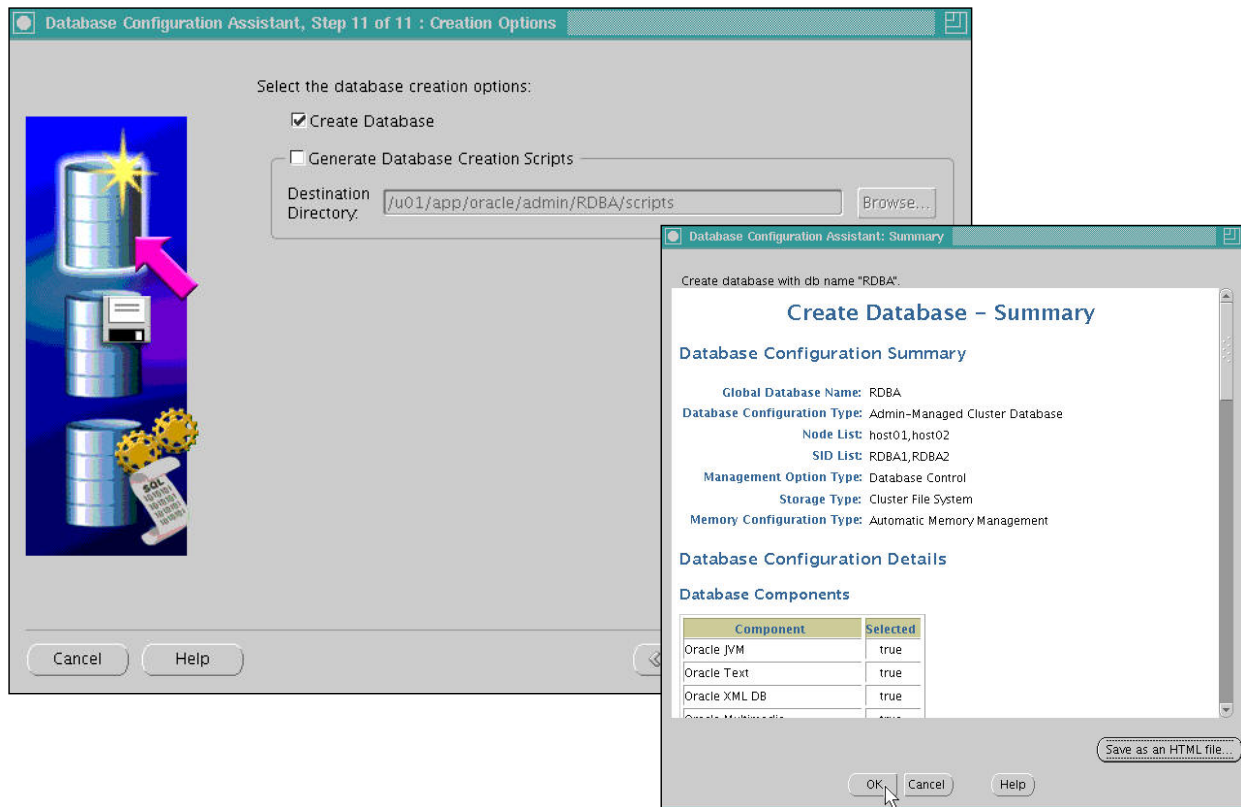
Copyright © 2010, Oracle. All rights reserved.

Database Storage Options

The Database Storage window provides full control over all aspects of database storage, including tablespaces, data files, and log members. Size, location, and all aspects of extent management are under your control here.

When you have finished, click the Next button to continue to the next page.

Create the Database



ORACLE

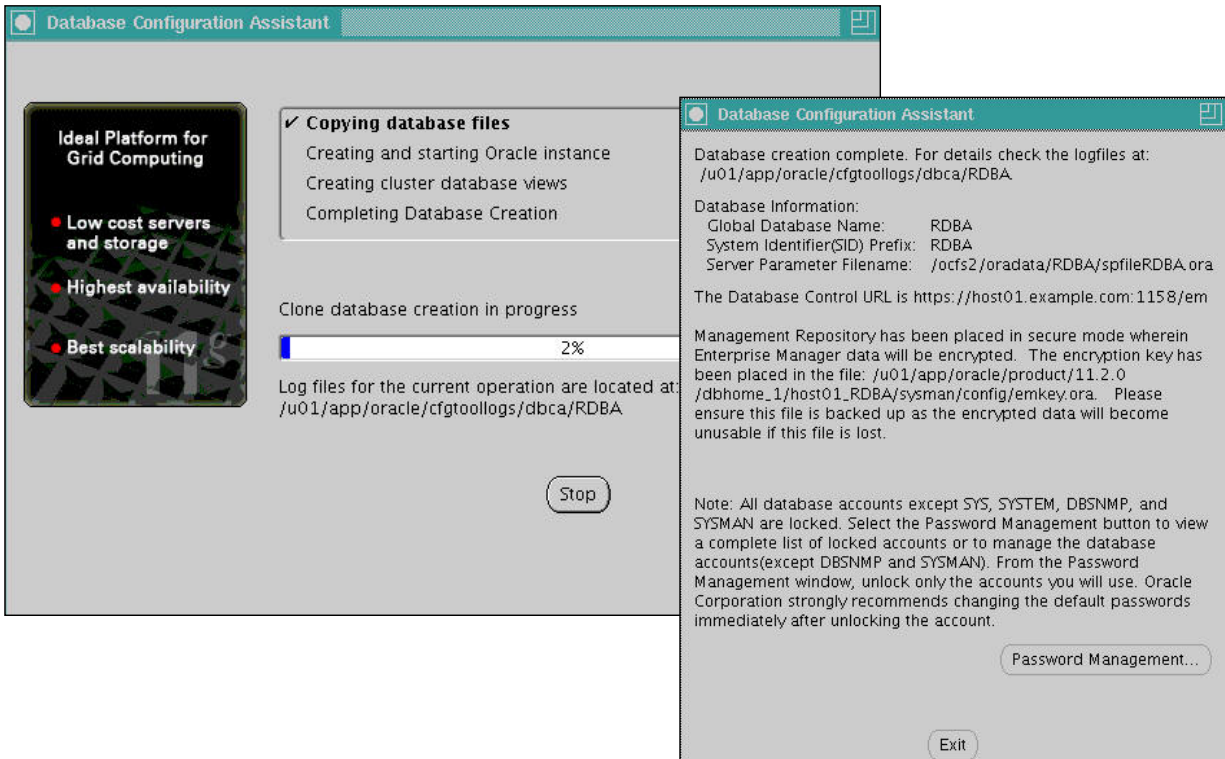
Copyright © 2010, Oracle. All rights reserved.

Create the Database

The Creation Options window appears. You can choose to create the database, or save your DBCA session as a database creation script by selecting the corresponding check box. Select the Create Database check box, and then click the Finish button. The DBCA displays the Summary screen, giving you a last chance to review all options, parameters, and so on that have been chosen for your database creation.

Review the summary data. When you are ready to proceed, close the Summary window by clicking the OK button.

Monitor Progress



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Monitor Progress

The Progress Monitor window appears next. In addition to informing you about how fast the database creation is taking place, it also informs you about the specific tasks being performed by the DBCA in real time. When the database creation progress reaches 100 percent, the DBCA displays a dialog box announcing the completion of the creation process. It also directs you to the installation log file location, parameter file location, and Enterprise Manager URL. By clicking the Password Management button, you can manage the database accounts created by the DBCA.

Postinstallation Tasks

- Back up the `root.sh` script.
- Download and install the required patch updates.
- Verify the cluster database configuration.

```
$ srvctl config database -d orcl
Database unique name: ORCL
Database name: ORCL
Oracle home: /u01/app/oracle/product/11.2.0/dbhome_1
Oracle user: oracle
Spfile: +DATA/orcl/spfileorcl.ora
Domain: example.com
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Server pools: orcl
Database instances: orcl1,orcl2, orcl3
Disk Groups: DATA,FRA
Services:
Database is administrator managed
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Postinstallation Tasks

After the cluster database has been successfully created, run the following command to verify the Oracle Cluster Registry configuration in your newly installed RAC environment:

```
$ srvctl config database -d db_name
```

It is also recommended that you back up the `root.sh` script after you complete an installation.

```
$ cd $ORACLE_HOME
```

```
$ cp root.sh root.sh.bak
```

If you install other products in the same Oracle Home directory, the OUI updates the contents of the existing `root.sh` script during the installation. If you require information contained in the original `root.sh` script, you can recover it from the `root.sh` file copy.

Check Managed Targets

<https://host01.example.com:1158/em>

Cluster: cluster01		Latest Data Collected From Target Jul 31, 2009 5:59:05 PM EDT Refresh						
Home	Performance	Targets	Administration	Interconnects	Topology			
Name ▾	Host	Oracle Home	Availability	Alerts	Policy Violations	Compliance Score (%)	Type	Last Load Time
racnode02.example.com	racnode02.example.com	Not Applicable		0 1	5 0 0	82	Host	Jul 23, 2009 7:57:19 PM
racnode01.example.com	racnode01.example.com	Not Applicable		3 4	5 0 0	82	Host	Jul 31, 2009 5:59:05 PM
RDBA_RDBA2	racnode02.example.com	/u01/app/oracle/product/11.2.0/dbhome_1		0 1	1 49 1	98	Database Instance	Jul 23, 2009 7:54:11 PM
RDBA_RDBA1	racnode01.example.com	/u01/app/oracle/product/11.2.0/dbhome_1		1 1	0 49 2	99	Database Instance	Jul 31, 2009 5:59:27 PM
RDBA	racnode01.example.com	/u01/app/oracle/product/11.2.0/dbhome_1		1 2	10 98 3	96	Cluster Database	Jul 31, 2009 5:58:13 PM
LISTENER racnode02.example.com	racnode02.example.com	/u01/ogi		0 0	8 5 0	83	Listener	Jul 23, 2009 7:56:49 PM
LISTENER racnode01.example.com	racnode01.example.com	/u01/ogi		0 0	9 5 0	83	Listener	Jul 31, 2009 5:58:59 PM
LISTENER SCAN3 cluster01	racnode01.example.com	/u01/ogi		2 0	9 4 0	84	Listener	Jul 31, 2009 5:58:56 PM
LISTENER SCAN2 cluster01	racnode02.example.com	/u01/ogi		1 0	8 4 0	84	Listener	Jul 23, 2009 7:56:46 PM
LISTENER SCAN1 cluster01	racnode01.example.com	/u01/ogi		1 0	9 4 0	84	Listener	Jul 31, 2009 5:58:53 PM
+ASM2 racnode02.example.com	racnode02.example.com	/u01/ogi		1 0	0 0 0		Automatic Storage Management	Jul 23, 2009 7:56:50 PM
+ASM1 racnode01.example.com	racnode01.example.com	/u01/ogi		0 0	0 0 0		Automatic Storage Management	Jul 31, 2009 5:58:58 PM

TIP * indicates the host that is currently elected to monitor the metrics for Cluster Database or Cluster target shown in the table.
TIP ** indicates targets marked for deletion. The corresponding crs resource was deleted and user can permanently delete the target.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Check Managed Targets

Another postinstallation task that you should perform if you are using Enterprise Manager Database Control or Grid Control is to check whether all the managed nodes and their managed resources are properly registered and available. Open a browser and enter the address for your Database Control console. Click the Targets tab to verify that all the targets appear here.

Note: If Enterprise Manager Database Control has not been started on your node, you can start it with the following commands logged in as the `oracle` user, or the account that owns the database software installation:

```
$ export ORACLE_UNQNAME=orcl
$ emctl start dbconsole
```

Background Processes Specific to Oracle RAC

- **ACMS**: Atomic Control file to Memory Service
- **GTX [0 - j]**: Global Transaction Process
- **LMON**: Global Enqueue Service Monitor
- **LMD**: Global Enqueue Service Daemon
- **LMS**: Global Cache Service Process
- **LCK0**: Instance Enqueue Process
- **RMS_n**: Oracle RAC Management Processes
- **RSMN**: Remote Slave Monitor

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Background Processes Specific to Oracle RAC

An Oracle RAC database has the same processes and memory structures as a single-instance Oracle database and additional process and memory structures that are specific to Oracle RAC. The global cache service and global enqueue service processes, and the global resource directory (GRD) collaborate to enable cache fusion. The Oracle RAC processes and their identifiers are as follows:

- **Atomic Control file to Memory Service (ACMS)**: In a RAC environment, the ACMS per-instance process is an agent that contributes to ensuring a distributed SGA memory update is either globally committed on success or globally aborted if a failure occurs.
- **Global Transaction Process (GTX [0 - j])**: The GTX [0 - j] processes provide transparent support for XA global transactions in a RAC environment. The database automatically tunes the number of these processes based on the workload of XA global transactions.
- **Global Enqueue Service Monitor (LMON)**: The LMON process monitors global enqueues and resources across the cluster and performs global enqueue recovery operations.
- **Global Enqueue Service Daemon (LMD)**: The LMD process manages incoming remote resource requests within each instance.
- **Global Cache Service Process (LMS)**: The LMS process maintains records of the data file statuses and each cached block by recording information in the GRD.

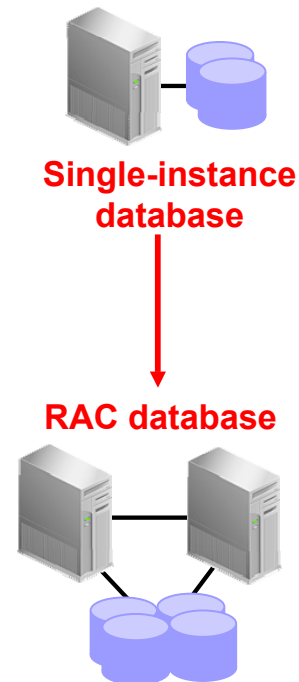
Background Processes Specific to Oracle RAC (continued)

The LMS process also controls the flow of messages to remote instances and manages global data block access and transmits block images between the buffer caches of different instances. This processing is part of the cache fusion feature.

- **LCK0: Instance Enqueue Process** The LCK0 process manages noncache fusion resource requests such as library and row cache requests.
- **RMSn: Oracle RAC Management Processes** The RMSn processes perform manageability tasks for Oracle RAC. Tasks accomplished by an RMSn process include creation of resources related to Oracle RAC when new instances are added to the clusters.
- **RSMN: Remote Slave Monitor** The RSMN process manages background slave process creation and communication on remote instances. These background slave processes perform tasks on behalf of a coordinating process running in another instance.

Single Instance to RAC Conversion

- Single-instance databases can be converted to RAC using:
 - DBCA
 - Enterprise Manager
 - RCONFIG utility
- DBCA automates most of the conversion tasks.
- Before conversion, ensure that:
 - Your hardware and operating system are supported
 - Your cluster nodes have access to shared storage



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Single Instance to RAC Conversion

You can use the Database Configuration Assistant (DBCA) to convert single-instance Oracle databases to RAC. The DBCA automates the configuration of the control file attributes, creates the undo tablespaces and the redo logs, and makes the initialization parameter file entries for cluster-enabled environments. It also configures Oracle Net Services, Oracle Clusterware resources, and the configuration for RAC database management for use by Oracle Enterprise Manager or the `srvctl` utility.

Before you use the DBCA to convert a single-instance database to a RAC database, ensure that your system meets the conditions:

- It is a supported hardware and operating system configuration.
- It has shared storage. A supported Cluster File System, NFS mount, or ASM is available and accessible from all nodes.
- Your applications have no design characteristics that preclude their use with cluster database processing.

You can also use Enterprise Manager and the `rconfig` utility to perform the single instance to RAC conversion.

Issues for Converting Single Instance Databases to Oracle RAC

- Backup procedures should be available before conversion takes place.
- Archiving in Oracle RAC environments requires a thread number in the archive file format.
- The archived logs from all instances of an Oracle RAC database are required for media recovery.
- By default, all database files are migrated to Oracle Managed Files (OMF).

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Issues for Converting Single Instance Databases to Oracle RAC

Whatever method you choose to use for the conversion, note the following administrative considerations before converting single-instance databases to Oracle RAC:

- Backup procedures should be available before converting from a single-instance Oracle Database to Oracle RAC. For archiving with Oracle RAC environments, the archive file format requires a thread number.
- The archived logs from all instances of an Oracle RAC database are required for media recovery. Because of this, if you archive to a file and you do not use a cluster file system, or some other means to provide shared file systems, then you require a method of accessing the archive logs from all nodes on which the cluster database has instances.
- By default, all database files are migrated to Oracle Managed Files (OMF). This feature simplifies table space creation, ensures data file location consistency and compliance with OFA rules, and reduces human error with data file management.

Single-Instance Conversion Using the DBCA

Conversion steps for a single-instance database on *nonclustered* hardware:

1. Back up the original single-instance database.
2. Complete the Oracle Clusterware installation.
3. Validate the cluster.
4. Copy the preconfigured database image.
5. Install the Oracle Database 11g software with RAC.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

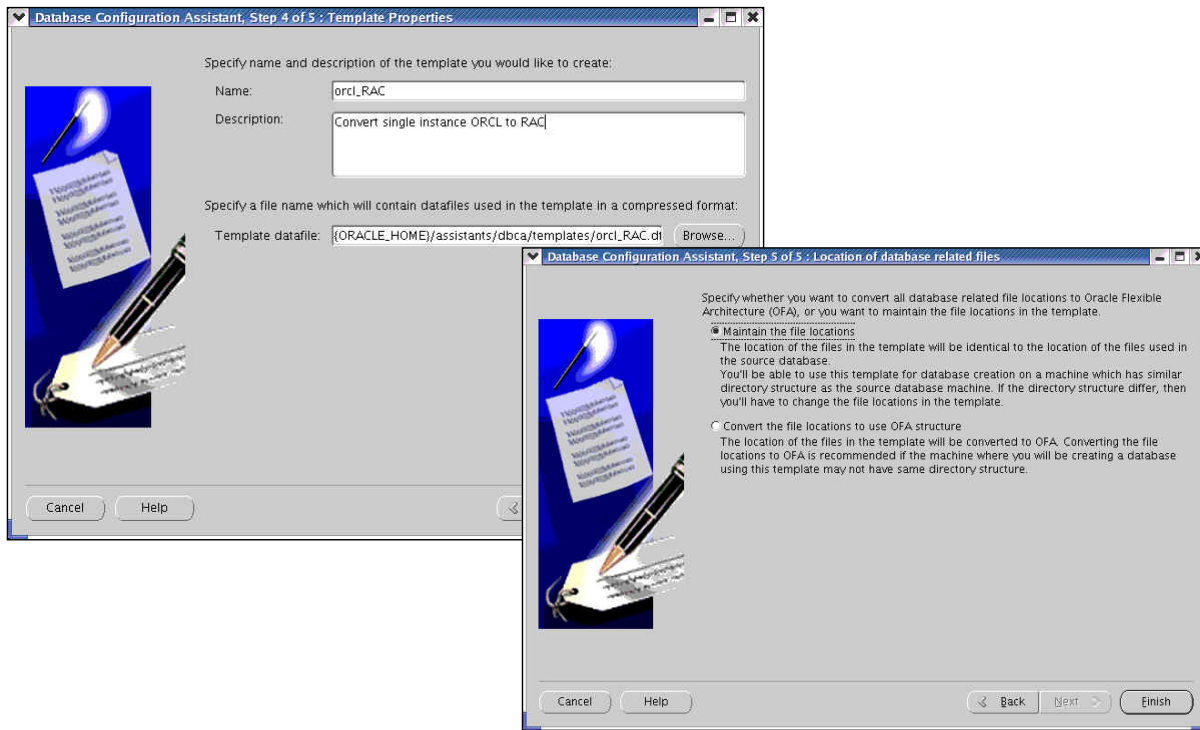
Single-Instance Conversion Using DBCA

To convert from a single-instance Oracle database that is on a noncluster computer to a RAC database, perform the following steps:

1. Back up the original single-instance database.
2. Complete the Oracle Clusterware installation.
3. Validate the cluster.
4. Copy the preconfigured database image.
5. Install the Oracle Database 11g software with RAC.

Conversion Steps

1. Back up the original single-instance database.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Conversion Steps

1. Back Up the Original Single-Instance Database.

Use the DBCA to create a preconfigured image of your single-instance database by using the following procedure:

1. Navigate to the `bin` directory in `$ORACLE_HOME`, and start the DBCA.
2. In the Welcome window, click Next.
3. In the Operations window, select Manage Templates, and click Next.
4. In the Template Management window, select "Create a database" template and "From an existing database (structure as well as data)," and click Next.
5. In the Source Database window, enter the database name in the Database instance field, and click Next.
6. In the Template Properties window, enter a template name in the Name field. By default, the template files are generated in `$ORACLE_HOME/assistants/dbca/templates`. Enter a description of the file in the Description field, and change the template file location in the Template data file field if you want. When you have finished, click Next.
7. In the Location of Database Related Files window, select "Maintain the file locations," so that you can restore the database to the current directory structure, and click Finish. The DBCA generates two files: a database structure file (`template_name.ct1`) and a database preconfigured image file (`template_name.dfb`).

Conversion Steps

2. Perform the preinstallation steps.
 - Tasks include kernel parameter configuration, hardware setup, network configuration, and shared storage setup
3. Set up and validate the cluster.
 - Create a cluster with the required number of nodes according to your hardware vendor's documentation.
 - Validate cluster components before installation.
 - Install Oracle Clusterware.
 - Validate the completed cluster installation by using `cluvfy`.
4. Copy the preconfigured database image.
 - The database structure *.dbc file
 - The preconfigured database image *.dfb file
5. Install the Oracle Database 11g Release 2 software with RAC.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Conversion Steps (continued)

2. Perform the Preinstallation Steps.

Complete the Oracle Clusterware installation, as described in the lesson titled “Grid Infrastructure Installation” or refer to the *Oracle Grid Infrastructure Installation Guide* for your platform.

3. Set Up and Validate the Cluster.

Form a cluster with the required number of nodes according to your business needs. When you have configured all the nodes in your cluster, validate cluster components by using the Cluster Verification Utility, and then install Oracle Clusterware. When the clusterware is installed, validate the completed cluster installation and configuration using the Cluster Verification Utility.

4. Copy the Preconfigured Database Image.

This includes copying the database structure *.dbc file and the database preconfigured image *.dfb file that the DBCA created in step one (Back Up the Original Single-Instance Database) to a temporary location on the node in the cluster from which you plan to run the DBCA.

Conversion Steps (continued)

5. Install the Oracle Database 11g Release 2 software with RAC.

1. Run the OUI to perform an Oracle database installation with RAC. Select Cluster Installation Mode and select the nodes to include in your RAC database.
2. In the OUI Database Configuration Types window, select “Advanced install.” After installing the software, the OUI runs postinstallation tools such as NETCA, DBCA, and so on.
3. In the DBCA Template Selection window, use the template that you copied to a temporary location in the “Copy the Preconfigured Database Image” step. Use the browse option to select the template location.
4. After creating the RAC database, the DBCA displays the Password Management window in which you must change the passwords for database privileged users. When the DBCA exits, the conversion is complete.

Single-Instance Conversion Using `rconfig`

1. Edit the `ConvertToRAC.xml` file located in the `$ORACLE_HOME/assistants/rconfig/sampleXMLs` directory.
2. Modify the `ConvertToRAC_AdminManaged.xml` or `ConvertToRAC_PolicyManaged.xml` file as required for your system.
3. Save the file under a different name.

```
$ cd $ORACLE_HOME/assistants/rconfig/sampleXMLs
$ vi ConvertToRAC.xml
$ rconfig my_rac_conversion.xml
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Single-Instance Conversion Using `rconfig`

You can use the `rconfig` command-line utility to convert a single-instance database to RAC. To use this feature, perform the following steps:

1. Go to the `$ORACLE_HOME/assistants/rconfig/sampleXMLs` directory as the `oracle` user and open the `ConvertToRAC_AdminManaged.xml` or the `ConvertToRAC_PolicyManaged.xml` file (depending on your desired management style) using a text editor, such as `vi`.
2. Review the XML file, and modify the parameters as required for your system. The XML sample file contains comment lines that provide instructions about how to configure the file. When you have completed making changes, save the file with the syntax `filename.xml`. Make a note of the name you select.
3. Assuming that you save your XML file as `my_rac_conversion.xml`, navigate to the `$ORACLE_HOME/bin` directory, and use the following syntax to run the `rconfig` command:
`$./rconfig my_rac_conversion.xml`

Note: The `Convert verify` option in the `xml` file has three options:

- `Convert verify="YES":` `rconfig` performs checks to ensure that the prerequisites for single-instance to RAC conversion have been met before it starts conversion.

Single-Instance Conversion Using `rconfig` (continued)

- Convert `verify="NO"`: `rconfig` does not perform prerequisite checks, and starts conversion.
- Convert `verify="ONLY"`: `rconfig` performs only prerequisite checks; it does not start conversion after completing prerequisite checks.

Quiz

The RAC database software installation is initiated by executing `runInstaller` from the root directory of the Oracle Database 11g Release 2 CD-ROM or from the software staging location.

1. True
2. False

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Answer: 1

The statement is true.

Quiz

A single instance database can be converted to a RAC database by using (choose the correct options):

- 1. rconfig
- 2. netca
- 3. dbca

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Answer: 1, 3

Choices a and c are correct.

Summary

In this lesson, you should have learned how to:

- Install the Oracle database software
- Create a cluster database
- Perform post-database-creation tasks
- Perform a single instance to RAC conversion

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Practice 11 Overview

This practice covers the following topics:

- Installing the Oracle database software
- Creating a RAC database

ORACLE

Copyright © 2010, Oracle. All rights reserved.

12

Oracle RAC Administration

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Objectives

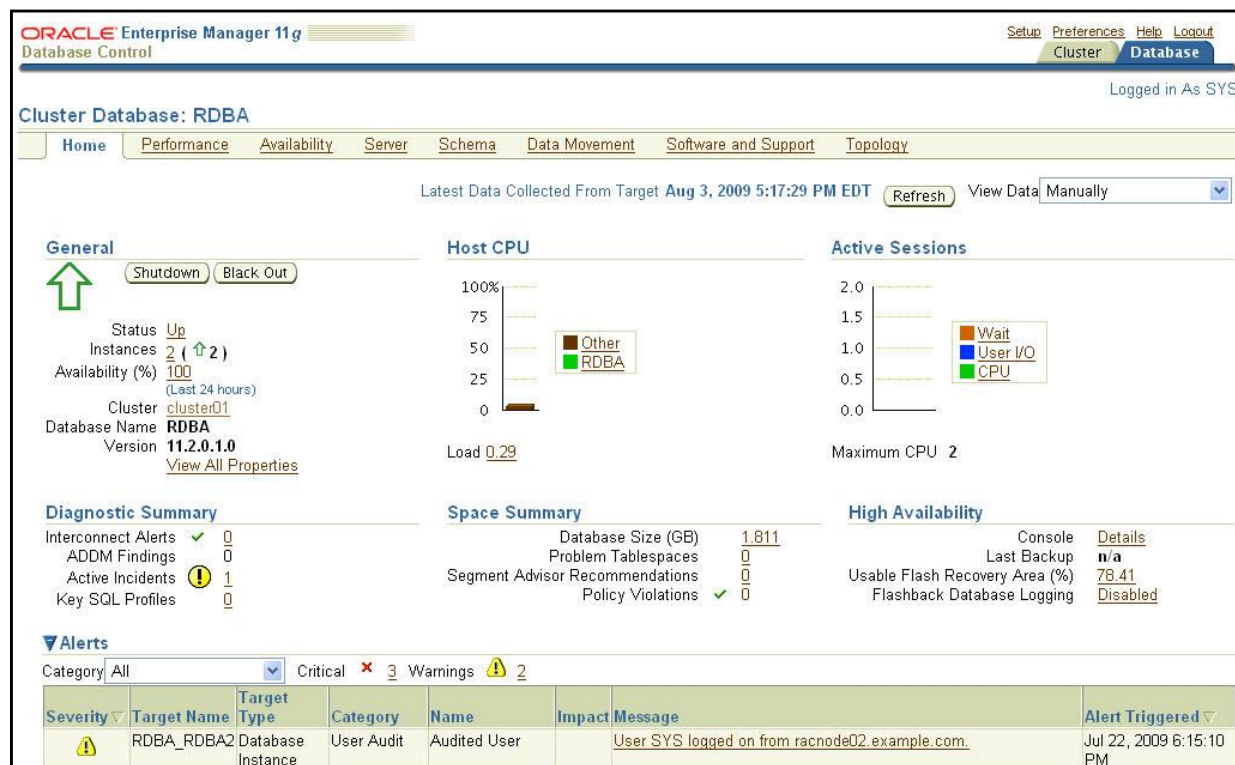
After completing this lesson, you should be able to:

- Use Enterprise Manager Cluster Database pages
- Define redo log files in a RAC environment
- Define undo tablespaces in a RAC environment
- Start and stop RAC databases and instances
- Modify initialization parameters in a RAC environment

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Cluster Database Home Page



Cluster Database Home Page

The Cluster Database Home page serves as a crossroad for managing and monitoring all aspects of your RAC database. From this page, you can access the other main cluster database tabs: Performance, Availability, Server, Schema, Data Movement, Software and Support, and Topology.

On this page, you find General, High Availability, Space Summary, and Diagnostic Summary sections for information that pertains to your cluster database as a whole. The number of instances is displayed for the RAC database, in addition to the status. A RAC database is considered to be up if at least one instance has the database open. You can access the Cluster Home page by clicking the Cluster link in the General section of the page.

Other items of interest include the date of the last RMAN backup, archiving information, space utilization, and an alert summary. By clicking the link next to the Flashback Database Logging label, you can open the Recovery Settings page from where you can change various recovery parameters.

The Alerts table shows all the recent alerts that are open. Click the alert message in the Message column for more information about the alert. When an alert is triggered, the name of the metric for which the alert was triggered is displayed in the Name column.

Cluster Database Home Page

Related Alerts
Critical ✖ 13 Warnings ⚠ 10

Policy Violations
All 10 Critical Rules Violated 9 Critical Security Patches 0 Compliance Score (%) 93

Security
Last Security Evaluation 📅 Aug 2, 2009 5:40:08 PM EDT Compliance Score (%) 90 [Enterprise Security At a Glance](#)

Job Activity
Create Job OS Command ▼ Go
Job executions scheduled to start no more than 7 days ago

Status	Submitted to the Cluster Database	Submitted to any member
Scheduled	0	0
Running	0	0
Suspended	0	0
Problem	0	0

Critical Patch Advisories for Oracle Homes
Patch Advisories 0
⚠ Patch Advisory information may be stale. Oracle MetaLink credentials are not configured.
Affected Oracle Homes 0
Oracle MetaLink Credentials Not Configured

[Home](#) [Performance](#) [Availability](#) [Server](#) [Schema](#) [Data Movement](#) [Software and Support](#) [Topology](#)

Instances

Name	Status	Host Name	Alerts	Policy Violations	Compliance Score (%)	ASM Instance	ADDM Findings
RDBA1	🔄	racnode01.example.com	3 1	0 49 2	99	+ASM1 racnode01.example.com 🔄 0 0	0
RDBA2	🔄	racnode02.example.com	0 1	1 49 1	98	+ASM2 racnode02.example.com 🔄 1 0	0

Related Links

Access	Add Exadata Cell Targets	Advisor Central
Alert History	All Metrics	Blackouts
Deployments	EM SQL History	Jobs
Metric and Policy Settings	Metric Collection Errors	Monitoring Configuration
Policy Groups	Scheduler Central	SQL Worksheet
Target Properties	User-Defined Metrics	

[Cluster](#) | [Database](#) | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Cluster Database Home Page (continued)

The Related Alerts table provides information about alerts for related targets, such as Listeners and Hosts, and contains details about the message, the time the alert was triggered, the value, and the time the alert was last checked.

The Policy Trend Overview page (accessed by clicking the Compliance Score link) provides a comprehensive view about a group or targets containing other targets with regard to compliance over a period of time. Using the tables and graphs, you can easily watch for trends in progress and changes.

The “Security At a Glance” page shows an overview of the security health of the enterprise for the targets or specific groups. This helps you to focus on security issues by showing statistics about security policy violations and noting critical security patches that have not been applied.

The Job Activity table displays a report of the job executions that shows the scheduled, running, suspended, and problem (stopped/failed) executions for all Enterprise Manager jobs on the cluster database.

The Instances table lists the instances for the cluster database, their availability, alerts, policy violations, performance findings, and related ASM Instance. Click an instance name to go to the Home page for that instance. Click the links in the table to get more information about a particular alert, advice, or metric.

Cluster Database Instance Home Page



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Cluster Database Instance Home Page

The Cluster Database Instance Home page enables you to view the current state of the instance by displaying a series of metrics that portray its overall health. This page provides a launch point for the performance, administration, and maintenance of the instance environment.

You can access the Cluster Database Instance Home page by clicking one of the instance names from the Instances section of the Cluster Database Home page. This page has basically the same sections as the Cluster Database Home page.

The difference is that tasks and monitored activities from these pages apply primarily to a specific instance. For example, clicking the Shutdown button on this page shuts down only this one instance. However, clicking the Shutdown button on the Cluster Database Home page gives you the option of shutting down all or specific instances.

By scrolling down on this page, you see the Alerts, Related Alerts, Policy Violations, Jobs Activity, and Related Links sections. These provide information similar to that provided in the same sections on the Cluster Database Home page.

Cluster Home Page

ORACLE Enterprise Manager 11g Database Control

Cluster: cluster01

Latest Data Collected From Target Aug 4, 2009 6:44:05 AM EDT Refresh

Home Performance Targets Administration Interconnects Topology

View Data Automatically (60 sec)

General

Status Up Startup/Shutdown Black Out

Hosts 2 (1) 1

Availability (%) 100.0 (Last 24 hours)

Cluster Name cluster01

Clusterware Status Up (1) 1

Clusterware Version 11.2.0.1.0

Oracle Home /u01/ogi

Reconfiguration Activities 2 View All Properties

Configuration

View Operating Systems

Operating Systems	Hosts/OS Patches
Enterprise Linux Enterprise Linux Server release 5 (Carthage)	1 Not available
2.6.18 8.1.15.0.1.e5xen	
Enterprise Linux Enterprise Linux Server release 5.2 (Carthage)	1 Not available
2.6.18 8.1.15.0.1.e5xen	

Diagnostic Summary

Interconnect Alerts 0

Resource Summary

Problem Resources 1

Cluster Databases

View Cluster Databases only

Name	Status	Alerts	Policy Violations	Compliance Score (%)	Version
RDBA	1	3 2	10 98 3	96	11.2.0.1.0

Alerts

Category All Critical 8 Warnings 10

Previous 1-10 of 18 Next 8

Severity	Target Name	Target Type	Category	Name	Impact	Message	Alert Triggered
Warning	racnode01.example.com	Host	Clusterware Alert Log	Clusterware Service Alert Log		[crsd(6416)]CRS-1012:The OCR service started on node racnode01. See /u01/ogi/log/racnode01...	Jul 31, 2009 5:41:25 PM

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Cluster Home Page

The slide shows you the Cluster Home page, which can be accessed by clicking the Cluster tab located in the top-right corner of Enterprise Manager. Even if the database is down, the cluster page is available to manage resources. The cluster is represented as a composite target composed of nodes and cluster databases. An overall summary of the cluster is provided here. The Cluster Home page displays several sections, including General, Configuration, Diagnostic Summary, Cluster Databases, Alerts, and Hosts. The General section provides a quick view of the status of the cluster, providing basic information such as current Status, Availability, Up nodes, Clusterware Version, and Oracle Home.

The Configuration section enables you to view the operating systems (including hosts and OS patches) and hardware (including hardware configuration and hosts) for the cluster.

The Cluster Databases table displays the cluster databases (optionally associated with corresponding services) associated with this cluster, their availability, and any alerts on those databases. The Alerts table provides information about any alerts that have been issued along with the severity rating of each.

The Hosts table (not shown in the screenshot) displays the hosts for the cluster, availability, corresponding alerts, CPU and memory utilization percentage, and total input/output (I/O) per second.

Configuration Section

Oracle Enterprise Manager 11g Database Control

Hardware: i686 GenuineIntel i686 in composite target cluster01

Host	Operating System	Hardware Details
racnode01.example.com	Enterprise Linux Enterprise Linux Server release 5.2 (Carthage) 2.6.18-8.1.15.0.1.el5xen	
racnode02.example.com	Enterprise Linux Enterprise Linux Server release 5 (Carthage) 2.6.18-8.1.15.0.1.el5xen	

Search Host Operating System and Hardware Summaries

Oracle Enterprise Manager 11g Database Control

Host: racnode01.example.com >

Hardware Details

Data Collected Aug 3, 2009 5:38:56 PM EDT

Hostname	racnode01.example.com	Local Disk Capacity (GB)	15.43
System Configuration	i686	Clock Frequency (MHz)	
Machine Architecture	GenuineIntel i686	Number of CPUs	1
Hardware Provider	Intel Based Hardware	Number of CPU boards	1
Memory Size (MB)	2048	Number of IO devices	0

CPUs

CPU speed (MHZ)	Vendor	PROM Revision	ECACHE (MB)	CPU Implementation	Mask
2327	GenuineIntel	23	6	Intel(R) Xeon(R) CPU E5410 @ 2.33GHz	6

IO Devices

Name	Vendor	Bus Type	Frequency (MHZ)	PROM Revision
No IO details found.				

Network Interfaces

Name	INET Address	Maximum Transfer Unit	Broadcast Address	Mask	Flags	MAC Address	Hostname Aliases
eth0	192.0.2.252	1500	192.0.2.255	255.255.255.0	BROADCAST,MULTICAST,RUNNING,UP	00:16:3E:33:81:6D	racnode01.example.com, racnode01
eth0.1	192.0.2.227	1500	192.0.2.255	255.255.255.0	BROADCAST,MULTICAST,RUNNING,UP	00:16:3E:33:81:6D	
eth0.2	192.0.2.247	1500	192.0.2.255	255.255.255.0	BROADCAST,MULTICAST,RUNNING,UP	00:16:3E:33:81:6D	
eth1	192.168.1.103	1500	192.168.1.255	255.255.255.0	BROADCAST,MULTICAST,RUNNING,UP	00:16:3E:27:61:BD	racnode01-priv.example.com, racnode01-priv
eth2	10.141.136.230	1500	10.141.136.255	255.255.254.0	BROADCAST,MULTICAST,RUNNING,UP	00:16:3E:2F:34:B8	stcvm03.us.oracle.com, stcvm03
lo	127.0.0.1	16436		255.0.0.0	LOOPBACK,RUNNING,UP		localhost.localdomain, localhost
sit0		1480			NOARP		

TIP Some information may not be available depending upon the Hardware platform.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Configuration Section

The Cluster Home page is invaluable for locating configuration-specific data. Locate the Configuration section on the Cluster Home page. The View drop-down list enables you to inspect hardware and operating system overview information.

Click the Hosts link, and then click the Hardware Details link of the host that you want. On the Hardware Details page, you find detailed information about your CPU, disk controllers, network adapters, and so on. This information can be very useful when determining the Linux patches for your platform.

Click History to access the hardware history information for the host.

Some hardware information is not available, depending on the hardware platform.

Note: The Local Disk Capacity (GB) field shows the disk space that is physically attached (local) to the host. This value does not include disk space that may be available to the host through networked file systems.

Configuration Section

Configuration

View: Operating Systems

Operating Systems	Hosts	Patches
Enterprise Linux Enterprise Linux Server release 5 (Carthage) 2.6.18.8.1.15.0.1.el5xen		
Enterprise Linux Enterprise Linux Server release 5.2 (Carthage) 2.6.18.8.1.15.0.1.el5xen		

Operating System: Enterprise Linux Enterprise Linux Server release 5 (Carthage) 2.6.18.8.1.15.0.1.el5xen in composite target cluster01

Host	Hardware	Operating System Details
racnode02.example.com	i686 GenuineIntel i686	

Operating System Details Data Collected Jul 23, 2009 2:48:31 PM EDT

Host: racnode02.example.com
Operating System: Enterprise Linux Enterprise Linux Server release 5 (Carthage) 2.6.18.8.1.15.0.1.el5xen
Vendor: Red Hat

General | **File Systems** | Packages

Distributor Version: Red Hat 4.1.1-52
Maximum Swap Space (MB): 3007.99

Operating System Properties

Name	Source	Value
vm.dirty_writeback_centisecs	/sbin/sysctl	500
vm.drop_caches	/sbin/sysctl	0
vm.laptop_mode	/sbin/sysctl	0
vm.legacy_va_layout		
vm.lowmem_reserve_ratio		
vm.max_map_count	/dev/mapper/VolGroup00-LogVol00	
vm.min_free_kbytes	/dev/vnda1	
vm.nr_pdflush_threads	/dev/mapper/VolGroup01-LogVol00	
vm.overcommit_memory		
vm.overcommit_ratio	/sbin/sysctl	50
vm.page-cluster	/sbin/sysctl	3
vm.panic_on_oom	/sbin/sysctl	0
vm.percpu_pagelist_fraction	/sbin/sysctl	0
vm.swap_token_timeout	/sbin/sysctl	300
vm.swappiness	/sbin/sysctl	60
vm.vdso_enabled	/sbin/sysctl	1

File Systems

Resource Name	Type	Mount Location	Mount Options
/dev/mapper/VolGroup00-LogVol00	ext3	/	rw
/dev/vnda1	ext3	/boot	rw
/dev/mapper/VolGroup01-LogVol00	ext3	/u01	rw

Related Link: Search File Systems on Hosts

Packages

GConf2 2.14.0-9.el5
MAKEDDEV 3.23-1.2
NetworkManager-glib 1:0.6.4-6.el5
ORBit2 2.14.3-4.el5
OpenIPMI-libs 2.0.6-5.el5.3
PyXML 0.8.4-4
a2ps 4.13b-57.1.el5

Previous 1-25 of 610 Next 25

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Configuration Section (continued)

The Operating System Details General page displays the operating system details for a host, including:

- General information, such as the distributor version and the maximum swap space of the operating system
- Information about operating system properties

The Source column displays where Enterprise Manager obtained the value for each operating system property.

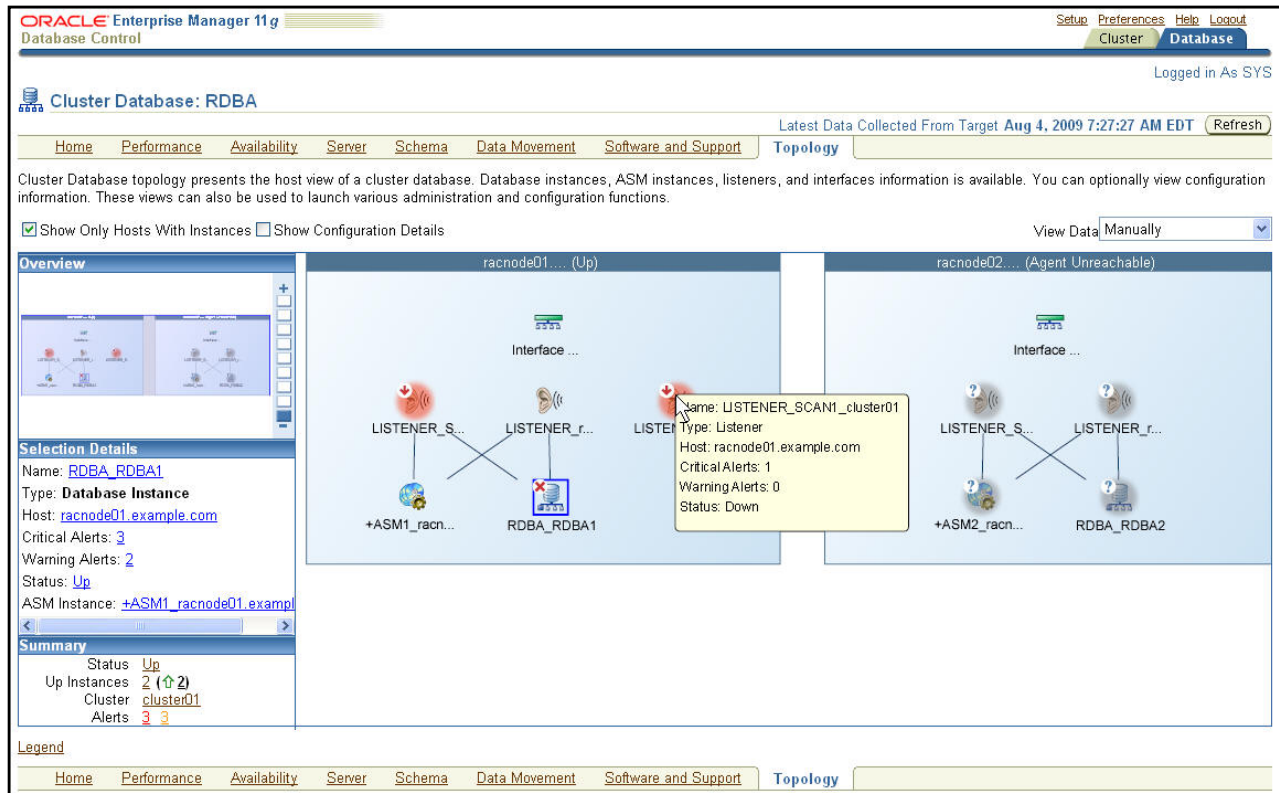
To see a list of changes to the operating system properties, click History.

The Operating System Details File Systems page displays information about one or more file systems for the selected hosts:

- Name of the file system on the host
- Type of mounted file system—for example, ufs or nfs
- Directory where the file system is mounted
- The mount options for the file system—for example, ro, nosuid, or nobrowse

The Operating System Details Packages page displays information about the operating system packages that have been installed on a host.

Topology Viewer



Copyright © 2010, Oracle. All rights reserved.

Topology Viewer

The Oracle Enterprise Manager Topology Viewer enables you to visually see the relationships between target types for each host of your cluster database. You can zoom in or out, pan, and see selection details. These views can also be used to launch various administration functions.

The Topology Viewer populates icons on the basis of your system configuration. If a listener is serving an instance, a line connects the listener icon and the instance icon. Possible target types are:

- Interface
- Listener
- ASM Instance
- Database Instance

If the Show Configuration Details option is not selected, the topology shows the monitoring view of the environment, which includes general information such as alerts and overall status. If you select the Show Configuration Details option, additional details are shown in the Selection Details window, which are valid for any topology view. For instance, the Listener component would also show the machine name and port number.

You can click an icon and then right-click to display a menu of available actions.

Enterprise Manager Alerts and RAC

	RDBA_RDBA1	Database Instance	Waits by Wait Class	Database Time Spent Waiting (%)	Metrics "Database Time Spent Waiting (%)" is at 86.667 for event class "Other"	Aug 4, 2009 7:18:33 AM
	RDBA_RDBA1	Database Instance	User Audit	Audited User	User SYS logged on from racnode01.example.com.	Aug 4, 2009 6:54:08 AM
	RDBA_RDBA2	Database Instance	User Audit	Audited User	User SYS logged on from racnode02.example.com.	Jul 22, 2009 6:15:10 PM

Related Alerts
Critical 13 Warnings 10

Policy Violations
All 10 Critical Rules Violated 9 Critical Security Patches 0 Compliance Score (%) 93

Security
Last Security Evaluation Aug 3, 2009 5:40:59 PM EDT Compliance Score (%) 90 [Enterprise Security At a Glance](#)

Job Activity
Create Job
Job executions scheduled to start no more than 7 days ago

Status	Submitted to the Cluster Database	Submitted to any member
Scheduled	0	0
Running	0	0
Suspended	0	0
Problem	0	0

Critical Patch Advisories for Oracle Homes
Patch Advisories 0
 Patch Advisory information may be stale. Oracle MetaLink credentials are not configured.
Affected Oracle Homes 0
Oracle MetaLink Credentials [Not Configured](#)

[Home](#) [Performance](#) [Availability](#) [Server](#) [Schema](#) [Data Movement](#) [Software and Support](#) [Topology](#)

Instances

Name	Status	Host Name	Alerts	Policy Violations	Compliance Score (%)	ASM Instance	ADDM Findings
RDBA_RDBA1		racnode01.example.com	3 2	0 49 2	99	+ASM1 racnode01.example.com 0 0	0
RDBA_RDBA2		racnode02.example.com	0 1	1 49 1	98	+ASM2 racnode02.example.com 1 0	0

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Enterprise Manager Alerts and RAC

You can use Enterprise Manager to administer alerts for RAC environments. Enterprise Manager distinguishes between database- and instance-level alerts in RAC environments.

Enterprise Manager also responds to metrics from across the entire RAC database and publishes alerts when thresholds are exceeded. Enterprise Manager interprets both predefined and customized metrics. You can also copy customized metrics from one cluster database instance to another, or from one RAC database to another. A recent alert summary can be found on the Database Control Home page. Notice that alerts are sorted by relative time and target name.

Enterprise Manager Metrics and RAC

Cluster Database: RDBA >

Metric and Policy Settings

Database Instance: RDBA_RDBA1 >

Metric and Policy Settings

Metric Thresholds Policies

View Metrics with thresholds

Metric	Comparison Operator	Warning Threshold	Critical Threshold	Corrective Actions	Collection Schedule	Edit
Access Violation	Matches		*	None	Every 5 Minutes	
Access Violation Status	>	0	0	None	Every 5 Minutes	
Archive Area Used (%)	>	80		None	Every 15 Minutes	
Archiver Hung	Matches		*	None	Every 5 Minutes	
Archiver Hung Status	>		0	None	Every 5 Minutes	
Audited User	=	SYS		None	Every 15 Minutes	
Average Users Waiting Count						
Owner's Invalid Object Count	>	0		None	Every 24 Hours	
Segments Approaching Maximum Extents Count	>	0		None	Every 24 Hours	
Segments Not Able to Extend Count	>	0		None	Every 24 Hours	
Service Status	=		Down	None	Every 5 Minutes	
Status			Down	None	Event-Driven	
Streams Apply - (%)Messages in Waiting State	>	75	90	None	Every 5 Minutes	
Streams Apply - (%)Spilled Messages	>	60	80	None	Every 5 Minutes	
Streams Capture - (%)Spilled Messages	>	60	80	None	Every 5 Minutes	
Streams Process Errors	>	0		None	Every 5 Minutes	
Streams Process Status	=	DISABLI	ABORTE	None	Every 5 Minutes	
Streams Prop - (%)Messages in Waiting State	>	75	90	None	Every 5 Minutes	
Tablespace Space Used (%)						

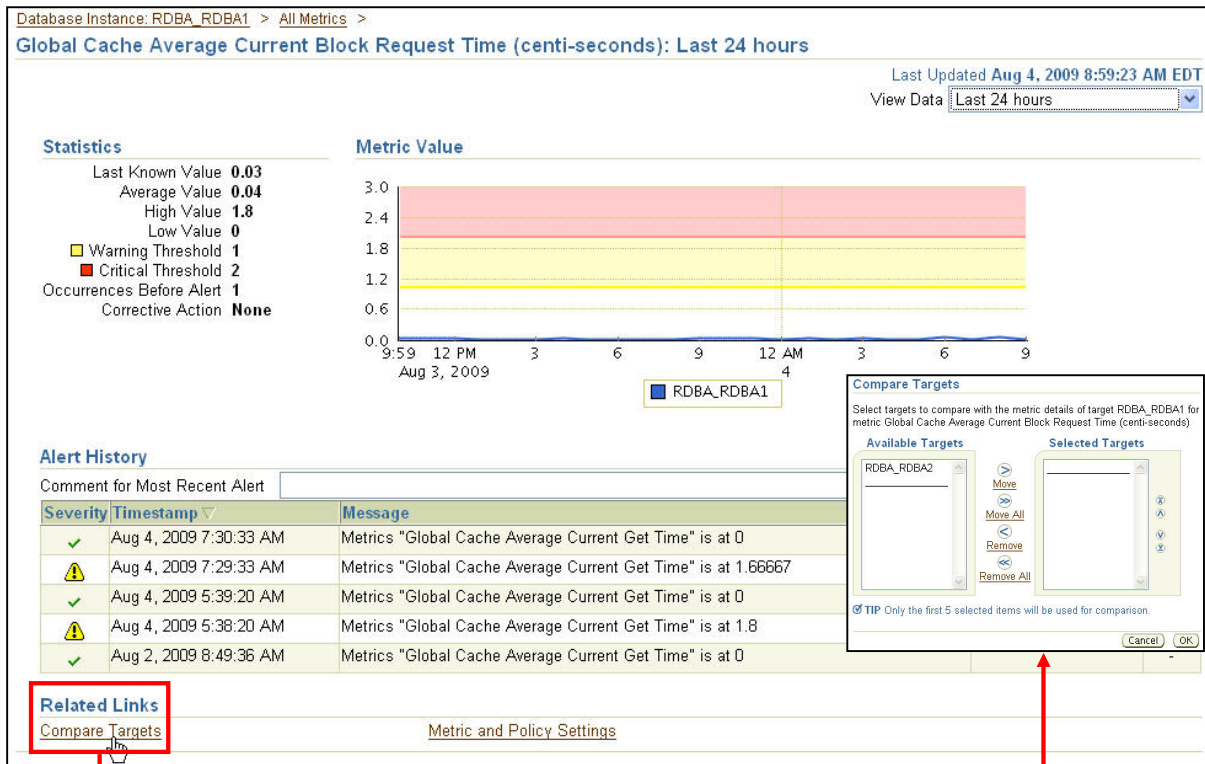
ORACLE

Copyright © 2010, Oracle. All rights reserved.

Enterprise Manager Metrics and RAC

Alert thresholds for instance-level alerts, such as archive log alerts, can be set at the instance target level. This enables you to receive alerts for the specific instance if performance exceeds your threshold. You can also configure alerts at the database level, such as setting alerts for tablespaces. This enables you to avoid receiving duplicate alerts at each instance.

Enterprise Manager Metrics and RAC



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Enterprise Manager Metrics and RAC (continued)

It is also possible to view the metric across the cluster in a comparative or overlay fashion. To view this information, click the Compare Targets link at the bottom of the corresponding metric page.

When the Compare Targets page appears, choose the instance targets that you want to compare by selecting them and then clicking the Move button. If you want to compare the metric data from all targets, click the Move All button. After making your selections, click the OK button to continue.

The Metric summary page appears next. Depending on your needs, you can accept the default timeline of 24 hours or select a more suitable value from the View Data drop-down list. If you want to add a comment regarding the event for future reference, enter a comment in the "Comment for Most Recent Alert" field, and then click the Add Comment button.

Enterprise Manager Alert History and RAC



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Enterprise Manager Alert History and RAC

In a RAC environment, you can see a summary of the alert history for each participating instance directly from the Cluster Database Home page. The drill-down process is shown in the slide. You click the Alert History link in the Related Links section of the Cluster Database Home page. This takes you to the Alert History page on which you can see the summary for both instances in the example. You can then click one of the instance's links to go to the corresponding Alert History page for that instance. From there, you can access a corresponding alert page by choosing the alert of your choice.

Oracle Internal & Oracle Academy Use Only

Enterprise Manager Blackouts and RAC

ORACLE Enterprise Manager 11g Database Control

Enterprise Manager Configuration | Management Services and Repository | Agents

Blackouts

Blackouts allow you to suspend monitoring on one or more targets in order to perform maintenance operations. To blackout a target, you need to have at least Operator privileges on the target.

View By: Targets in Blackout

Search: All target types

Parent: None

Show: Active ☒ Show all included targets

Create

No Blackouts found.

Create Blackout: Properties

Name: Blackout-Nov 7 2007 8:15:29 AM

Comments:

Reason: DB: Database Patch/Maintenance

☒ Run jobs during the blackout

Targets

Select the targets to be blacked out. You only see the targets for which you have Operator privileges. Targets that are not shown because they can only be blacked out as part of a full host blackout.

Available Targets: Cluster Database

Selected Targets:

Create Blackout: Member Targets

Targets such as groups, web applications, and application servers have member targets. Specify the member targets that should be included in the blackout. Agents that are members are ignored and can only be blacked out as part of a full host blackout.

Name	Type	Blackout
Selected Composite Targets		
▼ RDBB	Cluster Database	Full blackout (all members at blackout start time)
RDBB_RDBB1	Database Instance	<input checked="" type="checkbox"/>
RDBB_RDBB2	Database Instance	<input checked="" type="checkbox"/>

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Enterprise Manager Blackouts and RAC

You can use Enterprise Manager to define blackouts for all managed targets of your RAC database to prevent alerts from being recorded. Blackouts are useful when performing scheduled maintenance or other tasks that might trigger extraneous or unwanted events. You can define blackouts for an entire cluster database or for specific cluster database instances.

To create a blackout event, click the Setup link at the top of any Enterprise Manager page. Then click the Blackouts link on the left. The Blackouts page appears.

Click the Create button. The Create Blackout: Properties page appears. You must enter a name or tag in the Name field. If you want, you can also enter a descriptive comment in the Comments field. This is optional. Enter a reason for the blackout in the Reason field.

In the Targets area of the Properties page, you must choose a target Type from the drop-down list. In the example in the slide, the entire cluster database RDBB is chosen. Click the cluster database in the Available Targets list, and then click the Move button to move your choice to the Selected Targets list. Click the Next button to continue.

The Member Targets page appears next. Expand the Selected Composite Targets tree and ensure that all targets that must be included appear in the list. Continue and define your schedule as you normally would.

Redo Log Files and RAC

Cluster Database: RDBB > Redo Log Groups > Logged in As SYS

Create Redo Log Group

Show SQL Cancel OK

* Group # 5

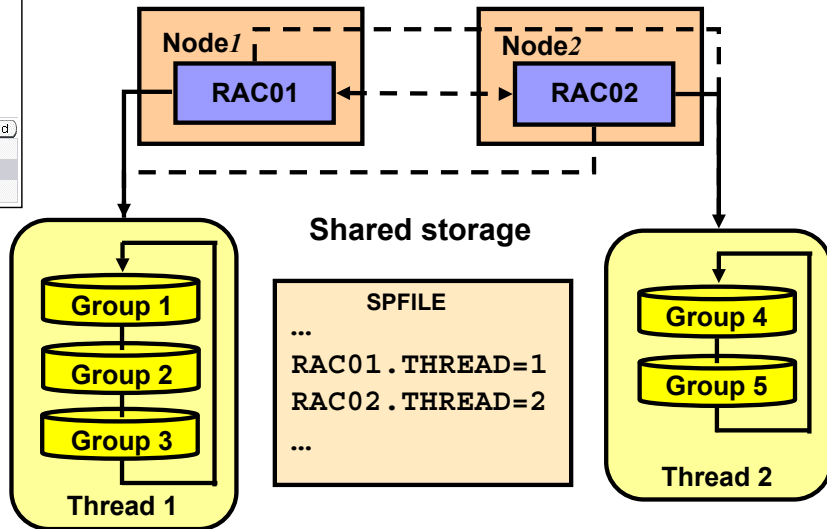
* File size 51200 KB

* Thread # 1

Redo Log Members

Edit Remove Add

Select File Name	File Directory
<ASM Generated>	+DATA/



```
ALTER DATABASE ADD LOGFILE THREAD 2 GROUP 4;
ALTER DATABASE ADD LOGFILE THREAD 2 GROUP 5;
```

```
ALTER DATABASE ENABLE THREAD 2;
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Internal & Oracle Academy Use Only

Redo Log Files and RAC

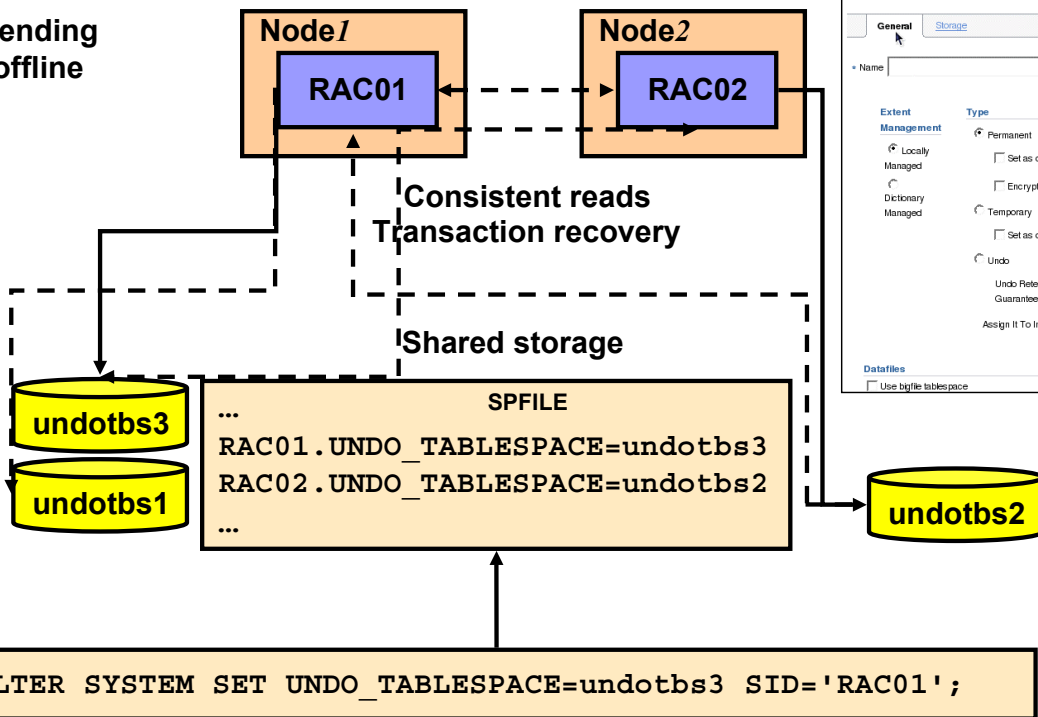
If you increase the cardinality of a server pool in a policy-managed database, and a new server is allocated to the server pool, then Oracle Clusterware starts an instance on the new server if you have Oracle Managed Files (OMF) enabled. If the instance starts and there is no thread or redo log file available, then Oracle Clusterware automatically enables a thread of redo and allocates the associated redo log files and undo if the database uses Oracle ASM or any cluster file system.

You should create redo log groups only if you are using administrator-managed databases. For administrator-managed databases, each instance has its own online redo log groups. Create these redo log groups and establish group members. To add a redo log group to a specific instance, specify the `INSTANCE` clause in the `ALTER DATABASE ADD LOGFILE` statement. If you do not specify the instance when adding the redo log group, the redo log group is added to the instance to which you are currently connected. Each instance must have at least two groups of redo log files. You must allocate the redo log groups before enabling a new instance with the `ALTER DATABASE ENABLE INSTANCE instance_name` command. When the current group fills, an instance begins writing to the next log file group. If your database is in `ARCHIVELOG` mode, each instance must save full online log groups as archived redo log files that are tracked in the control file.

Note: You can use Enterprise Manager to administer redo log groups in a RAC environment.

Automatic Undo Management and RAC

Pending
offline



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Automatic Undo Management in RAC

The Oracle database automatically manages undo segments within a specific undo tablespace that is assigned to an instance. Under normal circumstances, only the instance assigned to the undo tablespace can modify the contents of that tablespace. However, all instances can always read all undo blocks for consistent-read purposes. Also, any instance can update any undo tablespace during transaction recovery, as long as that undo tablespace is not currently used by another instance for undo generation or transaction recovery.

You assign undo tablespaces in your RAC database by specifying a different value for the `UNDO_TABLESPACE` parameter for each instance in your `SPFILE` or individual `PFILES`. If you do not set the `UNDO_TABLESPACE` parameter, each instance uses the first available undo tablespace. For policy-managed databases, Oracle automatically allocates the undo tablespace when the instance starts if you have OMF enabled.

You can dynamically switch undo tablespace assignments by executing the `ALTER SYSTEM SET UNDO_TABLESPACE` statement. You can run this command from any instance. In the example above, the previously used undo tablespace assigned to the `RAC01` instance remains assigned to it until `RAC01`'s last active transaction commits. The pending offline tablespace may be unavailable for other instances until all transactions against that tablespace are committed. You cannot simultaneously use Automatic Undo Management (AUM) and manual undo management in a RAC database. It is highly recommended that you use the AUM mode.

Starting and Stopping RAC Instances

- Multiple instances can open the same database simultaneously.
- Shutting down one instance does not interfere with other running instances.
- `SHUTDOWN TRANSACTIONAL LOCAL` does not wait for other instances' transactions to finish.
- RAC instances can be started and stopped by using:
 - Enterprise Manager
 - The Server Control (`srvctl`) utility
 - SQL*Plus
- Shutting down a RAC database means shutting down all instances accessing the database.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Starting and Stopping RAC Instances

In a RAC environment, multiple instances can have the same RAC database open at the same time. Also, shutting down one instance does not interfere with the operation of other running instances.

The procedures for starting up and shutting down RAC instances are identical to the procedures used in single-instance Oracle, with the following exception:

The `SHUTDOWN TRANSACTIONAL` command with the `LOCAL` option is useful to shut down an instance after all active transactions on the instance have either committed or rolled back. Transactions on other instances do not block this operation. If you omit the `LOCAL` option, this operation waits until transactions on all other instances that started before the shutdown are issued either a `COMMIT` or a `ROLLBACK`.

You can start up and shut down instances by using Enterprise Manager, SQL*Plus, or Server Control (`srvctl`). Both Enterprise Manager and `srvctl` provide options to start up and shut down all the instances of a RAC database with a single step.

Shutting down a RAC database mounted or opened by multiple instances means that you need to shut down every instance accessing that RAC database. However, having only one instance opening the RAC database is enough to declare the RAC database open.

Starting and Stopping RAC Instances with SQL*Plus

```
[host01] $ echo $ORACLE_SID
orcl1
sqlplus / as sysdba
SQL> startup
SQL> shutdown
```

```
[host02] $ echo $ORACLE_SID
orcl2
sqlplus / as sysdba
SQL> startup
SQL> shutdown
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Starting and Stopping RAC Instances with SQL*Plus

If you want to start up or shut down just one instance, and you are connected to your local node, then you must first ensure that your current environment includes the system identifier (SID) for the local instance.

To start up or shut down your local instance, initiate a SQL*Plus session connected as SYSDBA or SYSOPER, and then issue the required command (for example, STARTUP).

You can start multiple instances from a single SQL*Plus session on one node by way of Oracle Net Services. To achieve this, you must connect to each instance by using a Net Services connection string, typically an instance-specific alias from your `tnsnames.ora` file. For example, you can use a SQL*Plus session on a local node to shut down two instances on remote nodes by connecting to each using the instance's individual alias name.

It is not possible to start up or shut down more than one instance at a time in SQL*Plus, so you cannot start or stop all the instances for a cluster database with a single SQL*Plus command.

To verify that instances are running, on any node, look at `V$ACTIVE_INSTANCES`.

Note: SQL*Plus is integrated with Oracle Clusterware to make sure that corresponding resources are correctly handled when starting up and shutting down instances via SQL*Plus.

Starting and Stopping RAC Instances with `srvctl`

- start/stop syntax:

```
srvctl start|stop instance -d <db_name> -i <inst_name_list>  
[-o open|mount|nomount|normal|transactional|immediate|abort>]  
[-c <connect_str> | -q]
```

```
srvctl start|stop database -d <db_name>  
[-o open|mount|nomount|normal|transactional|immediate|abort>]  
[-c <connect_str> | -q]
```

- Examples:

```
$ srvctl start instance -d orcl -i orcl1,orcl2
```

```
$ srvctl stop instance -d orcl -i orcl1,orcl2
```

```
$ srvctl start database -d orcl -o open
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Starting and Stopping RAC Instances with `srvctl`

The `srvctl start database` command starts a cluster database, its enabled instances, and services.

The `srvctl stop database` command stops a database, its instances, and its services.

The `srvctl start instance` command starts instances of a cluster database. This command also starts all enabled and nonrunning services that have the listed instances either as preferred or as available instances.

The `srvctl stop instance` command stops instances, and all enabled and running services that have these instances as either preferred or available instances.

You must disable an object that you intend to keep stopped after you issue a `srvctl stop` command, otherwise Oracle Clusterware can restart it as a result of another planned operation. For the commands that use a connect string, if you do not provide a connect string, then `srvctl` uses `/ as sysdba` to perform the operation. The `-q` option asks for a connect string from standard input.

`srvctl` does not support concurrent executions of commands on the same object. Therefore, run only one `srvctl` command at a time for each database, service, or other object. In order to use the `START` or `STOP` options of the `SRVCTL` command, your service must be an Oracle Clusterware-enabled, nonrunning service.

Note: For more information, refer to the *Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide*.

Switch Between the Automatic and Manual Policies

```
$ srvctl config database -d orcl -a
Database unique name: orcl
Database name: orcl
Oracle home: /u01/app/oracle/product/11.2.0/dbhome_1
Oracle user: oracle
Spfile: +DATA/orcl/spfileorcl.ora
Domain:
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Server pools: orcl
Database instances: orcl1,orcl2
Disk Groups: DATA, FRA
Services:
Database is enabled
Database is administrator managed
```

```
srvctl modify database -d orcl -y MANUAL;
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Switch Between the Automatic and Manual Policies

By default, Oracle Clusterware is configured to start the VIP, listener, instance, ASM, database services, and other resources during system boot. It is possible to modify some resources to have their profile parameter `AUTO_START` set to the value 2. This means that after node reboot, or when Oracle Clusterware is started, resources with `AUTO_START=2` need to be started manually via `srvctl`. This is designed to assist in troubleshooting and system maintenance. When changing resource profiles through `srvctl`, the command tool automatically modifies the profile attributes of other dependent resources given the current prebuilt dependencies. The command to accomplish this is:

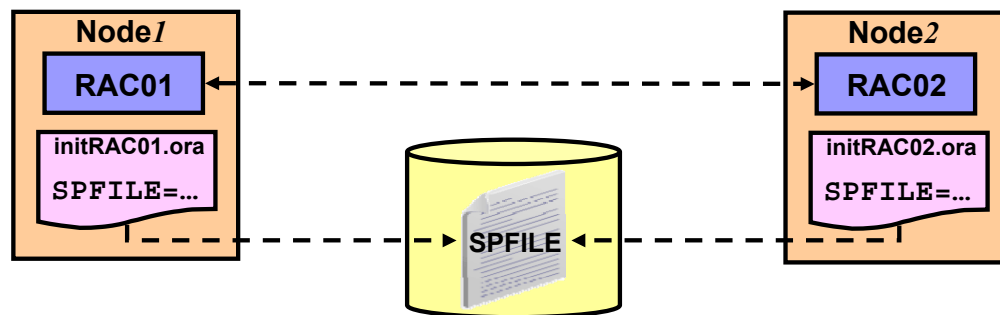
```
srvctl modify database -d <dbname> -y AUTOMATIC|MANUAL
```

To implement Oracle Clusterware and Real Application Clusters, it is best to have Oracle Clusterware start the defined Oracle Clusterware resources during system boot, which is the default. The first example in the slide uses the `srvctl config database` command to display the current policy for the `orcl` database. As you can see, it is currently set to its default: `AUTOMATIC`. The second statement uses the `srvctl modify database` command to change the current policy to `MANUAL` for the `orcl` database. When you add a new database by using the `srvctl add database` command, by default, that database is placed under the control of Oracle Clusterware using the `AUTOMATIC` policy. However, you can use the following statement to directly set the policy to `MANUAL`: `srvctl add database -d orcl -y MANUAL`.

Note: You can also use this procedure to configure your system to prevent Oracle Clusterware from auto-restarting failed database instances more than once.

RAC Initialization Parameter Files

- An SPFILE is created if you use the DBCA.
- The SPFILE must be created on a shared volume or shared raw device.
- All instances use the same SPFILE.
- If the database is created manually, create an SPFILE from a PFILE.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

RAC Initialization Parameter Files

When you create the database, DBCA creates an SPFILE in the file location that you specify. This location can be an ASM disk group, cluster file system file, or a shared raw device. If you manually create your database, it is recommended to create an SPFILE from a PFILE.

All instances in the cluster database use the same SPFILE. Because the SPFILE is a binary file, do not edit it. Change the SPFILE settings by using EM or ALTER SYSTEM SQL statements.

RAC uses a traditional PFILE only if an SPFILE does not exist or if you specify PFILE in your STARTUP command. Using SPFILE simplifies administration, maintaining parameter settings consistent, and guarantees parameter settings persistence across database shutdown and startup. In addition, you can configure RMAN to back up your SPFILE.

SPFILE Parameter Values and RAC

- You can change parameter settings using the ALTER SYSTEM SET command from any instance:

```
ALTER SYSTEM SET <dpname> SCOPE=MEMORY sid='<sid|*>';
```

- SPFILE entries such as:
 - *.<pname> apply to all instances
 - <sid>.<pname> apply only to <sid>
 - <sid>.<pname> takes precedence over *.<pname>
- Use current or future *.<dpname> settings for <sid>:

```
ALTER SYSTEM RESET <dpname> SCOPE=MEMORY sid='<sid>';
```

- Remove an entry from your SPFILE:

```
ALTER SYSTEM RESET <dpname> SCOPE=SPFILE sid='<sid|*>';
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Internal & Oracle Academy Use Only

SPFILE Parameter Values and RAC

You can modify the value of your initialization parameters by using the ALTER SYSTEM SET command. This is the same as with a single-instance database except that you have the possibility to specify the SID clause in addition to the SCOPE clause.

By using the SID clause, you can specify the SID of the instance where the value takes effect. Specify SID='*' if you want to change the value of the parameter for all instances. Specify SID='sid' if you want to change the value of the parameter only for the instance sid. This setting takes precedence over previous and subsequent ALTER SYSTEM SET statements that specify SID='*'. If the instances are started up with an SPFILE, then SID='*' is the default if you do not specify the SID clause.

If you specify an instance other than the current instance, then a message is sent to that instance to change the parameter value in its memory if you are not using the SPFILE scope.

The combination of SCOPE=MEMORY and SID='sid' of the ALTER SYSTEM RESET command allows you to override the precedence of a currently used <sid>.<dparam> entry. This allows for the current *.<dparam> entry to be used, or for the next created *.<dparam> entry to be taken into account on that particular SID.

Using the last example, you can remove a line from your SPFILE.

EM and SPFILE Parameter Values

Oracle Enterprise Manager 11g Database Control

Cluster Database: RDBB

Home Performance Availability Server Schema Data Movement

Storage Database Configuration

Control Files Initialization Parameters View Database Feature Usage

Temporary Tablespace Groups

Initialization Parameters

Current SPFILE

The parameter values listed here are currently used by the running instance(s). You can change static parameters in SPFILE mode.

Name Basic Modified Dynamic Category

open All All All All Go

Filter on a name or partial name

☐ Apply changes in current running instance(s) mode to SPFILE. For static parameters, you must restart the database.

Add Reset

Select	Instance	Name	Help	Revisions	Value	Comments	Type	Basic	Modified	Dynamic	Category
<input checked="" type="radio"/>	*	open_cursors			300		Integer	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Cursors and Library Cache
<input type="radio"/>	*	open_links					Integer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Distributed, Replication and Snapshot
<input type="radio"/>	*	open_links_per_instance			4		Integer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Distributed, Replication and Snapshot
<input type="radio"/>	*	read_only_open_delayed			FALSE		Boolean	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Memory
<input type="radio"/>	*	session_max_open_files			10		Integer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Objects and LOBs

Current SPFILE

Related Links

[Search for Changes Across Instances](#)

[Search for Latest Values Across Instances](#)

Show SQL Revert Apply

SCOPE=MEMORY

Copyright © 2010, Oracle. All rights reserved.

EM and SPFILE Parameter Values

You can access the Initialization Parameters page by clicking the Initialization Parameters link on the Cluster Database Server page.

The Current tabbed page shows you the values currently used by the initialization parameters of all the instances accessing the RAC database. You can filter the Initialization Parameters page to show only those parameters that meet the criteria of the filter that you entered in the Name field.

The Instance column shows the instances for which the parameter has the value listed in the table. An asterisk (*) indicates that the parameter has the same value for all remaining instances of the cluster database.

Choose a parameter from the Select column and perform one of the following steps:

- Click Add to add the selected parameter to a different instance. Enter a new instance name and value in the newly created row in the table.
- Click Reset to reset the value of the selected parameter. Note that you can reset only those parameters that do not have an asterisk in the Instance column. The value of the selected column is reset to the value of the remaining instances.

Note: For both Add and Reset buttons, the ALTER SYSTEM command uses SCOPE=MEMORY.

EM and SPFILE Parameter Values

ORACLE Enterprise Manager 11g Database Control

Cluster Database: RDBB >

Logged in As SYS

Show SQL Revert Apply

Initialization Parameters

Current SPFile

The parameter values listed here are from the SPFILE +DATA/rdbb/spfile1rdbb.ora

Name Basic Dynamic Category

open All All All Go

Filter on a name or partial name

☒ Apply changes in SPFILE mode to the current running instance(s). For static parameters, you must restart the database.

SCOPE=BOTH

Add Reset

Select	Instance	Name	Help	Value	Comments	Type	Constraint	Basic	Dynamic	Category
<input checked="" type="checkbox"/>	*	open_cursors	?	300		Integer	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Cursors and Library Cache
<input type="checkbox"/>	*	open_links	?			Integer	None	<input type="checkbox"/>	<input type="checkbox"/>	Distributed, Replication and Snapshot
<input type="checkbox"/>	*	open_links_per_instance	?			Integer	None	<input type="checkbox"/>	<input type="checkbox"/>	Distributed, Replication and Snapshot
<input type="checkbox"/>	*	read_only_open_delayed	?			Boolean	None	<input type="checkbox"/>	<input type="checkbox"/>	Memory
<input type="checkbox"/>	*	session_max_open_files	?			Integer	None	<input type="checkbox"/>	<input type="checkbox"/>	Objects and LOBs

SCOPE=SPFILE

Current SPFile

Related Links

[Search for Changes Across Instances](#)

[Search for Latest Values Across Instances](#)

Show SQL Revert Apply

EM and SPFILE Parameter Values (continued)

The SPFile tabbed page displays the current values stored in your SPFILE.

As on the Current tabbed page, you can add or reset parameters. However, if you select the “Apply changes in SPFILE mode” check box, the ALTER SYSTEM command uses SCOPE=BOTH. If this check box is not selected, SCOPE=SPFILE is used.

Click Apply to accept and generate your changes.

RAC Initialization Parameters

Initialization Parameters

Current [SPFile](#)

The parameter values listed here are currently used by the running instance(s). You can change static parameters in SPFile mode.

Name: Basic: Modified: Dynamic: Category:

Filter on a name or partial name

☒ Apply changes in current running instance(s) mode to SPFile. For static parameters, you must restart the database.

Select	Instance	Name	Help	Revisions	Value	Comments	Type	Basic	Modified	Dynamic	Category
<input type="radio"/>	*	cluster_database	?		TRUE		Boolean	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Cluster Database
<input type="radio"/>	*	cluster_database_instances	?		2		Integer		<input checked="" type="checkbox"/>		Cluster Database
<input type="radio"/>	*	cluster_interconnects	?				String				Cluster Database
<input type="radio"/>	*	db_name	?		RDBB		String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Database Identification
<input type="radio"/>	*	dispatchers	?		(PROTOCOL=TCP) (SER		String		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Shared Server
<input type="radio"/>	*	spfile	?		+DATA/rcdbb/spfilerdbb.ora		String		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Miscellaneous
<input type="radio"/>	RDBB1	thread	?		1		Integer		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Cluster Database
<input type="radio"/>	RDBB2	thread	?		2		Integer		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Cluster Database

ORACLE

Copyright © 2010, Oracle. All rights reserved.

RAC Initialization Parameters

CLUSTER_DATABASE: Enables a database to be started in cluster mode. Set this to TRUE.

CLUSTER_DATABASE_INSTANCES: Sets the number of instances in your RAC environment. A proper setting for this parameter can improve memory use.

CLUSTER_INTERCONNECTS: Specifies the cluster interconnect when there is more than one interconnect. Refer to your Oracle platform-specific documentation for the use of this parameter, its syntax, and its behavior. You typically do not need to set the CLUSTER_INTERCONNECTS parameter. For example, do not set this parameter for the following common configurations:

- If you have only one cluster interconnect
- If the default cluster interconnect meets the bandwidth requirements of your RAC database, which is typically the case
- If NIC bonding is being used for the interconnect
- When OIFCFG's global configuration can specify the right cluster interconnects. It only needs to be specified as an override for OIFCFG.

DB_NAME: If you set a value for DB_NAME in instance-specific parameter files, the setting must be identical for all instances.

DISPATCHERS: Set this parameter to enable a shared-server configuration, that is, a server that is configured to allow many user processes to share very few server processes.

RAC Initialization Parameters (continued)

With shared-server configurations, many user processes connect to a dispatcher. The `DISPATCHERS` parameter may contain many attributes. Oracle recommends that you configure at least the `PROTOCOL` and `LISTENER` attributes.

`PROTOCOL` specifies the network protocol for which the dispatcher process generates a listening end point. `LISTENER` specifies an alias name for the Oracle Net Services listeners. Set the alias to a name that is resolved through a naming method, such as a `tnsnames.ora` file.

THREAD : If specified, this parameter must have unique values on all instances. The `THREAD` parameter specifies the number of the redo thread to be used by an instance. You can specify any available redo thread number as long as that thread number is enabled and is not used.

Other parameters that can affect RAC database configurations include:

- **ASM_PREFERRED_READ_FAILURE_GROUPS** : Specifies a set of disks to be the preferred disks from which to read mirror data copies. The values that you set for this parameter are instance specific and need not be the same on all instances.
- **GCS_SERVER_PROCESSES** : This static parameter specifies the initial number of server processes for an Oracle RAC instance's Global Cache Service (GCS). The GCS processes manage the routing of interinstance traffic among Oracle RAC instances. The default number of GCS server processes is calculated based on system resources. For systems with one to three CPUs, there are two GCS server processes (`LMSn`). For systems with four to fifteen CPUs, there are two GCS server processes (`LMSn`). For systems with sixteen more CPUs, the number of GCS server processes equals (the number of CPUs divided by 32) + 2, dropping any fractions. You can set this parameter to different values on different instances.
- **INSTANCE_NAME** : The instance's SID. The SID identifies the instance's shared memory on a host. Any alphanumeric characters can be used. The value for this parameter is automatically set to the database unique name followed by an incrementing number during the creation of the database when using DBCA.
- **INSTANCE_NUMBER** : An Oracle RAC parameter that specifies a unique number that maps the instance to one free list group for each database object. This parameter must be set for every instance in the cluster. It is automatically defined during the creation of the database when using DBCA.
- **REMOTE_LISTENER** : This dynamic parameter specifies a network name that resolves to an address or address list of Oracle Net remote listeners (that is, listeners that are not running on the same machine as this instance). The address or address list is specified in the `TNSNAMES.ORA` file or other address repository as configured for your system.

Parameters That Require Identical Settings

- ACTIVE_INSTANCE_COUNT
- ARCHIVE_LAG_TARGET
- COMPATIBLE
- CLUSTER_DATABASE/CLUSTER_DATABASE_INSTANCES
- CONTROL_FILES
- DB_BLOCK_SIZE
- DB_DOMAIN
- DB_FILES
- DB_NAME
- DB_RECOVERY_FILE_DEST/DB_RECOVERY_FILE_DEST_SIZE
- DB_UNIQUE_NAME
- INSTANCE_TYPE (RDBMS or ASM)
- PARALLEL_EXECUTION_MESSAGE_SIZE
- REMOTE_LOGIN_PASSWORD_FILE
- UNDO_MANAGEMENT

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Parameters That Require Identical Settings

Certain initialization parameters that are critical at database creation or that affect certain database operations must have the same value for every instance in RAC. Specify these parameter values in the SPFILE, or within each `init_dbname.ora` file on each instance. In the above list, each parameter must have the same value on all instances.

Note: The setting for `DML_LOCKS` and `RESULT_CACHE_MAX_SIZE` must be identical on every instance only if set to zero. Disabling the result cache on some instances may lead to incorrect results.

Parameters That Require Unique Settings

Instance settings:

- THREAD
- ROLLBACK_SEGMENTS
- INSTANCE_NAME
- ASM_PREFERRED_READ_FAILURE_GROUPS
- INSTANCE_NUMBER
- UNDO_TABLESPACE
- CLUSTER_INTERCONNECTS

RDBB1	instance_name		RDBB1	String			Instance Identification
RDBB2	instance_name		RDBB2	String			Instance Identification
RDBB1	instance_number		1	Integer	✓	✓	Cluster Database
RDBB2	instance_number		2	Integer	✓	✓	Cluster Database

RDBB1	thread		1	Integer		✓	Cluster Database
RDBB2	thread		2	Integer		✓	Cluster Database

RDBB1	undo_tablespace		UNDOTBS1	String	✓	✓	Automatic Undo Management
RDBB2	undo_tablespace		UNDOTBS2	String	✓	✓	Automatic Undo Management

RDBB1	asm_preferred_read_failure_groups		DATA.SITEA	String		✓	Automatic Storage Management
RDBB2	asm_preferred_read_failure_groups		DATA.SITEB	String		✓	Automatic Storage Management

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Parameters That Require Unique Settings

If you use the `THREAD` or `ROLLBACK_SEGMENTS` parameters, it is recommended that you set unique values for them by using the `SID` identifier in the `SPFILE`. However, you must set a unique value for `INSTANCE_NUMBER` for each instance and you cannot use a default value.

The Oracle server uses the `INSTANCE_NUMBER` parameter to distinguish among instances at startup. The Oracle server uses the `THREAD` number to assign redo log groups to specific instances. To simplify administration, use the same number for both the `THREAD` and `INSTANCE_NUMBER` parameters.

If you specify `UNDO_TABLESPACE` with Automatic Undo Management enabled, set this parameter to a unique undo tablespace name for each instance.

Using the `ASM_PREFERRED_READ_FAILURE_GROUPS` initialization parameter, you can specify a list of preferred read failure group names. The disks in those failure groups become the preferred read disks. Thus, every node can read from its local disks. The setting for this parameter is instance specific, and the values need not be the same on all instances.

Quiescing RAC Databases

- Use the `ALTER SYSTEM QUIESCE RESTRICTED` statement from a single instance:

```
SQL> ALTER SYSTEM QUIESCE RESTRICTED;
```

- You must have the Database Resource Manager feature activated to issue the statement above.
- The database cannot be opened by other instances after the `ALTER SYSTEM QUIESCE...` statement starts.
- The `ALTER SYSTEM QUIESCE RESTRICTED` and `ALTER SYSTEM UNQUIESCE` statements affect all instances in a RAC environment.
- Cold backups cannot be taken when the database is in a quiesced state.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Quiescing RAC Databases

To quiesce a RAC database, use the `ALTER SYSTEM QUIESCE RESTRICTED` statement from one instance. It is not possible to open the database from any instance while the database is in the process of being quiesced from another instance. After all the non-DBA sessions become inactive, the `ALTER SYSTEM QUIESCE RESTRICTED` statement executes and the database is considered to be quiesced. In a RAC environment, this statement affects all instances.

To issue the `ALTER SYSTEM QUIESCE RESTRICTED` statement in a RAC environment, you must have the Database Resource Manager feature activated, and it must have been activated since instance startup for all instances in the cluster database. It is through the Database Resource Manager that non-DBA sessions are prevented from becoming active. The following conditions apply to RAC:

- If you had issued the `ALTER SYSTEM QUIESCE RESTRICTED` statement, but the Oracle server has not finished processing it, then you cannot open the database.
- You cannot open the database if it is already in a quiesced state.
- The `ALTER SYSTEM QUIESCE RESTRICTED` and `ALTER SYSTEM UNQUIESCE` statements affect all instances in a RAC environment, not just the instance that issues the command.

Cold backups cannot be taken when the database is in a quiesced state because the Oracle background processes may still perform updates for internal purposes even when the database is in a quiesced state. Also, the file headers of online data files continue to appear as if they are being accessed. They do not look the same as if a clean shutdown were done.

Terminating Sessions on a Specific Instance

```
SQL> SELECT SID, SERIAL#, INST_ID
       2 FROM GV$SESSION WHERE USERNAME='JMW';

       SID      SERIAL#      INST_ID
-----
       140        3340         2

SQL> ALTER SYSTEM KILL SESSION '140,3340,@2';
System altered.

SQL>
```

```
ALTER SYSTEM KILL SESSION '140,3340,@2'
*
ERROR at line 1:

ORA-00031: session marked for kill
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Terminating Sessions on a Specific Instance

Starting with Oracle Database 11g Release 1, you can use the `ALTER SYSTEM KILL SESSION` statement to terminate a session on a specific instance.

The slide above illustrates this by terminating a session started on a different instance than the one used to terminate the problematic session.

If the session is performing some activity that must be completed, such as waiting for a reply from a remote database or rolling back a transaction, then Oracle Database waits for this activity to complete, marks the session as terminated, and then returns control to you. If the waiting lasts a minute, Oracle Database marks the session to be terminated and returns control to you with a message that the session is marked to be terminated. The PMON background process then marks the session as terminated when the activity is complete.

Note: You can also use the `IMMEDIATE` clause at the end of the `ALTER SYSTEM` command to immediately terminate the session without waiting for outstanding activity to complete.

How SQL*Plus Commands Affect Instances

SQL*Plus Command	Associated Instance
ARCHIVE LOG	Generally affects the current instance
CONNECT	Affects the default instance if no instance is specified in the CONNECT command
HOST	Affects the node running the SQL*Plus session
RECOVER	Does not affect any particular instance, but rather the database
SHOW PARAMETER and SHOW SGA	Show the current instance parameter and SGA information
STARTUP and SHUTDOWN	Affect the current instance
SHOW INSTANCE	Displays information about the current instance

ORACLE

Copyright © 2010, Oracle. All rights reserved.

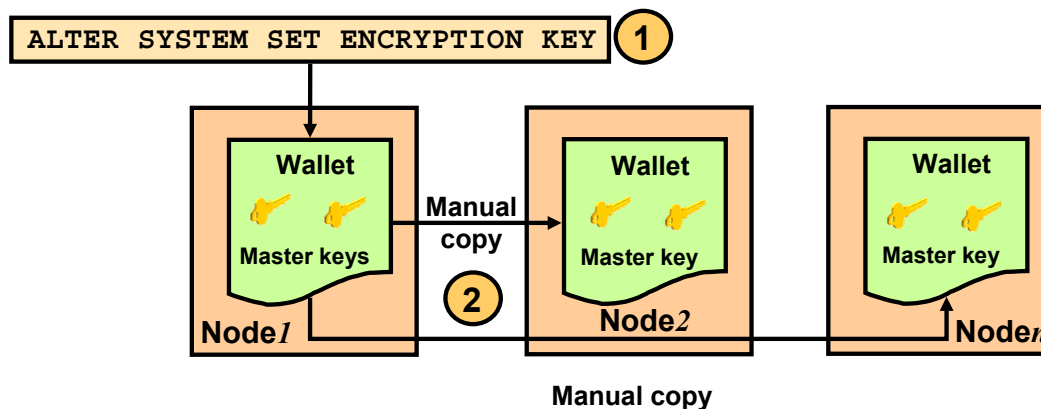
How SQL*Plus Commands Affect Instances

Most SQL statements affect the current instance. You can use SQL*Plus to start and stop instances in the RAC database. You do not need to run SQL*Plus commands as `root` on UNIX-based systems or as Administrator on Windows-based systems. You need only the proper database account with the privileges that you normally use for single-instance Oracle database administration. The following are some examples of how SQL*Plus commands affect instances:

- The `ALTER SYSTEM SET CHECKPOINT LOCAL` statement affects only the instance to which you are currently connected, rather than the default instance or all instances.
- `ALTER SYSTEM CHECKPOINT LOCAL` affects the current instance.
- `ALTER SYSTEM CHECKPOINT` or `ALTER SYSTEM CHECKPOINT GLOBAL` affects all instances in the cluster database.
- `ALTER SYSTEM SWITCH LOGFILE` affects only the current instance.
- To force a global log switch, use the `ALTER SYSTEM ARCHIVE LOG CURRENT` statement.
- The `INSTANCE` option of `ALTER SYSTEM ARCHIVE LOG` enables you to archive each online redo log file for a specific instance.

Transparent Data Encryption and Wallets in RAC

- One wallet shared by all instances on shared storage:
 - No additional administration is required.
- One copy of the wallet on each local storage:
 - Local copies need to be synchronized each time master key is changed.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Transparent Data Encryption and Wallets in RAC

Wallets used by RAC instances for Transparent Database Encryption may be a local copy of a common wallet shared by multiple nodes, or a shared copy residing on shared storage that all of the nodes can access.

A deployment with a single wallet on a shared disk requires no additional configuration to use Transparent Data Encryption.

If you want to use local copies, you must copy the wallet and make it available to all of the other nodes after initial configuration. For systems using Transparent Data Encryption with encrypted wallets, you can use any standard file transport protocol. For systems using Transparent Data Encryption with obfuscated wallets, file transport through a secured channel is recommended. The wallet must reside in the directory specified by the setting for the `WALLET_LOCATION` or `ENCRYPTION_WALLET_LOCATION` parameter in `sqlnet.ora`. The local copies of the wallet need not be synchronized for the duration of Transparent Data Encryption usage until the server key is rekeyed through the `ALTER SYSTEM SET KEY` SQL statement. Each time you run the `ALTER SYSTEM SET KEY` statement at a database instance, you must again copy the wallet residing on that node and make it available to all of the other nodes. To avoid unnecessary administrative overhead, reserve rekeying for exceptional cases where you are certain that the server master key is compromised and that not rekeying it would cause a serious security problem.

Quiz

If an instance starts in a policy-managed RAC environment and no thread or redo log file is available, then Oracle Clusterware automatically enables a thread of redo and allocates the redo log files and undo if the database uses Oracle ASM or any cluster file system and OMF is enabled.

1. True
2. False

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Answer: 1

The statement is true.

Quiz

Which of the following statements is *not* true:

1. Multiple instances can open the same database simultaneously.
2. Shutting down one instance does not interfere with other running instances.
3. SHUTDOWN TRANSACTIONAL LOCAL will wait for other instances' transactions to finish.
4. Shutting down a RAC database means shutting down all instances accessing the database.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Answer: 3

Statement 3 is not true.

Summary

In this lesson, you should have learned how to:

- Use Enterprise Manager Cluster Database pages
- Define redo log files in a RAC environment
- Define undo tablespaces in a RAC environment
- Start and stop RAC databases and instances
- Modify initialization parameters in a RAC environment

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Practice 12 Overview

This practice covers the following topics:

- Using operating system and password file authenticated connections
- Using Oracle Database authenticated connections
- Stopping a complete ORACLE_HOME component stack

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Managing Backup and Recovery for RAC

13

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Objectives

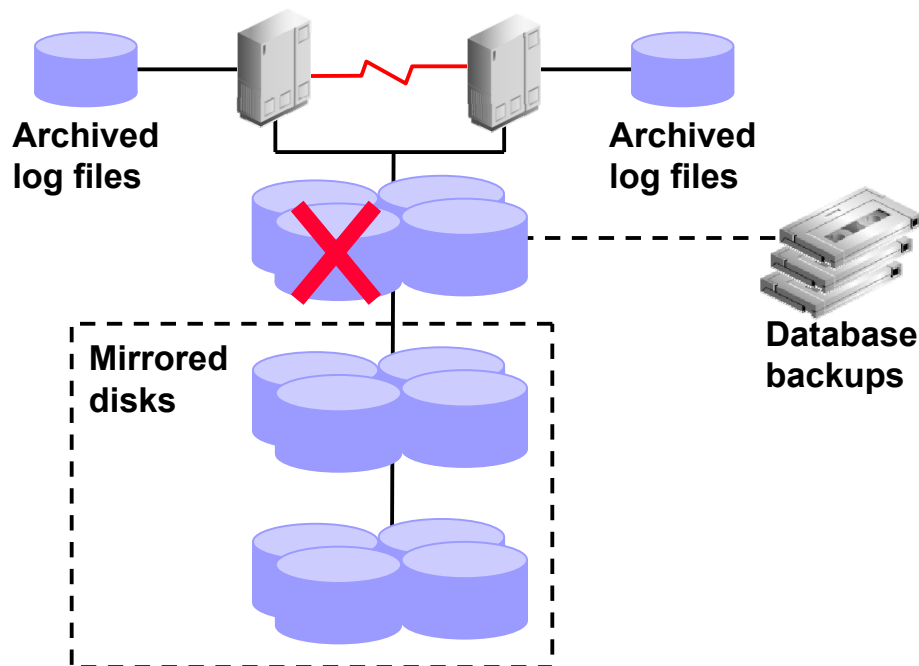
After completing this lesson, you should be able to configure the following:

- The RAC database to use `ARCHIVELOG` mode and the fast recovery area
- RMAN for the RAC environment

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Protecting Against Media Failure



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Protecting Against Media Failure

Although RAC provides you with methods to avoid or to reduce down time due to a failure of one or more (but not all) of your instances, you must still protect the database itself, which is shared by all the instances. This means that you need to consider disk backup and recovery strategies for your cluster database just as you would for a nonclustered database.

To minimize the potential loss of data due to disk failures, you may want to use disk mirroring technology (available from your server or disk vendor). As in nonclustered databases, you can have more than one mirror if your vendor allows it, to help reduce the potential for data loss and to provide you with alternative backup strategies. For example, with your database in ARCHIVELOG mode and with three copies of your disks, you can remove one mirror copy and perform your backup from it while the two remaining mirror copies continue to protect ongoing disk activity. To do this correctly, you must first put the tablespaces into backup mode and then, if required by your cluster or disk vendor, temporarily halt disk operations by issuing the `ALTER SYSTEM SUSPEND` command. After the statement completes, you can break the mirror and then resume normal operations by executing the `ALTER SYSTEM RESUME` command and taking the tablespaces out of backup mode.

Media Recovery in Oracle RAC

- Media recovery must be user initiated through a client application.
- In these situations, use RMAN to restore backups of the data files and then recover the database.
- RMAN media recovery procedures for RAC do not differ substantially from those for single-instance environments.
- The node that performs the recovery must be able to restore all of the required data files.
 - That node must also be able to either read all required archived redo logs on disk or restore them from backups.
- When recovering a database with encrypted tablespaces, the Oracle Wallet must be opened after database mount and before you open the database.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Media Recovery in Oracle RAC

Media recovery must be user initiated through a client application, whereas instance recovery is automatically performed by the database. In these situations, use RMAN to restore backups of the data files and then recover the database. The procedures for RMAN media recovery in Oracle RAC environments do not differ substantially from the media recovery procedures for single-instance environments.

The node that performs the recovery must be able to restore all of the required data files. That node must also be able to either read all the required archived redo logs on disk or be able to restore them from backups. Each instance generates its own archive logs that are copies of its dedicated redo log group threads. It is recommended that Automatic Storage Management (ASM) or a cluster file system be used to consolidate these files.

When recovering a database with encrypted tablespaces (for example, after a SHUTDOWN ABORT or a catastrophic error that brings down the database instance), you must open the Oracle Wallet after database mount and before you open the database, so the recovery process can decrypt data blocks and redo.

Parallel Recovery in RAC

- Oracle Database automatically selects the optimum degree of parallelism for:
 - Instance recovery
 - Crash recovery
- Archived redo logs are applied using an optimal number of parallel processes based on the availability of CPUs.
- With RMAN's `RESTORE` and `RECOVER` commands, the following three stages of recovery are performed in parallel:
 - Restoring data files
 - Applying incremental backups
 - Applying archived redo logs
- To disable parallel instance and crash recovery, set the `RECOVERY_PARALLELISM` parameter to 0.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Parallel Recovery in RAC

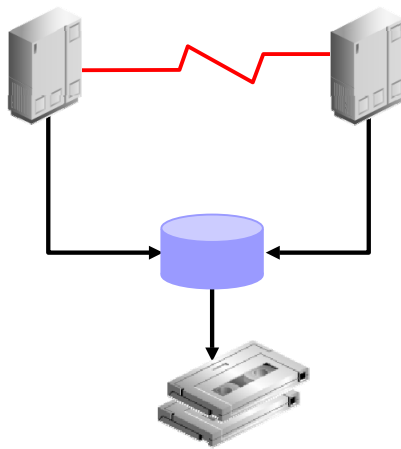
Oracle Database automatically selects the optimum degree of parallelism for instance and crash recovery. Oracle Database applies archived redo logs using an optimal number of parallel processes based on the availability of CPUs. With RMAN's `RESTORE` and `RECOVER` commands, Oracle Database automatically performs in parallel the following three stages of recovery:

- **Restoring Data Files:** When restoring data files, the number of channels you allocate in the RMAN recover script effectively sets the parallelism that RMAN uses. For example, if you allocate five channels, you can have up to five parallel streams restoring data files.
- **Applying Incremental Backups:** Similarly, when you are applying incremental backups, the number of channels you allocate determines the potential parallelism.
- **Applying Archived Redo Logs with RMAN:** The application of archived redo logs is performed in parallel. Oracle Database automatically selects the optimum degree of parallelism based on available CPU resources.

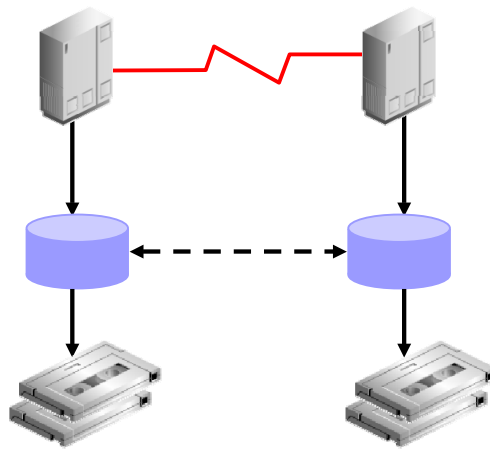
To disable parallel instance and crash recovery on a system with multiple CPUs, set the `RECOVERY_PARALLELISM` parameter to 0.

Use the `NOPARALLEL` clause of the RMAN `RECOVER` command or `ALTER DATABASE RECOVER` statement to force the RAC database to use nonparallel media recovery.

Archived Log File Configurations



Cluster file system scheme:
Archive logs from each instance are written to the same file location.



Local archive with NFS scheme: Each instance can read mounted archive destinations of all instances.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Archived Log File Configurations

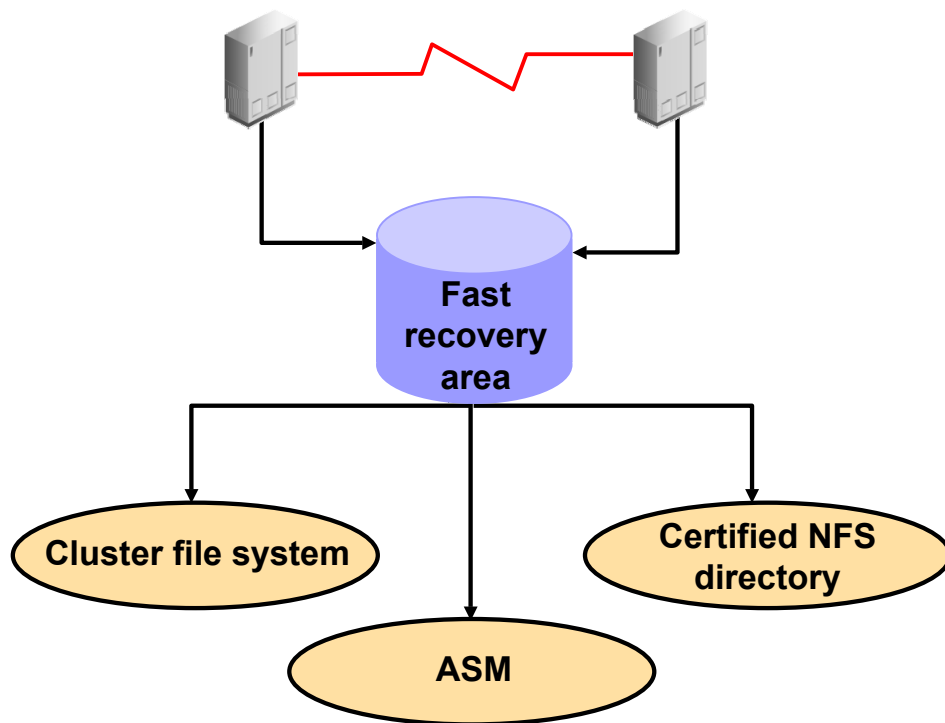
During backup and recovery operations involving archived log files, the Oracle server determines the file destinations and names from the control file. If you use RMAN, the archived log file path names can also be stored in the optional recovery catalog. However, the archived log file path names do not include the node name, so RMAN expects to find the files it needs on the nodes where the channels are allocated.

If you use a cluster file system, your instances can all write to the same archive log destination. This is known as the cluster file system scheme. Backup and recovery of the archive logs are easy because all logs are located in the same directory.

If a cluster file system is not available, Oracle recommends that local archive log destinations be created for each instance with NFS-read mount points to all other instances. This is known as the local archive with network file system (NFS) scheme. During backup, you can either back up the archive logs from each host or select one host to perform the backup for all archive logs. During recovery, one instance may access the logs from any host without having to first copy them to the local destination. The `LOG_ARCHIVE_FORMAT` parameter supports the `%t` variable that embeds the unique thread number into the name of the archive logs so that each node generates unique names.

Using either scheme, you may want to provide a second archive destination to avoid single points of failure.

RAC and the Fast Recovery Area



ORACLE

Copyright © 2010, Oracle. All rights reserved.

RAC and the Fast Recovery Area

To use a fast recovery area in RAC, you must place it on an ASM disk group, a cluster file system, or on a shared directory that is configured through certified NFS for each RAC instance. That is, the fast recovery area must be shared among all the instances of a RAC database. In addition, set the `DB_RECOVERY_FILE_DEST` parameter to the same value on all instances.

Oracle Enterprise Manager enables you to set up a fast recovery area. To use this feature:

1. From the Cluster Database Home page, click the Maintenance tab.
2. Under the Backup/Recovery options list, click Configure Recovery Settings.
3. Specify your requirements in the Flash Recovery Area section of the page.

Note: For Oracle Database 11g Release 2 (11.2), the flash recovery area has been renamed fast recovery area. Oracle Enterprise Manager, however, still uses the older vocabulary on its Web pages.

RAC Backup and Recovery Using EM



RAC Backup and Recovery Using EM

You can access the Cluster Database backup and recovery–related tasks by clicking the Availability tab on the Cluster Database Home page. On the Availability tabbed page, you can perform a range of backup and recovery operations using RMAN, such as scheduling backups, performing recovery when necessary, and configuring backup and recovery settings. Also, there are links related to Oracle Secure Backup and Service management.

Configure RAC Recovery Settings with EM

Cluster Database: RDBA > Recovery Settings

Logged in As SYS

Show SQL Revert Apply

Instance Recovery

The fast-start checkpointing feature is enabled by specifying a non-zero desired mean-time to recover (MTTR) value, which will be used to set the FAST_START_MTTR_TARGET initialization parameter. This parameter controls the amount of time the database takes to perform crash recovery for a single instance. When fast-start checkpointing is enabled, Oracle automatically maintains the speed of checkpointing so that the requested MTTR is achieved. Setting the value to 0 will disable this functionality.

Current Estimated Mean Time To Recover (seconds) 0

Maximum value across all instances:

Desired Mean Time To Recover 0 Minutes

Media Recovery

The database is currently in NOARCHIVELOG mode. In ARCHIVELOG mode, hot backups and recovery to the latest time are possible, but you must provide space for archived redo log files. If you change the database to ARCHIVELOG mode, you should perform a backup immediately. In NOARCHIVELOG mode, only cold backups are possible and data may be lost.

☐ ARCHIVELOG Mode*

Log Archive Filename Format* RDBA_%.t_%.s_%.r.dbf

Number	Archived Redo Log Destination
1	USE_DB_RECOVERY_FILE_DEST

[Add Another Row](#)

☒ TIP It is recommended that archived redo log files be written to multiple destinations.

☒ TIP You can specify up to 10 archived redo log destinations.

☐ Enable Minimal Supplemental Logging

Minimal supplemental logging logs the minimal amount of information needed for LogMiner (operations associated with DML changes).

Flash Recovery

Flash Recovery Area is enabled for this database. The chart shows space used by each file type that is not reclaimable by Oracle. Performing backups to a tertiary storage is one way to make space reclaimable. Usable Flash Recovery Area includes free and reclaimable space.

Flash Recovery Area Location +FRA

Flash Recovery Area Size 3 GB

Flash Recovery Area Size must be set when the location is set

Reclaimable Flash Recovery Area (MB) 18

Free Flash Recovery Area (GB) 2.48

☒ Enable Flashback Database - flashback logging can be used for fast database point-in-time recovery*

The flashback recovery area must be set to enable flashback logging. When using flashback logs, you may recover your entire database to a prior point-in-time without restoring files. Flashback is the preferred point-in-time recovery method in the recovery wizard when appropriate.

Flashback Retention Time 24 Hours

Current size of the flashback logs (MB) 15.625

Lowest SCN in the flashback data: 684307

Flashback Time Nov 21, 2008 7:57:56 AM

☐ Apply changes to SPFILE only. Otherwise the changes will be made to both SPFILE and the running instance which requires that you restart the database to invoke static parameters.

Flash Recovery Area Usage

File Type	Size (GB)	Percentage
Usable	~ 2.5 GB	83.3%
BACKUP PIECE	0.23 GB	7.6%
REDO LOG	0.2 GB	6.6%
ARCHIVED LOG	0.04 GB	1.4%
CONTROL FILE	0.02 GB	0.6%
FLASHBACK LOG	0.02 GB	0.5%

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Configure RAC Recovery Settings with EM

You can use Enterprise Manager to configure important recovery settings for your cluster database. On the Database Home page, click the Availability tab, and then click the Recovery Settings link. From here, you can ensure that your database is in ARCHIVELOG mode and configure flash recovery settings.

With a RAC database, if the Archive Log Destination setting is not the same for all instances, the field appears blank, with a message indicating that instances have different settings for this field. In this case, entering a location in this field sets the archive log location for all instances of database. You can assign instance-specific values for an archive log destination by using the Initialization Parameters page.

Note: You can run the ALTER DATABASE SQL statement to change the archiving mode in RAC as long as the database is mounted by the local instance but not open in any instances. You do not need to modify parameter settings to run this statement. Set the initialization parameters DB_RECOVERY_FILE_DEST and DB_RECOVERY_FILE_DEST_SIZE to the same values on all instances to configure a flash recovery area in a RAC environment.

Archived Redo File Conventions in RAC

Variable	Description	Example
%t	Thread number, not padded	log_1
%T	Thread number, left-zero-padded	log_0001
%s	Log sequence number, not padded	log_251
%S	Log sequence number, left-zero-padded	log_000000251
%r	Resetlogs identifier	log_23452345
%R	Padded resetlogs identifier	log_0023452345
%t_%s_%r	Using multiple variables	log_1_251_23452345

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Archived Redo File Conventions in RAC

For any archived redo log configuration, uniquely identify the archived redo logs with the `LOG_ARCHIVE_FORMAT` parameter. The format of this parameter is operating system-specific and it can include text strings, one or more variables, and a file name extension.

All of the thread parameters, in either uppercase or lowercase, are mandatory for RAC. This enables the Oracle database to create unique names for archive logs across the incarnation. This requirement is in effect when the `COMPATIBLE` parameter is set to 10.0 or greater. Use the `%R` or `%r` parameter to include the resetlogs identifier to avoid overwriting the logs from a previous incarnation. If you do not specify a log format, the default is operating system specific and includes `%t`, `%s`, and `%r`.

As an example, if the instance associated with redo thread number 1 sets `LOG_ARCHIVE_FORMAT` to `log_%t_%s_%r.arc`, then its archived redo log files are named as:

```
log_1_1000_23435343.arc
log_1_1001_23452345.arc
log_1_1002_23452345.arc
...
```


Configure RAC Backup Settings with EM

ORACLE

Copyright © 2010, Oracle. All rights reserved.

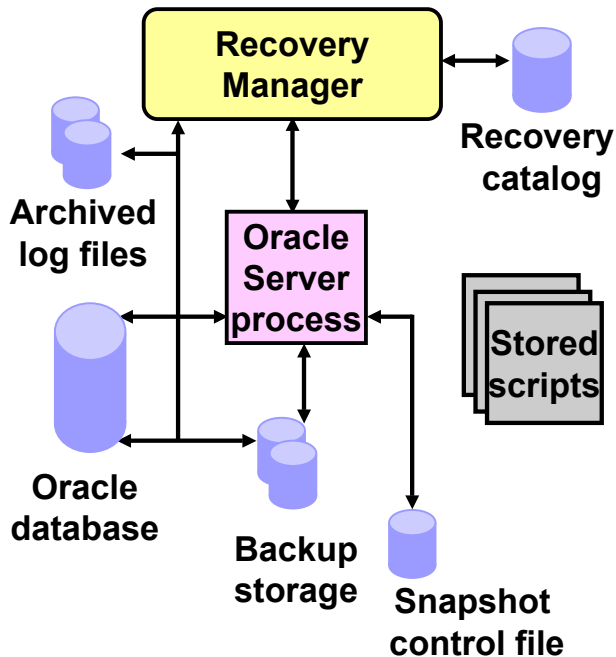
Configure RAC Backup Settings with EM

Persistent backup settings can be configured using Enterprise Manager. On the Database Control Home page, click the Availability tab, and then click the Backup Settings link. You can configure disk settings, such as the directory location of your disk backups and level of parallelism. You can also choose the default backup type:

- Backup set
- Compressed backup set
- Image copy

You can also specify important tape-related settings, such as the number of available tape drives and vendor-specific media management parameters.

Oracle Recovery Manager



RMAN provides the following benefits for Real Application Clusters:

- Can read cluster files, ASM files, or raw partitions with no configuration changes
- Can access multiple archive log destinations

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Recovery Manager

Oracle Database provides RMAN for backing up and restoring the database. RMAN enables you to back up, restore, and recover data files, control files, SPFILEs, and archived redo logs. You can run RMAN from the command line or you can use it from the Backup Manager in Enterprise Manager. In addition, RMAN is the recommended backup and recovery tool if you are using ASM. RMAN can use stored scripts, interactive scripts, or an interactive GUI front end. When using RMAN with your RAC database, use stored scripts to initiate the backup and recovery processes from the most appropriate node.

If you use different Oracle Home locations for your RAC instances on each of your nodes, create a snapshot control file in a location that exist on all your nodes. The snapshot control file is only needed on the nodes on which RMAN performs backups. The snapshot control file does not need to be globally available to all instances in a RAC environment though.

You can use either a cluster file or a shared raw device as well as a local directory that exists on each node in your cluster. Here is an example:

```
RMAN> CONFIGURE SNAPSHOT CONTROLFILE TO  
'/oracle/db_files/snaps/snap_prod1.cf';
```

For recovery, you must ensure that each recovery node can access the archive log files from all instances by using one of the archive schemes discussed earlier, or make the archived logs available to the recovering instance by copying them from another location.

Configure RMAN Snapshot Control File Location

- The snapshot control file path must be valid on every node from which you might initiate an RMAN backup.
- Configure the snapshot control file location in RMAN.
 - Determine the current location:

```
RMAN> SHOW SNAPSHOT CONTROLFILE NAME;  
/u01/app/oracle/product/11.2.0/dbhome_1/dbs/snap_prod.f
```

- You can use ASM, a shared file system location, or a shared block device if you prefer:

```
RMAN> CONFIGURE SNAPSHOT CONTROLFILE NAME TO  
'+FRA/SNAP/snap_prod.cf';  
RMAN> CONFIGURE SNAPSHOT CONTROLFILE NAME TO  
'/ocfs2/oradata/dbs/scf/snap_prod.cf';  
RMAN> CONFIGURE SNAPSHOT CONTROLFILE NAME TO  
'/dev/sdj2';
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Configure RMAN Snapshot Control File Location

The snapshot control file is a temporary file that RMAN creates to resynchronize from a read-consistent version of the control file. RMAN needs a snapshot control file only when resynchronizing with the recovery catalog or when making a backup of the current control file.

In a RAC database, the snapshot control file is created on the node that is making the backup. You need to configure a default path and file name for these snapshot control files that are valid on every node from which you might initiate an RMAN backup.

Run the following RMAN command to determine the configured location of the snapshot control file:

```
SHOW SNAPSHOT CONTROLFILE NAME
```

You can change the configured location of the snapshot control file. For example, on UNIX-based systems, you can specify the snapshot control file location as `snap_prod.cf` located in the ASM disk group `+FRA` by entering the following at the RMAN prompt:

```
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '+FRA/SNAP/snap_prod.cf'
```

This command globally sets the configuration for the location of the snapshot control file throughout your cluster database.

Note: The `CONFIGURE` command creates persistent settings across RMAN sessions.

Configure Control File and SPFILE Autobackup

- RMAN automatically creates a control file and SPFILE backup after the BACKUP or COPY command:

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

- Change default location:

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR  
DEVICE TYPE DISK TO '+DATA';
```

- Location must be available to all nodes in your RAC database.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Configure Control File and SPFILE Autobackup

If you set `CONFIGURE CONTROLFILE AUTOBACKUP` to ON, RMAN automatically creates a control file and an SPFILE backup after you run the `BACKUP` or `COPY` command. RMAN can also automatically restore an SPFILE if this is required to start an instance to perform recovery. This means that the default location for the SPFILE must be available to all nodes in your RAC database.

These features are important in disaster recovery because RMAN can restore the control file even without a recovery catalog. RMAN can restore an autobackup of the control file even after the loss of both the recovery catalog and the current control file.

You can change the default location that RMAN gives to this file with the `CONFIGURE CONTROLFILE AUTOBACKUP FORMAT` command. If you specify an absolute path name in this command, this path must exist identically on all nodes that participate in backups.

Note: RMAN performs `CONTROL FILE AUTOBACKUP` on the first allocated channel. When you allocate multiple channels with different parameters (especially if you allocate a channel with the `CONNECT` command), you must determine which channel will perform the automatic backup. Always allocate the channel for the connected node first.

Crosschecking on Multiple RAC Clusters Nodes

When crosschecking on multiple nodes make sure that all backups can be accessed by every node in the cluster.

- This allows you to allocate channels at any node in the cluster during restore or crosscheck operations.
- Otherwise you must allocate channels on multiple nodes by providing the `CONNECT` option to the `CONFIGURE CHANNEL` command.
- If backups are not accessible because no channel was configured on the node that can access those backups, then those backups are marked `EXPIRED`.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Crosschecking on Multiple RAC Clusters Nodes

When crosschecking on multiple RAC nodes, configure the cluster so that all backups can be accessed by every node, regardless of which node created the backup. When the cluster is configured this way, you can allocate channels at any node in the cluster during restore or crosscheck operations.

If you cannot configure the cluster so that each node can access all backups, then during restore and crosscheck operations, you must allocate channels on multiple nodes by providing the `CONNECT` option to the `CONFIGURE CHANNEL` command, so that every backup can be accessed by at least one node. If some backups are not accessible during crosscheck because no channel was configured on the node that can access those backups, then those backups are marked `EXPIRED` in the RMAN repository after the crosscheck.

For example, you can use `CONFIGURE CHANNEL . . . CONNECT` in an Oracle RAC configuration in which tape backups are created on various nodes in the cluster and each backup is only accessible on the node on which it is created.

Channel Connections to Cluster Instances

- When backing up, each allocated channel can connect to a different instance in the cluster.
- Instances to which the channels connect must be either all mounted or all open.
- When choosing a channel to use, RMAN gives preference to the nodes with faster access to the data files that you want to back up.

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE DEVICE TYPE sbt PARALLELISM 3;
CONFIGURE CHANNEL 1 DEVICE TYPE sbt CONNECT='sys/rac@orcl1';
CONFIGURE CHANNEL 2 DEVICE TYPE sbt CONNECT='sys/rac@orcl2';
CONFIGURE CHANNEL 3 DEVICE TYPE sbt CONNECT='sys/rac@orcl3';
```

OR

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE DEVICE TYPE sbt PARALLELISM 3;
CONFIGURE CHANNEL DEVICE TYPE sbt CONNECT='sys/rac@bkp_serv';
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Channel Connections to Cluster Instances

When making backups in parallel, RMAN channels can connect to a different instance in the cluster. The examples in the slide illustrate two possible configurations:

- If you want to dedicate channels to specific instances, you can control at which instance the channels are allocated by using separate connect strings for each channel configuration as shown by the first example.
- If you define a special service for your backup and recovery jobs, you can use the second example shown in the slide. If you configure this service with load balancing turned on, then the channels are allocated at a node as decided by the load balancing algorithm.

During a backup, the instances to which the channels connect must be either all mounted or all open. For example, if the `orcl1` instance has the database mounted whereas the `orcl2` and `orcl3` instances have the database open, then the backup fails.

In some cluster database configurations, some nodes of the cluster have faster access to certain data files than to other data files. RMAN automatically detects this, which is known as node affinity awareness. When deciding which channel to use to back up a particular data file, RMAN gives preference to the nodes with faster access to the data files that you want to back up. For example, if you have a three-node cluster, and if node 1 has faster read/write access to data files 7, 8, and 9 than the other nodes, then node 1 has greater node affinity to those files than nodes 2 and 3 and RMAN will take advantage of this automatically.

RMAN Channel Support for the Grid

- RAC allows the use of nondeterministic connect strings.
- RMAN can use connect strings that are not bound to a specific instance in the Grid environment.
- It simplifies the use of parallelism with RMAN in a RAC environment.
- It uses the load-balancing characteristics of the Grid environment.
 - Channels connect to RAC instances that are the least loaded.

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;  
CONFIGURE DEVICE TYPE sbt PARALLELISM 3;
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

RMAN Channel Support for the Grid

RAC allows the use of nondeterministic connect strings that can connect to different instances based on RAC features, such as load balancing. Therefore, to support RAC, the RMAN polling mechanism no longer depends on deterministic connect strings, and makes it possible to use RMAN with connect strings that are not bound to a specific instance in the Grid environment. Previously, if you wanted to use RMAN parallelism and spread a job between many instances, you had to manually allocate an RMAN channel for each instance. To use dynamic channel allocation, you do not need separate `CONFIGURE CHANNEL CONNECT` statements anymore. You only need to define your degree of parallelism by using a command such as `CONFIGURE DEVICE TYPE disk PARALLELISM`, and then run backup or restore commands. RMAN then automatically connects to different instances and does the job in parallel. The Grid environment selects the instances that RMAN connects to, based on load balancing. As a result of this, configuring RMAN parallelism in a RAC environment becomes as simple as setting it up in a non-RAC environment. By configuring parallelism when backing up or recovering a RAC database, RMAN channels are dynamically allocated across all RAC instances.

Note: RMAN has no control over the selection of the instances. If you require a guaranteed connection to an instance, you should provide a connect string that can connect only to the required instance.

RMAN Default Autolocation

- Recovery Manager autolocates the following files:
 - Backup pieces
 - Archived redo logs during backup
 - Data file or control file copies
- If local archiving is used, a node can read only those archived logs that were generated on that node.
- When restoring, a channel connected to a specific node restores only those files that were backed up to the node.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

RMAN Default Autolocation

Recovery Manager automatically discovers which nodes of a RAC configuration can access the files that you want to back up or restore. Recovery Manager autolocates the following files:

- Backup pieces during backup or restore
- Archived redo logs during backup
- Data file or control file copies during backup or restore

If you use a noncluster file system local archiving scheme, a node can read only those archived redo logs that were generated by an instance on that node. RMAN never attempts to back up archived redo logs on a channel that it cannot read.

During a restore operation, RMAN automatically performs the autolocation of backups. A channel connected to a specific node attempts to restore only those files that were backed up to the node. For example, assume that log sequence 1001 is backed up to the drive attached to node 1, whereas log 1002 is backed up to the drive attached to node 2. If you then allocate channels that connect to each node, the channel connected to node 1 can restore log 1001 (but not 1002), and the channel connected to node 2 can restore log 1002 (but not 1001).

Distribution of Backups

Several possible backup configurations for RAC:

- A dedicated backup server performs and manages backups for the cluster and the cluster database.
- One node has access to a local backup appliance and performs and manages backups for the cluster database.
- Each node has access to a local backup appliance and can write to its own local backup media.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Internal & Oracle Academy Use Only

Distribution of Backups

When configuring the backup options for RAC, you have several possible configurations:

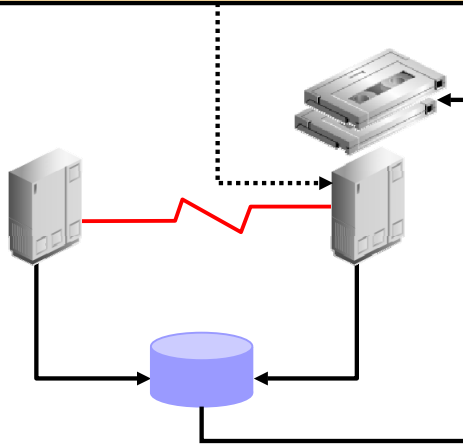
- **Network backup server:** A dedicated backup server performs and manages backups for the cluster and the cluster database. None of the nodes have local backup appliances.
- **One local drive:** One node has access to a local backup appliance and performs and manages backups for the cluster database. All nodes of the cluster should be on a cluster file system to be able to read all data files, archived redo logs, and SPFILEs. It is recommended that you do not use the noncluster file system archiving scheme if you have backup media on only one local drive.
- **Multiple drives:** Each node has access to a local backup appliance and can write to its own local backup media.

In the cluster file system scheme, any node can access all the data files, archived redo logs, and SPFILEs. In the noncluster file system scheme, you must write the backup script so that the backup is distributed to the correct drive and path for each node. For example, node 1 can back up the archived redo logs whose path names begin with `/arc_dest_1`, node 2 can back up the archived redo logs whose path names begin with `/arc_dest_2`, and node 3 can back up the archived redo logs whose path names begin with `/arc_dest_3`.

One Local Drive CFS Backup Scheme

```
RMAN> CONFIGURE DEVICE TYPE sbt PARALLELISM 1;  
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO sbt;
```

```
RMAN> BACKUP DATABASE PLUS ARCHIVELOG DELETE INPUT;
```



ORACLE

Copyright © 2010, Oracle. All rights reserved.

One Local Drive CFS Backup Scheme

In a cluster file system (CFS) backup scheme, each node in the cluster has read access to all the data files, archived redo logs, and SPFILEs. This includes Automated Storage Management (ASM), cluster file systems, and Network Attached Storage (NAS).

When backing up to only one local drive in the cluster file system backup scheme, it is assumed that only one node in the cluster has a local backup appliance such as a tape drive. In this case, run the following one-time configuration commands:

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 1;  
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
```

Because any node performing the backup has read/write access to the archived redo logs written by the other nodes, the backup script for any node is simple:

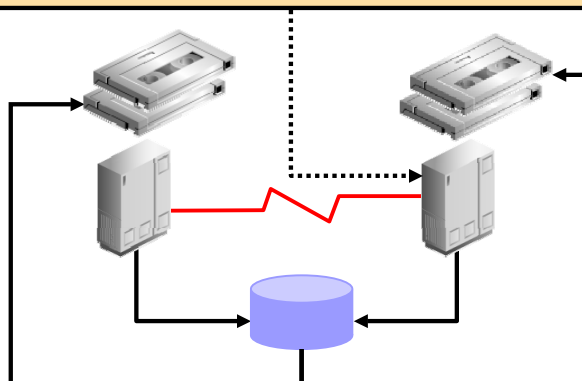
```
BACKUP DATABASE PLUS ARCHIVELOG DELETE INPUT;
```

In this case, the tape drive receives all data files, archived redo logs, and SPFILEs.

Multiple Drives CFS Backup Scheme

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 2;  
CONFIGURE DEFAULT DEVICE TYPE TO sbt;  
CONFIGURE CHANNEL 1 DEVICE TYPE sbt CONNECT 'usr1/pwd1@n1';  
CONFIGURE CHANNEL 2 DEVICE TYPE sbt CONNECT 'usr2/pwd2@n2';
```

```
BACKUP DATABASE PLUS ARCHIVELOG DELETE INPUT;
```



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Multiple Drives CFS Backup Scheme

When backing up to multiple drives in the cluster file system backup scheme, it is assumed that each node in the cluster has its own local tape drive. Perform the following one-time configuration so that one channel is configured for each node in the cluster. For example, enter the following at the RMAN prompt:

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 2;  
CONFIGURE DEFAULT DEVICE TYPE TO sbt;  
CONFIGURE CHANNEL 1 DEVICE TYPE sbt CONNECT 'user1/passwd1@node1';  
CONFIGURE CHANNEL 2 DEVICE TYPE sbt CONNECT 'user2/passwd2@node2';
```

Similarly, you can perform this configuration for a device type of DISK. The following backup script, which you can run from any node in the cluster, distributes the data files, archived redo logs, and SPFILE backups among the backup drives:

```
BACKUP DATABASE PLUS ARCHIVELOG DELETE INPUT;
```

Restoring and Recovering

- Media recovery may require one or more archived log files from each thread.
- The `RMAN RECOVER` command automatically restores and applies the required archived logs.
- Archive logs may be restored to any node performing the restore and recover operation.
- Logs must be readable from the node performing the restore and recovery activity.
- Recovery processes request additional threads enabled during the recovery period.
- Recovery processes notify you of threads no longer needed because they were disabled.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Restoring and Recovering

Media recovery of a database that is accessed by RAC may require at least one archived log file for each thread. However, if a thread's online redo log contains enough recovery information, restoring archived log files for any thread is unnecessary.

If you use RMAN for media recovery and you share archive log directories, you can change the destination of the automatic restoration of archive logs with the `SET` clause to restore the files to a local directory of the node where you begin recovery. If you backed up the archive logs from each node without using a central media management system, you must first restore all the log files from the remote nodes and move them to the host from which you will start recovery with RMAN.

However, if you backed up each node's log files using a central media management system, you can use RMAN's `AUTOLOCATE` feature. This enables you to recover a database using the local tape drive on the remote node.

If recovery reaches a time when an additional thread was enabled, the recovery process requests the archived log file for that thread. If you are using a backup control file, when all archive log files are exhausted, you may need to redirect the recovery process to the online redo log files to complete recovery. If recovery reaches a time when a thread was disabled, the process informs you that the log file for that thread is no longer needed.

Quiz

Which of the following statements regarding media recovery in RAC is *not* true?

1. Media recovery must be user initiated through a client application.
2. RMAN media recovery procedures for RAC are quite different from those for single-instance environments.
3. The node that performs the recovery must be able to restore all the required data files.
4. The recovering node must be able to either read all required archived redo logs on disk or restore them from backups.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Answer: 2

Statement 2 is not correct.

Quiz

To use a fast recovery area in RAC, you must place it on an ASM disk group, a cluster file system, or on a shared directory that is configured through certified NFS for each RAC instance.

1. True
2. False

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Answer: 1

The statement above is true.

Summary

In this lesson, you should have learned how to configure the following:

- The RAC database to use `ARCHIVELOG` mode and the fast recovery area
- RMAN for the RAC environment

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Practice 13 Overview

This practice covers the following topics:

- Configuring the archive log mode
- Configuring specific instance connection strings
- Configuring RMAN and performing parallel backups

ORACLE

Copyright © 2010, Oracle. All rights reserved.

14

RAC Database Monitoring and Tuning

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Objectives

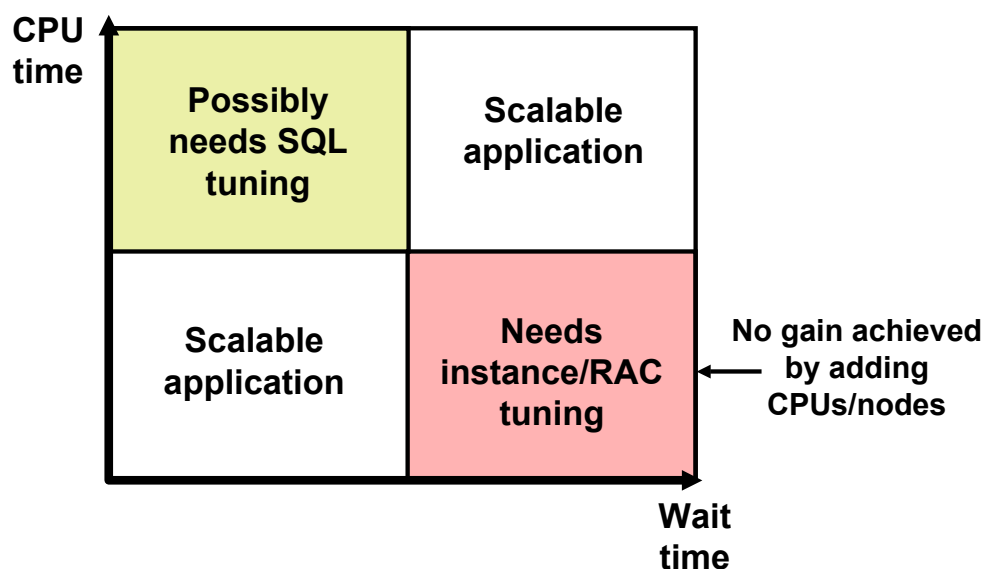
After completing this lesson, you should be able to:

- Determine RAC-specific tuning components
- Tune instance recovery in RAC
- Determine RAC-specific wait events, global enqueues, and system statistics
- Implement the most common RAC tuning tips
- Use the Cluster Database Performance pages
- Use the Automatic Workload Repository (AWR) in RAC
- Use Automatic Database Diagnostic Monitor (ADDM) in RAC

ORACLE

Copyright © 2010, Oracle. All rights reserved.

CPU and Wait Time Tuning Dimensions



ORACLE

Copyright © 2010, Oracle. All rights reserved.

CPU and Wait Time Tuning Dimensions

When tuning your system, it is important that you compare the CPU time with the wait time of your system. Comparing CPU time with wait time helps to determine how much of the response time is spent on useful work and how much on waiting for resources potentially held by other processes.

As a general rule, the systems where CPU time is dominant usually need less tuning than the ones where wait time is dominant. Alternatively, heavy CPU usage can be caused by badly written SQL statements.

Although the proportion of CPU time to wait time always tends to decrease as load on the system increases, steep increases in wait time are a sign of contention and must be addressed for good scalability.

Adding more CPUs to a node, or nodes to a cluster, would provide very limited benefit under contention. Conversely, a system where the proportion of CPU time to wait time does not decrease significantly as load increases can scale better, and would most likely benefit from adding CPUs or Real Application Clusters (RAC) instances if needed.

Note: Automatic Workload Repository (AWR) reports display CPU time together with wait time in the **Top 5 Timed Events** section, if the CPU time portion is among the top five events.

RAC-Specific Tuning

- Tune for a single instance first.
- Tune for RAC:
 - Instance recovery
 - Interconnect traffic
 - Point of serialization can be exacerbated
- RAC-reactive tuning tools:
 - Specific wait events
 - System and enqueue statistics
 - Enterprise Manager performance pages
 - Statspack and AWR reports
- RAC-proactive tuning tools:
 - AWR snapshots
 - ADDM reports

} Certain combinations
are characteristic of
well-known tuning cases.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

RAC-Specific Tuning

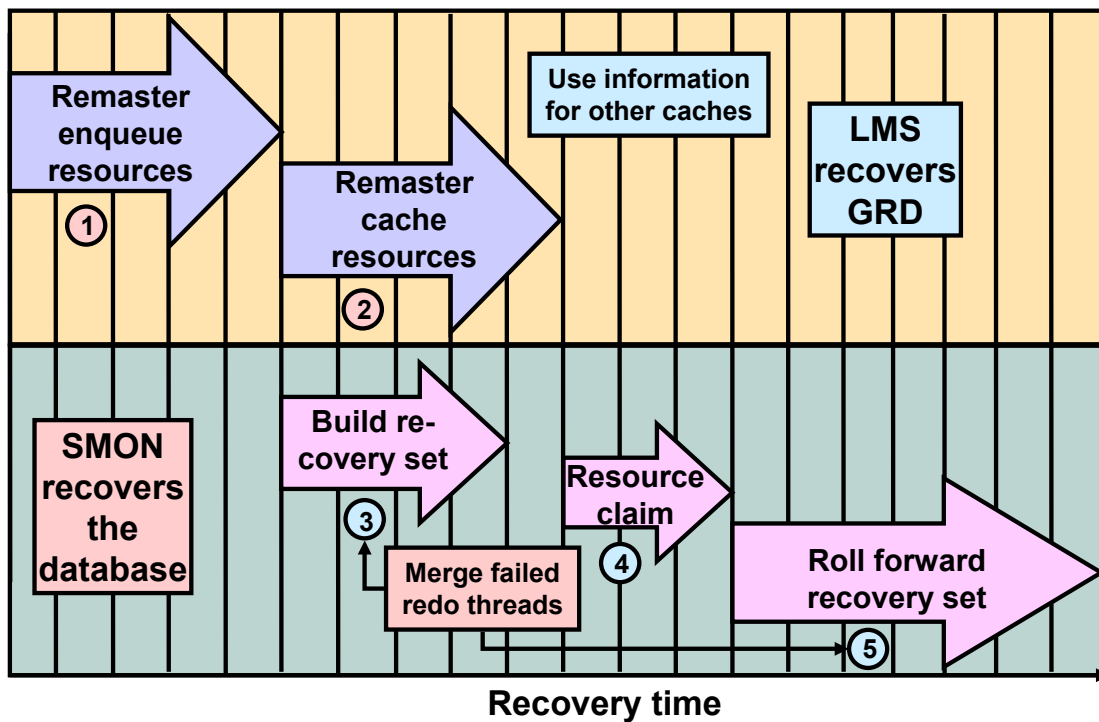
Although there are specific tuning areas for RAC, such as instance recovery and interconnect traffic, you get most benefits by tuning your system like a single-instance system. At least, this must be your starting point.

Obviously, if you have serialization issues in a single-instance environment, these may be exacerbated with RAC.

As shown in the slide, you have basically the same tuning tools with RAC as with a single-instance system. However, certain combinations of specific wait events and statistics are well-known RAC tuning cases.

In this lesson, you see some of those specific combinations, as well as the RAC-specific information that you can get from the Enterprise Manager performance pages, and Statspack and AWR reports. Finally, you see the RAC-specific information that you can get from the Automatic Database Diagnostic Monitor (ADDM).

RAC and Instance or Crash Recovery



Copyright © 2010, Oracle. All rights reserved.

RAC and Instance or Crash Recovery

When an instance fails and the failure is detected by another instance, the second instance performs the following recovery steps:

1. During the first phase of recovery, Global Enqueue Services remasters the enqueues.
2. The Global Cache Services (GCS) remasters its resources. The GCS processes remaster only those resources that lose their masters. During this time, all GCS resource requests and write requests are temporarily suspended. However, transactions can continue to modify data blocks as long as these transactions have already acquired the necessary resources.
3. After enqueues are reconfigured, one of the surviving instances can grab the Instance Recovery enqueue. Therefore, at the same time as GCS resources are remastered, SMON determines the set of blocks that need recovery. This set is called the recovery set. Because, with Cache Fusion, an instance ships the contents of its blocks to the requesting instance without writing the blocks to the disk, the on-disk version of the blocks may not contain the changes that are made by either instance. This implies that SMON needs to merge the content of all the online redo logs of each failed instance to determine the recovery set. This is because one failed thread might contain a hole in the redo that needs to be applied to a particular block. So, redo threads of failed instances cannot be applied serially. Also, redo threads of surviving instances are not needed for recovery because SMON could use past or current images of their corresponding buffer caches.

RAC and Instance or Crash Recovery (continued)

4. Buffer space for recovery is allocated and the resources that were identified in the previous reading of the redo logs are claimed as recovery resources. This is done to avoid other instances to access those resources.
5. All resources required for subsequent processing have been acquired and the Global Resource Directory (GRD) is now unfrozen. Any data blocks that are not in recovery can now be accessed. Note that the system is already partially available.

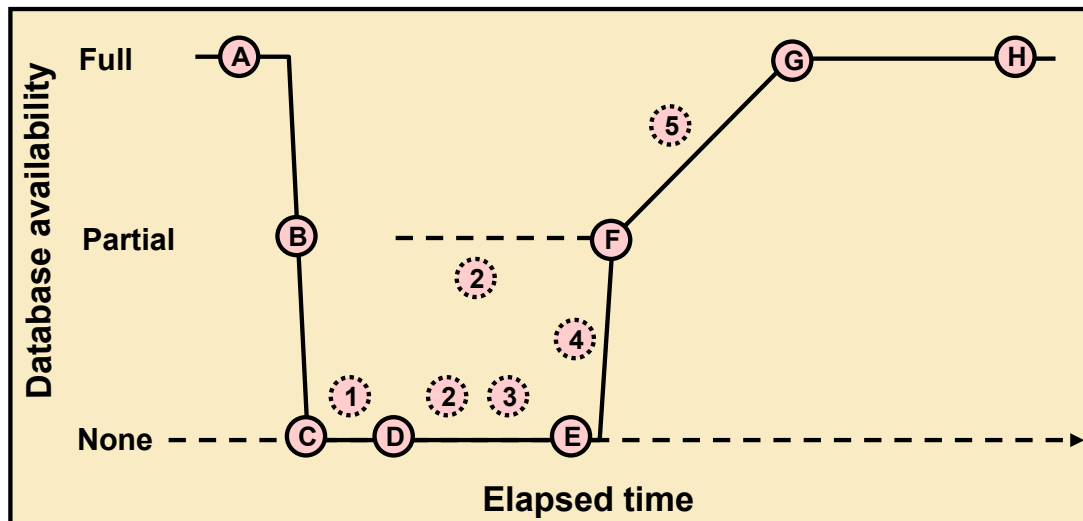
Then, assuming that there are past images or current images of blocks to be recovered in other caches in the cluster database, the most recent image is the starting point of recovery for these particular blocks. If neither the past image buffers nor the current buffer for a data block is in any of the surviving instances' caches, then SMON performs a log merge of the failed instances. SMON recovers and writes each block identified in step 3, releasing the recovery resources immediately after block recovery so that more blocks become available as recovery proceeds. Refer to the section titled "Global Cache Coordination: Example" in this lesson for more information about past images.

After all the blocks have been recovered and the recovery resources have been released, the system is again fully available.

In summary, the recovered database or the recovered portions of the database becomes available earlier, and before the completion of the entire recovery sequence. This makes the system available sooner and it makes recovery more scalable.

Note: The performance overhead of a log merge is proportional to the number of failed instances and to the size of the amount of redo written in the redo logs for each instance.

Instance Recovery and Database Availability



ORACLE

Copyright © 2010, Oracle. All rights reserved.

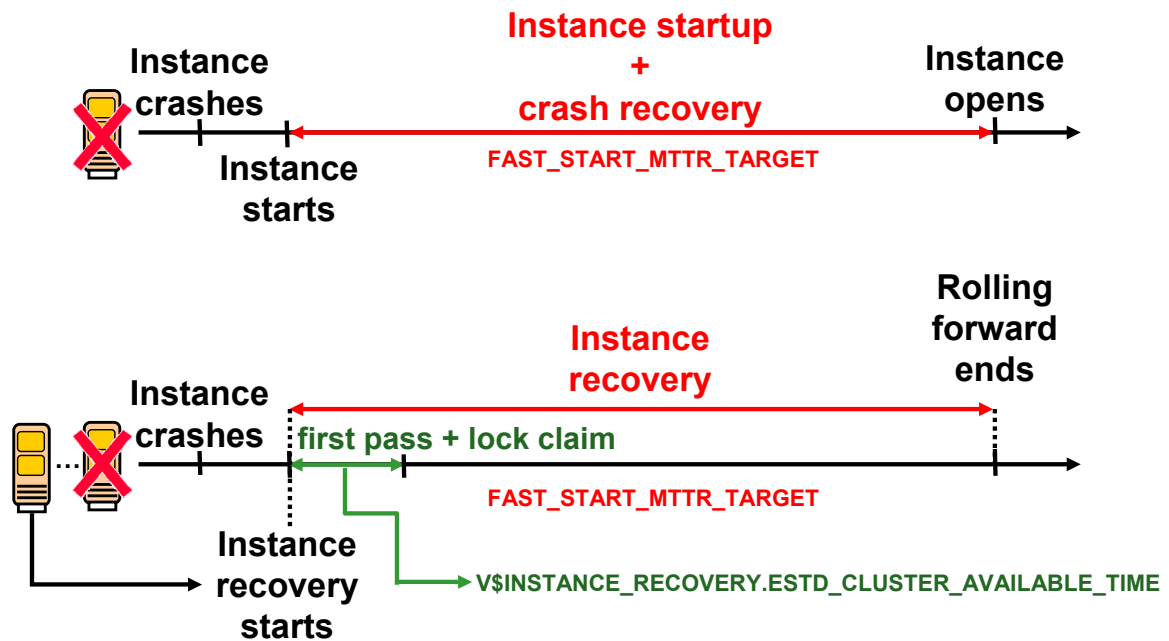
Instance Recovery and Database Availability

The graphic illustrates the degree of database availability during each step of Oracle instance recovery:

- A. Real Application Clusters is running on multiple nodes.
- B. Node failure is detected.
- C. The enqueue part of the GRD is reconfigured; resource management is redistributed to the surviving nodes. This operation occurs relatively quickly.
- D. The cache part of the GRD is reconfigured and SMON reads the redo log of the failed instance to identify the database blocks that it needs to recover.
- E. SMON issues the GRD requests to obtain all the database blocks it needs for recovery. After the requests are complete, all other blocks are accessible.
- F. The Oracle server performs roll forward recovery. Redo logs of the failed threads are applied to the database, and blocks are available right after their recovery is completed.
- G. The Oracle server performs rollback recovery. Undo blocks are applied to the database for all uncommitted transactions.
- H. Instance recovery is complete and all data is accessible.

Note: The dashed line represents the blocks identified in step 2 in the previous slide. Also, the dotted steps represent the ones identified in the previous slide.

Instance Recovery and RAC



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Internal & Oracle Academy Use Only

Instance Recovery and RAC

In a single-instance environment, the instance startup combined with the crash recovery time is controlled by the setting of the `FAST_START_MTTR_TARGET` initialization parameter. You can set its value if you want incremental checkpointing to be more aggressive than the autotuned checkpointing. However, this is at the expense of a much higher I/O overhead.

In a RAC environment, including the startup time of the instance in this calculation is useless because one of the surviving instances is doing the recovery.

In a RAC environment, it is possible to monitor the estimated target, in seconds, for the duration from the start of instance recovery to the time when GCD is open for lock requests for blocks not needed for recovery. This estimation is published in the `V$INSTANCE_RECOVERY` view through the `ESTD_CLUSTER_AVAILABLE_TIME` column. Basically, you can monitor the time your cluster is frozen during instance recovery situations.

In a RAC environment, the `FAST_START_MTTR_TARGET` initialization parameter is used to bound the entire instance recovery time, assuming it is instance recovery for single instance death.

Note: If you really want to have small instance recovery time by setting `FAST_START_MTTR_TARGET`, you can safely ignore the alert log messages indicating to raise its value.

Instance Recovery and RAC

- Use parallel instance recovery.
- Set `PARALLEL_MIN_SERVERS`.
- Use asynchronous input/output (I/O).
- Increase the size of the default buffer cache.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Instance Recovery and RAC (continued)

Here are some guidelines you can use to make sure that instance recovery in your RAC environment is faster:

- Use parallel instance recovery by setting `RECOVERY_PARALLISM`.
- Set `PARALLEL_MIN_SERVERS` to `CPU_COUNT-1`. This will prespawn recovery slaves at startup time.
- Using asynchronous I/O is one of the most crucial factors in recovery time. The first-pass log read uses asynchronous I/O.
- Instance recovery uses 50 percent of the default buffer cache for recovery buffers. If this is not enough, some of the steps of instance recovery will be done in several passes. You should be able to identify such situations by looking at your `alert.log` file. In that case, you should increase the size of your default buffer cache.

Analyzing Cache Fusion Impact in RAC

- The cost of block access and cache coherency is represented by:
 - Global Cache Services statistics
 - Global Cache Services wait events
- The response time for cache fusion transfers is determined by:
 - Overhead of the physical interconnect components
 - IPC protocol
 - GCS protocol
- The response time is not generally affected by disk I/O factors.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Analyzing Cache Fusion Impact in RAC

The effect of accessing blocks in the global cache and maintaining cache coherency is represented by:

- The Global Cache Services statistics for current and cr blocks—for example, gc current blocks received, gc cr blocks received, and so on
- The Global Cache Services wait events for gc current block 3-way, gc cr grant 2-way, and so on

The response time for cache fusion transfers is determined by the messaging time and processing time imposed by the physical interconnect components, the IPC protocol, and the GCS protocol. It is not affected by disk input/output (I/O) factors other than occasional log writes. The cache fusion protocol does not require I/O to data files in order to guarantee cache coherency, and RAC inherently does not cause any more I/O to disk than a nonclustered instance.

Typical Latencies for RAC Operations

AWR Report Latency Name	Lower Bound	Typical	Upper Bound
Average time to process cr block request	0.1	1	10
Avg global cache cr block receive time (ms)	0.3	4	12
Average time to process current block request	0.1	3	23
Avg global cache current block receive time (ms)	0.3	8	30

Global Cache and Enqueue Services - Workload Characteristics	
Avg global enqueue get time (ms):	4.5
Avg global cache cr block receive time (ms):	0.6
Avg global cache current block receive time (ms):	1.1
Avg global cache cr block build time (ms):	0.0
Avg global cache cr block send time (ms):	0.1
Global cache log flushes for cr blocks served %:	3.2
Avg global cache cr block flush time (ms):	4.0
Avg global cache current block pin time (ms):	0.4
Avg global cache current block send time (ms):	0.1
Global cache log flushes for current blocks served %:	2.9
Avg global cache current block flush time (ms):	35.5

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Typical Latencies for RAC Operations

In a RAC AWR report, there is a table in the RAC Statistics section containing average times (latencies) for some Global Cache Services and Global Enqueue Services operations. This table is shown in the slide and is called “Global Cache and Enqueue Services: Workload Characteristics.” Those latencies should be monitored over time, and significant increases in their values should be investigated. The table presents some typical values, based on empirical observations. Factors that may cause variations to those latencies include:

- Utilization of the IPC protocol. User-mode IPC protocols are faster, but only Tru64’s RDG is recommended for use.
- Scheduling delays, when the system is under high CPU utilization
- Log flushes for current blocks served

Other RAC latencies in AWR reports are mostly derived from `V$GES_STATISTICS` and may be useful for debugging purposes, but do not require frequent monitoring.

Note: The time to process consistent read (CR) block request in the cache corresponds to (build time + flush time + send time), and the time to process current block request in the cache corresponds to (pin time + flush time + send time).

Wait Events for RAC

- Wait events help to analyze what sessions are waiting for.
- Wait times are attributed to events that reflect the outcome of a request:
 - Placeholders while waiting
 - Precise events after waiting
- Global cache waits are summarized in a broader category called Cluster Wait Class.
- These wait events are used in ADDM to enable Cache Fusion diagnostics.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Wait Events for RAC

Analyzing what sessions are waiting for is an important method to determine where time is spent. In RAC, the wait time is attributed to an event that reflects the exact outcome of a request. For example, when a session on an instance is looking for a block in the global cache, it does not know whether it will receive the data cached by another instance or whether it will receive a message to read from disk. The wait events for the global cache convey precise information and wait for global cache blocks or messages. They are mainly categorized by the following:

- Summarized in a broader category called Cluster Wait Class
- Temporarily represented by a placeholder event that is active while waiting for a block
- Attributed to precise events when the outcome of the request is known

The wait events for RAC convey information valuable for performance analysis. They are used in ADDM to enable precise diagnostics of the impact of cache fusion.

Wait Event Views

Total waits for an event	V\$SYSTEM_EVENT
Waits for a wait event class by a session	V\$SESSION_WAIT_CLASS
Waits for an event by a session	V\$SESSION_EVENT
Activity of recent active sessions	V\$ACTIVE_SESSION_HISTORY
Last 10 wait events for each active session	V\$SESSION_WAIT_HISTORY
Events for which active sessions are waiting	V\$SESSION_WAIT
Identify SQL statements impacted by interconnect latencies	V\$SQLSTATS

ORACLE

Copyright © 2010, Oracle. All rights reserved.

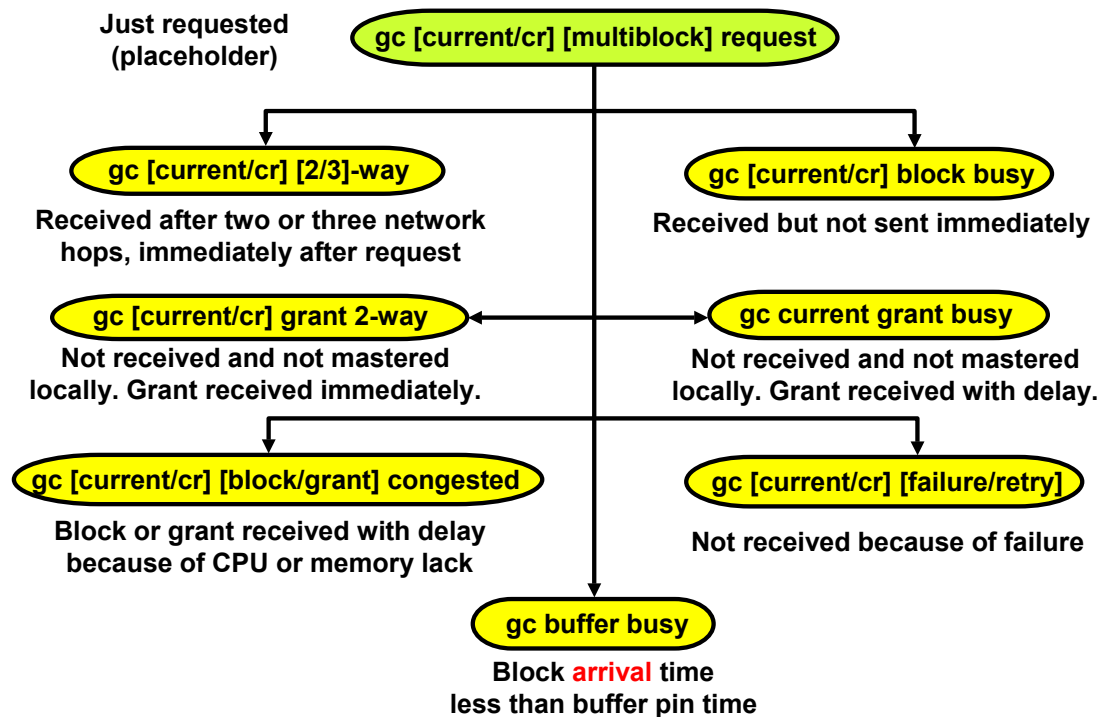
Wait Event Views

When it takes some time to acquire resources because of the total path length and latency for requests, processes sleep to avoid spinning for indeterminate periods of time. When the process decides to wait, it wakes up either after a specified timer value expires (timeout) or when the event it is waiting for occurs and the process is posted. The wait events are recorded and aggregated in the views shown in the slide. The first three are aggregations of wait times, timeouts, and the number of times waited for a particular event, whereas the rest enable the monitoring of waiting sessions in real time, including a history of recent events waited for.

The individual events distinguish themselves by their names and the parameters that they assume. For most of the global cache wait events, the parameters include file number, block number, the block class, and access mode dispositions, such as mode held and requested. The wait times for events presented and aggregated in these views are very useful when debugging response time performance issues. Note that the time waited is cumulative, and that the event with the highest score is not necessarily a problem. However, if the available CPU power cannot be maximized, or response times for an application are too high, the top wait events provide valuable performance diagnostics.

Note: Use the `CLUSTER_WAIT_TIME` column in `V$SQLSTATS` to identify SQL statements impacted by interconnect latencies, or run an ADDM report on the corresponding AWR snapshot.

Global Cache Wait Events: Overview



Copyright © 2010, Oracle. All rights reserved.

ORACLE

Oracle Internal & Oracle Academy Use Only

Global Cache Wait Events: Overview

The main global cache wait events are described briefly in the slide:

- **gc current/cr request:** These wait events are relevant only while a gc request for a cr block or current buffer is in progress. They act as placeholders until the request completes.
- **gc [current/cr] [2/3]-way:** A current or cr block is requested and received after two or three network hops. The request is processed immediately; the block is not busy or congested.
- **gc [current/cr] block busy:** A current or cr block is requested and received, but is not sent immediately by LMS because some special condition that delayed the sending was found.
- **gc [current/cr] grant 2-way:** A current or cr block is requested and a grant message received. The grant is given without any significant delays. If the block is not in its local cache, a current or cr grant is followed by a disk read on the requesting instance.
- **gc current grant busy:** A current block is requested and a grant message received. The busy hint implies that the request is blocked because others are ahead of it or it cannot be handled immediately.

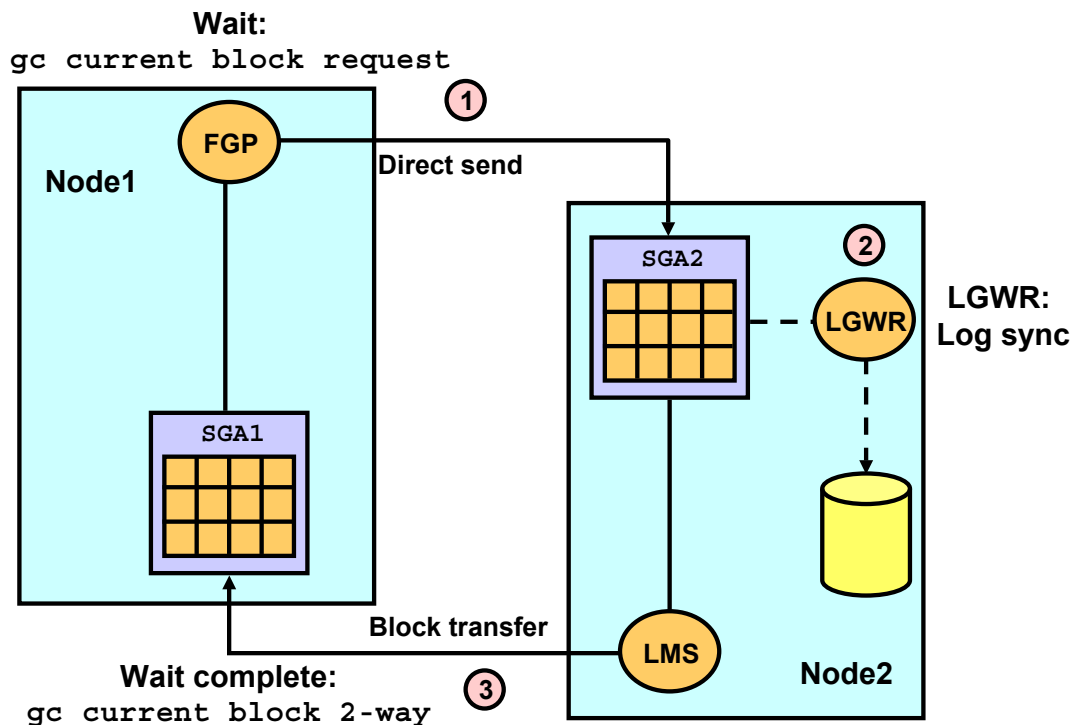
Note: For dynamic remastering, two events are of most importance: `gc remaster` and `gc quiesce`. They can be symptoms of the impact of remastering on the running processes.

Global Cache Wait Events: Overview (continued)

- **gc [current/cr] [block/grant] congested:** A current or cr block is requested and a block or grant message received. The congested hint implies that the request spent more than 1 ms in internal queues.
- **gc [current/cr] [failure/retry]:** A block is requested and a failure status received or some other exceptional event has occurred.
- **gc buffer busy:** If the time between buffer accesses becomes less than the time the buffer is pinned in memory, the buffer containing a block is said to become busy and as a result interested users may have to wait for it to be unpinned.

Note: For more information, refer to *Oracle Database Reference*.

2-way Block Request: Example



ORACLE

Copyright © 2010, Oracle. All rights reserved.

2-way Block Request: Example

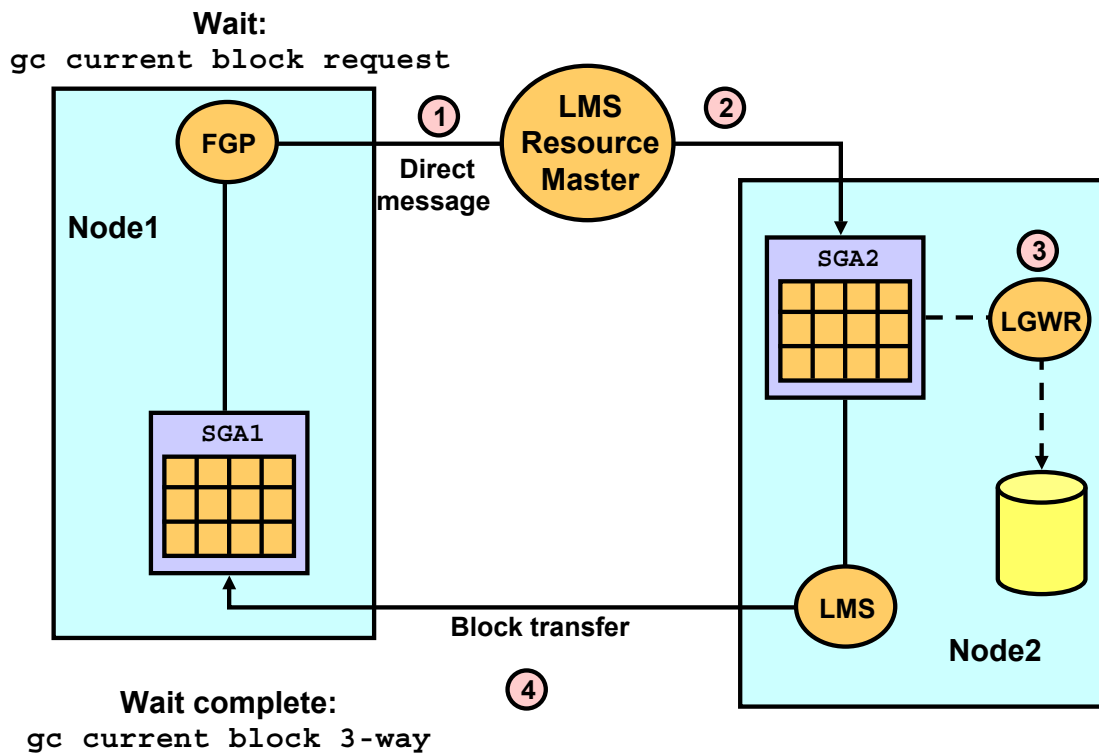
This slide shows you what typically happens when the master instance requests a block that is not cached locally. Here it is supposed that the master instance is called SGA1, and SGA2 contains the requested block. The scenario is as follows:

1. SGA1 sends a direct request to SGA2. So SGA1 waits on the `gc current block request` event.
2. When SGA2 receives the request, its local LGWR process may need to flush some recovery information to its local redo log files. For example, if the cached block is frequently changed, and the changes have not been logged yet, LMS would have to ask LGWR to flush the log before it can ship the block. This may add a delay to the serving of the block and may show up in the requesting node as a busy wait.
3. Then, SGA2 sends the requested block to SGA1. When the block arrives in SGA1, the wait event is complete, and is reflected as `gc current block 2-way`.

Note: Using the notation R = time at requestor, W = wire time and transfer delay, and S = time at server, the total time for a round-trip would be:

$R(\text{send}) + W(\text{small msg}) + S(\text{process msg, process block, send}) + W(\text{block}) + R(\text{receive block})$

3-way Block Request: Example



ORACLE

Copyright © 2010, Oracle. All rights reserved.

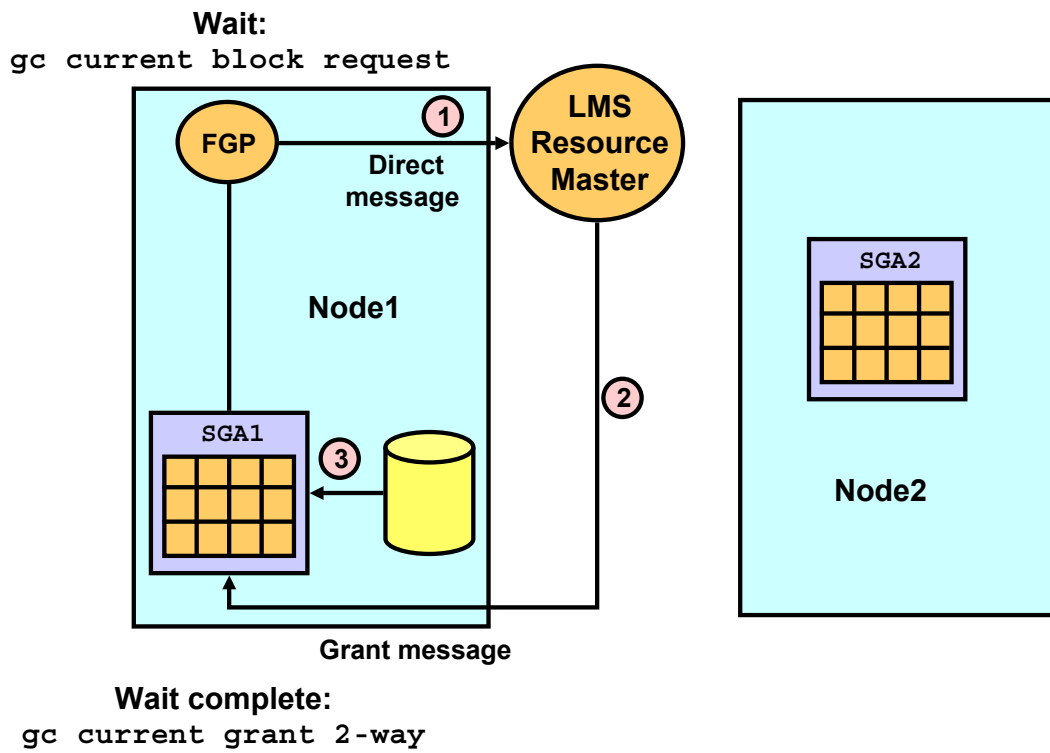
3-way Block Request: Example

This is a modified scenario for a cluster with more than two nodes. It is very similar to the previous one. However, the master for this block is on a node that is different from that of the requestor, and where the block is cached. Thus, the request must be forwarded. There is an additional delay for one message and the processing at the master node:

$R(\text{send}) + W(\text{small msg}) + S(\text{process msg, send}) + W(\text{small msg}) + S(\text{process msg, process block, send}) + W(\text{block}) + R(\text{receive block})$

While a remote read is pending, any process on the requesting instance that is trying to write or read the data cached in the buffer has to wait for a `gc buffer busy`. The buffer remains globally busy until the block arrives.

2-way Grant: Example



ORACLE

Copyright © 2010, Oracle. All rights reserved.

2-way Grant: Example

In this scenario, a grant message is sent by the master because the requested block is not cached in any instance.

If the local instance is the resource master, the grant happens immediately. If not, the grant is always 2-way, regardless of the number of instances in the cluster.

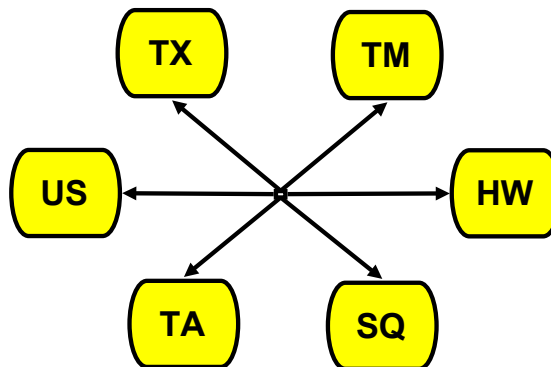
The grant messages are small. For every block read from the disk, a grant has to be received before the I/O is initiated, which adds the latency of the grant round-trip to the disk latency:

$R(\text{send}) + W(\text{small msg}) + S(\text{process msg, send}) + W(\text{small msg}) + R(\text{receive block})$

The round-trip looks similar to a 2-way block round-trip, with the difference that the wire time is determined by a small message, and the processing does not involve the buffer cache.

Global Enqueue Waits: Overview

- Enqueues are synchronous.
- Enqueues are global resources in RAC.
- The most frequent waits are for:



- The waits may constitute serious serialization points.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Global Enqueue Waits: Overview

An enqueue wait is not RAC specific, but involves a global lock operation when RAC is enabled. Most of the global requests for enqueues are synchronous, and foreground processes wait for them. Therefore, contention on enqueues in RAC is more visible than in single-instance environments. Most waits for enqueues occur for enqueues of the following types:

- **TX:** Transaction enqueue; used for transaction demarcation and tracking
- **TM:** Table or partition enqueue; used to protect table definitions during DML operations
- **HW:** High-water mark enqueue; acquired to synchronize a new block operation
- **SQ:** Sequence enqueue; used to serialize incrementing of an Oracle sequence number
- **US:** Undo segment enqueue; mainly used by the Automatic Undo Management (AUM) feature
- **TA:** Enqueue used mainly for transaction recovery as part of instance recovery

In all of the cases above, the waits are synchronous and may constitute serious serialization points that can be exacerbated in a RAC environment.

Note: The enqueue wait events specify the resource name and a reason for the wait—for example, “TX Enqueue index block split.” This makes diagnostics of enqueue waits easier.

Session and System Statistics

- Use V\$SYSSTAT to characterize the workload.
- Use V\$SESSTAT to monitor important sessions.
- V\$SEGMENT_STATISTICS includes RAC statistics.
- RAC-relevant statistic groups are:
 - Global Cache Service statistics
 - Global Enqueue Service statistics
 - Statistics for messages sent
- V\$ENQUEUE_STATISTICS determines the enqueue with the highest impact.
- V\$INSTANCE_CACHE_TRANSFER breaks down GCS statistics into block classes.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Session and System Statistics

Using system statistics based on V\$SYSSTAT enables characterization of the database activity based on averages. It is the basis for many metrics and ratios used in various tools and methods, such as AWR, Statspack, and Database Control.

In order to drill down to individual sessions or groups of sessions, V\$SESSTAT is useful when the important session identifiers to monitor are known. Its usefulness is enhanced if an application fills in the MODULE and ACTION columns in V\$SESSION.

V\$SEGMENT_STATISTICS is useful for RAC because it also tracks the number of CR and current blocks received by the object.

The RAC-relevant statistics can be grouped into:

- **Global Cache Service statistics:** *gc cr blocks received*, *gc cr block receive time*, and so on
- **Global Enqueue Service statistics:** *global enqueue gets*, and so on
- **Statistics for messages sent:** *gcs messages sent* and *ges messages sent*

V\$ENQUEUE_STATISTICS can be queried to determine which enqueue has the highest impact on database service times and eventually, response times.

V\$INSTANCE_CACHE_TRANSFER indicates how many current and CR blocks per block class are received from each instance, including how many transfers incurred a delay.

Note: For more information about statistics, refer to *Oracle Database Reference*.

Most Common RAC Tuning Tips

Application tuning is often the most beneficial!

- Resize and tune the buffer cache.
- Reduce long full-table scans in OLTP systems.
- Use Automatic Segment Space Management (ASSM).
- Increase sequence caches.
- Use partitioning to reduce interinstance traffic.
- Avoid unnecessary parsing.
- Minimize locking usage.
- Remove unselective indexes.
- Configure interconnect properly.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Most Common RAC Tuning Tips

In any database system, RAC or single instance, the most significant performance gains are usually obtained from traditional application-tuning techniques. The benefits of those techniques are even more remarkable in a RAC database. In addition to traditional application tuning, some of the techniques that are particularly important for RAC include the following:

- Try to avoid long full-table scans to minimize GCS requests. The overhead caused by the global CR requests in this scenario is because when queries result in local cache misses, an attempt is first made to find the data in another cache, based on the assumption that the chance is high that another instance has cached the block.
- Automatic Segment Space Management can provide instance affinity to table blocks.
- Increasing sequence caches improves instance affinity to index keys deriving their values from sequences. That technique may result in significant performance gains for multi-instance insert-intensive applications.
- Range or list partitioning may be very effective in conjunction with data-dependent routing, if the workload can be directed to modify a particular range of values from a particular instance.
- Hash partitioning may help to reduce buffer busy contention by making buffer access distribution patterns sparser, enabling more buffers to be available for concurrent access.

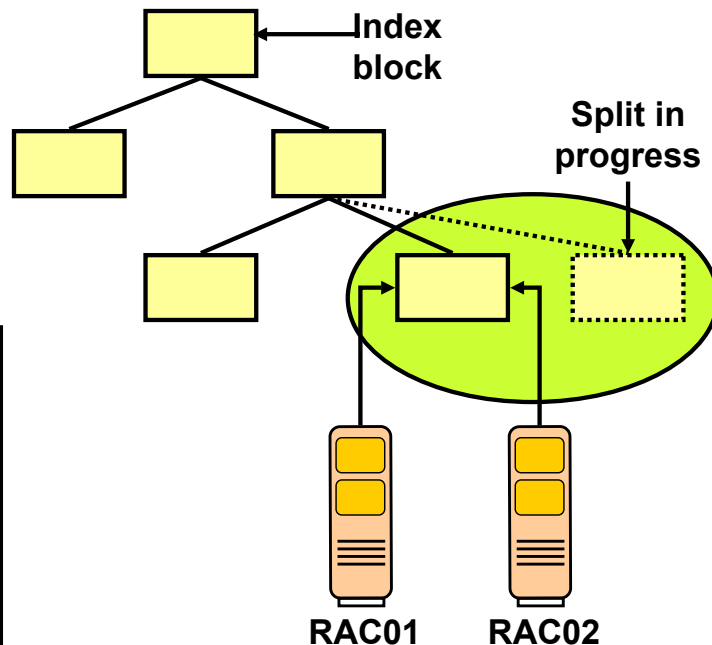
Most Common RAC Tuning Tips (continued)

- In RAC, library cache and row cache operations are globally coordinated. So, excessive parsing means additional interconnect traffic. Library cache locks are heavily used, in particular by applications using PL/SQL or Advanced Queuing. Library cache locks are acquired in exclusive mode whenever a package or procedure has to be recompiled.
- Because transaction locks are globally coordinated, they also deserve special attention in RAC. For example, using tables instead of Oracle sequences to generate unique numbers is not recommended because it may cause severe contention even for a single instance system.
- Indexes that are not selective do not improve query performance, but can degrade DML performance. In RAC, unselective index blocks may be subject to inter-instance contention, increasing the frequency of cache transfers for indexes belonging to insert-intensive tables.
- Always verify that you use a private network for your interconnect, and that your private network is configured properly. Ensure that a network link is operating in full duplex mode. Ensure that your network interface and Ethernet switches support MTU size of 9 KB. Note that a single gigabit Ethernet interface can scale up to ten thousand 8-KB blocks per second before saturation.

Index Block Contention: Considerations

Wait events
enq: TX - index contention
gc buffer busy
gc current block busy
gc current split

System statistics
Leaf node splits
Branch node splits
Exchange deadlocks
gcs refuse xid
gcs ast xid
Service ITL waits



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Index Block Contention: Considerations

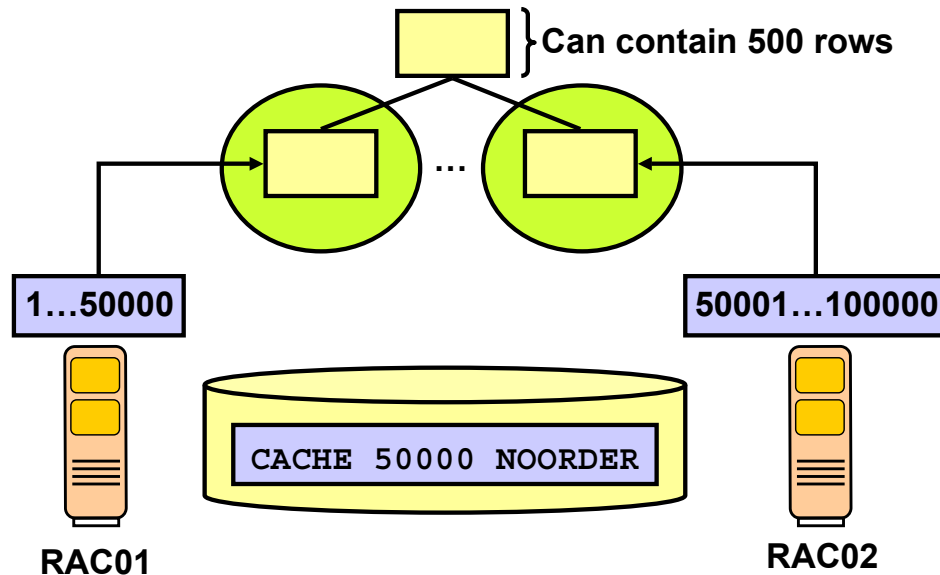
In application systems where the loading or batch processing of data is a dominant business function, there may be performance issues affecting response times because of the high volume of data inserted into indexes. Depending on the access frequency and the number of processes concurrently inserting data, indexes can become hot spots and contention can be exacerbated by:

- Ordered, monotonically increasing key values in the index (right-growing trees)
- Frequent leaf block splits
- Low tree depth: All leaf block access go through the root block.

A leaf or branch block split can become an important serialization point if the particular leaf block or branch of the tree is concurrently accessed. The tables in the slide sum up the most common symptoms associated with the splitting of index blocks, listing wait events and statistics that are commonly elevated when index block splits are prevalent. As a general recommendation, to alleviate the performance impact of globally hot index blocks and leaf block splits, a more uniform, less skewed distribution of the concurrency in the index tree should be the primary objective. This can be achieved by:

- Global index hash partitioning
- Increasing the sequence cache, if the key value is derived from a sequence
- Using natural keys as opposed to surrogate keys
- Using reverse key indexes

Oracle Sequences and Index Contention



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Sequences and Index Contention

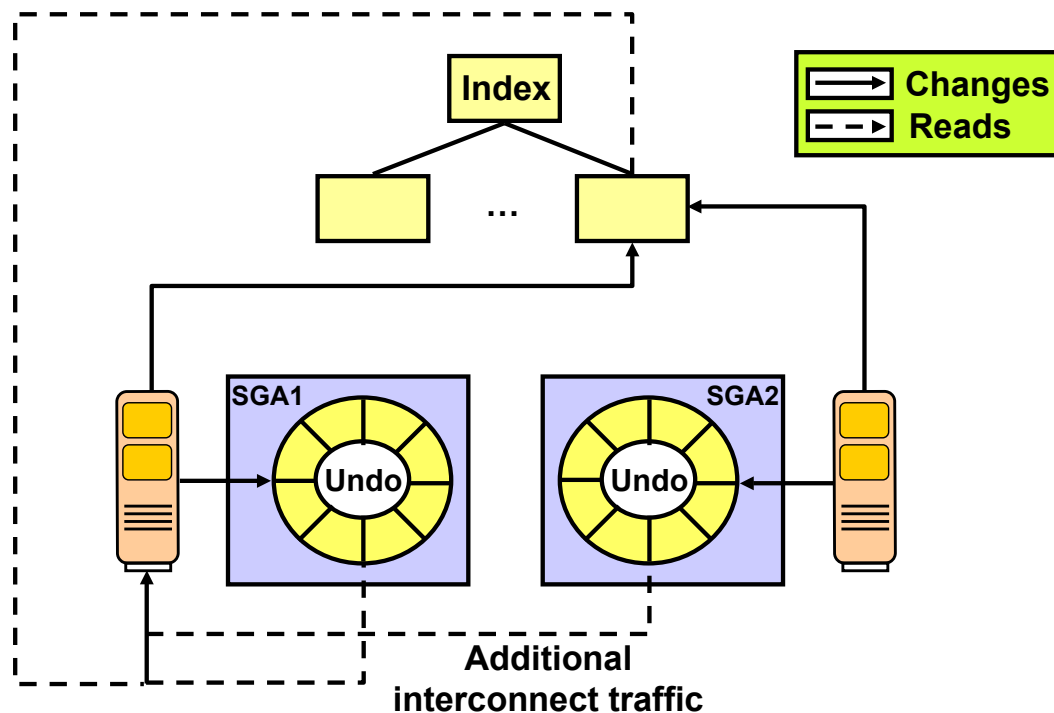
Indexes with key values generated by sequences tend to be subject to leaf block contention when the insert rate is high. That is because the index leaf block holding the highest key value is changed for every row inserted, as the values are monotonically ascending. In RAC, this may lead to a high rate of current and CR blocks transferred between nodes.

One of the simplest techniques that can be used to limit this overhead is to increase the sequence cache, if you are using Oracle sequences. Because the difference between sequence values generated by different instances increases, successive index block splits tend to create instance affinity to index leaf blocks. For example, suppose that an index key value is generated by a `CACHE NOORDER` sequence and each index leaf block can hold 500 rows. If the sequence cache is set to 50000, while instance 1 inserts values 1, 2, 3, and so on, instance 2 concurrently inserts 50001, 50002, and so on. After some block splits, each instance writes to a different part of the index tree.

So, what is the ideal value for a sequence cache to avoid inter-instance leaf index block contention, yet minimizing possible gaps? One of the main variables to consider is the insert rate: the higher it is, the higher must be the sequence cache. However, creating a simulation to evaluate the gains for a specific configuration is recommended.

Note: By default, the cache value is 20. Typically, 20 is too small for the preceding example.

Undo Block Considerations



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Undo Block Considerations

Excessive undo block shipment and contention for undo buffers usually happens when index blocks containing active transactions from multiple instances are read frequently.

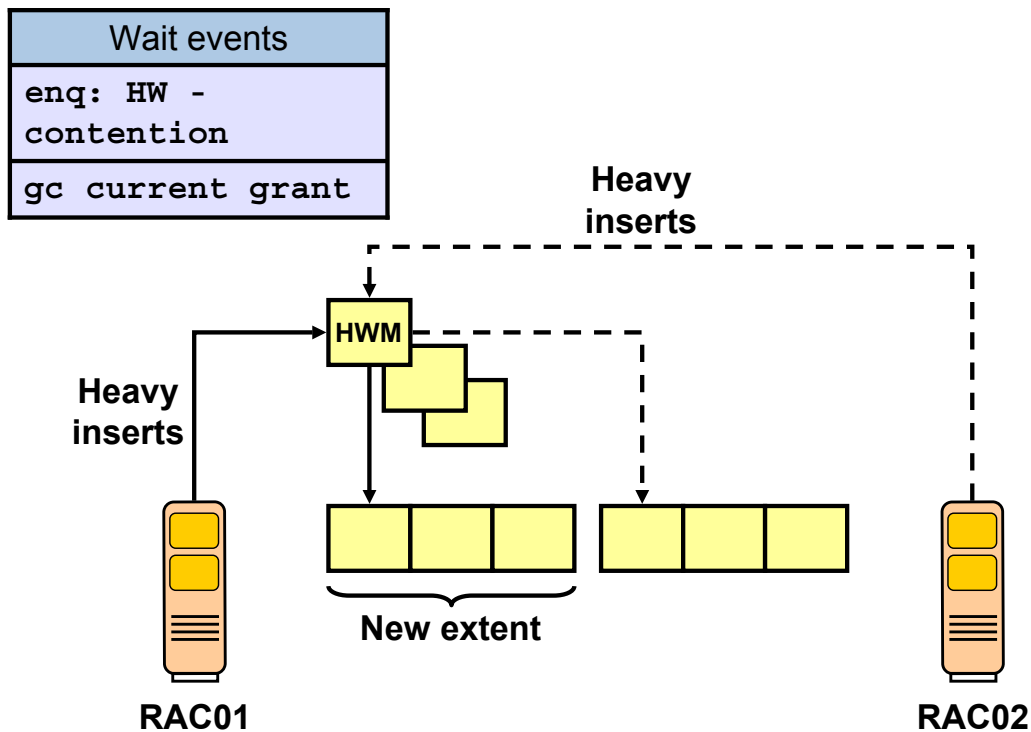
When a `SELECT` statement needs to read a block with active transactions, it has to undo the changes to create a CR version. If the active transactions in the block belong to more than one instance, there is a need to combine local and remote undo information for the consistent read. Depending on the amount of index blocks changed by multiple instances and the duration of the transactions, undo block shipment may become a bottleneck.

Usually this happens in applications that read recently inserted data very frequently, but commit infrequently. Techniques that can be used to reduce such situations include the following:

- Shorter transactions reduce the likelihood that any given index block in the cache contains uncommitted data, thereby reducing the need to access undo information for consistent read.
- As explained earlier, increasing sequence cache sizes can reduce inter-instance concurrent access to index leaf blocks. CR versions of index blocks modified by only one instance can be fabricated without the need of remote undo information.

Note: In RAC, the problem is exacerbated by the fact that a subset of the undo information has to be obtained from remote instances.

High-Water Mark Considerations



ORACLE

Copyright © 2010, Oracle. All rights reserved.

High-Water Mark Considerations

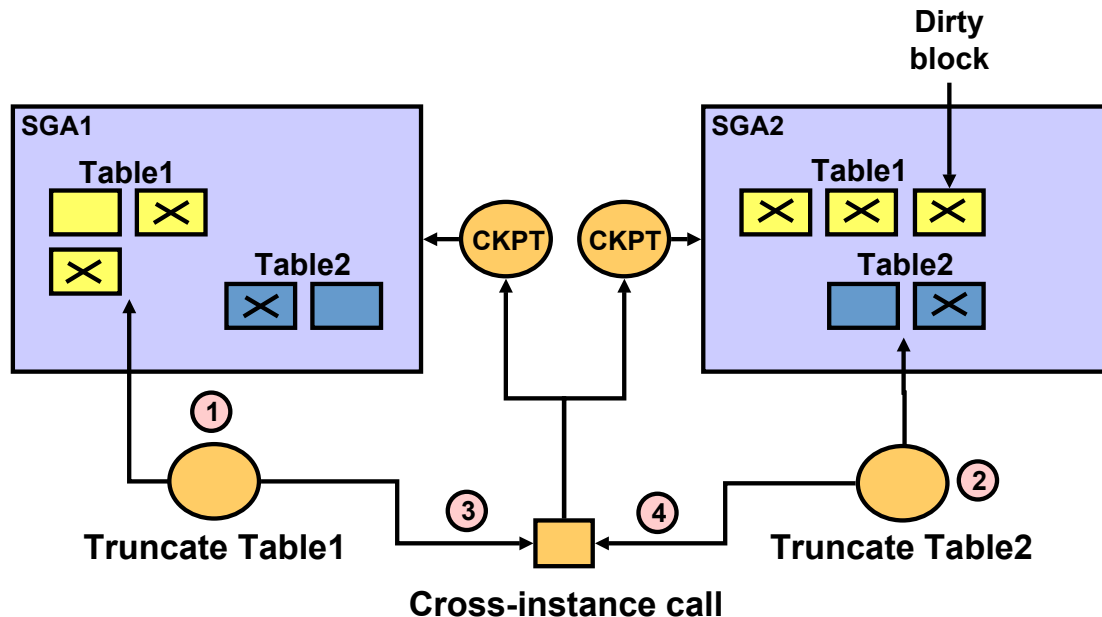
A certain combination of wait events and statistics presents itself in applications where the insertion of data is a dominant business function and new blocks have to be allocated frequently to a segment. If data is inserted at a high rate, new blocks may have to be made available after unfruitful searches for free space. This has to happen while holding the high-water mark (HWM) enqueue.

Therefore, the most common symptoms for this scenario include:

- A high percentage of wait time for enq: HW - contention
- A high percentage of wait time for gc current grant events

The former is a consequence of the serialization on the HWM enqueue, and the latter is because of the fact that current access to the new data blocks that need formatting is required for the new block operation. In a RAC environment, the length of this space management operation is proportional to the time it takes to acquire the HWM enqueue and the time it takes to acquire global locks for all the new blocks that need formatting. This time is small under normal circumstances because there is never any access conflict for the new blocks. Therefore, this scenario may be observed in applications with business functions requiring a lot of data loading, and the main recommendation to alleviate the symptoms is to define uniform and large extent sizes for the locally managed and automatic space managed segments that are subject to high-volume inserts.

Concurrent Cross-Instance Calls: Considerations



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Concurrent Cross-Instance Calls: Considerations

In data warehouse and data mart environments, it is not uncommon to see a lot of TRUNCATE operations. These essentially happen on tables containing temporary data.

In a RAC environment, truncating tables concurrently from different instances does not scale well, especially if, in conjunction, you are also using direct read operations such as parallel queries.

As shown in the slide, a truncate operation requires a cross-instance call to flush dirty blocks of the table that may be spread across instances. This constitutes a point of serialization. So, while the first TRUNCATE command is processing, the second has to wait until the first one completes.

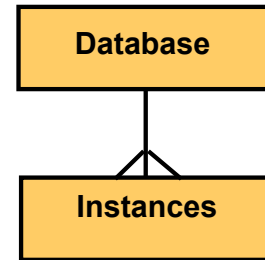
There are different types of cross-instance calls. However, all use the same serialization mechanism.

For example, the cache flush for a partitioned table with many partitions may add latency to a corresponding parallel query. This is because each cross-instance call is serialized at the cluster level, and one cross-instance call is needed for each partition at the start of the parallel query for direct read purposes.

Monitoring RAC Database and Cluster Performance

Directly from Database Control and Grid Control:

- View the status of each node in the cluster.
- View the aggregated alert messages across all the instances.
- Review the issues that are affecting the entire cluster or each instance.
- Monitor the cluster cache coherency statistics.
- Determine whether any of the services for the cluster database are having availability problems.
- Review any outstanding Clusterware interconnect alerts.



ORACLE

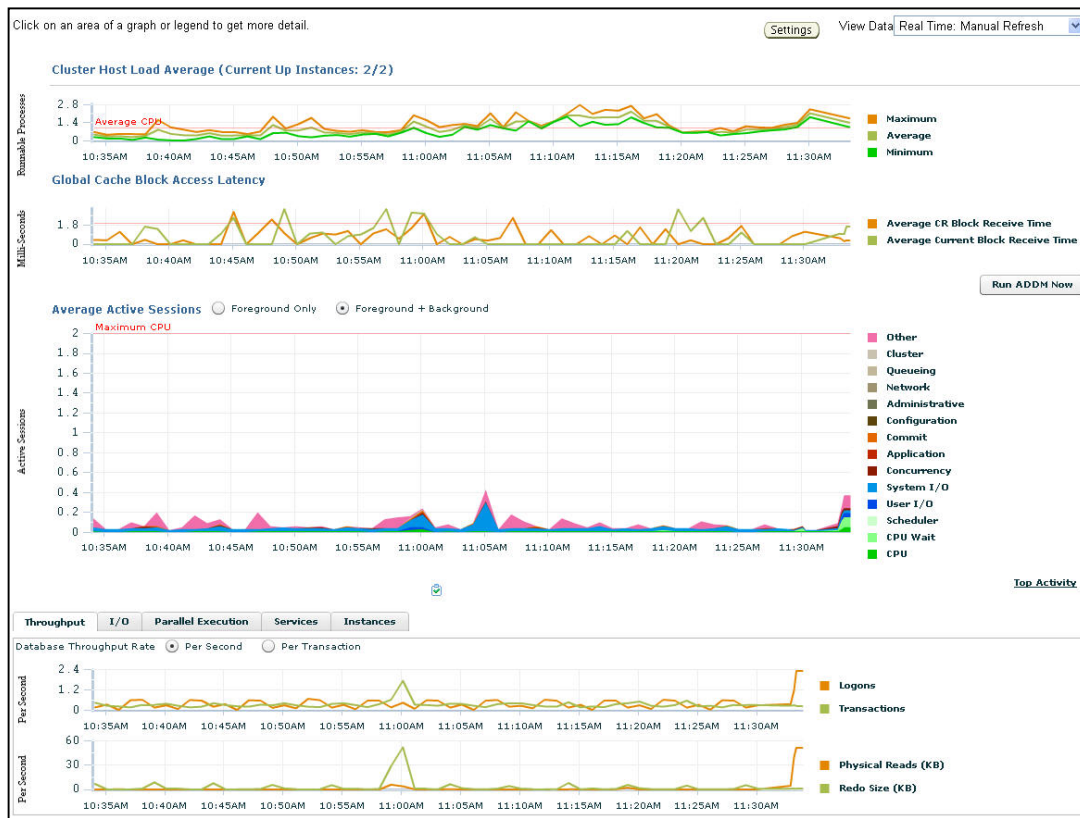
Copyright © 2010, Oracle. All rights reserved.

Monitoring RAC Database and Cluster Performance

Both Oracle Enterprise Manager Database Control and Grid Control are cluster-aware and provide a central console to manage your cluster database. From the Cluster Database Home page, you can do all of the following:

- View the overall system status, such as the number of nodes in the cluster and their current status, so you do not have to access each individual database instance for details
- View the alert messages aggregated across all the instances with lists for the source of each alert message.
- Review the issues that are affecting the entire cluster as well as those that are affecting individual instances.
- Monitor cluster cache coherency statistics to help you identify processing trends and optimize performance for your Oracle RAC environment. Cache coherency statistics measure how well the data in caches on multiple instances is synchronized.
- Determine whether any of the services for the cluster database are having availability problems. A service is deemed to be a problem service if it is not running on all preferred instances, if its response time thresholds are not met, and so on.
- Review any outstanding Clusterware interconnect alerts.

Cluster Database Performance Page



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Cluster Database Performance Page

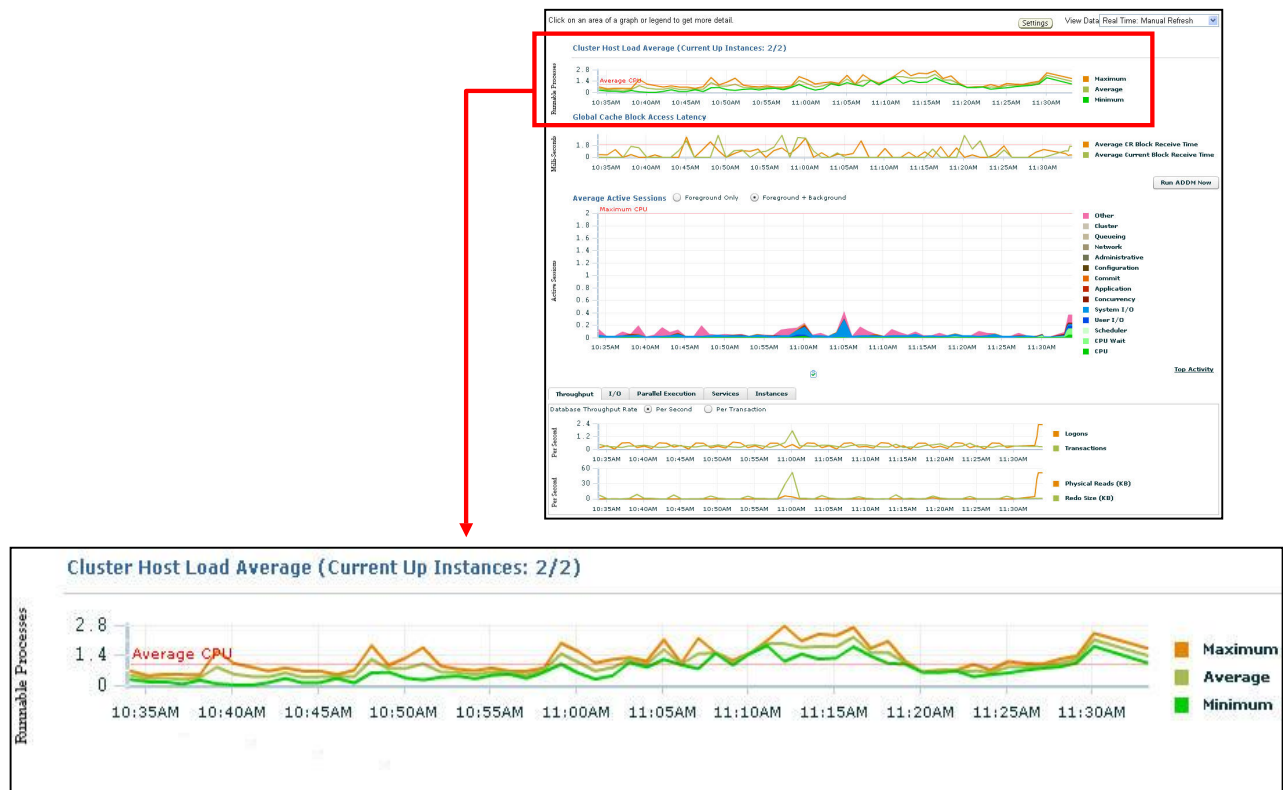
The Cluster Database Performance page provides a quick glimpse of the performance statistics for a database. Enterprise Manager accumulates data from each instance over specified periods of time, called collection-based data. Enterprise Manager also provides current data from each instance, known as real-time data.

Statistics are rolled up across all the instances in the cluster database. 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 need to be added or redistributed.
- Tune your SQL plan and schema for better optimization.
- Resolve performance issues.

The screenshot in the slide shows a partial view of the Cluster Database Performance page. You access this page by clicking the Performance tab from the Cluster Database Home page.

Determining Cluster Host Load Average



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Determining Cluster Host Load Average

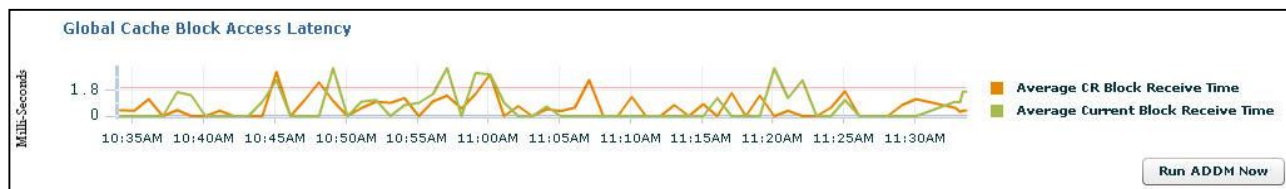
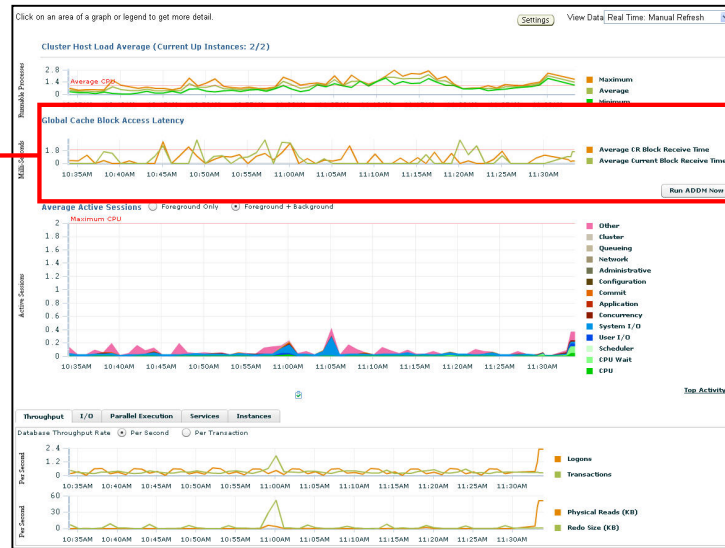
The Cluster Host Load Average chart in the Cluster Database Performance page shows potential problems that are outside the database. The chart shows maximum, average, and minimum load values for available nodes in the cluster for the previous hour.

If the load average is higher than the average of the total number of CPUs across all the hosts in the cluster, then too many processes are waiting for CPU resources. SQL statements that are not tuned often cause high CPU usage. Compare the load average values with the values displayed for CPU Used in the Average Active Sessions chart. If the sessions value is low and the load average value is high, this indicates that something else on the host, other than your database, is consuming the CPU.

You can click any of the load value labels for the Cluster Host Load Average chart to view more detailed information about that load value. For example, if you click the Average label, the Hosts: Average Load page appears, displaying charts that depict the average host load for up to four nodes in the cluster.

You can select whether the data is displayed in a summary chart, combining the data for each node in one display, or using tile charts, where the data for each node is displayed in its own chart. You can click Customize to change the number of tile charts displayed in each row or the method of ordering the tile charts.

Determining Global Cache Block Access Latency



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Determining Global Cache Block Access Latency

The Global Cache Block Access Latency chart shows the latency for each type of data block requests: current and consistent-read (CR) blocks. That is the elapsed time it takes to locate and transfer consistent-read and current blocks between the buffer caches.

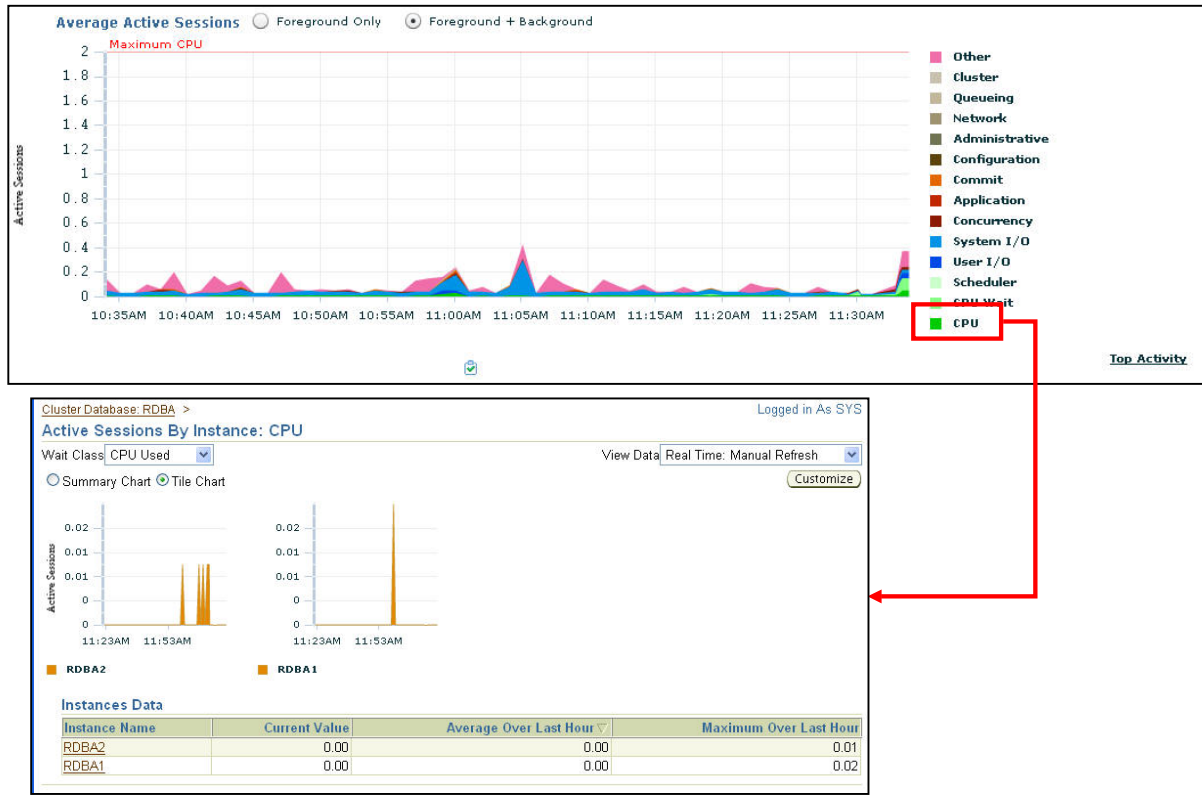
You can click either metric for the Global Cache Block Access Latency chart to view more detailed information about that type of cached block.

If the Global Cache Block Access Latency chart shows high latencies (high elapsed times), this can be caused by any of the following:

- A high number of requests caused by SQL statements that are not tuned
- A large number of processes in the queue waiting for the CPU, or scheduling delays
- Slow, busy, or faulty interconnects. In these cases, check your network connection for dropped packets, retransmittals, or cyclic redundancy check (CRC) errors.

Concurrent read and write activity on shared data in a cluster is a frequently occurring activity. Depending on the service requirements, this activity does not usually cause performance problems. However, when global cache requests cause a performance problem, optimizing SQL plans and the schema to improve the rate at which data blocks are located in the local buffer cache, and minimizing I/O is a successful strategy for performance tuning. If the latency for consistent-read and current block requests reaches 10 milliseconds, then see the Cluster Cache Coherency page for more detailed information.

Determining Average Active Sessions



Copyright © 2010, Oracle. All rights reserved.

ORACLE

Determining Average Active Sessions

The Average Active Sessions chart on 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 with 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.

At the cluster database level, this chart shows the aggregate wait class statistics across all the instances. For a more detailed analysis, you can click the Clipboard icon at the bottom of the chart to view the ADDM analysis for the database for that time period.

If you click the wait class legends beside the Average Active Sessions chart, you can view instance-level information stored in "Active Sessions by Instance" pages. You can use the Wait Class action list on the "Active Sessions by Instance" page to view the different wait classes. The "Active Sessions by Instance" pages show the service times for up to four instances. Using the Customize button, you can select the instances that are displayed. You can view the data for the instances separately by using tile charts, or you can combine the data into a single summary chart.

Determining Database Throughput



Copyright © 2010, Oracle. All rights reserved.

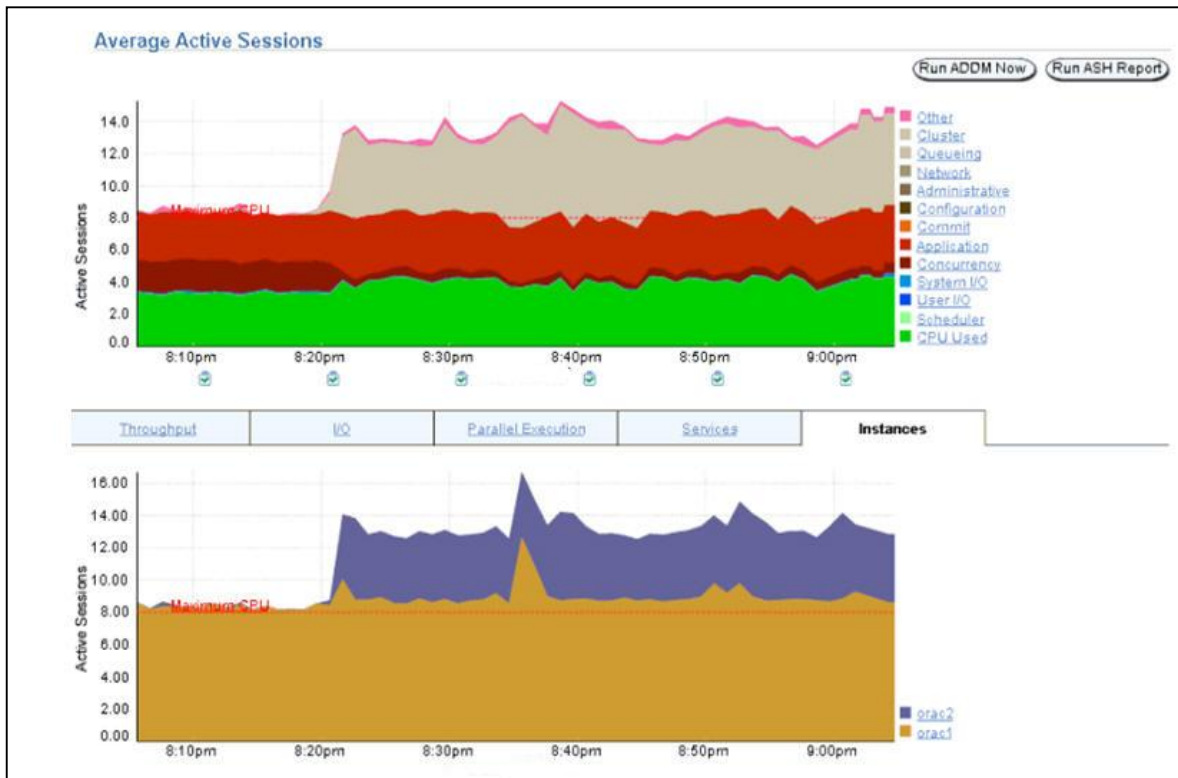
Determining Database Throughput

The last chart on the Performance page monitors the usage of various database resources. Click the Throughput tab at the top of this chart to view the Database Throughput chart. Compare the peaks on the Average Active Sessions chart with those on the Database Throughput charts. If internal contention is high and throughput is low, consider tuning the database.

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 (not shown here) the number of physical reads compared to the redo size per second. The Per Transaction view shows the number of physical reads compared to the redo size per transaction. Logons is the number of users that are logged on to the database.

To obtain information at the instance level, access the “Database Throughput by Instance” page by clicking one of the legends to the right of the charts. This page shows the breakdown of the aggregated Database Throughput chart for up to four instances. You can select the instances that are displayed. You can drill down further on the “Database Throughput by Instance” page to see the sessions of an instance consuming the greatest resources. Click an instance name legend under the chart to go to the Top Sessions sub-page of the Top Consumers page for that instance.

Determining Database Throughput



ORACLE

Copyright © 2010, Oracle. All rights reserved.

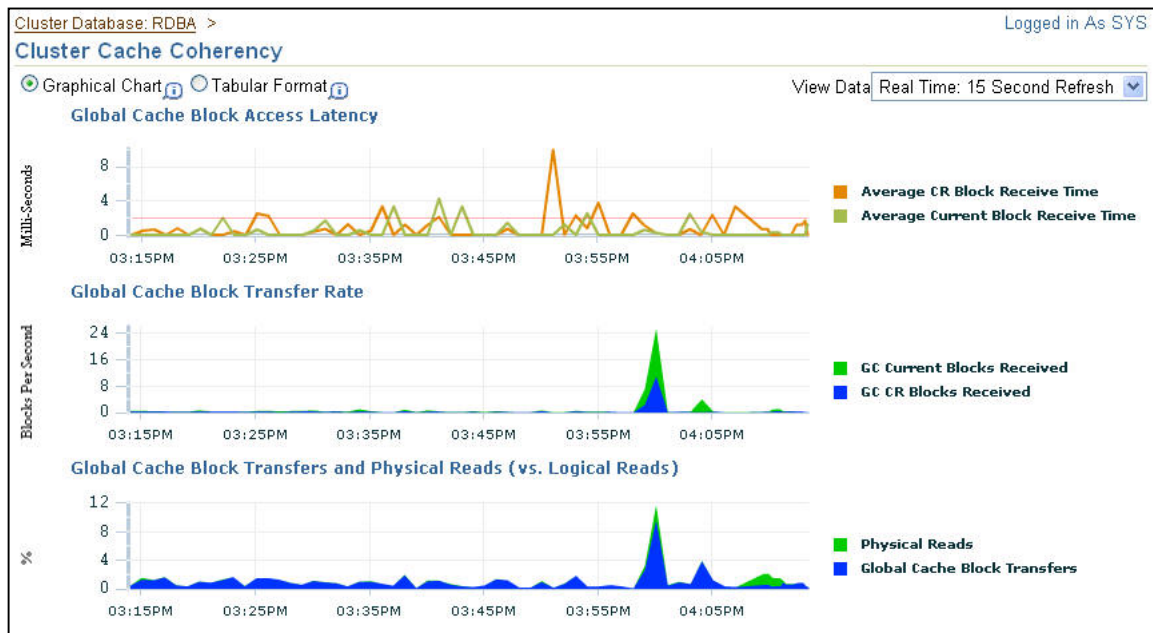
Determining Database Throughput (continued)

The last chart on the Performance page monitors the usage of various database resources. By clicking the Instances tab at the top of this chart, you can view the “Active Sessions by Instance” chart.

The “Active Sessions by Instance” chart summarizes any resource contention that appears in the Average Active Sessions chart. Using this chart, you can quickly determine how much of the database work is being performed on each instance.

You can also obtain information at the instance level by clicking one of the legends to the right of the chart to access the Top Sessions page. On the Top Session page, you can view real-time data showing the sessions that consume the greatest system resources. In the graph in the slide above, the orac2 instance after 8:20 PM is consistently showing more active sessions than the orac1 instance.

Accessing the Cluster Cache Coherency Page



Block
Class

Segment
name

Segment
name

ORACLE

Copyright © 2010, Oracle. All rights reserved.

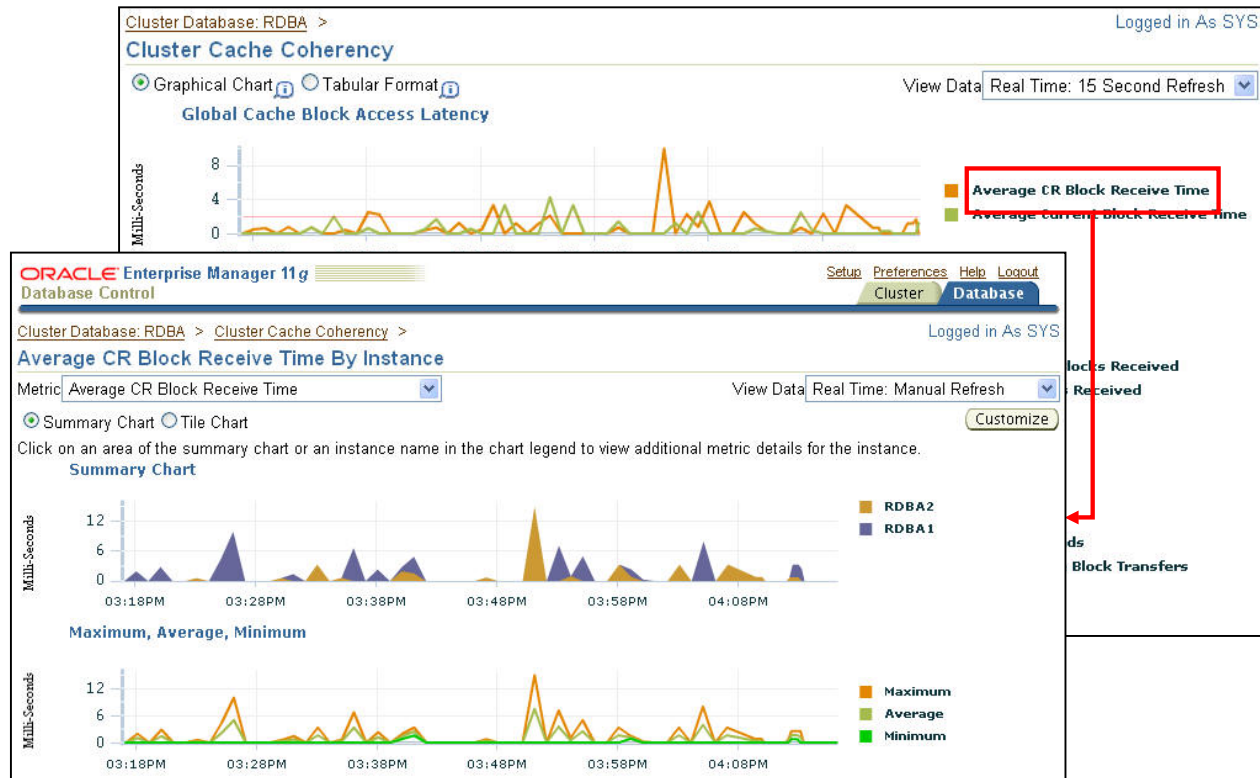
Accessing the Cluster Cache Coherency Page

To access the Cluster Cache Coherency page, click the Performance tab on the Cluster Database Home page, and click Cluster Cache Coherency in the Additional Monitoring Links section at the bottom of the page. Alternatively, click either of the legends to the right of the Global Cache Block Access Latency chart.

The Cluster Cache Coherency page contains summary charts for cache coherency metrics for the cluster:

- Global Cache Block Access Latency:** Shows the total elapsed time, or latency, for a block request. Click one of the legends to the right of the chart to view the average time it takes to receive data blocks for each block type (current or CR) by instance. On the “Average Block Receive Time by Instance” page, you can click an instance legend under the chart to go to the “Block Transfer for Local Instance” page, where you can identify which block classes, such as undo blocks, data blocks, and so on, are subject to intense global cache activity. This page displays the block classes that are being transferred, and which instances are transferring most of the blocks. Cache transfer indicates how many current and CR blocks for each block class were received from remote instances, including how many transfers incurred a delay (busy) or an unexpected longer delay (congested).

Accessing the Cluster Cache Coherency Page



Accessing the Cluster Cache Coherency Page (continued)

- Global Cache Block Transfer Rate:** Shows the total aggregated number of blocks received by all instances in the cluster by way of an interconnect. Click one of the legends to the right of the chart to go to the “Global Cache Blocks Received by Instance” page for that type of block. From there, you can click an instance legend under the chart to go to the “Segment Statistics by Instance” page, where you can see which segments are causing cache contention.
- Global Cache Block Transfers and Physical Reads:** Shows the percentage of logical read operations that retrieved data from the buffer cache of other instances by way of Direct Memory Access and from disk. It is essentially a profile of how much work is performed in the local buffer cache, rather than the portion of remote references and physical reads, which both have higher latencies. Click one of the legends to the right of the chart to go to the “Global Cache Block Transfers vs. Logical Reads by Instance” and “Physical Reads vs. Logical Reads by Instance” pages. From there, you can click an instance legend under the chart to go to the “Segment Statistics by Instance” page, where you can see which segments are causing cache contention.

Viewing the Cluster Interconnects Page

Cluster: cluster01 Latest Data Collected From Target **Aug 5, 2009 4:19:05 PM EDT** [Refresh](#)

[Home](#) [Performance](#) [Targets](#) [Administration](#) [Interconnects](#) [Topology](#)

The interconnect configuration and internode communication will influence the performance of cluster databases. The tables below show network interfaces on all hosts and network interfaces currently in use by cluster databases. It is important that cluster databases are configured to use a private interconnect for message and block transfers.

Private Interconnect Transfer Rate (MB/Sec) **0.055 ***
Transfer rate on the private network in the last 5 minutes.

View Data **Manually**

Interfaces by Hosts

View **Private**

[Expand All](#) | [Collapse All](#)

Name	Type	Subnet	Interface Type	Total I/O Rate (MB/Sec) (Last 5 Minutes)	Total Error Rate (%) (Last 5 Minutes)
cluster01	Cluster				
racnode01.example.com	Host				
eth1	Interface	192.168.1.0	Private	.073	0
racnode02.example.com	Host				
eth1	Interface	192.168.1.0	Private	.03 *	0 *

Interfaces in Use by Cluster Databases

[Expand All](#) | [Collapse All](#)

Name	Target Type	Interface Name	Host Name	IP Address	Interface Type	Source	Transfer Rate (MB/Sec) (Last 5 Minutes)
RDBA	Cluster Database						
RDBA1	Database Instance	eth1	racnode01.example.com	192.168.1.103	Private	Oracle Cluster Repository	.012
RDBA2	Database Instance	eth1	racnode02.example.com	192.168.1.107	Private	Oracle Cluster Repository	.0051 *

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Viewing the Cluster Interconnects Page

The Cluster Interconnects page is useful for monitoring the interconnect interfaces, determining configuration issues, and identifying transfer rate–related issues including excess traffic. This page helps to determine the load added by instances and databases on the interconnect. Sometimes you can quickly identify interconnect delays that are due to applications outside Oracle.

You can use this page to perform the following tasks:

- View all interfaces that are configured across the cluster
- View statistics for the interfaces, such as absolute transfer rates and errors
- Determine the type of interfaces, such as private or public
- Determine whether the instance is using a public or private network
- Determine which database instance is currently using which interface
- Determine how much the instance is contributing to the transfer rate on the interface

The Private Interconnect Transfer Rate value shows a global view of the private interconnect traffic, which is the estimated traffic on all the private networks in the cluster. The traffic is calculated as the summary of the input rate of all private interfaces known to the cluster.

From the Cluster Interconnects page, you can access the Hardware Details page, on which you can get more information about all the network interfaces defined on each node of your cluster.

Viewing the Cluster Interconnects Page (continued)

Similarly, you can access the “Transfer Rate metric” page, which collects the internode communication traffic of a cluster database instance. The critical and warning thresholds of this metric are not set by default. You can set them according to the speed of your cluster interconnects.

Note: You can query the `GV$CLUSTER_INTERCONNECTS` view to see information about the private interconnect:

```
SQL> select * from GV$CLUSTER_INTERCONNECTS;
```

INST_ID	NAME	IP_ADDRESS	IS_PUBLIC	SOURCE
1	eth1	192.0.2.110	NO	Oracle Cluster Repository
2	eth1	192.0.2.111	NO	Oracle Cluster Repository
3	eth1	192.0.2.112	NO	Oracle Cluster Repository

Viewing the Database Locks Page

Cluster Database: RDBA > Logged in As SYS

Database Locks Page Refreshed Aug 5, 2009 4:34:51 PM EDT [Refresh](#)

View: **All Database Locks**
☐ Blocking Locks
☐ User Locks
☒ All Database Locks

[Kill Session](#) [Session Details](#) [View Object](#) [View SQL](#)

[Expand All](#) | [Collapse All](#)

Select Username	Sessions	Instance	Blocked Name	Session ID	Serial Number	Process ID	SOL Hash Value	Lock Type	Mode Held	Mode Requested	Object Type	Object Owner	Object Name	ROWID	Time in current mode (seconds)
All Database Locks															
GEN0	0	RDBA2		4	1	4100		XR	NULL	NONE					689986
GEN0	0	RDBA1		4	1	6221		XR	NULL	NONE					689902
DBW0	0	RDBA1		17	1	6249		RT	EXCLUSIVE	NONE					689894
DBW0	0	RDBA1		17	1	6249		DM	SHARE	NONE					689893
DBW0	0	RDBA2		18	1	4137		DM	SHARE	NONE					689974
DBW0	0	RDBA2		18	1	4137		RT	EXCLUSIVE	NONE					689975
DBSNMP	0	RDBA2		34	45336	24475	1tu4ybcx7vks8	PS	SHARE	NONE					0
DBSNMP	0	RDBA1		41	3015	18368		AE	SHARE	NONE					428174
DBSNMP	0	RDBA1		43	53999	19695	1tu4ybcx7vks8	PS	SHARE	NONE					1
SYS	0	RDBA2		46	1	4323		AE	SHARE	NONE					689965
SYSMAN	0	RDBA1		46	1839	18972		AE	SHARE	NONE					428133
SYSMAN	0	RDBA2		48	35028	24448		TO	ROW EXCLUSIVE	NONE					426919
DBSNMP	0	RDBA2		59	11707	30013	1tu4ybcx7vks8	PS	SHARE	NONE					0
DBSNMP	0	RDBA2		59	11707	30013	1tu4ybcx7vks8	PS	SHARE	NONE					0
DBSNMP	0	RDBA2		59	11707	30013	1tu4ybcx7vks8	PS	SHARE	NONE					0
DBSNMP	0	RDBA2		59	11707	30013	1tu4ybcx7vks8	PS	SHARE	NONE					0
DBSNMP	0	RDBA2		59	11707	30013	1tu4ybcx7vks8	AE	SHARE	NONE					39

ORACLE

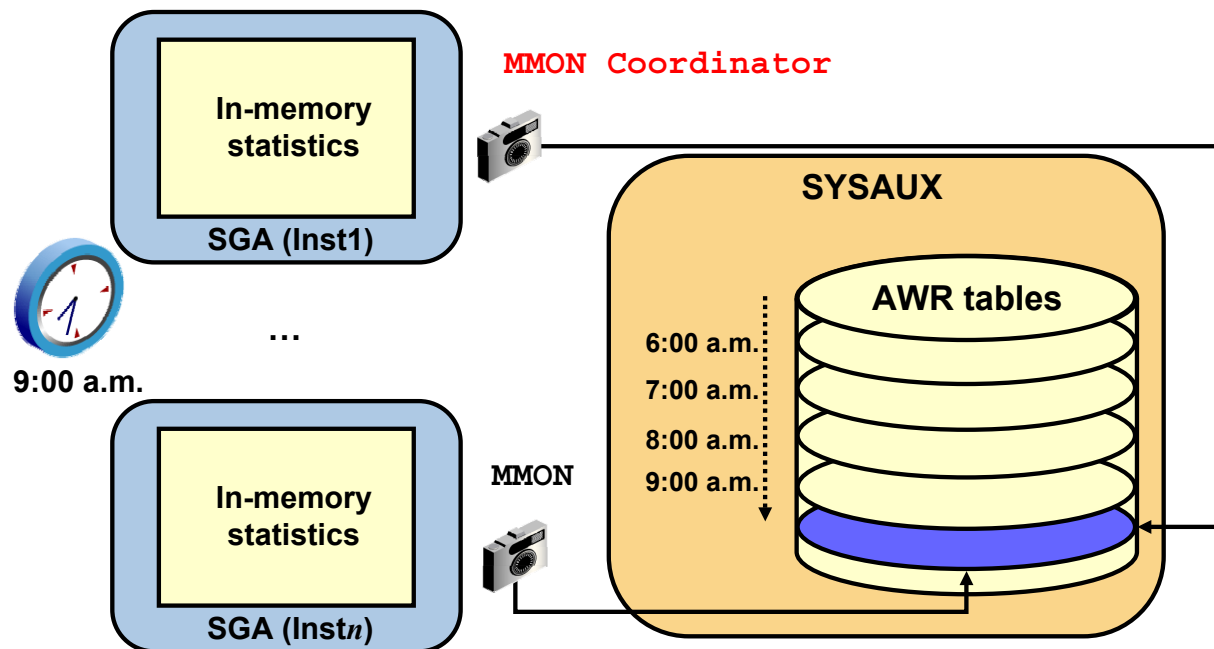
Copyright © 2010, Oracle. All rights reserved.

Viewing the Database Locks Page

Use the Database Locks page to determine whether multiple instances are holding locks for the same object. The page shows user locks, all database locks, or locks that are blocking other users or applications. You can use this information to stop a session that is unnecessarily locking an object.

To access the Database Locks page, select Performance on the Cluster Database Home page, and click Database Locks in the Additional Monitoring Links section at the bottom of the Performance subpage.

AWR Snapshots in RAC



ORACLE

Copyright © 2010, Oracle. All rights reserved.

AWR Snapshots in RAC

AWR automatically generates snapshots of the performance data once every hour and collects the statistics in the workload repository. In RAC environments, each AWR snapshot captures data from all active instances within the cluster. The data for each snapshot set that is captured for all active instances is from roughly the same point in time. In addition, the data for each instance is stored separately and is identified with an instance identifier. For example, the `buffer_busy_wait` statistic shows the number of buffer waits on each instance. The AWR does not store data that is aggregated from across the entire cluster. That is, the data is stored for each individual instance.

The statistics snapshots generated by the AWR can be evaluated by producing reports displaying summary data such as load and cluster profiles based on regular statistics and wait events gathered on each instance.

The AWR functions in a similar way as Statspack. The difference is that the AWR automatically collects and maintains performance statistics for problem detection and self-tuning purposes. Unlike in Statspack, in the AWR, there is only one `snapshot_id` per snapshot across instances.

AWR Reports and RAC: Overview

WORKLOAD REPOSITORY report for

DB Name	DB Id	Instance	Inst num	Startup Time	Release	RAC
RDBA	901665511	RDBA1	1	28-Jul-09 16:07	11.2.0.1.0	YES

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
racnode01.example.com	Linux IA (32-bit)	1			2.00

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	263	06-Aug-09 06:00:12	45	2.4
End Snap:	264	06-Aug-09 07:00:34	44	2.5
Elapsed:		60.38 (mins)		
DB Time:		0.84 (mins)		

Report Summary

Instance Efficiency Percentages (Target 100%)

Buffer Nowait %:	100.00	Redo NoWait %:	100.00
Buffer Hit %:	99.87	In-memory Sort %:	100.00
Library Hit %:	99.67	Soft Parse %:	98.75
Execute to Parse %:	62.60	Latch Hit %:	100.00
Parse CPU to Parse Elapsed %:	43.02	% Non-Parse CPU:	89.60

Shared Pool Statistics

	Begin	End
Memory Usage %:	84.60	84.50
% SQL with executions>1:	81.88	89.03
% Memory for SQL w/exec>1:	89.39	92.03

Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
relable message	1,613	32	20	64.01	Other
PX Deq: Slave Session Stats	948	10	11	20.56	Other
DB CPU		7		14.64	
log file sync	298	7	22	12.93	Commit
library cache pin	5,330	2	0	4.10	Concurrency

RAC Statistics

	Begin	End
Number of Instances:	2	2

Global Cache Load Profile

Global Cache Efficiency Percentages (Target local+remote 100%)

Global Cache and Enqueue Services - Workload Characteristics

Global Cache and Enqueue Services - Messaging Statistics

More RAC Statistics

- ◆ [RAC Report Summary](#)
- ◆ [Global Messaging Statistics](#)
- ◆ [Global CR Served Stats](#)
- ◆ [Global CURRENT Served Stats](#)
- ◆ [Global Cache Transfer Stats](#)
- ◆ [Interconnect Stats](#)
- ◆ [Dynamic Remastering Statistics](#)

Segment Statistics

- ◆ [Segments by Logical Reads](#)
- ◆ [Segments by Physical Reads](#)
- ◆ [Segments by Physical Read Requests](#)
- ◆ [Segments by UnOptimized Reads](#)
- ◆ [Segments by Optimized Reads](#)
- ◆ [Segments by Direct Physical Reads](#)
- ◆ [Segments by Physical Writes](#)
- ◆ [Segments by Physical Write Requests](#)
- ◆ [Segments by Direct Physical Writes](#)
- ◆ [Segments by Table Scans](#)
- ◆ [Segments by DB Blocks Changes](#)
- ◆ [Segments by Row Lock Waits](#)
- ◆ [Segments by ITL Waits](#)
- ◆ [Segments by Buffer Busy Waits](#)
- ◆ [Segments by Global Cache Buffer Busy](#)
- ◆ [Segments by CR Blocks Received](#)
- ◆ [Segments by Current Blocks Received](#)

ORACLE

Copyright © 2010, Oracle. All rights reserved.

AWR Reports and RAC

The RAC-related statistics in an AWR report are organized in different sections. A RAC statistics section appears after the Top 5 Timed Events. This section contains:

- The number of instances open at the time of the begin snapshot and the end snapshot to indicate whether instances joined or left between the two snapshots
- The Global Cache Load Profile, which essentially lists the number of blocks and messages that are sent and received, as well as the number of fusion writes
- The Global Cache Efficiency Percentages, which indicate the percentage of buffer gets broken up into buffers received from the disk, local cache, and remote caches. Ideally, the percentage of disk buffer access should be close to zero.
- GCS and GES Workload Characteristics, which gives you an overview of the more important numbers first. Because the global enqueue convert statistics have been consolidated with the global enqueue get statistics, the report prints only the average global enqueue get time. The round-trip times for CR and current block transfers follow, as well as the individual sender-side statistics for CR and current blocks. The average log flush times are computed by dividing the total log flush time by the number of actual log flushes. Also, the report prints the percentage of blocks served that actually incurred a log flush.

AWR Reports and RAC (continued)

- GCS and GES Messaging Statistics. The most important statistic here is the *average message sent queue time on kxsp*, which indicates how well the IPC works. Average numbers should be less than 1 ms.

Additional RAC statistics are then organized in the following sections:

- The Global Enqueue Statistics section contains data extracted from V\$GES_STATISTICS.
- The Global CR Served Stats section contains data from V\$CR_BLOCK_SERVER.
- The Global CURRENT Served Stats section contains data from V\$CURRENT_BLOCK_SERVER.
- The Global Cache Transfer Stats section contains data from V\$INSTANCE_CACHE_TRANSFER.

The Segment Statistics section also includes the GC Buffer Busy Waits, CR Blocks Received, and CUR Blocks Received information for relevant segments.

Note: For more information about wait events and statistics, refer to *Oracle Database Reference*.

Active Session History Reports for RAC

- Active Session History (ASH) report statistics provide details about the RAC Database session activity.
- The database records information about active sessions for all active RAC instances.

Report Results

ASH Report For RDBA/RDBA1

DB Name	DB Id	Instance	Inst num	Release	RAC	Host
RDBA	901665511	RDBA1	1	11.2.0.1.0	YES	racnode01.example.com

CPU#	SGA Size	Buffer Cache	Shared Pool	ASH Buffer Size
1	818M (100%)	184M (22.5%)	292M (35.7%)	2.0M (0.2%)

	Sample Time	Data Source
Analysis Begin Time:	07-Aug-09 08:45:49	V\$ACTIVE_SESSION_HISTORY
Analysis End Time:	07-Aug-09 09:50:49	V\$ACTIVE_SESSION_HISTORY
Elapsed Time:	65.0 (mins)	
Sample Count:	261	
Average Active Sessions:	0.07	
Avg. Active Session per CPU:	0.07	
Report Target:	None specified	

ASH Report

- [Top Events](#)
- [Load Profile](#)
- [Top SQL](#)
- [Top PL/SQL](#)
- [Top Java](#)
- [Top Call Types](#)
- [Top Sessions](#)
- [Top Objects/Files/Latches](#)
- [Activity Over Time](#)

- Two ASH report sections specific to Oracle RAC are **Top Cluster Events** and **Top Remote Instance**.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Active Session History Reports for RAC

Active Session History (ASH) is an integral part of the Oracle Database self-management framework and is useful for diagnosing performance problems in Oracle RAC environments. ASH report statistics provide details about Oracle Database session activity. Oracle Database records information about active sessions for all active Oracle RAC instances and stores this data in the System Global Area (SGA). Any session that is connected to the database and using CPU is considered an active session. The exception to this is sessions that are waiting for an event that belongs to the idle wait class.

ASH reports present a manageable set of data by capturing only information about active sessions. The amount of the data is directly related to the work being performed, rather than the number of sessions allowed on the system. ASH statistics that are gathered over a specified duration can be put into ASH reports.

Each ASH report is divided into multiple sections to help you identify short-lived performance problems that do not appear in the ADDM analysis. Two ASH report sections that are specific to Oracle RAC are Top Cluster Events and Top Remote Instance.

Active Session History Reports for RAC (continued)

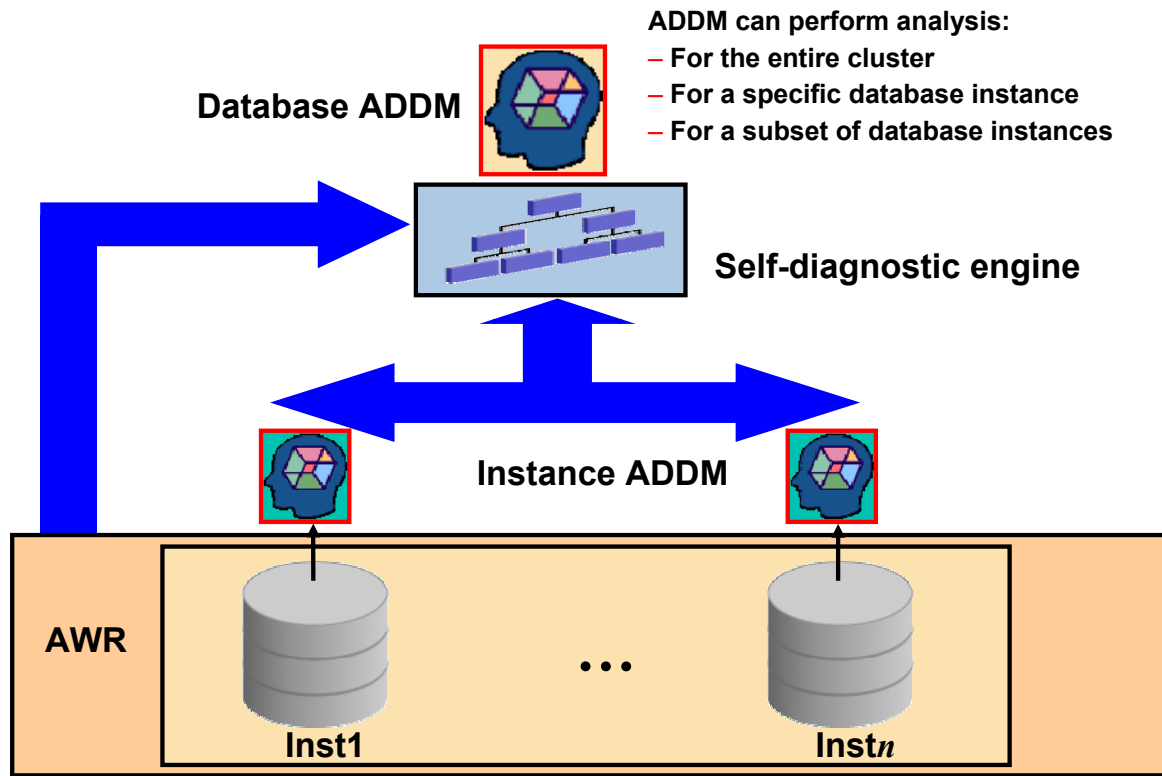
Top Cluster Events

The ASH report Top Cluster Events section is part of the Top Events report that is specific to Oracle RAC. The Top Cluster Events report lists events that account for the highest percentage of session activity in the cluster wait class event along with the instance number of the affected instances. You can use this information to identify which events and instances caused a high percentage of cluster wait events.

Top Remote Instance

The ASH report Top Remote Instance section is part of the Top Load Profile report that is specific to Oracle RAC. The Top Remote Instance report shows cluster wait events along with the instance numbers of the instances that accounted for the highest percentages of session activity. You can use this information to identify the instance that caused the extended cluster wait period.

Automatic Database Diagnostic Monitor for RAC



Copyright © 2010, Oracle. All rights reserved.

Automatic Database Diagnostic Monitor for RAC

Using the Automatic Database Diagnostic Monitor (ADDM), you can analyze the information collected by AWR for possible performance problems with your Oracle database. ADDM presents performance data from a clusterwide perspective, thus enabling you to analyze performance on a global basis. In an Oracle RAC environment, ADDM can analyze performance using data collected from all instances and present it at different levels of granularity, including:

- Analysis for the entire cluster
- Analysis for a specific database instance
- Analysis for a subset of database instances

To perform these analyses, you can run the ADDM Advisor in Database ADDM for RAC mode to perform an analysis of the entire cluster, in Local ADDM mode to analyze the performance of an individual instance, or in Partial ADDM mode to analyze a subset of instances. Database ADDM for RAC is not just a report of reports but has independent analysis that is appropriate for RAC. You activate ADDM analysis using the advisor framework through Advisor Central in Oracle Enterprise Manager, or through the `DBMS_ADVISOR` and `DBMS_ADDM` PL/SQL packages.

Note: Database ADDM report is generated on AWR snapshot coordinator.

Automatic Database Diagnostic Monitor for RAC

- Identifies the most critical performance problems for the entire RAC cluster database
- Runs automatically when taking AWR snapshots
- Performs database-wide analysis of:
 - Global resources (for example I/O and global locks)
 - High-load SQL and hot blocks
 - Global cache interconnect traffic
 - Network latency issues
 - Skew in instance response times
- Is used by DBAs to analyze cluster performance
- No need to investigate *n* reports to spot common problems

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Automatic Database Diagnostic Monitor for RAC (continued)

In Oracle Database 11g, you can create a period analysis mode for ADDM that analyzes the throughput performance for an entire cluster. When the advisor runs in this mode, it is called *database ADDM*. You can run the advisor for a single instance, which is called *instance ADDM*.

Database ADDM has access to AWR data generated by all instances, thereby making the analysis of global resources more accurate. Both database and instance ADDM run on continuous time periods that can contain instance startup and shutdown. In the case of database ADDM, there may be several instances that are shut down or started during the analysis period. However, you must maintain the same database version throughout the entire time period.

Database ADDM runs automatically after each snapshot is taken. You can also perform analysis on a subset of instances in the cluster. This is called *partial analysis ADDM*.

I/O capacity finding (the I/O system is overused) is a global finding because it concerns a global resource affecting multiple instances. A local finding concerns a local resource or issue that affects a single instance. For example, a CPU-bound instance results in a local finding about the CPU.

Although ADDM can be used during application development to test changes to either the application, the database system, or the hosting machines, database ADDM is targeted at DBAs.

What Does ADDM Diagnose for RAC?

- Latency problems in interconnect
- Congestion (identifying top instances affecting the entire cluster)
- Contention (buffer busy, top objects, etc.)
- Top consumers of multiblock requests
- Lost blocks
- Reports information about interconnect devices; warns about using PUBLIC interfaces
- Reports throughput of devices, and how much of it is used by Oracle and for what purpose (GC, locks, PQ)

ORACLE

Copyright © 2010, Oracle. All rights reserved.

What Does ADDM Diagnose for RAC?

Data sources are:

- Wait events (especially Cluster class and buffer busy)
- Active Session History (ASH) reports
- Instance cache transfer data
- Interconnect statistics (throughput, usage by component, pings)

ADDM analyzes the effects of RAC for both the entire database (DATABASE analysis mode) and for each instance (INSTANCE analysis mode).

EM Support for ADDM for RAC



ORACLE

Copyright © 2010, Oracle. All rights reserved.

EM Support for ADDM for RAC

Oracle Database 11g Enterprise Manager displays the ADDM analysis on the Cluster Database Home page.

On the Automatic Database Diagnostic Monitor (ADDM) page, the Database Activity chart (not shown here) plots the database activity during the ADDM analysis period. Database activity types are defined in the legend based on its corresponding color in the chart. Each icon below the chart represents a different ADDM task, which in turn corresponds to a pair of individual Oracle Database snapshots saved in the Workload Repository.

In the ADDM Performance Analysis section, the ADDM findings are listed in descending order, from highest impact to least impact. For each finding, the Affected Instances column displays the number (*m* of *n*) of instances affected. Drilling down further on the findings takes you to the Performance Findings Detail page. The Informational Findings section lists the areas that do not have a performance impact and are for informational purpose only.

The Affected Instances chart shows how much each instance is impacted by these findings. The display indicates the percentage impact for each instance.

Quiz

Although there are specific tuning areas for RAC, such as instance recovery and interconnect traffic, you get most benefits by tuning your system like a single-instance system.

1. True
2. False

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Answer: 1

Quiz

Which of the following RAC tuning tips are correct?

1. Application tuning is often the most beneficial!
2. Reduce long full-table scans in OLTP systems
3. Eliminate sequence caches
4. Use partitioning to reduce inter-instance traffic
5. Configure the interconnects properly

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Answer: 1, 2 , 4, 5

Statement 3 is incorrect.

Summary

In this lesson, you should have learned how to:

- Determine RAC-specific tuning components
- Tune instance recovery in RAC
- Determine RAC-specific wait events, global enqueues, and system statistics
- Implement the most common RAC tuning tips
- Use the Cluster Database Performance pages
- Use the Automatic Workload Repository (AWR) in RAC
- Use Automatic Database Diagnostic Monitor (ADDM) in RAC

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Practice 14 Overview

This practice covers discovering performance issues manually using the EM performance pages as well as ADDM.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

15

Services

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Configure and manage services in a RAC environment
- Use services with client applications
- Use services with the Database Resource Manager
- Use services with the Scheduler
- Configure services aggregation and tracing

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Services

- To manage workloads or a group of applications, you can define services for a particular application or a subset of an application's operations.
- You can also group work by type under services.
- For example OLTP users can use one service while batch processing can use another to connect to the database.
- Users who share a service should have the same service-level requirements.
- Use `srvctl` or Enterprise Manager to manage services, **not** `DBMS_SERVICE`.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle Services

To manage workloads or a group of applications, you can define services that you assign to a particular application or to a subset of an application's operations. You can also group work by type under services. For example, online users can use one service, while batch processing can use another, and reporting can use yet another service to connect to the database.

It is recommended that all users who share a service have the same service-level requirements. You can define specific characteristics for services and each service can be a separate unit of work. There are many options that you can take advantage of when using services. Although you do not have to implement these options, using them helps optimize application performance. You can define services for both policy-managed and administrator-managed databases.

Do not use `DBMS_SERVICE` with cluster-managed services. When Oracle Clusterware starts a service, it updates the database with the attributes stored in the CRS resource. If you use `DBMS_SERVICE` to modify the service and do not update the CRS resource, the next time CRS resource is started, it will override the database attributes set by `DBMS_SERVICE`.

Services for Policy- and Administrator-Managed Databases

- You can define services for both policy-managed and administrator-managed databases.
- Services for a policy-managed database are defined to a server pool where the database is running.
- Services for policy-managed databases can be defined as:
 - **UNIFORM** (running on all instances in the server pool)
 - **SINGLETON** (running on only one instance in the pool)
 - For singleton services, RAC chooses on which instance in the server pool the service is active.
- Services for an administrator-managed database define which instances normally support that service.
 - These are known as the **PREFERRED** instances.
 - Instances defined to support a service if the preferred instance fails are known as **AVAILABLE** instances.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Services for Policy- and Administrator-Managed Databases

It is recommended that all users who share a service have the same service-level requirements. You can define specific characteristics for services and each service can be a separate unit of work. There are many options that you can take advantage of when using services. Although you do not have to implement these options, they help optimize application performance. You can define services for both policy-managed and administrator-managed databases.

- **Policy-managed database:** When you define services for a policy-managed database, you define the service to a server pool where the database is running. You can define the service as either uniform (running on all instances in the server pool) or singleton (running on only one instance in the server pool). For singleton services, RAC chooses on which instance in the server pool the service is active. If that instance fails, then the service fails over to another instance in the pool. A service can only run in one server pool.
- **Administrator-managed database:** When you define a service for an administrator-managed database, you define which instances support that service. These are known as the **PREFERRED** instances. You can also define other instances to support a service if the service's preferred instance fails. These are known as **AVAILABLE** instances.

Default Service Connections

- Application services:
 - Limit of 115 services per database
- Internal services:
 - SYS\$BACKGROUND
 - SYS\$USERS
 - Cannot be deleted or changed
- A special Oracle database service is created by default for the Oracle RAC database.
- This default service is always available on all instances in an Oracle RAC environment.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Default Service Connections

There are two broad types of services: application services and internal services. Application services are mainly functional maps to workloads. Sessions doing work for a common business function are grouped together. For Oracle E-Business Suite, AP, AR, GL, MFG, WIP, and so on create a functional division of work within the database and can be categorized as services.

The RDBMS also supports two internal services. SYS\$BACKGROUND is used by the background processes only. SYS\$USERS is the default service for user sessions that are not associated with any application service. Both internal services support all the workload management features and neither one can be stopped or disabled. A special Oracle database service is created by default for your Oracle RAC database. This default service is always available on all instances in an Oracle RAC environment, unless an instance is in restricted mode. You cannot alter this service or its properties. There is a limitation of 115 application services per database that you can create. Also, a service name is restricted to 64 characters.

Note: Shadow services are also included in the application service category. For more information about shadow services, see the lesson titled “High Availability of Connections.” In addition, a service is also created for each Advanced Queue created. However, these types of services are not managed by Oracle Clusterware. Using service names to access a queue provides location transparency for the queue within a RAC database.

Create Service with Enterprise Manager

Administration-Managed

Policy-Managed

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Create Services with Enterprise Manager

From your Cluster Database home page, click the Availability tab, and then click Cluster Managed Database Services. On the Cluster Managed Database Services page, click Create Service.

Use the Create Service page to configure a new service in which you do the following:

- Select the desired service policy for each instance configured for the cluster database.
- Select the desired service properties. Refer to the section “Service Attributes” in this lesson for more information about the properties you can specify on this page.

If your database is administration managed, the High Availability Configuration section allows you to configure preferred and available servers. If your database employs policy-managed administration, you can configure the service cardinality to be UNIFORM or SINGLETON and assign the service to a server pool.

You can also define the management policy for a service. You can choose either an automatic or a manual management policy.

- **Automatic:** The service always starts when the database starts.
- **Manual:** Requires that the service be started manually. Prior to Oracle RAC 11g Release 2, all services worked as though they were defined with a manual management policy.

Note: Enterprise Manager now generates the corresponding entries in your `tnsnames.ora` files for your services. Just click the “Update local naming parameter (tnsnames.ora) file” check box when creating the service.

Create Services with SRVCTL

- To create a service called GL with preferred instance RAC02 and an available instance RAC01:

```
$ srvctl add service -d PROD1 -s GL -r RAC02 -a RAC01
```

- To create a service called AP with preferred instance RAC01 and an available instance RAC02:

```
$ srvctl add service -d PROD1 -s AP -r RAC01 -a RAC02
```

- To create a SINGLETON service called BATCH using server pool SP1 and a UNIFORM service called ERP using server pool SP2 :

```
$ srvctl add service -d PROD2 -s BATCH -g SP1 \  
-c singleton -y manual
```

```
$ srvctl add service -d PROD2 -s ERP -g SP2 \  
-c UNIFORM -y manual
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Create Services with SRVCTL

For the example in the slide, assume a two-node, administration-managed database called PROD1 with an instance named RAC01 on one node and an instance called RAC02 on the other. Two services are created, AP and GL, to be managed by Oracle Clusterware. The AP service is defined with a preferred instance of RAC01 and an available instance of RAC02.

If RAC01 dies, the AP service member on RAC01 is restored automatically on RAC02. A similar scenario holds true for the GL service.

Note that it is possible to assign more than one instance with both the `-r` and `-a` options. However, `-r` is mandatory but `-a` is optional.

Next, assume a policy-managed cluster database called PROD2. Two services are created, a SINGLETON service called BATCH and a UNIFORM service called ERP. SINGLETON services run on one of the active servers and UNIFORM services run on all active servers of the server pool. The characteristics of the server pool determines how resources are allocated to the service.

Note: When services are created with `srvctl`, `tnsnames.ora` is not updated and the service is not started.

Manage Services with Enterprise Manager

ORACLE Enterprise Manager 11g Database Control

Cluster Database: RDBA >

Cluster Managed Database Services

The following shows the status of all cluster managed services defined for the current database. Select a service to manage the states of its instances. Page Refreshed 8/11/09 8:18 AM Refresh

Create Service

Start Stop Test Connection Show All TNS Strings Actions Manage Go

Select	Service Name	Status	Running Instances	Response Time (m)	Service related alerts among all Instances	Status Details
<input checked="" type="radio"/>	ERP	↑	RDBA1, RDBA2	0.00 0.04 3.5	0 0	✓ Service is running on all preferred instances.
<input type="radio"/>	GL	↑	RDBA2	0.00 0.00 0.00 0.00	0 0	✓ Service is running on all preferred instances.

TIP Response Time and % CPU Load data is average over the last 5 minutes

Return

Manage Services with Enterprise Manager

You can use Enterprise Manager to manage services within a GUI framework. The screenshot in the slide shows the main page for administering services within RAC. It shows you some basic status information about a defined service.

To access this page, click the Cluster Managed Database Services link on the Cluster Database Availability page.

You can perform simple service management such as enabling, disabling, starting, stopping, and relocating services. All possible operations are shown in the slide.

If you choose to start a service on the Cluster Managed Database Services page, then EM attempts to start the service on every preferred instance. Stopping the service stops it on all instances that it is currently running.

To relocate a service, select the service that you want to administer, select the Manage option from the Actions drop-down list, and then click Go.

Note: On the Cluster Managed Database Services page, you can test the connection for a service.

Managing Services with EM

Cluster Managed Database Service: ERP

The service has been configured to run on the following instances. A service may have been stopped on an instance if the instance was down or the service was disabled. Starting a service on a down instance will first bring up the down instance.

Page Refreshed 8/11/09 9:05 AM [Refresh](#)

Service Status **Service is running on all preferred instances.**

% CPU Load [0.00](#)

Transparent Application Failover (TAF) Policy **BASIC**

Top Consumers [Details](#)

Service Properties [Edit](#)

Instances

[Enable](#) [Disable](#) [Start](#) [Stop](#) [Relocate](#)

Select	Node Name	Instance Name	Service Status for Node	Instance Status	Service Policy	Response Time (per user call) (milliseconds)	CPU Time (per user call) (milliseconds)	Status Details
	racnode01	RDBA1	Running		Preferred	0.00	0.00	
	racnode02	RDBA2	Running		Preferred	0.00	0.00	

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Manage Services with Enterprise Manager (continued)

To access the Cluster Managed Database Service page for an individual service, you must select a service from the Cluster Managed Database Services page, select the Manage option from the Actions drop-down list, and then click Go.

This is the Cluster Managed Database Service page for an individual service. It offers you the same functionality as the previous page, except that actions performed here apply to specific instances of a service.

This page also offers you the added functionality of relocating a service to an available instance. Relocating a service from one instance to another stops the service on the first instance and then starts it on the second.

Manage Services with `srvctl`

- Start a named service on all configured instances:

```
$ srvctl start service -d orcl -s AP
```

- Stop a service:

```
$ srvctl stop service -d orcl -s AP
```

- Disable a service at a named instance:

```
$ srvctl disable service -d orcl -s AP -i orcl4
```

- Set an available instance as a preferred instance:

```
$ srvctl modify service -d orcl -s AP -i orcl5 -r
```

- Relocate a service from one instance to another:

```
$ srvctl relocate service -d orcl -s AP -i orcl5 -t orcl4
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Manage Services with `srvctl`

The slide demonstrates some management tasks with services by using `SRVCTL`.

Assume that an AP service has been created with four preferred instances: `orcl1`, `orcl2`, `orcl3`, and `orcl4`. An available instance, `orcl5`, has also been defined for AP.

In the first example, the AP service is started on all instances. If any of the preferred or available instances that support AP are not running but are enabled, then they are started.

The `stop` command stops the AP service on instances `orcl3` and `orcl4`. The instances themselves are not shut down, but remain running possibly supporting other services. The AP service continues to run on `orcl1` and `orcl2`. The intention might have been to perform maintenance on `orcl4`, and so the AP service was disabled on that instance to prevent automatic restart of the service on that instance. The OCR records the fact that AP is disabled for `orcl4`. Thus, Oracle Clusterware will not run AP on `orcl4` until the service is enabled.

The next command in the slide changes `orcl5` from being an available instance to a preferred one. This is beneficial if the intent is to always have four instances run the service because `orcl4` was previously disabled. The last example relocates the AP service from instance `orcl5` to `orcl4`. Do not perform other service operations while the online service modification is in progress.

Note: For more information, refer to the *Oracle Real Application Clusters Administrator's Guide*.

Use Services with Client Applications

```
ERP= (DESCRIPTION=                ## Using the SCAN ##
      (LOAD_BALANCE=on)
        (ADDRESS= (PROTOCOL=TCP) (HOST=cluster01-scan) (PORT=1521))
        (CONNECT_DATA= (SERVICE_NAME=ERP)) )
```

```
ERP= (DESCRIPTION=                ## Using VIPs ##
      (LOAD_BALANCE=on)
        (ADDRESS= (PROTOCOL=TCP) (HOST=node-1vip) (PORT=1521))
        (ADDRESS= (PROTOCOL=TCP) (HOST=node-2vip) (PORT=1521))
        (ADDRESS= (PROTOCOL=TCP) (HOST=node-3vip) (PORT=1521))
        (CONNECT_DATA= (SERVICE_NAME=ERP)) )
```

```
url="jdbc:oracle:oci:@ERP"        ## Thick JDBC ##
```

```
url="jdbc:oracle:thin:@(DESCRIPTION=    ## Thin JDBC ##
      (LOAD_BALANCE=on)
        (ADDRESS= (PROTOCOL=TCP) (HOST=cluster01-scan) (PORT=1521))
        (CONNECT_DATA= (SERVICE_NAME=ERP)) ) "
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Use Services with Client Applications

The first example in the slide shows the TNS connect descriptor that can be used to access the ERP service. It uses the cluster's Single Client Access Name (SCAN). The SCAN provides a single name to the clients connecting to Oracle RAC that does not change throughout the life of the cluster, even if you add or remove nodes from the cluster. Clients connecting with SCAN can use a simple connection string, such as a thin JDBC URL or EZConnect, and still achieve the load balancing and client connection failover. The second example uses virtual IP addresses as in previous versions of the Oracle Database.

The third example shows the thick JDBC connection description using the previously defined TNS connect descriptor.

The third example shows the thin JDBC connection description using the same TNS connect descriptor as the first example.

Note: The `LOAD_BALANCE=ON` clause is used by Oracle Net to randomize its progress through the protocol addresses of the connect descriptor. This feature is called client connection load balancing.

Services and Connection Load Balancing

- The two load balancing methods that you can implement are:
 - **Client-side load balancing:** Balances the connection requests across the listeners
 - **Server-side load balancing:** The listener directs a connection request to the best instance currently providing the service by using the load balancing advisory (LBA).
- FAN, Fast Connection Failover, and LBA depend on a connection load balancing configuration that includes setting the connection load balancing goal for the service.
- The load balancing goal for the service can be either:
 - **LONG:** For applications having long-lived connections. This is typical for connection pools and SQL*Forms sessions.
 - **SHORT:** For applications that have short-lived connections

```
srvctl modify service -s service_name -j LONG|SHORT
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Services and Connection Load Balancing

Oracle Net Services provides the ability to balance client connections across the instances in an Oracle RAC configuration. You can implement two types of load balancing: client-side and server-side. Client-side load balancing balances the connection requests across the listeners. With server-side load balancing, the listener directs a connection request to the best instance currently providing the service by using the load balancing advisory. In a RAC database, client connections should use both types of connection load balancing.

FAN, Fast Connection Failover, and the load balancing advisory depend on an accurate connection load balancing configuration that includes setting the connection load balancing goal for the service. You can use a goal of either LONG or SHORT for connection load balancing. These goals have the following characteristics:

- **LONG:** Use the LONG load balancing method for applications that have long-lived connections. This is typical for connection pools and SQL*Forms sessions. LONG is the default connection load balancing goal. The following is an example of modifying a service, POSTMAN, with the `srvctl` utility to define the connection load balancing goal for long-lived sessions:
- **SHORT:** Use the SHORT connection load balancing method for applications that have short-lived connections. The following example modifies the ORDER service, using `srvctl` to set the goal to SHORT:

```
srvctl modify service -s POSTMAN -j LONG
```

```
srvctl modify service -s ORDER -j SHORT
```


Services and Transparent Application Failover

- Services simplify the deployment of Transparent Application Failover (TAF).
- You can define a TAF policy for a service and all connections using this service will automatically have TAF enabled.
- The TAF setting on a service can be **NONE**, **BASIC**, or **PRECONNECT** and overrides any TAF setting in the client connection definition.
- To define a TAF policy for a service, the `srvctl` utility can be used as shown below:

```
srvctl modify service -s gl.example.com -q TRUE -P
BASIC -e SELECT -z 180 -w 5 -j LONG
```

Where *-z* is the number of retries, *-w* is the delay between retry attempts and *-j* is the connection load balancing goal.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Services and Transparent Application Failover

When Oracle Net Services establishes a connection to an instance, the connection remains open until the client closes the connection, the instance is shut down, or a failure occurs. If you configure TAF for the connection, then Oracle Database moves the session to a surviving instance when an outage occurs.

TAF can restart a query after failover has completed but for other types of transactions, such as INSERT, UPDATE, or DELETE, the application must roll back the failed transaction and resubmit the transaction. You must re-execute any session customizations, in other words, ALTER SESSION statements, after failover has occurred. However, with TAF, a connection is not moved during normal processing, even if the workload changes over time.

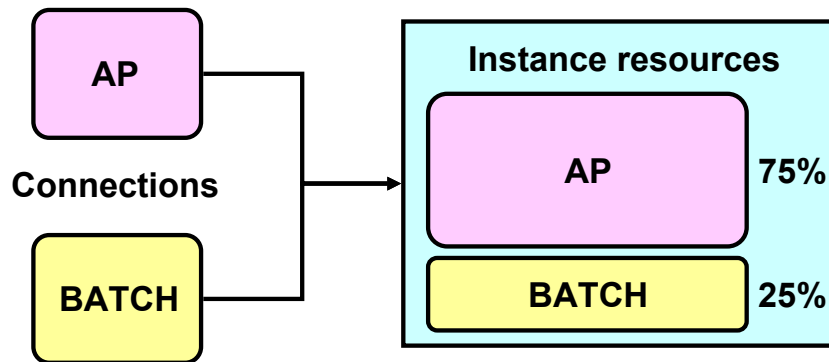
Services simplify the deployment of TAF. You can define a TAF policy for a service, and all connections using this service will automatically have TAF enabled. This does not require any client-side changes. The TAF setting on a service overrides any TAF setting in the client connection definition. To define a TAF policy for a service, use the `srvctl` utility as in the following example:

```
srvctl modify service -s gl.example.com -q TRUE -P BASIC -e
SELECT -z 180 -w 5 -j LONG
```

Note: TAF applies only to an admin-managed database and not to policy-managed databases.

Use Services with the Resource Manager

- Consumer groups are automatically assigned to sessions based on session services.
- Work is prioritized by service inside one instance.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

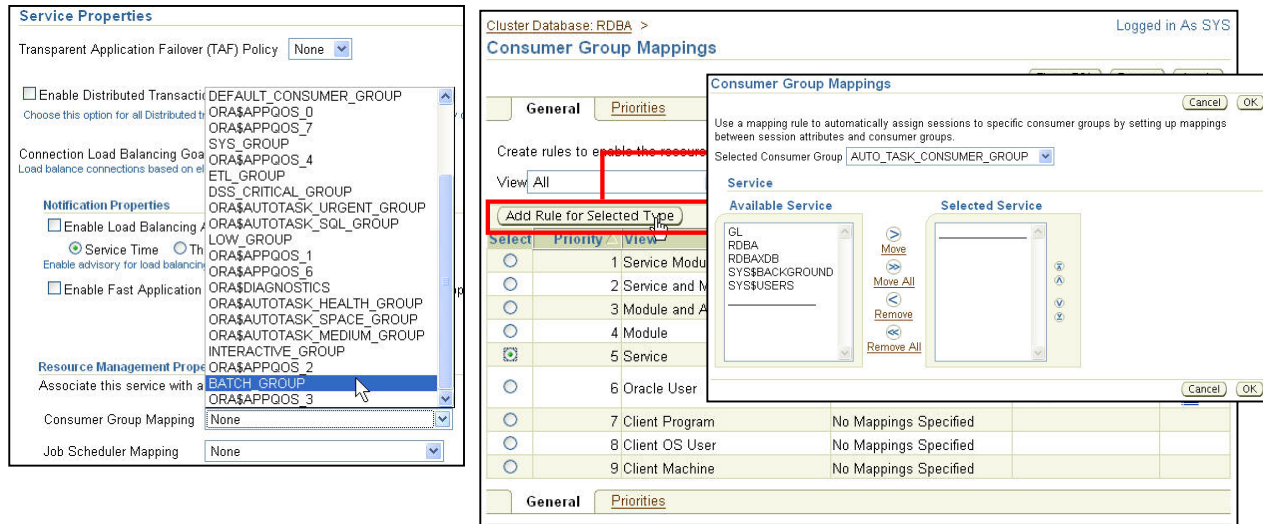
Use Services with the Resource Manager

The Database Resource Manager (also called Resource Manager) enables you to identify work by using services. It manages the relative priority of services within an instance by binding services directly to consumer groups. When a client connects by using a service, the consumer group is assigned transparently at connect time. This enables the Resource Manager to manage the work requests by service in the order of their importance.

For example, you define the AP and BATCH services to run on the same instance, and assign AP to a high-priority consumer group and BATCH to a low-priority consumer group. Sessions that connect to the database with the AP service specified in their TNS connect descriptor get priority over those that connect to the BATCH service.

This offers benefits in managing workloads because priority is given to business functions rather than the sessions that support those business functions.

Services and Resource Manager with EM



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Services and Resource Manager with EM

Enterprise Manager (EM) presents a GUI through the Consumer Group Mapping page to automatically map sessions to consumer groups. You can access this page by clicking the Consumer Group Mappings link on the Server page.

Using the General tabbed page of the Consumer Group Mapping page, you can set up a mapping of sessions connecting with a service name to consumer groups as illustrated on the right part of the slide.

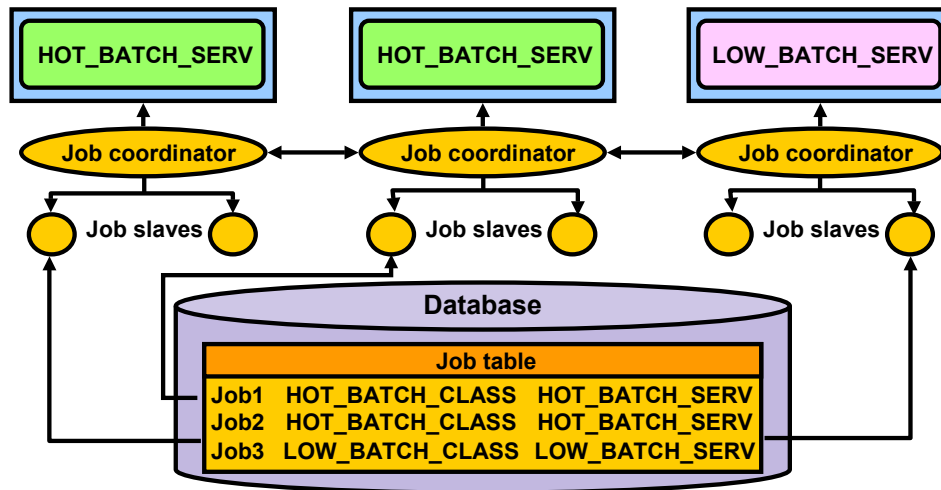
With the ability to map sessions to consumer groups by service, module, and action, you have greater flexibility when it comes to managing the performance of different application workloads.

Using the Priorities tabbed page of the Consumer Group Mapping page, you can change priorities for the mappings that you set up on the General tabbed page. The mapping options correspond to columns in `V$SESSION`. When multiple mapping columns have values, the priorities you set determine the precedence for assigning sessions to consumer groups.

Note: You can also map a service to a consumer group directly from the Create Service page as shown on the left part of the slide.

Use Services with the Scheduler

- Services are associated with Scheduler classes.
- Scheduler jobs have service affinity:
 - High Availability
 - Load balancing



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Use Services with the Scheduler

Just as in other environments, the Scheduler in a RAC environment uses one job table for each database and one job coordinator (CJQ0 process) for each instance. The job coordinators communicate with each other to keep information current.

The Scheduler can use the services and the benefits they offer in a RAC environment. The service that a specific job class uses is defined when the job class is created. During execution, jobs are assigned to job classes and job classes run within services. Using services with job classes ensures that the work of the Scheduler is identified for workload management and performance tuning. For example, jobs inherit server-generated alerts and performance thresholds for the service they run under.

For High Availability, the Scheduler offers service affinity instead of instance affinity. Jobs are not scheduled to run on any specific instance. They are scheduled to run under a service. So, if an instance dies, the job can still run on any other instance in the cluster that offers the service.

Note: By specifying the service where you want the jobs to run, the job coordinators balance the load on your system for better performance.

Services and the Scheduler with EM

Cluster Database: RDBA > Scheduler Job Classes > Logged in As SYS

Create Job Class Show SQL Cancel OK

* Name

Description

Logging Level Log job runs only (RUNS)

Log Retention Period (Days)

Resource Consumer Group

Service Name

Enable advisory for load balancing: None

☐ Enable Fast Application

ODP.NET Applications

Resource Management Properties

Associate this service with a

Consumer Group Mapping

Job Scheduler Mapping None

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Services and the Scheduler with EM

To configure a job to run under a specific service, click the Job Classes link in the Database Scheduler section of the Server page. This opens the Scheduler Job Classes page. On the Scheduler Job Classes page, you can see services assigned to job classes.

When you click the Create button on the Scheduler Job Classes page, the Create Job Class page is displayed. On this page, you can enter details of a new job class, including which service it must run under.

Note: Similarly, you can map a service to a job class on the Create Service page as shown at the bottom of the slide.

Services and the Scheduler with EM

The screenshot shows the 'Create Job' page in Oracle Enterprise Manager 11g, specifically the 'Options' tab. The page is titled 'ORACLE Enterprise Manager 11g Database Control'. The breadcrumb navigation shows 'Cluster Database: RDBA > Scheduler Jobs > Create Job'. The user is logged in as 'SYS'. The 'Options' tab is selected, and the 'Instance Stickiness' dropdown is highlighted with a red box, showing 'TRUE'. Other options include 'Raise Events' (checkboxes for Job Started, Job Succeeded, Job Failed, Job Stopped, Job Broken, Job Disabled, Job Completed, Job Chain Stalled, Job Schedule Limit Reached), 'Maximum Run Duration (minutes)', 'Priority' (Medium), 'Schedule Limit (minutes)', 'Maximum Runs', 'Maximum Failures', 'Job Weight', and 'Job which includes parallel queries'. The 'Instance Stickiness' dropdown is set to 'TRUE', and a note below it states: 'For use in RAC, if instance_stickiness is set to TRUE, the Oracle Scheduler will attempt to execute the job on the same instance as the previous run'.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Services and the Scheduler with EM (continued)

After your job class is set up with the service that you want it to run under, you can create the job. To create the job, click the Jobs link on the Server page. The Scheduler Jobs page appears, on which you can click the Create button to create a new job. When you click the Create button, the Create Job page is displayed. This page has different tabs: General, Schedule, and Options. Use the General tabbed page to assign your job to a job class.

Use the Options page (displayed in the slide) to set the Instance Stickiness attribute for your job. Basically, this attribute causes the job to be load balanced across the instances for which the service of the job is running. The job can run only on one instance. If the Instance Stickiness value is set to TRUE, which is the default value, the Scheduler runs the job on the instance where the service is offered with the lightest load. If Instance Stickiness is set to FALSE, then the job is run on the first available instance where the service is offered.

Note: It is possible to set job attributes, such as `INSTANCE_STICKINESS`, by using the `SET_ATTRIBUTE` procedure of the `DBMS_SCHEDULER` PL/SQL package.

Using Distributed Transactions with RAC

- An XA transaction can span RAC instances, allowing any application that uses XA to take full advantage of the Oracle RAC environment.
- Tightly coupled XA transactions no longer require the special type of singleton services (DTP).
- XA transactions are transparently supported on Oracle RAC databases with any type of services configuration.
- However, DTP services will improve performance for many distributed transaction scenarios.
- DTP services allow you to direct all branches of a distributed transaction to a single instance in the cluster.
- To load balance, it is better to have several groups of smaller application servers with each group directing its transactions to a single service.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Using Distributed Transactions with RAC

An XA transaction can span Oracle RAC instances by default, allowing any application that uses XA to take full advantage of the Oracle RAC environment to enhance the availability and scalability of the application. This is controlled through the `GLOBAL_TXN_PROCESSES` initialization parameter, which is set to 1 by default. This parameter specifies the initial number of `GTxn` background processes for each Oracle RAC instance. Keep this parameter at its default value clusterwide to allow distributed transactions to span multiple Oracle RAC instances. This allows the units of work performed across these Oracle RAC instances to share resources and act as a single transaction (that is, the units of work are tightly coupled). It also allows 2PC requests to be sent to any node in the cluster. Tightly coupled XA transactions no longer require the special type of singleton services (that is, Oracle Distributed Transaction Processing [DTP] services) to be deployed on Oracle RAC database. XA transactions are transparently supported on Oracle RAC databases with any type of services configuration.

To provide improved application performance with distributed transaction processing in Oracle RAC, you may want to take advantage of the specialized service referred to as a DTP Service. Using DTP services, you can direct all branches of a distributed transaction to a single instance in the cluster. To load balance across the cluster, it is better to have several groups of smaller application servers with each group directing its transactions to a single service, or set of services, than to have one or two larger application servers.

Distributed Transactions and Services

To leverage all instances, create one or more DTP services for each instance that hosts distributed transactions.

- Use EM or `srvctl` to create singleton services XA1, XA2 and XA3 for database CRM, enabling DTP for the service:

Create Service

Define a highly available service by specifying preferred and available instances. You can also specify service properties to customize failover mechanisms, monitoring thresholds and resource management.

* Service Name

☒ Start service after creation

☒ Update local naming parameter (tnsnames.ora) file

High Availability Configuration

Instance Name	Service Policy
orcl1	Preferred
orcl2	Available
orcl3	Available

☒ **TIP** Must select at least one preferred instance.

Service Properties

Transparent Application Failover (TAF) Policy

☒ **Enable Distributed Transaction Processing**
Choose this option for all Distributed transactions including XA, JTA. Services with exactly one preferred instance can enable this.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Distributed Transactions and Services

To enhance the performance of distributed transactions, you can use services to manage DTP environments. By defining the DTP property of a service, the service is guaranteed to run on one instance at a time in an Oracle RAC database. All global distributed transactions performed through the DTP service are ensured to have their tightly coupled branches running on a single Oracle RAC instance. This has the following benefits:

- The changes are available locally within one Oracle RAC instance when tightly coupled branches need information about changes made by each other.
- Relocation and failover of services are fully supported for DTP.
- By using more DTP services than there are Oracle RAC instances, Oracle Database can balance the load by services across all the Oracle RAC database instances.

To leverage all the instances in a cluster, create one or more DTP services for each Oracle RAC instance that hosts distributed transactions. Choose one DTP service for one distributed transaction. Choose different DTP services for different distributed transactions to balance the workload among the Oracle RAC database instances.

Because all the branches of a distributed transaction are on one instance, you can leverage all the instances to balance the load of many DTP transactions through multiple singleton services, thereby maximizing application throughput.

Distributed Transactions and Services (continued)

An external transaction manager, such as OraMTS, coordinates DTP/XA transactions. However, an internal Oracle transaction manager coordinates distributed SQL transactions. Both DTP/XA and distributed SQL transactions must use the DTP service in Oracle RAC.

For services that you are going to use for distributed transaction processing, create the service using Oracle Enterprise Manager or `srvctl`, and define only one instance as the preferred instance. You can have as many AVAILABLE instances as you want. For example, the following `srvctl` command creates a singleton service for database CRM, `xa1.service.example.com`, whose preferred instance is `orcl1`:

```
srvctl add service -d CRM -s xa1.service.example.com -r orcl1 -a  
orcl2, orcl3
```

Then mark the service for distributed transaction processing by setting the DTP parameter to TRUE; the default is FALSE. Oracle Enterprise Manager enables you to set this parameter on the Cluster Managed Database Services: Create Service or Modify Service page.

If, for example, `orcl1` (that provides service XA1) fails, then the singleton service that it provided fails over to another instance, such as `orcl2` or `orcl3`. If services migrate to other instances after the cold-start of the Oracle RAC database, then you might have to force the relocation of the service to evenly re-balance the load on all the available hardware. Use data from the `GV$ACTIVE_SERVICES` view to determine whether to do this.

Service Thresholds and Alerts

- Service-level thresholds enable you to compare achieved service levels against accepted minimum required levels.
- You can explicitly specify two performance thresholds for each service:
 - **SERVICE_ELAPSED_TIME**: The response time for calls
 - **SERVICE_CPU_TIME**: The CPU time for calls

Service Properties

Transparent Application Failover (TAF) Policy: Basic

☐ Enable Distributed Transaction Processing
Choose this option for all Distributed transactions including XA, JTA. Services with exactly one preferred instance can enable this.

Connection Load Balancing Goal: ☒ Short ☐ Long
Load balance connections based on elapsed time (Short) or number of sessions (Long).

Notification Properties

☒ Enable Load Balancing Advisory
☒ Service Time ☐ Throughput
Enable advisory for load balancing based on service quality.

☒ Enable Fast Application Notification (FAN) for OCI and ODP.NET Applications

Service Threshold Levels
If thresholds are specified, alerts will be published when the service elapsed response time and/or CPU time exceed the threshold.

	Warning	Critical
Elapsed Time Threshold (milliseconds)	500000	750000
CPU Time Threshold (milliseconds)		

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Service Thresholds and Alerts

Service-level thresholds enable you to compare achieved service levels against accepted minimum required levels. This provides accountability for the delivery or the failure to deliver an agreed service level. The end goal is a predictable system that achieves service levels. There is no requirement to perform as fast as possible with minimum resource consumption; the requirement is to meet the quality of service.

You can explicitly specify two performance thresholds for each service: the response time for calls, or **SERVICE_ELAPSED_TIME**, and the CPU time for calls, or **SERVICE_CPU_TIME**. The response time goal indicates that the elapsed time should not exceed a certain value, and the response time represents wall clock time. Response time is a fundamental measure that reflects all delays and faults that might be blocking the call from running on behalf of the user. Response time can also indicate differences in node power across the nodes of an Oracle RAC database.

The service time and CPU time are calculated as the moving average of the elapsed, server-side call time. The AWR monitors the service time and CPU time and publishes AWR alerts when the performance exceeds the thresholds. You can then respond to these alerts by changing the priority of a job, stopping overloaded processes, or by relocating, expanding, shrinking, starting, or stopping a service. This permits you to maintain service availability despite changes in demand.

Services and Thresholds Alerts: Example

```
EXECUTE DBMS_SERVER_ALERT.SET_THRESHOLD(  
METRICS_ID => DBMS_SERVER_ALERT.ELAPSED_TIME_PER_CALL  
, warning_operator => DBMS_SERVER_ALERT.OPERATOR_GE  
, warning_value => '500000'  
, critical_operator => DBMS_SERVER_ALERT.OPERATOR_GE  
, critical_value => '750000'  
, observation_period => 30  
, consecutive_occurrences => 5  
, instance_name => NULL  
, object_type => DBMS_SERVER_ALERT.OBJECT_TYPE_SERVICE  
, object_name => 'servall');
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Services and Thresholds Alerts: Example

To check the thresholds for the `servall` service, use the AWR report. You should record output from the report over several successive intervals during which time the system is running optimally. For example, assume that for an email server, the AWR report runs each Monday during the peak usage times of 10:00 AM to 2:00 PM. The AWR report would contain the response time, or DB time, and the CPU consumption time, or CPU time, for calls for each service. The AWR report would also provide a breakdown of the work done and the wait times that are contributing to the response times. Using `DBMS_SERVER_ALERT`, set a warning threshold for the `servall` service at 0.5 seconds and a critical threshold for the payroll service at 0.75 seconds. You must set these thresholds at all instances within an Oracle RAC database. The parameter `instance_name` can be set to a `NULL` value to indicate database-wide alerts. You can schedule actions using Enterprise Manager jobs for alerts, or you can schedule actions to occur programmatically when the alert is received. In this example, thresholds are added for the `servall` service and set as shown above.

Verify the threshold configuration using the following `SELECT` statement:

```
SELECT metrics_name, instance_name, warning_value,  
critical_value, observation_period FROM dba_thresholds;
```

Service Aggregation and Tracing

- Statistics are always aggregated by service to measure workloads for performance tuning.
- Statistics can be aggregated at finer levels:
 - MODULE
 - ACTION
 - Combination of `SERVICE_NAME`, `MODULE`, `ACTION`
- Tracing can be done at various levels:
 - `SERVICE_NAMES`
 - MODULE
 - ACTION
 - Combination of `SERVICE_NAME`, `MODULE`, `ACTION`
- This is useful for tuning systems that use shared sessions.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

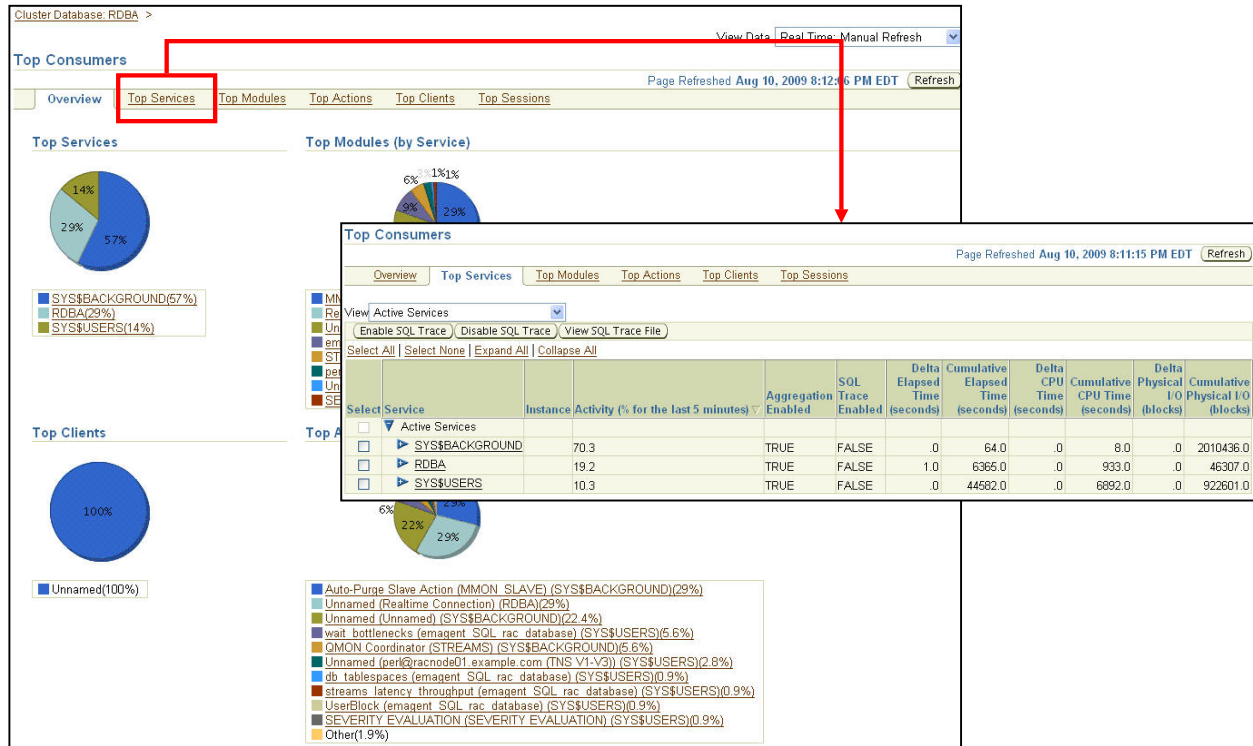
Service Aggregation and Tracing

By default, important statistics and wait events are collected for the work attributed to every service. An application can further qualify a service by `MODULE` and `ACTION` names to identify the important transactions within the service. This enables you to locate exactly the poorly performing transactions for categorized workloads. This is especially important when monitoring performance in systems by using connection pools or transaction processing monitors. For these systems, the sessions are shared, which makes accountability difficult.

`SERVICE_NAME`, `MODULE`, and `ACTION` are actual columns in `V$SESSION`. `SERVICE_NAME` is set automatically at login time for the user. `MODULE` and `ACTION` names are set by the application by using the `DBMS_APPLICATION_INFO` PL/SQL package or special OCI calls. `MODULE` should be set to a user-recognizable name for the program that is currently executing. Likewise, `ACTION` should be set to a specific action or task that a user is performing within a module (for example, entering a new customer).

Another aspect of this workload aggregation is tracing by service. The traditional method of tracing each session produces trace files with SQL commands that can span workloads. This results in a hit-or-miss approach to diagnose problematic SQL. With the criteria that you provide (`SERVICE_NAME`, `MODULE`, or `ACTION`), specific trace information is captured in a set of trace files and combined into a single output trace file. This enables you to produce trace files that contain SQL that is relevant to a specific workload being done.

Top Services Performance Page



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Top Services Performance Page

From the Performance page, you can access the Top Consumers page by clicking the Top Consumers link.

The Top Consumers page has several tabs for displaying your database as a single-system image. The Overview tabbed page contains four pie charts: Top Clients, Top Services, Top Modules, and Top Actions. Each chart provides a different perspective regarding the top resource consumers in your database.

The Top Services tabbed page displays performance-related information for the services that are defined in your database. Using this page, you can enable or disable tracing at the service level, as well as view the resulting SQL trace file.

Service Aggregation Configuration

- Automatic service aggregation level of statistics
- DBMS_MONITOR used for finer granularity of service aggregations:
 - SERV_MOD_ACT_STAT_ENABLE
 - SERV_MOD_ACT_STAT_DISABLE
- Possible additional aggregation levels:
 - SERVICE_NAME/MODULE
 - SERVICE_NAME/MODULE/ACTION
- Tracing services, modules, and actions:
 - SERV_MOD_ACT_TRACE_ENABLE
 - SERV_MOD_ACT_TRACE_DISABLE
- Database settings persist across instance restarts.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Service Aggregation Configuration

On each instance, important statistics and wait events are automatically aggregated and collected by service. You do not have to do anything to set this up, except connect with different connect strings using the services you want to connect to. However, to achieve a finer level of granularity of statistics collection for services, you must use the SERV_MOD_ACT_STAT_ENABLE procedure in the DBMS_MONITOR package. This procedure enables statistics gathering for additional hierarchical combinations of SERVICE_NAME/MODULE and SERVICE_NAME/MODULE/ACTION. The SERV_MOD_ACT_STAT_DISABLE procedure stops the statistics gathering that was turned on.

The enabling and disabling of statistics aggregation within the service applies to every instance accessing the database. These settings are persistent across instance restarts.

The SERV_MOD_ACT_TRACE_ENABLE procedure enables tracing for services with three hierarchical possibilities: SERVICE_NAME, SERVICE_NAME/MODULE, and SERVICE_NAME/MODULE/ACTION. The default is to trace for all instances that access the database. A parameter is provided that restricts tracing to specified instances where poor performance is known to exist. This procedure also gives you the option of capturing relevant waits and bind variable values in the generated trace files. SERV_MOD_ACT_TRACE_DISABLE disables the tracing at all enabled instances for a given combination of service, module, and action. Like the statistics gathering mentioned previously, service tracing persists across instance restarts.

Service, Module, and Action Monitoring

- For the ERP service, enable monitoring for the exceptions pay action in the PAYROLL module.

```
EXEC DBMS_MONITOR.SERV_MOD_ACT_STAT_ENABLE(  
service_name => 'ERP', module_name=> 'PAYROLL',  
action_name => 'EXCEPTIONS PAY')
```

- For the ERP service, enable monitoring for all the actions in the PAYROLL module:

```
EXEC DBMS_MONITOR.SERV_MOD_ACT_STAT_ENABLE(service_name =>  
'ERP', module_name=> 'PAYROLL', action_name => NULL);
```

- For the HOT_BATCH service, enable monitoring for all actions in the posting module:

```
EXEC DBMS_MONITOR.SERV_MOD_ACT_STAT_ENABLE(service_name =>  
'HOT_BATCH', module_name => 'POSTING', action_name => NULL);
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Service, Module, and Action Monitoring

You can enable performance data tracing for important modules and actions within each service. The performance statistics are available in the V\$SERV_MOD_ACT_STATS view. Consider the following actions, as implemented in the slide above:

- For the ERP service, enable monitoring for the exceptions pay action in the payroll module.
- Under the ERP service, enable monitoring for all the actions in the payroll module.
- Under the HOT_BATCH service, enable monitoring for all actions in the posting module.

Verify the enabled service, module, action configuration with the SELECT statement below:

```
COLUMN AGGREGATION_TYPE FORMAT A21 TRUNCATED HEADING 'AGGREGATION'  
COLUMN PRIMARY_ID FORMAT A20 TRUNCATED HEADING 'SERVICE'  
COLUMN QUALIFIER_ID1 FORMAT A20 TRUNCATED HEADING 'MODULE'  
COLUMN QUALIFIER_ID2 FORMAT A20 TRUNCATED HEADING 'ACTION'  
SELECT * FROM DBA_ENABLED_AGGREGATIONS ;
```

The output might appear as follows:

AGGREGATION	SERVICE	MODULE	ACTION
SERVICE_MODULE_ACTION	ERP	PAYROLL	EXCEPTIONS PAY
SERVICE_MODULE_ACTION	ERP	PAYROLL	
SERVICE_MODULE_ACTION	HOT_BATCH	POSTING	

Service Performance Views

- Service, module, and action information in:
 - V\$SESSION
 - V\$ACTIVE_SESSION_HISTORY
- Service performance in:
 - V\$SERVICE_STATS
 - V\$SERVICE_EVENT
 - V\$SERVICE_WAIT_CLASS
 - V\$SERVICEMETRIC
 - V\$SERVICEMETRIC_HISTORY
 - V\$SERV_MOD_ACT_STATS
 - DBA_ENABLED_AGGREGATIONS
 - DBA_ENABLED_TRACES

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Service Performance Views

The service, module, and action information are visible in V\$SESSION and V\$ACTIVE_SESSION_HISTORY.

The call times and performance statistics are visible in V\$SERVICE_STATS, V\$SERVICE_EVENT, V\$SERVICE_WAIT_CLASS, V\$SERVICEMETRIC, and V\$SERVICEMETRIC_HISTORY.

When statistics collection for specific modules and actions is enabled, performance measures are visible at each instance in V\$SERV_MOD_ACT_STATS.

There are more than 600 performance-related statistics that are tracked and visible in V\$SYSSTAT. Of these, 28 statistics are tracked for services. To see the statistics measured for services, run the following query: `SELECT DISTINCT stat_name FROM v$service_stats`

Of the 28 statistics, DB time and DB CPU are worth mentioning. DB time is a statistic that measures the average response time per call. It represents the actual wall clock time for a call to complete. DB CPU is an average of the actual CPU time spent per call. The difference between response time and CPU time is the wait time for the service. After the wait time is known, and if it consumes a large percentage of response time, then you can trace at the action level to identify the waits.

Note: DBA_ENABLED_AGGREGATIONS displays information about enabled on-demand statistic aggregation. DBA_ENABLED_TRACES displays information about enabled traces.

Quiz

Which of the following statements regarding Oracle Services is *not* correct?

1. You can group work by type under services.
2. Users who share a service should have the same service-level requirements.
3. Use `DBMS_SERVICE` to manage services, not `srvctl` or Enterprise Manager.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Answer: 3

Statement 3 is not correct.

Quiz

Is the following statement regarding performance thresholds true or false? The two performance thresholds that can be explicitly set for each service are:

- (a) SERVICE_ELAPSED_TIME: The response time for calls
- (b) SERVICE_CPU_TIME: The CPU time for calls

1. True
2. False

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Answer: 1

The statement is true.

Summary

In this lesson, you should have learned how to:

- Configure and manage services in a RAC environment
- Use services with client applications
- Use services with the Database Resource Manager
- Use services with the Scheduler
- Set performance-metric thresholds on services
- Configure services aggregation and tracing

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Practice 15 Overview

This practice covers the following topics:

- Creating and managing services using EM
- Using server-generated alerts in combination with services

ORACLE

Copyright © 2010, Oracle. All rights reserved.

16

Design for High Availability

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Objectives

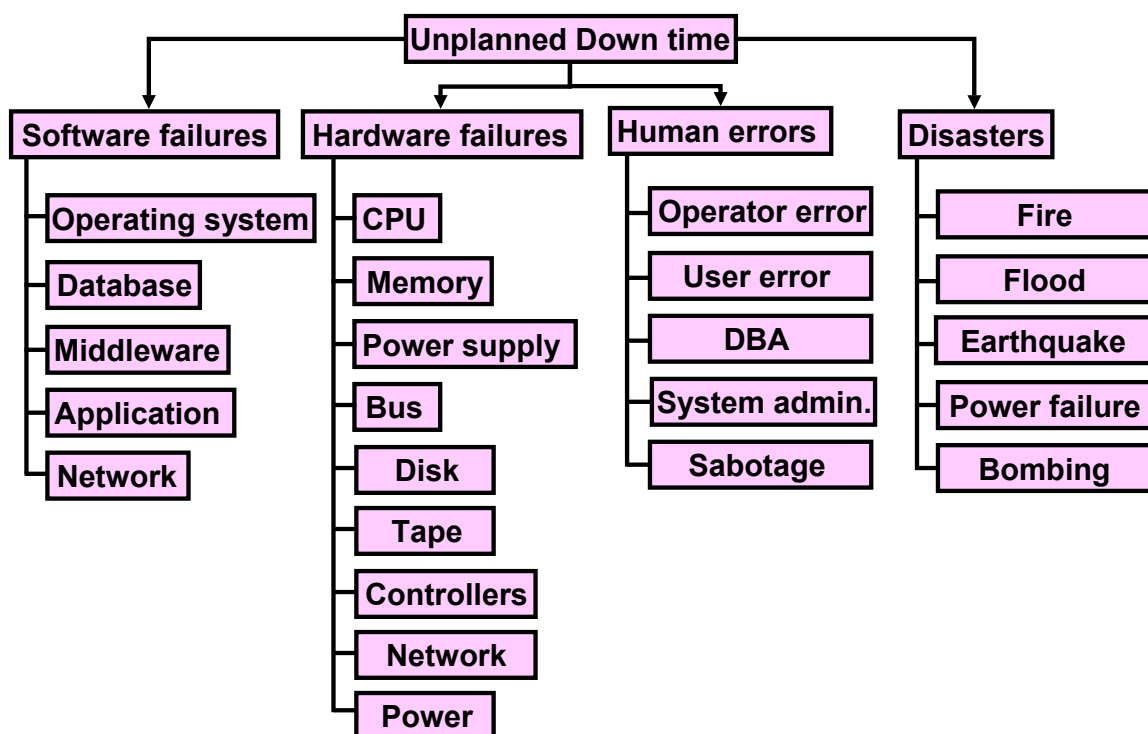
After completing this lesson, you should be able to:

- Design a Maximum Availability Architecture in your environment
- Determine the best RAC and Data Guard topologies for your environment
- Configure the Data Guard Broker configuration files in a RAC environment
- Identify successful disk I/O strategies

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Causes of Unplanned Down Time



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Causes of Unplanned Down Time

One of the true challenges in designing a highly available solution is examining and addressing all the possible causes of down time. It is important to consider causes of both unplanned and planned down time. The diagram shown in the slide, which is a taxonomy of unplanned failures, classifies failures as software failures, hardware failures, human error, and disasters. Under each category heading is a list of possible causes of failures related to that category.

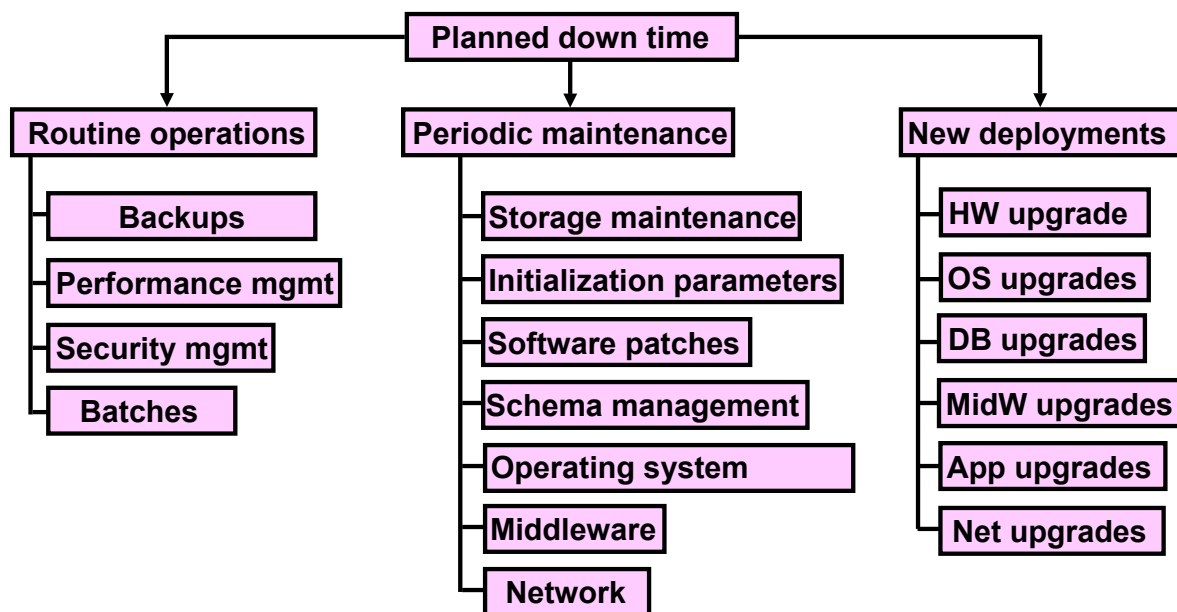
Software failures include operating system, database, middleware, application, and network failures. A failure of any one of these components can cause a system fault.

Hardware failures include system, peripheral, network, and power failures.

Human error, which is a leading cause of failures, includes errors by an operator, user, database administrator, or system administrator. Another type of human error that can cause unplanned down time is sabotage.

The final category is disasters. Although infrequent, these can have extreme impacts on enterprises, because of their prolonged effect on operations. Possible causes of disasters include fires, floods, earthquakes, power failures, and bombings. A well-designed high-availability solution accounts for all these factors in preventing unplanned down time.

Causes of Planned Down Time



ORACLE

Copyright © 2010, Oracle. All rights reserved.

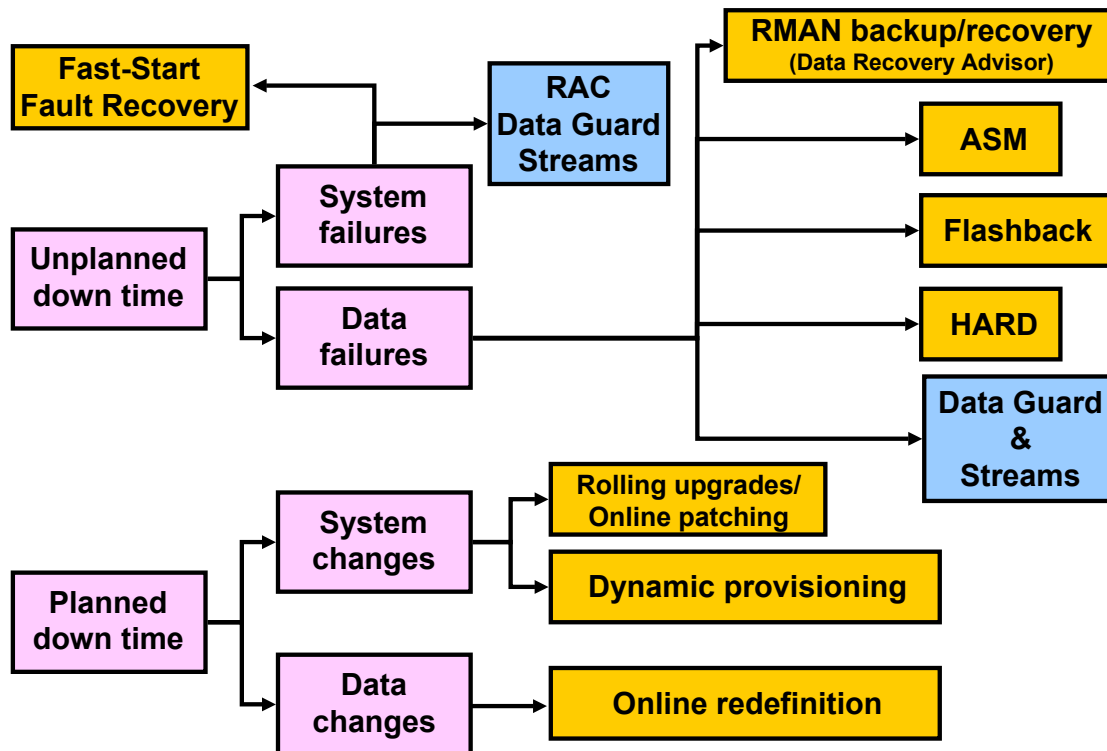
Causes of Planned Down Time

Planned down time can be just as disruptive to operations, especially in global enterprises that support users in multiple time zones, up to 24 hours per day. In these cases, it is important to design a system to minimize planned interruptions. As shown by the diagram in the slide, causes of planned down time include routine operations, periodic maintenance, and new deployments. Routine operations are frequent maintenance tasks that include backups, performance management, user and security management, and batch operations.

Periodic maintenance, such as installing a patch or reconfiguring the system, is occasionally necessary to update the database, application, operating system middleware, or network.

New deployments describe major upgrades to the hardware, operating system, database, application, middleware, or network. It is important to consider not only the time to perform the upgrade, but also the effect the changes may have on the overall application.

Oracle's Solution to Down Time



ORACLE

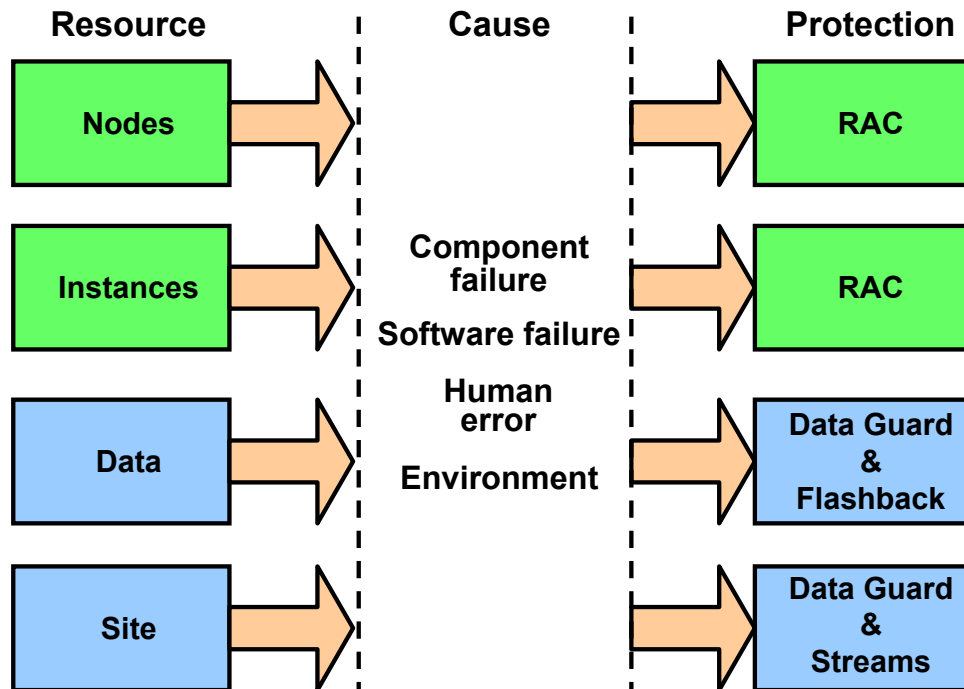
Copyright © 2010, Oracle. All rights reserved.

Oracle's Solution to Down Time

Unplanned down time is primarily the result of computer failures or data failures. Planned down time is primarily due to data changes or system changes:

- RAC provides optimal performance, scalability, and availability gains.
- Fast-Start Fault Recovery enables you to bound the crash/recovery time. The database self-tunes checkpoint processing to safeguard the desired recovery time objective.
- ASM provides a higher level of availability using online provisioning of storage.
- Flashback provides a quick resolution to human errors.
- Oracle Hardware Assisted Resilient Data (HARD) is a comprehensive program designed to prevent data corruptions before they happen.
- Recovery Manager (RMAN) automates database backup and recovery. Data Recovery Advisor (not supported for RAC) diagnoses data failures and presents repair options.
- Data Guard must be the foundation of any Oracle database disaster-recovery plan.
- The increased flexibility and capability of Streams over Data Guard with SQL Apply requires more expense and expertise to maintain an integrated high availability solution.
- With online redefinition, the Oracle database supports many maintenance operations without disrupting database operations, or users updating or accessing data.
- Oracle Database continues to broaden support for dynamic reconfiguration, enabling it to adapt to changes in demand and hardware with no disruption of service.
- Oracle Database supports the application of patches to the nodes of a RAC system, as well as database software upgrades, in a rolling fashion.

RAC and Data Guard Complementarity



ORACLE

Copyright © 2010, Oracle. All rights reserved.

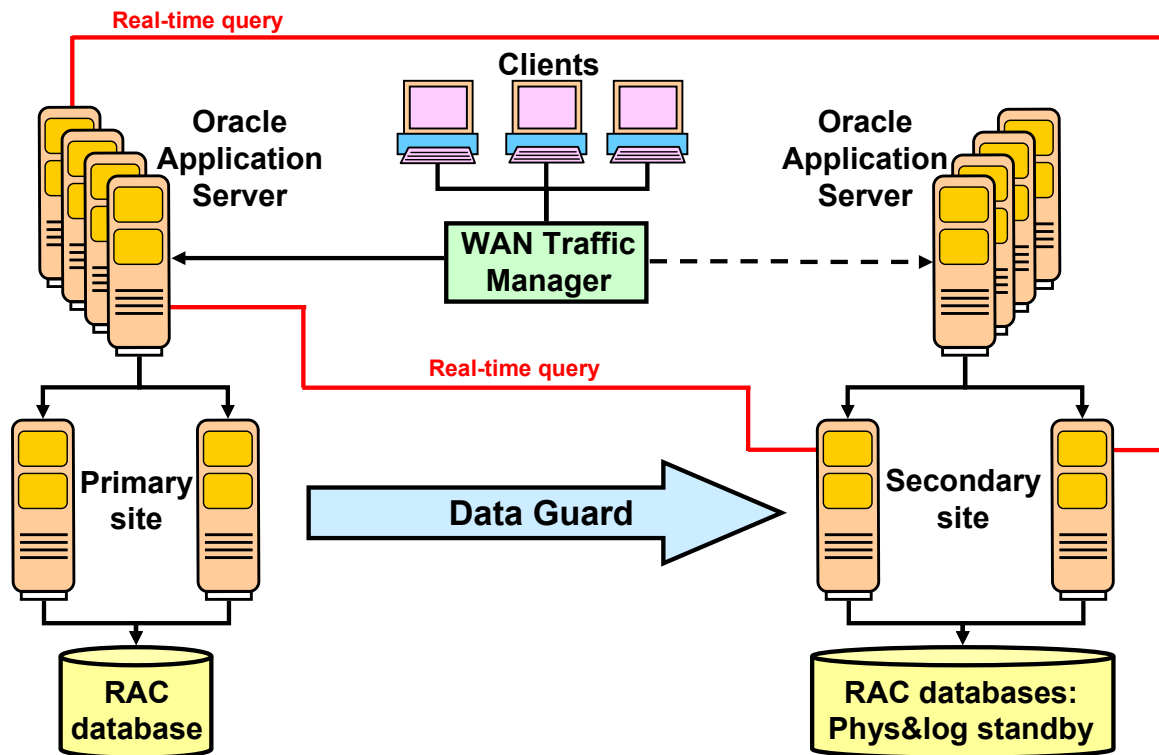
RAC and Data Guard Complementarity

RAC and Data Guard together provide the benefits of system-level, site-level, and data-level protection, resulting in high levels of availability and disaster recovery without loss of data.

- RAC addresses system failures by providing rapid and automatic recovery from failures, such as node failures and instance crashes.
- Data Guard addresses site failures and data protection through transactionally consistent primary and standby databases that do not share disks, enabling recovery from site disasters and data corruption.

Note: Unlike Data Guard using SQL Apply, Oracle Streams enables updates on the replica and provides support for heterogeneous platforms with different database releases. Therefore, Oracle Streams may provide the fastest approach for database upgrades and platform migration.

Maximum Availability Architecture



Copyright © 2010, Oracle. All rights reserved.

Maximum Availability Architecture (MAA)

RAC and Data Guard provide the basis of the database MAA solution. MAA provides the most comprehensive architecture for reducing down time for scheduled outages and preventing, detecting, and recovering from unscheduled outages. The recommended MAA has two identical sites. The primary site contains the RAC database, and the secondary site contains both a physical standby database and a logical standby database on RAC. Identical site configuration is recommended to ensure that performance is not sacrificed after a failover or switchover. Symmetric sites also enable processes and procedures to be kept the same between sites, making operational tasks easier to maintain and execute.

The graphic illustrates identically configured sites. Each site consists of redundant components and redundant routing mechanisms, so that requests are always serviceable even in the event of a failure. Most outages are resolved locally. Client requests are always routed to the site playing the production role.

After a failover or switchover operation occurs due to a serious outage, client requests are routed to another site that assumes the production role. Each site contains a set of application servers or mid-tier servers. The site playing the production role contains a production database using RAC to protect from host and instance failures. The site playing the standby role contains one standby database, and one logical standby database managed by Data Guard. Data Guard switchover and failover functions allow the roles to be traded between sites.

Note: For more information, see the following Web site:
<http://otn.oracle.com/deploy/availability/htdocs/maa.htm>

RAC and Data Guard Topologies

- Symmetric configuration with RAC at all sites:
 - Same number of instances
 - Same service preferences
- Asymmetric configuration with RAC at all sites:
 - Different number of instances
 - Different service preferences
- Asymmetric configuration with mixture of RAC and single instance:
 - All sites running under Oracle Clusterware
 - Some single-instance sites not running under Oracle Clusterware

ORACLE

Copyright © 2010, Oracle. All rights reserved.

RAC and Data Guard Topologies

You can configure a standby database to protect a primary database in a RAC environment. Basically, all kinds of combinations are supported. For example, it is possible to have your primary database running under RAC, and your standby database running as a single-instance database. It is also possible to have both the primary and standby databases running under RAC.

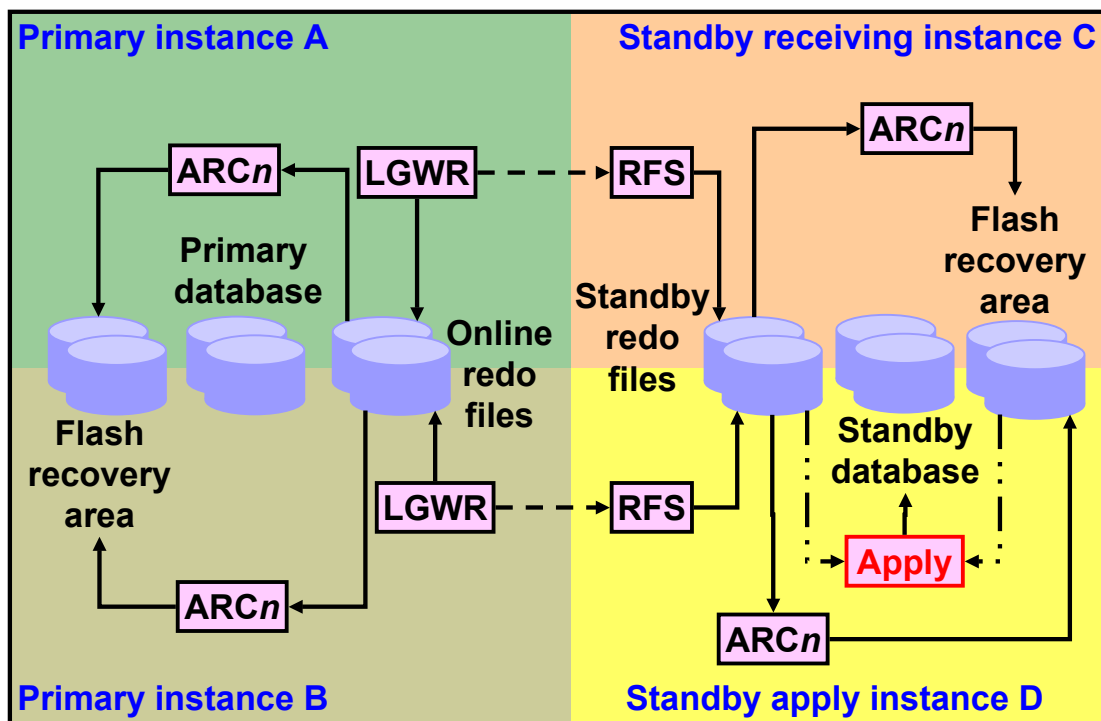
The slide explains the distinction between symmetric environments and asymmetric ones.

If you want to create a symmetric environment running RAC, then all databases need to have the same number of instances and the same service preferences. As the DBA, you need to make sure that this is the case by manually configuring them in a symmetric way.

However, if you want to benefit from the tight integration of Oracle Clusterware and Data Guard Broker, make sure that both the primary site and the secondary site are running under Oracle Clusterware, and that both sites have the same services defined.

Note: Beginning with Oracle Database 11g, the primary and standby systems in a Data Guard configuration can have different CPU architectures, operating systems (for example, Windows and Linux for physical standby database only with no EM support for this combination), operating system binaries (32-bit and 64-bit), and Oracle database binaries (32-bit and 64-bit).

RAC and Data Guard Architecture



ORACLE

Copyright © 2010, Oracle. All rights reserved.

RAC and Data Guard Architecture

Although it is perfectly possible to use a “RAC to single-instance Data Guard (DG)” configuration, you also have the possibility to use a RAC-to-RAC DG configuration. In this mode, although multiple standby instances can receive redo from the primary database, only one standby instance can apply the redo stream generated by the primary instances.

A RAC-to-RAC DG configuration can be set up in different ways, and the slide shows you one possibility with a symmetric configuration where each primary instance sends its redo stream to a corresponding standby instance using standby redo log files. It is also possible for each primary instance to send its redo stream to only one standby instance that can also apply this stream to the standby database. However, you can get performance benefits by using the configuration shown in the slide. For example, assume that the redo generation rate on the primary is too great for a single receiving instance on the standby side to handle. Suppose further that the primary database is using the SYNC redo transport mode. If a single receiving instance on the standby cannot keep up with the primary, then the primary’s progress is going to be throttled by the standby. If the load is spread across multiple receiving instances on the standby, then this is less likely to occur.

If the standby can keep up with the primary, another approach is to use only one standby instance to receive and apply the complete redo stream. For example, you can set up the primary instances to remotely archive to the same Oracle Net service name.

RAC and Data Guard Architecture (continued)

You can then configure one of the standby nodes to handle that service. This instance then both receives and applies redo from the primary. If you need to do maintenance on that node, then you can stop the service on that node and start it on another node. This approach allows for the primary instances to be more independent of the standby configuration because they are not configured to send redo to a particular instance.

Note: For more information, refer to the *Oracle Data Guard Concepts and Administration* guide.

Data Guard Broker (DGB) and Oracle Clusterware (OC) Integration

- OC manages intrasite HA operations.
- OC manages intrasite planned HA operations.
- OC notifies when manual intervention is required.
- DBA receives notification.
- DBA decides to switch over or fail over using DGB.
- DGB manages intersite planned HA operations.
- DGB takes over from OC for intersite failover, switchover, and protection mode changes:
 - DMON notifies OC to stop and disable the site, leaving all or one instance.
 - DMON notifies OC to enable and start the site according to the DG site role.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Guard Broker (DGB) and Oracle Clusterware Integration

DGB is tightly integrated with Oracle Clusterware. Oracle Clusterware manages individual instances to provide unattended high availability of a given clustered database. DGB manages individual databases (clustered or otherwise) in a Data Guard configuration to provide disaster recovery in the event that Oracle Clusterware is unable to maintain availability of the primary database.

For example, Oracle Clusterware posts NOT_RESTARTING events for the database group and service groups that cannot be recovered. These events are available through Enterprise Manager, ONS, and server-side callouts. As a DBA, when you receive those events, you might decide to repair and restart the primary site, or to invoke DGB to fail over if not using Fast-Start Failover.

DGB and Oracle Clusterware work together to temporarily suspend service availability on the primary database, accomplish the actual role change for both databases during which Oracle Clusterware works with the DGB to properly restart the instances as necessary, and then to resume service availability on the new primary database. The broker manages the underlying Data Guard configuration and its database roles, whereas Oracle Clusterware manages service availability that depends upon those roles. Applications that rely upon Oracle Clusterware for managing service availability will see only a temporary suspension of service as the role change occurs within the Data Guard configuration.

Fast-Start Failover: Overview

- Fast-Start Failover implements automatic failover to a standby database:
 - Triggered by failure of site, hosts, storage, data file offline immediate, or network
 - Works with and supplements RAC server failover
- Failover occurs in seconds (< 20 seconds).
 - Comparable to cluster failover
- Original production site automatically rejoins the configuration after recovery.
- Automatically monitored by an Observer process:
 - Locate it on a distinct server on a distinct data center
 - Enterprise Manager can restart it on failure
 - Installed through Oracle Client Administrator

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Fast-Start Failover: Overview

Fast-Start Failover is a feature that automatically, quickly, and reliably fails over to a designated, synchronized standby database in the event of loss of the primary database, without requiring manual intervention to execute the failover. In addition, following a fast-start failover, the original primary database is automatically reconfigured as a new standby database upon reconnection to the configuration. This enables Data Guard to restore disaster protection in the configuration as soon as possible.

Fast-Start Failover is used in a Data Guard configuration under the control of the Data Guard Broker, and may be managed using either `dgmgrl` or Oracle Enterprise Manager Grid Control. There are three essential participants in a Fast-Start Failover configuration:

- The primary database, which can be a RAC database
- A target standby database, which becomes the new primary database following a fast-start failover
- The Fast-Start Failover Observer, which is a separate process incorporated into the `dgmgrl` client that continuously monitors the primary database and the target standby database for possible failure conditions. The underlying rule is that out of these three participants, whichever two can communicate with each other will determine the outcome of the fast-start failover. In addition, a fast-start failover can occur only if there is a guarantee that no data will be lost.

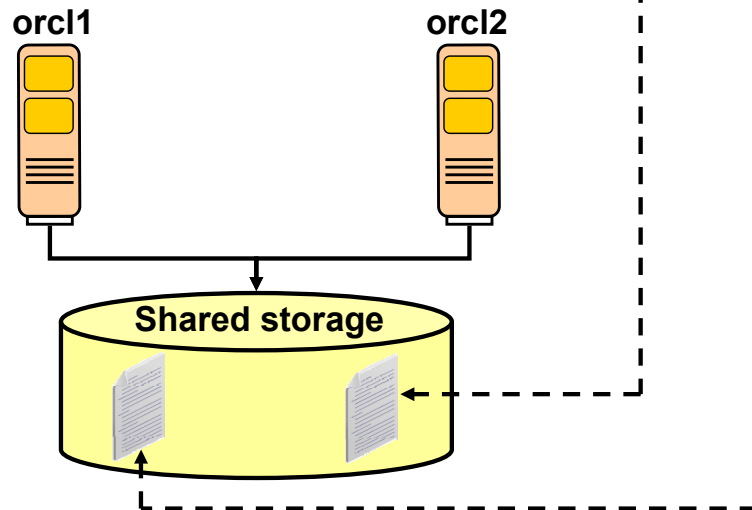
Fast-Start Failover: Overview (continued)

For disaster recovery requirements, install the Observer in a location separate from the primary and standby data centers. If the designated Observer fails, Enterprise Manager can detect the failure and can be configured to automatically restart the Observer on the same host.

You can install the Observer by installing the Oracle Client Administrator (choose the Administrator option from the Oracle Universal Installer). Installing the Oracle Client Administrator results in a small footprint because an Oracle instance is not included on the Observer system. If Enterprise Manager is used, also install the Enterprise Manager Agent on the Observer system.

Data Guard Broker Configuration Files

```
*.DG_BROKER_CONFIG_FILE1=+DG1/orcl/dr1config.dat -  
*.DG_BROKER_CONFIG_FILE2=+DG1/orcl/dr2config.dat
```



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Data Guard Broker Configuration Files

Two copies of the Data Guard Broker (DGB) configuration files are maintained for each database so as to always have a record of the last-known valid state of the configuration. When the broker is started for the first time, the configuration files are automatically created and named using a default path name and file name that is operating system specific.

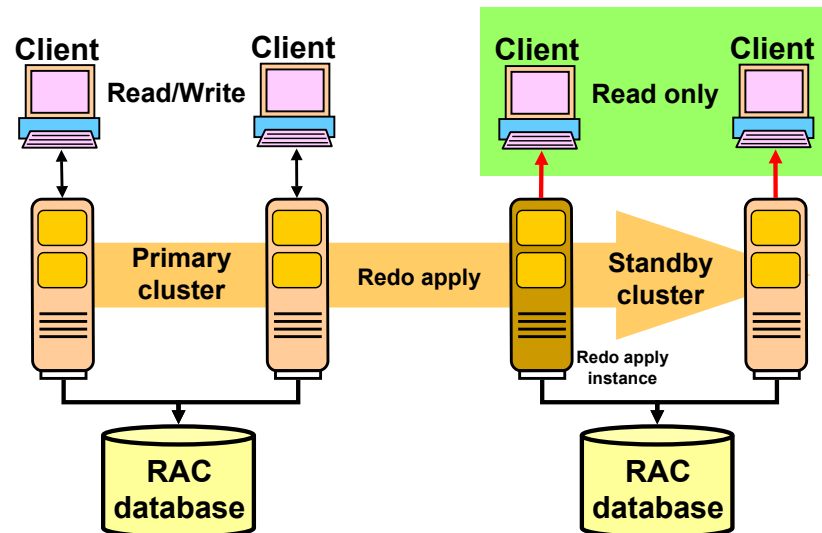
When using a RAC environment, the DGB configuration files must be shared by all instances of the same database. You can override the default path name and file name by setting the following initialization parameters for that database: `DG_BROKER_CONFIG_FILE1`, `DG_BROKER_CONFIG_FILE2`.

You have three possible options to share those files:

- Cluster file system
- Raw devices
- ASM

The example in the slide illustrates a case where those files are stored in an ASM disk group called `DG1`. It is assumed that you have already created a directory called `orcl` in `DG1`.

Real-Time Query Physical Standby Database



ORACLE

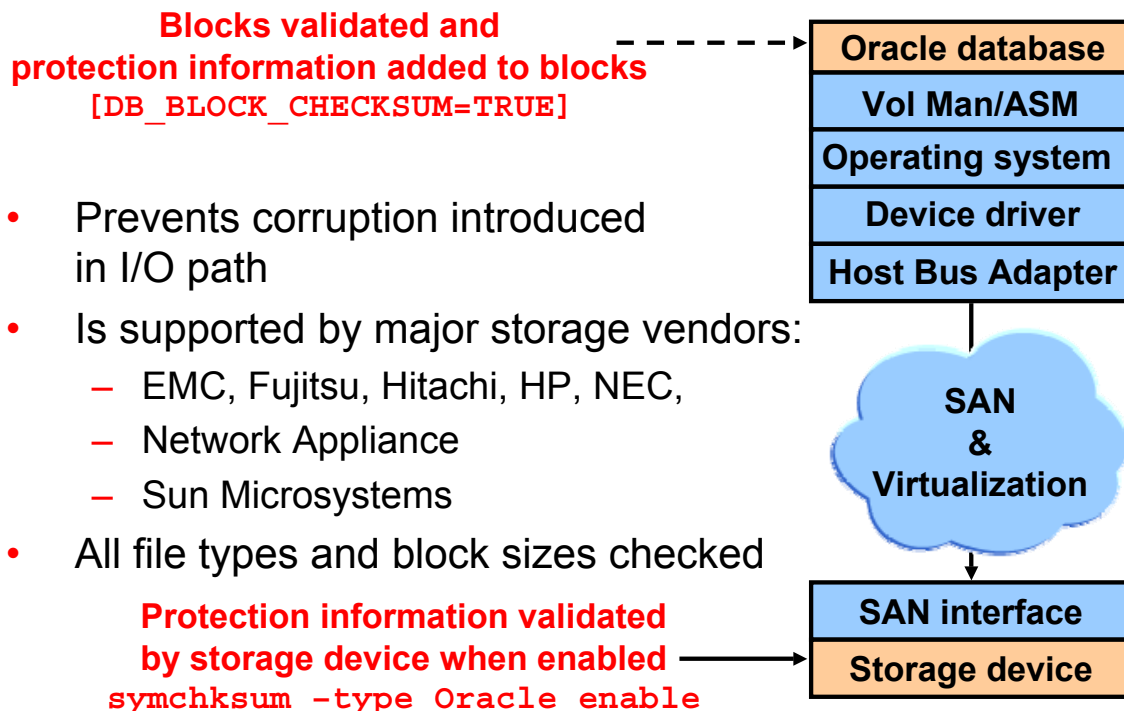
Copyright © 2010, Oracle. All rights reserved.

Real-Time Query Physical Standby Database

Data Guard Redo Apply (physical standby database) has proven to be a popular solution for disaster recovery due to its relative simplicity, high performance, and superior level of data protection. Beginning with Oracle Database 11g, a physical standby database can be open read-only while redo apply is active. This means that you can run queries and reports against an up-to-date physical standby database without compromising data protection or extending recovery time in the event a failover is required. This makes every physical standby database able to support productive uses even while in standby role. To enable real-time query, open the database in read-only mode and then issue the `ALTER DATABASE RECOVER MANAGED STANDBY` statement. Real-time query provides an ultimate high availability solution because it:

- Is totally transparent to applications
- Supports Oracle RAC on the primary and standby databases: Although Redo Apply can be running on only one Oracle RAC instance, you can have all of the instances running in read-only mode while Redo Apply is running on one instance.
- Returns transactionally consistent results that are very close to being up-to-date with the primary database.
- Enables you to use fast-start failover to allow for automatic fast failover in the case the primary database fails

Hardware Assisted Resilient Data



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Hardware Assisted Resilient Data

One problem that can cause lengthy outages is data corruption. Today, the primary means for detecting corruptions caused by hardware or software outside of the Oracle database, such as an I/O subsystem, is the Oracle database checksum. However, after a block is passed to the operating system, through the volume manager and out to disk, the Oracle database itself cannot validate whether the block being written is still correct.

With disk technologies expanding in complexity, and with configurations such as Storage Area Networks (SANs) becoming more popular, the number of layers between the host processor and the physical spindle continues to increase. With more layers, the chance of any problem increases. With the HARD initiative, it is possible to enable the verification of database block checksum information by the storage device. Verifying that the block is still the same at the end of the write as it was in the beginning gives you an additional level of security.

By default, the Oracle database automatically adds checksum information to its blocks. These checksums can be verified by the storage device if you enable this possibility. In case a block is found to be corrupted by the storage device, the device logs an I/O corruption, or it cancels the I/O and reports the error back to the instance.

Note: The way you enable the checksum validation at the storage device side is vendor specific. The example given in the slide was used with EMC Symmetrix storage.

Database High Availability: Best Practices

Use SPFILE.	Create two or more control files.	Set CONTROL_FILE_RECOVER_TIME long enough.	Multiplex production and standby redo logs
Log checkpoints to the alert log.	Use auto-tune checkpointing.	Enable ARCHIVELOG mode and use a flash recovery area.	Enable Flashback Database.
Enable block checking.	Use Automatic Undo Management.	Use locally managed tablespaces.	Use Automatic Segment Space Management.
Use resumable space allocation.	Use Database Resource Manager.	Register all instances with remote listeners.	Use temporary tablespaces.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Database High Availability: Best Practices

The table in the slide gives you a short summary of the recommended practices that apply to single-instance databases, RAC databases, and Data Guard standby databases.

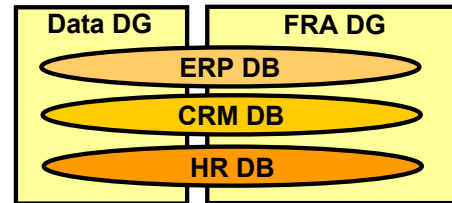
These practices affect the performance, availability, and mean time to recover (MTTR) of your system. Some of these practices may reduce performance, but they are necessary to reduce or avoid outages. The minimal performance impact is outweighed by the reduced risk of corruption or the performance improvement for recovery.

Note: For more information about how to set up the features listed in the slide, refer to the following documents:

- *Administrator's Guide*
- *Data Guard Concepts and Administration*
- *Net Services Administrator's Guide*

How Many ASM Disk Groups Per Database?

- Two disk groups are recommended.
 - Leverage maximum of LUNs.
 - Backups can be stored on one FRA disk group.
 - Lower performance may be used for FRA (or inner tracks).
- Exceptions:
 - Additional disk groups for different capacity or performance characteristics
 - Different ILM storage tiers



ORACLE

Copyright © 2010, Oracle. All rights reserved.

How Many ASM Disk Groups Per Database?

Most of the time, only two disk groups are enough to share the storage between multiple databases.

That way you can maximize the number of Logical Unit Numbers (LUNs) used as ASM disks, which gives you the best performance, especially if these LUNs are carved on the outer edge of your disks.

Using a second disk group allows you to have a backup of your data by using it as your common fast recovery area (FRA). You can put the corresponding LUNs on the inner edge of your disks because less performance is necessary.

The two noticeable exceptions to this rule are whenever you are using disks with different capacity or performance characteristics, or when you want to archive your data on lower-end disks for Information Lifecycle Management (ILM) purposes.

Which RAID Configuration for High Availability?

- A. ASM mirroring
- B. Hardware RAID 1 (mirroring)
- C. Hardware RAID 5 (parity protection)
- ~~D. Both ASM mirroring and hardware RAID~~

Answer: **Depends on business requirement and budget (cost, availability, performance, and utilization)**

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Which RAID Configuration for Best Availability?

To favor availability, you have multiple choices as shown in the slide.

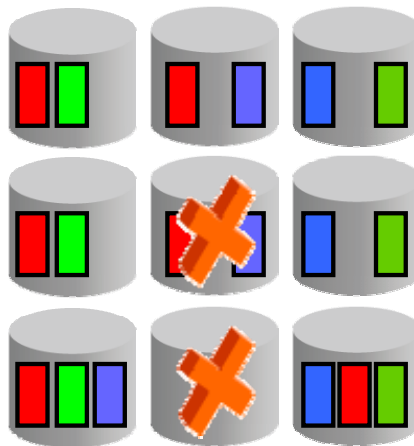
You could just use ASM mirroring capabilities, or hardware RAID 1 (Redundant Array of Inexpensive Disks) which is a hardware mirroring technique, or hardware RAID 5. The last possible answer, which is definitely not recommended, is to use both ASM mirroring and hardware mirroring. Oracle recommends the use of external redundancy disk groups when using hardware mirroring techniques to avoid an unnecessary overhead.

Therefore, between A, B, and C, the choice depends on your business requirements and budget.

RAID 1 has the best performance but requires twice the storage capacity. RAID 5 is a much more economical solution but with a performance penalty essentially for write-intensive workloads.

Should You Use ASM Mirroring Protection?

- Best choice for low-cost storage
- Enables extended clustering solutions
- No hardware mirroring



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Should You Use ASM Mirroring Protection?

Basically, leverage the storage array hardware RAID-1 mirroring protection when possible to offload the mirroring overhead from the server. Use ASM mirroring in the absence of a hardware RAID capability.

However hardware RAID 1 in most Advanced Technology Attachment (ATA) storage technologies is inefficient and degrades the performance of the array even more. Using ASM redundancy has proven to deliver much better performance in ATA arrays.

Because the storage cost can grow very rapidly whenever you want to achieve extended clustering solutions, ASM mirroring should be used as an alternative to hardware mirroring for low-cost storage solutions.

What Type of Striping Works Best?

- A. ASM only striping (no RAID 0)
- B. RAID 0 and ASM striping
- ~~C. Use LVM~~
- ~~D. No striping~~

Answer: **A and B**

ASM and RAID striping are complementary.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

What Type of Striping Works Best?

As shown in the slide, you can use ASM striping only, or you can use ASM striping in combination with RAID 0.

With RAID 0, multiple disks are configured together as a set, or a bank, and data from any one data file is spread, or striped, across all the disks in the bank.

Combining both ASM striping and RAID striping is called stripe-on-stripe. This combination offers good performance too.

However, there is no longer a need to use a Logical Volume Manager (LVM) for your database files, nor is it recommended to not use any striping at all.

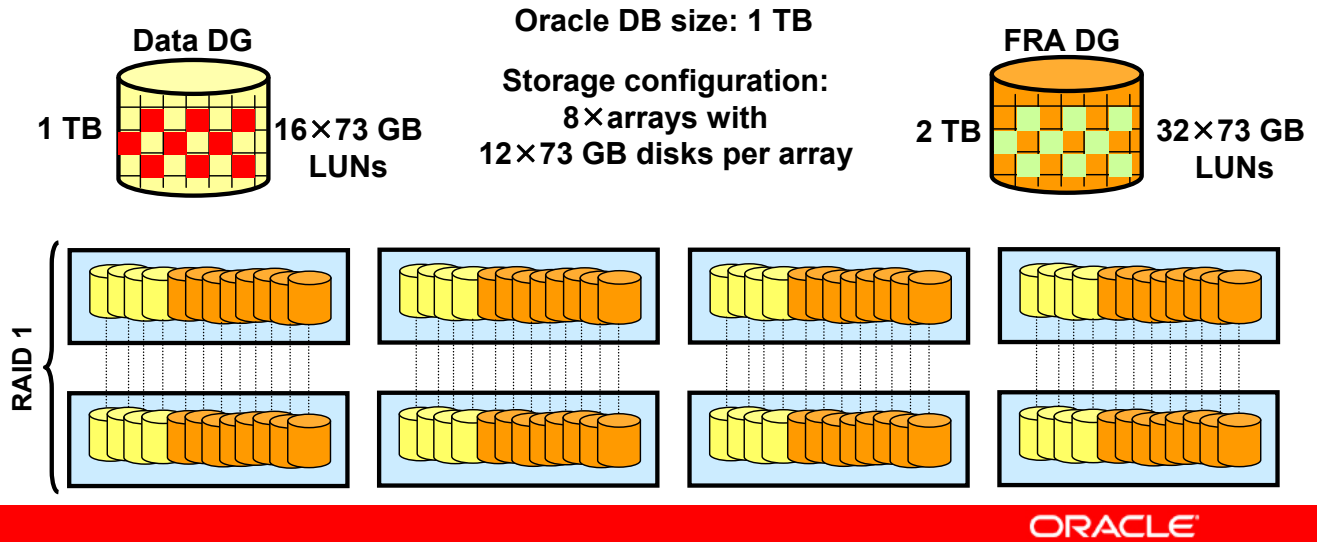
ASM Striping Only

Pros:

- Drives evenly distributed for Data & FRA
- Higher bandwidth
- Allows small incremental growth (73 GB)
- No drive contention

Cons:

- Not well balanced across ALL disks
- LUN size limited to disk size



Copyright © 2010, Oracle. All rights reserved.

ASM Striping Only

In the case shown in this slide, you want to store a one-terabyte database with a corresponding two-terabyte flash recovery area. You use RAID 1 to mirror each disk. In total, you have eight arrays of twelve disks, with each disk being 73 GB. ASM mirroring and hardware RAID 0 are not used.

In addition, each ASM disk is represented by one entire LUN of 73 GB. This means that the Data disk group (DG) is allocated 16 LUNs of 73 GB each.

On the other side, the Fast Recovery Area disk group is assigned 32 LUNs of 73 GB each.

This configuration enables you to evenly distribute disks for your data and backups, achieving good performance and allowing you to manage your storage in small incremental chunks.

However, using a restricted number of disks in your pool does not balance your data well across all your disks. In addition, you have many LUNs to manage at the storage level.

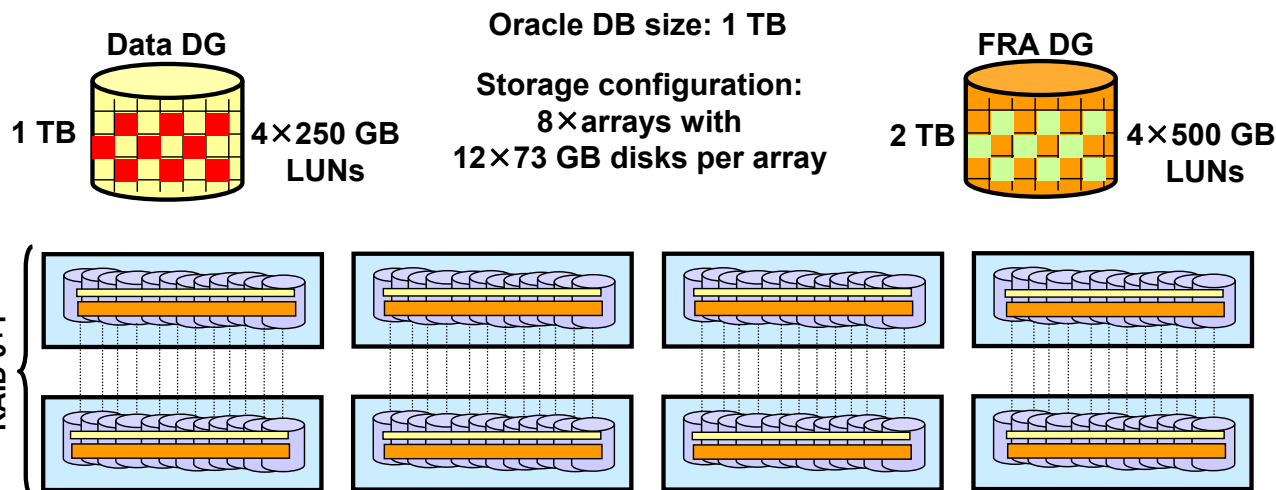
Hardware RAID–Striped LUNs

Pros:

- Fastest region for Data DG
- Balanced data distribution
- Fewer LUNs to manage while max spindles

Cons:

- Large incremental growth
- Data & FRA “contention”



Copyright © 2010, Oracle. All rights reserved.

Hardware RAID–Striped LUNs

In the case shown in this slide, you want to store a one-terabyte database with a corresponding two-terabyte flash recovery area. You use RAID 0+1, which is a combination of hardware striping and mirroring to mirror and stripe each disk. In total, you have eight arrays of twelve disks, with each disk being 73 GB. ASM mirroring is not used.

Here, you can define bigger LUNs not restricted to the size of one of your disks. This allows you to put the Data LUNs on the fastest region of your disks, and the backup LUNs on slower parts. By doing this, you achieve a better data distribution across all your disks, and you end up managing a significantly less number of LUNs.

However, you must manipulate your storage in much larger chunks than in the previous configuration.

Note: The hardware stripe size you choose is also very important because you want 1 MB alignment as much as possible to keep in sync with ASM AUs. Therefore, selecting power-of-two stripe sizes (128 KB or 256 KB) is better than selecting odd numbers. Storage vendors typically do not offer many flexible choices depending on their storage array RAID technology and can create unnecessary I/O bottlenecks if not carefully considered.

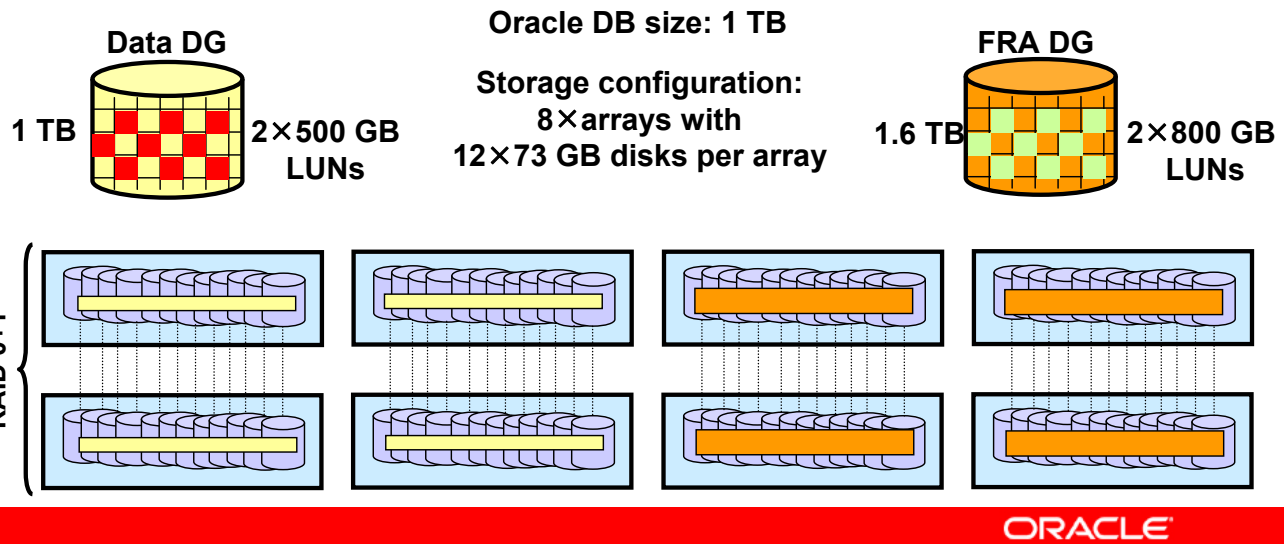
Hardware RAID–Striped LUNs HA

Pros:

- Fastest region for Data DG
- Balanced data distribution
- Fewer LUNs to manage
- More high available

Cons:

- Large incremental growth
- Might waste space



Copyright © 2010, Oracle. All rights reserved.

Hardware RAID–Striped LUNs HA

In the case shown in this slide, you want to store a one-terabyte database with a corresponding 1.6-TB fast recovery area. You use RAID 0+1, which is a combination of hardware striping and mirroring to mirror and stripe each disk. In total, you have eight arrays of twelve disks, with each disk being 73 GB. ASM mirroring is not used.

Compared to the previous slide, you use bigger LUNs for both the Data disk group and the Fast Recovery Area disk group. However, the presented solution is more highly available than the previous architecture because you separate the data from the backups into different arrays and controllers to reduce the risk of down time in case one array fails.

By doing this, you still have a good distribution of data across your disks, although not as much as in the previous configuration. You still end up managing a significantly less number of LUNs than in the first case.

However, you might end up losing more space than in the previous configuration. Here, you are using the same size and number of arrays to be consistent with the previous example.

Disk I/O Design Summary

- Use external RAID protection when possible.
- Create LUNs by using:
 - Outside half of disk drives for highest performance
 - Small disk, high rpm (that is, 73 GB/15k rpm)
- Use LUNs with the same performance characteristics.
- Use LUNs with the same capacity.
- Maximize the number of spindles in your disk group.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

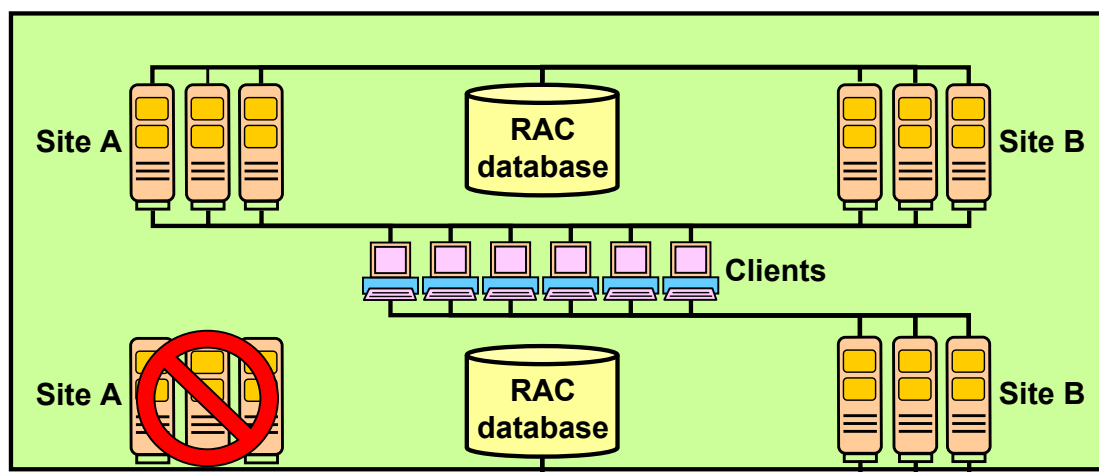
Disk I/O Design Summary

Use ASM for volume and file management to equalize the workload across disks and eliminate hot spots. The following are simple guidelines and best practices when configuring ASM disk groups:

- Use external RAID protection when possible.
- Create LUNs using:
 - Outside half of disk drives for highest performance
 - Small disk with high rpm (for example, 73 GB with 15k rpm). The reason why spindle (platter) speed is so important is that it directly impacts both positioning time and data transfer. This means that faster spindle speed drives have improved performance regardless of whether they are used for many small, random accesses, or for streaming large contiguous blocks from the disk. The stack of platters in a disk rotates at a constant speed. The drive head, while positioned close to the center of the disk, reads from a surface that is passing by more slowly than the surface at the outer edges.
- Maximize the number of spindles in your disk group.
- LUNs provisioned to ASM disk groups should have the same storage performance and availability characteristics. Configuring mixed speed drives will default to the lowest common denominator.
- ASM data distribution policy is capacity based. Therefore, LUNs provided to ASM should have the same capacity for each disk group to avoid imbalance and hot spots.

Extended RAC: Overview

- Full utilization of resources, no matter where they are located



- Fast recovery from site failure

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Extended RAC: Overview

Typically, RAC databases share a single set of storage and are located on servers in the same data center.

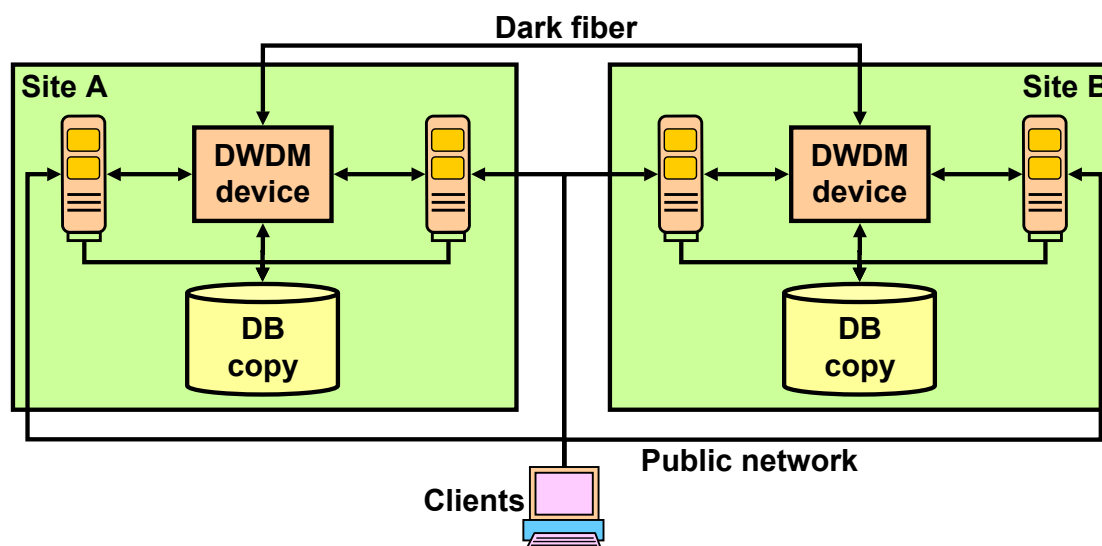
With extended RAC, you can use disk mirroring and Dense Wavelength Division Multiplexing (DWDM) equipment to extend the reach of the cluster. This configuration allows two data centers, separated by up to 100 kilometers, to share the same RAC database with multiple RAC instances spread across the two sites.

As shown in the slide, this RAC topology is very interesting, because the clients' work gets distributed automatically across all nodes independently of their location, and in case one site goes down, the clients' work continues to be executed on the remaining site. The types of failures that extended RAC can cover are mainly failures of an entire data center due to a limited geographic disaster. Fire, flooding, and site power failure are just a few examples of limited geographic disasters that can result in the failure of an entire data center.

Note: Extended RAC does not use special software other than the normal RAC installation.

Extended RAC Connectivity

- Distances over ten kilometers require dark fiber.
- Set up buffer credits for large distances.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Extended RAC Connectivity

In order to extend a RAC cluster to another site separated from your data center by more than ten kilometers, it is required to use DWDM over dark fiber to get good performance results.

DWDM is a technology that uses multiple lasers, and transmits several wavelengths of light simultaneously over a single optical fiber. DWDM enables the existing infrastructure of a single fiber cable to be dramatically increased. DWDM systems can support more than 150 wavelengths, each carrying up to 10 Gbps. Such systems provide more than a terabit per second of data transmission on one optical strand that is thinner than a human hair.

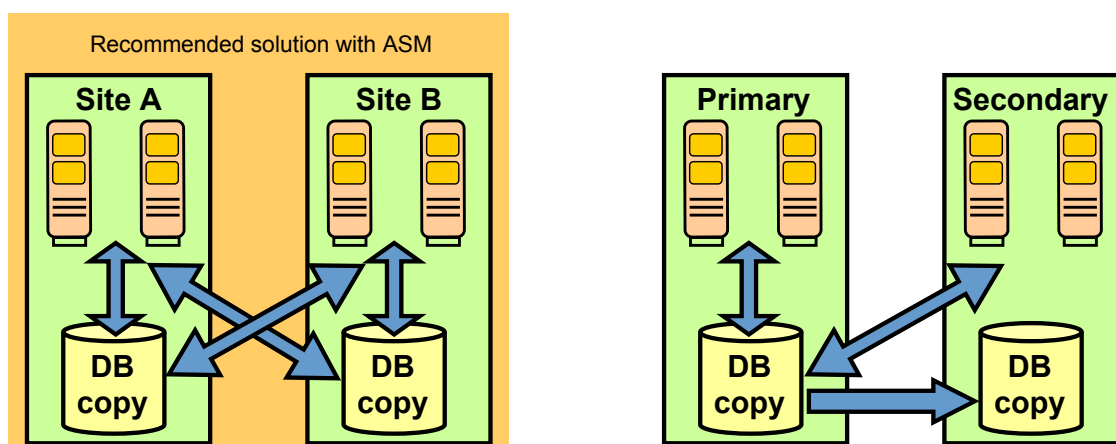
As shown in the slide, each site should have its own DWDM device connected together by a dark fiber optical strand. All traffic between the two sites is sent through the DWDM and carried on dark fiber. This includes mirrored disk writes, network and heartbeat traffic, and memory-to-memory data passage. Also shown on the graphic are the sets of disks at each site. Each site maintains a copy of the RAC database.

It is important to note that depending on the site's distance, you should tune and determine the minimum value of buffer credits in order to maintain the maximum link bandwidth. Buffer credit is a mechanism defined by the Fiber Channel standard that establishes the maximum amount of data that can be sent at any one time.

Note: Dark fiber is a single fiber optic cable or strand mainly sold by telecom providers.

Extended RAC Disk Mirroring

- Need copy of data at each location
- Two options:
 - Host-based mirroring
 - Remote array-based mirroring



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Extended RAC Disk Mirroring

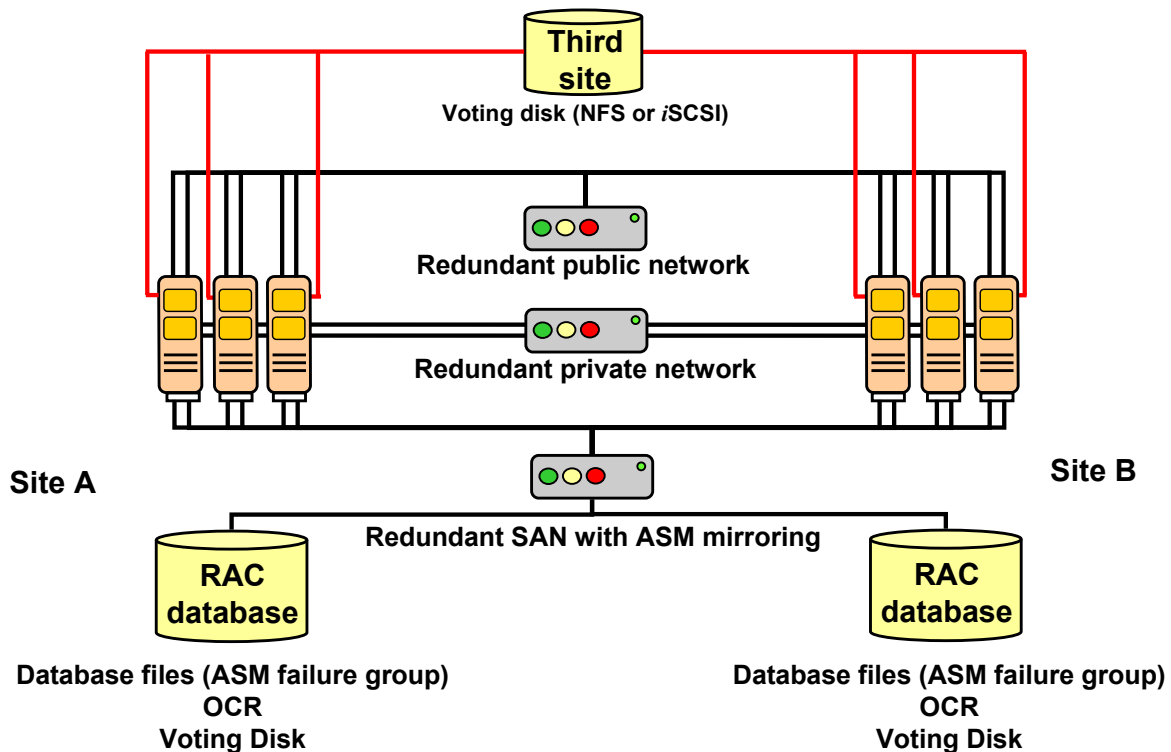
Although there is only one RAC database, each data center has its own set of storage that is synchronously mirrored using either a cluster-aware, host-based Logical Volume Manager (LVM) solution, such as SLVM with MirrorDiskUX, or an array-based mirroring solution, such as EMC SRDF.

With host-based mirroring, shown on the left of the slide, the disks appear as one set, and all I/Os get sent to both sets of disks. This solution requires closely integrated clusterware and LVM, and ASM is the recommended solution.

With array-based mirroring, shown on the right, all I/Os are sent to one site, and are then mirrored to the other. In fact, this solution is like a primary/secondary site setup. If the primary site fails, all access to primary disks is lost. An outage may be incurred before you can switch to the secondary site.

Note: With extended RAC, designing the cluster in a manner that ensures the cluster can achieve quorum after a site failure is a critical issue. For more information regarding this topic, refer to the *Oracle Technology Network* site.

Achieving Quorum with Extended RAC



Copyright © 2010, Oracle. All rights reserved.

Achieving Quorum with Extended RAC

As far as voting disks are concerned, a node must be able to access strictly more than half of the voting disks at any time, or that node will be evicted from the cluster. Extended clusters are generally implemented with only two storage systems, one at each site. This means that the site that houses the majority of the voting disks is a potential single point of failure for the entire cluster. To prevent this potential outage, Oracle Clusterware supports a third voting disk on an inexpensive, low-end, standard NFS-mounted device somewhere on the network. It is thus recommended to put this third NFS voting disk on a dedicated server visible from both sites. This situation is illustrated in the slide. The goal is that each site can run independently of the other when a site failure occurs.

Note: For more information about NFS configuration of the third voting disk, refer to the *Oracle Technology Network* site.

Additional Data Guard Benefits

- Greater disaster protection
 - Greater distance
 - Additional protection against corruptions
- Better for planned maintenance
 - Full rolling upgrades
- More performance neutral at large distances
 - Option to do asynchronous
- If you cannot handle the costs of a DWDM network, Data Guard still works over cheap, standard networks.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Additional Data Guard Benefits

Data Guard provides a greater disaster protection:

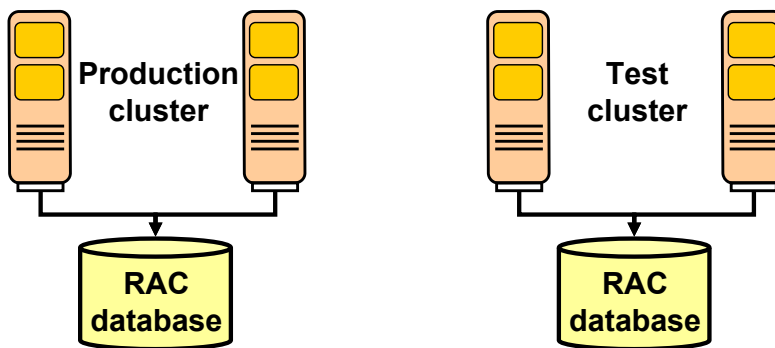
- Distance over 100 kilometers without performance hit
- Additional protection against corruptions because it uses a separate database
- Optional delay to protect against user errors

Data Guard also provides better planned maintenance capabilities by supporting full rolling upgrades.

Also, if you cannot handle the costs of a DWDM network, then Data Guard still works over cheap, standard networks.

Using a Test Environment

- The most common cause of down time is change.
- Test your changes on a separate test cluster before changing your production environment.



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Using a Test Environment

Change is the most likely cause of down time in a production environment. A proper test environment can catch more than 90 percent of the changes that could lead to a down time of the production environment, and is invaluable for quick test and resolution of issues in production.

When your production environment is RAC, your test environment should be a separate RAC cluster with all the identical software components and versions.

Without a test cluster, your production environment will not be highly available.

Note: Not using a test environment is one of the most common errors seen by Oracle Support Services.

Quiz

Which of the following statements regarding Disk I/O design are true?

1. Use external RAID protection when possible.
2. Use LUNs with the same performance characteristics.
3. Use LUNs with the same capacity.
4. Minimize the number of spindles in your disk group.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Answer: 1, 2, 3

Summary

In this lesson, you should have learned how to:

- Design a Maximum Availability Architecture in your environment
- Determine the best RAC and Data Guard topologies for your environment
- Configure the Data Guard Broker configuration files in a RAC environment
- Identify successful disk I/O strategies

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Appendix A

Practices and Solutions

Table of Contents

Practices for Lesson 11	3
Practice 11-1: Installing the Oracle Database Software	4
Practice 11-2: Creating a RAC Database.....	6
Practices for Lesson 12	8
Practice 12-1: Operating System and Password File Authenticated Connections.....	9
Practice 12-2: Oracle Database Authenticated Connections	12
Practice 12-3: Stopping a Complete ORACLE_HOME Component Stack	16
Practices for Lesson 13	19
Practice 13-1: Configuring the Archive Log Mode	20
Practice 13-2: Configuring Specific Instance Connection Strings	23
Practice 13-3: Configuring RMAN and Performing Parallel Backups.....	27
Practices for Lesson 14	33
Practice 14-1: ADDM and RAC Part I	34
Practice 14-2: ADDM and RAC Part II.....	41
Practice 14-3: ADDM and RAC Part III.....	45
Practices for Lesson 15	48
Practice 15-1: Working with Services	49
Practice 15-2: Monitoring Services	53
Practice 15-3: Services and Alert Thresholds.....	56

Practices for Lesson 11

In this practice, you will install the Oracle Database 11g Release 2 software and create a three-node cluster database.

You must have Oracle Database 11g Release 2 Grid Infrastructure installed on all three cluster nodes and the appropriate ASM disk groups available to support the cluster database. Before beginning this practice, the

`/home/oracle/solutions/catchup10/catchup.sh` script must execute successfully. Please refer to the introduction, Oracle 11g Release 2 RAC Administration Practice Notes before continuing.

Practice 11-1: Installing the Oracle Database Software

In this practice, you will install the Oracle Database 11g Release 2 software on three nodes.

- 1) Use the VNC session on ST_NODE1 on display :1. Click the VNC icon on your desktop, enter `Server ST_NODE1:1`, substituting the name of your first node for `ST_NODE1`. Enter the password `oracle`.
- 2) In the VNC session, confirm that you are connected as the `oracle` user. Change directory to the staged software location provided by your instructor and start the OUI by executing the `runInstaller` command from the `/staged_software_location/database/Disk1` directory.

```
$ id
uid=501(oracle) gid=502(oinstall)
groups=501(dba),502(oinstall),503(oper),505(asmdba)
$ cd /stage/database/Disk1

$ ./runInstaller
```

- a) On the Configure Security Updates page, deselect the “I wish to receive security updates” check box and click Next. **Note:** If this were a production machine or part of an important test environment, you might consider this option. A dialog box appears making sure that you want to remain uninformed about the updates. Click Yes to close the dialog box and continue.
- b) On the Select Installation Option page, select the “Install database software only” option and click Next.
- c) On the Node Selection page, select Real Application Clusters database installation. Select all three of your assigned hosts and click Next. If the ssh connectivity test fails, click the SSH Connectivity button. Enter the password `oracle` for the `oracle` user. Select the Reuse public and private keys check box and click the Setup button. After the setup completes, click Next to continue.
- d) On the Select Product Languages, promote all languages from the Available Languages window to the Selected Languages window on the right-hand side. Click Next to continue.
- e) On the Select Database Edition, select Enterprise Edition and click Next to continue.
- f) On the Specify Installation edition, the Oracle Base should be `/u01/app/oracle` and the Software Location should be `/u01/app/oracle/product/11.2.0/dbhome_1`. Do not install the database to a shared location. Click Next to continue.
- g) On the Privileged Operating System Groups, select `dba` as the Database Administrator Group and `oper` as the Database Operator Group. Click Next to continue.

Practice 11-1: Installing the Oracle Database Software (continued)

- h) When the prerequisites have successfully been checked on the Perform Prerequisites Check page, click Next to continue. If any checks fail, click the Fix and Check Again button. Run the scripts on the cluster nodes as root as directed, and then click Next.
- i) Check the information on the Summary page and click Finish.
- j) The Install Progress screen allows you to monitor the progression of the installation.
- k) When the files have been copied to all nodes, the Execute Configuration Scripts window is presented. Execute the `/u01/app/oracle/product/11.2.0/dbhome_1/root.sh` script on **all three nodes**.

```
# /u01/app/oracle/product/11.2.0/dbhome_1/root.sh

Running Oracle 11g root.sh script...

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /u01/app/oracle/product/11.2.0/dbhome_1

Enter the full pathname of the local bin directory:
[/usr/local/bin]:
The file "dbhome" already exists in /usr/local/bin.  Overwrite
it? (y/n)
[n]: n
The file "oraenv" already exists in /usr/local/bin.  Overwrite
it? (y/n)
[n]: n
The file "coraenv" already exists in /usr/local/bin.
Overwrite it? (y/n)
[n]: n

Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root.sh script.
Now product-specific root actions will be performed.
Finished product-specific root actions.
#
```

Run this script on the remaining two nodes before continuing.

- l) When you have run the `root.sh` scripts on all three nodes, click the OK button to close the Execute Configuration Scripts window.
- m) Click the Close button on the Finish page to complete the installation and exit the Installer.

Practice 11-2: Creating a RAC Database

In this practice, you will create a three-node RAC database.

- 1) From the oracle VNC session that you used to install the database software in the preceding practice, change directory to `/u01/app/oracle/product/11.2.0/dbhome_1/bin` and launch the Database Configuration Assistant by executing the `dbca` command.

```
$ id
uid=500 (oracle) gid=501 (oinstall)
groups=500 (dba) , 501 (oinstall) , 502 (oper) , 505 (asmdba)

$ cd /u01/app/oracle/product/11.2.0/dbhome_1/bin

$ ./dbca
```

- a) On the Welcome page, select Oracle Real Application Clusters database and click Next.
- b) On the Operations page, select “Create a Database” and click Next.
- c) On the Database Templates page, select the General Purpose or Transaction Processing template. Click Next to continue.
- d) On the Database Identification page, select Admin-Managed and enter **orcl** in the Global Database Name field. Click the Select All button to install the database to all three of your nodes and click Next.
- e) On the Management Options page, make sure that the Configure Enterprise Manager check box is selected and the “Configure Database Control for local management” option is selected. Click Next to continue.
- f) Select “Use the Same Administrative Password for All Accounts” on the Database Credentials page. Enter `oracle_4U` in the Password and Confirm Password fields and click Next.
- g) On Database File locations, you can specify your storage type and location for your database files. Select Automatic Storage Management (ASM) from the drop-down list for Storage Type. Select Use Oracle-Managed Files and make sure that the Database Area value is +DATA. Click Next to continue.
- h) When the ASM Credentials box appears, enter the ASM password that you used during the clusterware installation practice 2.2, step k. The password should be `oracle_4U`. Enter the correct password and click OK to close the box.
- i) On the Recovery Configuration page, enter `+FRA` in the Recovery Area field and accept the default for the Recovery Area Size. Click Next to continue.
- j) On the Database Content page, select the Sample Schemas check box and click Next.

Practice 11-2: Creating a RAC Database (continued)

- k) On the Memory tabbed page of the Initialization Parameters page, make sure that you enter 550 in the Memory Size (SGA and PGA) field. Then click the Character Sets tab, and select Use Unicode on the Database Character Set tabbed page. Click Next.
- l) Accept the default values on the Database Storage page and click Next to continue.
- m) Select Create Database on the Creation Options page and click Finish.
- n) On the Database Configuration Assistant: Summary page, click OK.
- o) You can monitor the database creation progress from the Database Configuration Assistant window.
- p) At the end of the installation, a dialog box with your database information including the Database Control URL is displayed. Click Exit. This will close the dialog box and end the Database Configuration Assistant.
- q) Open a browser and enter the Database Control URL displayed in the previous step. Verify that all instances are up. Verify that all cluster resources are up across all three nodes.

Practices for Lesson 12

In these practices, you will contrast operating system, password file authenticated connections, and Oracle database authenticated connections. You will also learn to stop a complete ORACLE_HOME component stack

Practice 12-1: Operating System and Password File Authenticated Connections

In this practice, you will make both operating system authenticated connections and password file authenticated connections to the database instance. You will also examine problems with the oraenv script.

- 1) Connect to your first node as the oracle user and set up your environment variables using the oraenv script.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
```

- 2) Identify all the database instance names that are currently executing on your machine using the Linux ps command. **Note:** All database instances have a mandatory background process named pmon, and the instance name will be part of the complete process name.

```
$ ps -ef | grep -i pmon
grid      4111      1  0 10:53 ?          00:00:00 asm_pmon_+ASM1
oracle    4987      1  0 10:56 ?          00:00:00 ora_pmon_orcl1
oracle    5299  5257  0 10:58 pts/0    00:00:00 grep -i pmon
```

- 3) Attempt to make a local connection to the orcl1 instance using SQL*Plus with the sysdba privilege. This is known as operating system authentication because a password is not needed. What happens when trying to connect to the instance?

```
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 10:59:04
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to an idle instance.

SQL> exit
Disconnected.
$
```

- 4) Attempt to connect to the instance using a network connection string @orcl with the sysdba privilege. This is known as password file authentication. Is the connection successful this time?

```
$ sqlplus sys@orcl as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 11:03:35
2009
```

Practice 12-1: Operating System and Password File Authenticated Connections (continued)

```
Copyright (c) 1982, 2009, Oracle. All rights reserved.

Enter password: oracle_4U << Password is not displayed

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> exit

$
```

- 5) Display the values of the environment variables (ORACLE_BASE, ORACLE_HOME, ORACLE_SID, PATH, and LD_LIBRARY_PATH) that were defined with the oraenv script in step 1.

```
$ env | grep ORA
ORACLE_SID=orcl
ORACLE_BASE=/u01/app/oracle
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1

$ env | grep PATH
LD_LIBRARY_PATH=/u01/app/oracle/product/11.2.0/dbhome_1/lib
PATH=/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/usr/NX/bin:/home/oracle/bin:/u01/app/oracle/product/11.2.0/dbhome_1/bin
```

- 6) Modify the ORACLE_SID environment variable to match the actual database instance name for the orcl database.

```
$ export ORACLE_SID=orcl1
```

- 7) Attempt the local connection with system authentication to the orcl1 instance using SQL*Plus with the sysdba privilege. This is the same command as in step 3.

```
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 11:01:48
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
```


Practice 12-1: Operating System and Password File Authenticated Connections (continued)

With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL>

- 8) Query the instance_name column of the v\$instance dynamic performance view to validate the instance that you connected with. Exit SQL*Plus when finished.

```
SQL> select instance_name from v$instance;
```

```
INSTANCE_NAME  
-----  
orcl1
```

```
SQL> exit
```

Practice 12-2: Oracle Database Authenticated Connections

In this practice, you will make multiple Oracle database authenticated connections to a database instance and notice the effects of load balanced connections.

- 1) From your first node, connected as the oracle user, validate the instance names on each host.

```
$ . /home/oracle/labs/st_env.sh
$ ssh $ST_NODE1 ps -ef | grep pmon
grid      4111      1  0 10:53 ?          00:00:00 asm_pmon_+ASM1
oracle    4987      1  0 10:56 ?          00:00:00 ora_pmon_orcl1
oracle    7932    7779  0 11:56 pts/0      00:00:00 grep pmon

$ ssh $ST_NODE2 ps -ef | grep pmon
grid      4104      1  0 10:53 ?          00:00:00 asm_pmon_+ASM2
oracle    4959      1  0 10:56 ?          00:00:00 ora_pmon_orcl2

$ ssh $ST_NODE3 ps -ef | grep pmon
grid      4096      1  0 10:53 ?          00:00:00 asm_pmon_+ASM3
oracle    4841      1  0 10:55 ?          00:00:00 ora_pmon_orcl3
```

- 2) Verify the current host name, and then set the environment variables using the oraenv script.

```
$ hostname
host01.example.com

$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
```

- 3) Connect to a database instance by using SQL*Plus with the system account. This is known as Oracle database authentication. After it is connected, query the instance_name column from the v\$instance dynamic performance view.
Note: Your instance names may vary from the ones displayed below.

```
$ sqlplus system@orcl

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 12:05:06
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Enter password: oracle_4U << Password is not displayed

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
```

Practice 12-2: Oracle Database Authenticated Connections (continued)

```
Data Mining and Real Application Testing options
```

```
SQL> select instance_name from v$instance;
```

```
INSTANCE_NAME  
-----  
orcl2
```

```
SQL>
```

- 4) Use the SQL*Plus host command to temporarily exit SQL*Plus and return to the operating system prompt. **Note:** SQL*Plus is still running when this is performed. Validate that you are still on your first node. Repeat step 3 from the operating system prompt to establish a second SQL*Plus session and database instance connection. What instance name did you connect to?

```
SQL> host
```

```
$ hostname
```

```
host01.example.com
```

```
$ sqlplus system@orcl
```

```
SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 12:13:02  
2009
```

```
Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

```
Enter password: oracle_4U << Password is not displayed
```

```
Connected to:
```

```
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -  
Production
```

```
With the Partitioning, Real Application Clusters, Automatic  
Storage Management, OLAP,  
Data Mining and Real Application Testing options
```

```
SQL> select instance_name from v$instance;
```

```
INSTANCE_NAME  
-----  
orcl1
```

```
SQL>
```

- 5) Use the SQL*Plus host command to temporarily exit SQL*Plus and return to the operating system prompt. **Note:** SQL*Plus is still running when this is performed. Validate that you are still on your first node. Repeat step 3 from the operating system prompt to establish a third SQL*Plus session and database instance connection. What instance name did you connect to?

Practice 12-2: Oracle Database Authenticated Connections (continued)

```
SQL> HOST
$ hostname
host01.example.com
[oracle@host01 ~]$ sqlplus system@orcl

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 12:15:52
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Enter password: oracle_4U << Password is not displayed

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> select instance_name from v$instance;

INSTANCE_NAME
-----
orcl3

SQL>
```

- 6) Exit the three SQL*Plus sessions that are currently executing on the first node.

```
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

$ exit
exit

SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

$ exit
exit
```

Practice 12-2: Oracle Database Authenticated Connections (continued)

```
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

$ exit << Optional. This will exit your terminal session.
```

Practice 12-3: Stopping a Complete ORACLE_HOME Component Stack

In this practice, you will use the `svrctl` utility to stop all resource components executing from a single home location.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$ . /home/oracle/labs/st env.sh
```

```
$ ssh $ST_NODE1 ps -ef | grep pmon
grid      4111      1  0 10:53 ?          00:00:00 asm_pmon_+ASM1
oracle    4987      1  0 10:56 ?          00:00:00 ora_pmon_orcl1
oracle    5321    5254  0 12:45 pts/0      00:00:00 grep pmon
```

>>>>>>>>>> or <<<<<<<<<<<<<<<<

```
$ srvctl status asm -a
ASM is running on host01,host02,host03
ASM is enabled.
```

```
$ /u01/app/11.2.0/grid/bin/crsctl status resource ora.asm
NAME=ora.asm
```

Practice 12-3: Stopping a Complete ORACLE_HOME Component Stack (continued)

```
TYPE=ora.asm.type
TARGET=ONLINE           , ONLINE           , ONLINE
STATE=ONLINE on host01, ONLINE on host02, ONLINE on host03
```

- 3) Display the syntax usage help for the `srvctl status home` command.

```
$ srvctl status home -help

Displays the current state of of all resources for the Oracle
home.

Usage: srvctl status home -o <oracle_home> -s <state_file> -n
<node_name>
    -o <oracle_home>          ORACLE_HOME path
    -s <state_file>          Specify a file path for the 'srvctl stop
home' command to store the state of the resources
    -n <node_name>            Node name
    -h                        Print usage
```

- 4) Use the `srvctl status home` command to check the state of all resources running from the `/u01/app/oracle/product/11.2.0/dbhome_1` home location. Create the required state file in the `/tmp` directory with the file name `host01_dbhome_state.dmp` for the first node only.

```
$ srvctl status home -o
/u01/app/oracle/product/11.2.0/dbhome_1 -s
/tmp/host01_dbhome_state.dmp -n $ST_NODE1

Database orcl is running on node host01
```

- 5) Display the syntax usage help for the `srvctl stop home` command.

```
$ srvctl stop home -help

Stops all Oracle clusterware resources that run from the
Oracle home.

Usage: srvctl stop home -o <oracle_home> -s <state_file> -n
<node_name> [-t <stop_options>] [-f]
    -o <oracle_home>          ORACLE_HOME path
    -s <state_file>          Specify a file path for the 'srvctl stop
home' command to store the state of the resources
    -n <node_name>            Node name
    -t <stop_options>          Stop options for the database.
Examples of shutdown options are normal, transactional,
immediate, or abort.
    -f                        Force stop
    -h                        Print usage
```

Practice 12-3: Stopping a Complete ORACLE_HOME Component Stack (continued)

- 6) Stop all resources executing in the /u01/app/oracle/product/11.2.0/dbhome_1 home using the state file created in step 4. Do not use the optional parameters identified by square brackets “[]” displayed in the syntax usage help.

```
$ srvctl stop home -o /u01/app/oracle/product/11.2.0/dbhome_1  
-s /tmp/host01_dbhome_state.dmp -n $ST_NODE1  
$
```

- 7) Check the status of the database instances on each node.

Note: There are several ways this step can be performed. Do not use the `srvctl status home` command with the same state file created above.

```
$ srvctl status database -d orcl -v  
Instance orcl1 is not running on node host01  
Instance orcl2 is running on node host02  
Instance orcl3 is running on node host03
```

- 8) Start all resources for the /u01/app/oracle/product/11.2.0/dbhome_1 home using the state file created by the stop command.

```
$ srvctl start home -o /u01/app/oracle/product/11.2.0/dbhome_1  
-s /tmp/host01_dbhome_state.dmp -n $ST_NODE1  
$
```

- 9) Check the status of the database instances on each node.

Note: There are several ways this step can be performed.

```
$ srvctl status database -d orcl -v  
Instance orcl1 is running on node host01  
Instance orcl2 is running on node host02  
Instance orcl3 is running on node host03
```


Practices for Lesson 13

In this practice, you will configure ARCHIVELOG mode for your RAC database, configure instance-specific connect strings for RMAN, and configure persistent RMAN settings.

Practice 13-1: Configuring the Archive Log Mode

In this practice, you will configure the archive log mode of a Real Applications Cluster database.

- 1) From the first node of your cluster, open a terminal session as the `oracle` user and set up the environment variables using the `oraenv` script for the database instance. Change the value of the `ORACLE_SID` variable to allow local system authenticated connections.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle

$ export ORACLE_SID=orcl1
```

- 2) Make a local connection using operating system authentication to the database instance and then use the `archive log list` SQL command to determine if the database is in archive log mode. Exit SQL*Plus when done.

```
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 15:48:58
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> archive log list
Database log mode                No Archive Mode
Automatic archival               Disabled
Archive destination              USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence       11
Current log sequence             12
SQL> exit
$
```

- 3) Stop the `orcl` database on each node of the cluster using the `srvctl stop database` command.

```
$ srvctl stop database -d orcl
```

- 4) Verify that the `orcl` database is not running on any node of the cluster using the `srvctl status database` command.

Practice 13-1: Configuring the Archive Log Mode (continued)

```
$ srvctl status database -d orcl -v
Instance orcl1 is not running on node host01
Instance orcl2 is not running on node host02
Instance orcl3 is not running on node host03
```

- 5) Make a local connection using operating system authentication to the database instance and then start up the database on the first node only with the mount option.

```
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 16:27:17
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to an idle instance.

SQL> startup mount
ORACLE instance started.

Total System Global Area  577511424 bytes
Fixed Size                  1338000 bytes
Variable Size              444597616 bytes
Database Buffers           125829120 bytes
Redo Buffers                5746688 bytes
Database mounted.
SQL>
```

- 6) Issue the alter database archivelog SQL command to change the archive mode of the database and then verify the results with the archive log list SQL command.

```
SQL> alter database archivelog;

Database altered.

SQL> archive log list
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     11
Next log sequence to archive   12
Current log sequence           12
```

- 7) Shut down the database instance with the immediate option and exit SQL*Plus. Use the srvctl utility to restart the database instances on all nodes of the cluster.

```
SQL> shutdown immediate
ORA-01109: database not open

Database dismounted.
```

Practice 13-1: Configuring the Archive Log Mode (continued)

```
ORACLE instance shut down.
```

```
SQL> exit
```

```
Disconnected from Oracle Database 11g Enterprise Edition  
Release 11.2.0.1.0 - Production  
With the Partitioning, Real Application Clusters, Automatic  
Storage Management, OLAP,  
Data Mining and Real Application Testing options
```

```
$ srvctl start database -d orcl
```

```
$
```

- 8) Verify that the orcl database is running on all the three nodes of your cluster by using the `srvctl status database` command.

```
$ srvctl status database -d orcl -v
```

```
Instance orcl1 is running on node host01
```

```
Instance orcl2 is running on node host02
```

```
Instance orcl3 is running on node host03
```

```
$
```

Practice 13-2: Configuring Specific Instance Connection Strings

In this practice, you will modify the `tnsnames.ora` file to disable connection load balancing and allow specific named instances to be used for connectivity.

- 1) Examine the `$ORACLE_HOME/network/admin/tnsnames.ora` file. There should be only one entry. This entry allows load balancing of connections as you observed in Practice 12-2.

```
$ cat
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/tnsnames
.ora

# tnsnames.ora Network Configuration File:
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/tnsnames
.ora
# Generated by Oracle configuration tools.

ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = cluster03-
scan.cluster03.example.com) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl.example.com)
    )
  )
```

- 2) Execute the `/home/oracle/labs/less_13/fix_tns.sh` script. This script will add three additional entries to the `tnsnames.ora` file that disable load balancing of connections by requiring a specific `INSTANCE_NAME` when used. Examine the changes made to the `tnsnames.ora` file.

```
$ /home/oracle/labs/less_13/fix_tns.sh

$ cat
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/tnsnames
.ora
# tnsnames.ora Network Configuration File:
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/tnsnames
.ora
# Generated by Oracle configuration tools.

ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = cluster03-
scan.cluster03.example.com) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )
```

Practice 13-2: Configuring Specific Instance Connection Strings (continued)

```
# Added by lab 13
orcl1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = host01) (PORT=1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = orcl)
      (INSTANCE_NAME = orcl1)
    )
  )
orcl2 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = host02) (PORT=1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = orcl)
      (INSTANCE_NAME = orcl2)
    )
  )
orcl3 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = host03) (PORT=1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = orcl)
      (INSTANCE_NAME = orcl3)
    )
  )
```

- 3) Using one of the three new entries in the `tnsnames.ora` file, connect to the system database account using SQL*Plus and verify the instance name to see that it matches the specific entry.

```
$ sqlplus system@orcl2

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 17:21:26
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Enter password: oracle_4U << Password is not displayed

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
```

Practice 13-2: Configuring Specific Instance Connection Strings (continued)

```
Data Mining and Real Application Testing options
```

```
SQL> select instance_name from v$instance;
```

```
INSTANCE_NAME  
-----  
orcl2
```

```
SQL>
```

- 4) Use the SQL*Plus host command to temporarily exit SQL*Plus and return to the operating system prompt. **Note:** SQL*Plus is still running when this is performed. Repeat step 3 from the operating system prompt to establish a second SQL*Plus session and database instance connection using the same connection string. Verify that the INSTANCE_NAME stays the same.

```
SQL> host
```

```
$ sqlplus system@orcl2
```

```
SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 17:26:30  
2009
```

```
Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

```
Enter password: oracle_4U << Password is not displayed
```

```
Connected to:
```

```
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -  
Production
```

```
With the Partitioning, Real Application Clusters, Automatic  
Storage Management, OLAP,  
Data Mining and Real Application Testing options
```

```
SQL> select instance_name from v$instance;
```

```
INSTANCE_NAME  
-----  
orcl2
```

- 5) Exit both SQL*Plus sessions.

```
SQL> exit
```

```
Disconnected from Oracle Database 11g Enterprise Edition  
Release 11.2.0.1.0 - Production  
With the Partitioning, Real Application Clusters, Automatic  
Storage Management, OLAP,  
Data Mining and Real Application Testing options
```

```
$ exit
```

Practice 13-2: Configuring Specific Instance Connection Strings (continued)

```
exit
```

```
SQL> exit
```

```
Disconnected from Oracle Database 11g Enterprise Edition
```

```
Release 11.2.0.1.0 - Production
```

```
With the Partitioning, Real Application Clusters, Automatic  
Storage Management, OLAP,
```

```
Data Mining and Real Application Testing options
```

```
$
```


Practice 13-3: Configuring RMAN and Performing Parallel Backups

In this practice, you will designate your first and second nodes of the cluster as nodes responsible for performing parallel backups of the database. The database will be backed up to the +FRA ASM disk group by default.

- 1) Using the recovery manager utility (RMAN), connect to the `orcl` database as the target database.

```
$ rman target /

Recovery Manager: Release 11.2.0.1.0 - Production on Tue Sep 8
17:31:34 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.

connected to target database: ORCL (DBID=1224399398)

RMAN>
```

- 2) Display all of the current RMAN settings.

```
RMAN> show all;

using target database control file instead of recovery catalog
RMAN configuration parameters for database with db_unique_name
ORCL are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP OFF; # default
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK
TO '%F'; # default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO
BACKUPSET; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; #
default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1;
# default
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE
'DEFAULT' OPTIMIZE FOR LOAD TRUE ; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'/u01/app/oracle/product/11.2.0/dbhome_1/dbs/snapcf_orcl1.f';
# default
```

- 3) Configure RMAN to automatically back up the control file and server parameter file each time any backup operation is performed.

Practice 13-3: Configuring RMAN and Performing Parallel Backups (continued)

```
RMAN> configure controlfile autobackup on;

new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored
```

- 4) Configure all backups done to disk to be done in parallel 2 degrees of parallelism.

```
RMAN> configure device type disk parallelism 2;

new RMAN configuration parameters:
CONFIGURE DEVICE TYPE DISK PARALLELISM 2 BACKUP TYPE TO
BACKUPSET;
new RMAN configuration parameters are successfully stored
```

- 5) Configure channel 1 and channel 2 to use the connect string 'sys/oracle_4U@orcl#' when performing a parallel backup to disk. Replace the pound sign (#) with 1 for channel 1 and 2 for channel 2, respectively. This will designate your first and second nodes as dedicated backup nodes for the cluster using the node specific connection strings created earlier. Without node specific connection strings, there would be no control over which nodes are being connected to in order to perform the backups.

```
RMAN> configure channel 1 device type disk
connect='sys/oracle_4U@orcl1';

new RMAN configuration parameters:
CONFIGURE CHANNEL 1 DEVICE TYPE DISK CONNECT '*';
new RMAN configuration parameters are successfully stored

RMAN> configure channel 2 device type disk
connect='sys/oracle_4U@orcl2';

new RMAN configuration parameters:
CONFIGURE CHANNEL 2 DEVICE TYPE DISK CONNECT '*';
new RMAN configuration parameters are successfully stored
RMAN>
```

- 6) Open a second terminal session as the oracle account and set up the environment variables for the orcl database. Navigate to the ~/labs/less_13 directory, invoke SQL*plus as the system user, and run the monitor_rman.sql script. Do not exit the first session with the RMAN prompt or this second session with the SQL prompt.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
```

Practice 13-3: Configuring RMAN and Performing Parallel Backups (continued)

```
$ cd /home/oracle/labs/less_13

$ sqlplus system@orcl

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 20:37:36
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Enter password: oracle_4U << Password is not displayed

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> @monitor_rman.sql

no rows selected
```

- 7) In the first session with the RMAN prompt, perform a full database backup with archive logs. The backup should happen only on the designated nodes (your first and second nodes) as the backup nodes. **Do not wait for this step to finish before proceeding to the next step!**

```
RMAN> backup database plus archivelog;

Starting backup at 08-SEP-09
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=31 instance=orcl1 device type=DISK
allocated channel: ORA_DISK_2
channel ORA_DISK_2: SID=55 instance=orcl2 device type=DISK
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=2 sequence=4 RECID=1 STAMP=697048583
input archived log thread=3 sequence=10 RECID=4
STAMP=697062223
channel ORA_DISK_1: starting piece 1 at 08-SEP-09
channel ORA_DISK_2: starting archived log backup set
channel ORA_DISK_2: specifying archived log(s) in backup set
input archived log thread=1 sequence=12 RECID=2
STAMP=697062220
input archived log thread=2 sequence=5 RECID=3 STAMP=697062220
channel ORA_DISK_2: starting piece 1 at 08-SEP-09
channel ORA_DISK_2: finished piece 1 at 08-SEP-09
```

Practice 13-3: Configuring RMAN and Performing Parallel Backups (continued)

```
piece
handle=+FRA/orcl/backupset/2009_09_08/annnf0_tag20090908t20234
7_0.268.697062231 tag=TAG20090908T202347 comment=NONE
channel ORA_DISK_2: backup set complete, elapsed time:
00:00:04
channel ORA_DISK_1: finished piece 1 at 08-SEP-09
piece
handle=+FRA/orcl/backupset/2009_09_08/annnf0_tag20090908t20234
7_0.267.697062229 tag=TAG20090908T202347 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:00:05
Finished backup at 08-SEP-09

Starting backup at 08-SEP-09
using channel ORA_DISK_1
using channel ORA_DISK_2
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001
name=+DATA/orcl/datafile/system.260.696618497
input datafile file number=00005
name=+DATA/orcl/datafile/example.264.696618709
input datafile file number=00004
name=+DATA/orcl/datafile/users.267.696618501
input datafile file number=00006
name=+DATA/orcl/datafile/undotbs2.259.696619015
channel ORA_DISK_1: starting piece 1 at 08-SEP-09
channel ORA_DISK_2: starting full datafile backup set
channel ORA_DISK_2: specifying datafile(s) in backup set
input datafile file number=00002
name=+DATA/orcl/datafile/sysaux.268.696618499
input datafile file number=00003
name=+DATA/orcl/datafile/undotbs1.263.696618499
input datafile file number=00007
name=+DATA/orcl/datafile/undotbs3.258.696619021
channel ORA_DISK_2: starting piece 1 at 08-SEP-09
channel ORA_DISK_2: finished piece 1 at 08-SEP-09
piece
handle=+FRA/orcl/backupset/2009_09_08/nnndf0_tag20090908t20235
4_0.270.697062239 tag=TAG20090908T202354 comment=NONE
channel ORA_DISK_2: backup set complete, elapsed time:
00:01:35
channel ORA_DISK_1: finished piece 1 at 08-SEP-09
piece
handle=+FRA/orcl/backupset/2009_09_08/nnndf0_tag20090908t20235
4_0.269.697062235 tag=TAG20090908T202354 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:01:47
Finished backup at 08-SEP-09
```

Practice 13-3: Configuring RMAN and Performing Parallel Backups (continued)

```
Starting backup at 08-SEP-09
current log archived
using channel ORA_DISK_1
using channel ORA_DISK_2
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=13 RECID=5
STAMP=697062345
input archived log thread=2 sequence=6 RECID=7 STAMP=697062345
channel ORA_DISK_1: starting piece 1 at 08-SEP-09
channel ORA_DISK_2: starting archived log backup set
channel ORA_DISK_2: specifying archived log(s) in backup set
input archived log thread=3 sequence=11 RECID=6
STAMP=697062345
channel ORA_DISK_2: starting piece 1 at 08-SEP-09
channel ORA_DISK_1: finished piece 1 at 08-SEP-09
piece
handle=+FRA/orcl/backupset/2009_09_08/annnf0_tag20090908t20254
6_0.274.697062347 tag=TAG20090908T202546 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:00:02
channel ORA_DISK_2: finished piece 1 at 08-SEP-09
piece
handle=+FRA/orcl/backupset/2009_09_08/annnf0_tag20090908t20254
6_0.275.697062347 tag=TAG20090908T202546 comment=NONE
channel ORA_DISK_2: backup set complete, elapsed time:
00:00:01
Finished backup at 08-SEP-09

Starting Control File and SPFILE Autobackup at 08-SEP-09
piece
handle=+FRA/orcl/autobackup/2009_09_08/s_697062349.276.6970623
53 comment=NONE
Finished Control File and SPFILE Autobackup at 08-SEP-09

RMAN>
```

- 8) While the backup is in progress, rerun the query on the second terminal window to monitor the RMAN backup session progress within the cluster. The backup should be done in parallel, with work distributed to both the backup nodes of the cluster. Enter the slash (/) symbol and press the Enter key to rerun the query. It may be necessary to do this multiple times until the output appears.

```
SQL> /

no rows selected

SQL> /
```

Practice 13-3: Configuring RMAN and Performing Parallel Backups (continued)

```
no rows selected
SQL> /

INST_ID  SID  SERIAL#  CONTEXT      SOFAR  TOTALWORK  %_COMPLETE
-----
         1   31      1913         1    13308    104960    12.68
         2   55       284         1    23934    101760    23.52

SQL> /

INST_ID  SID  SERIAL#  CONTEXT      SOFAR  TOTALWORK  %_COMPLETE
-----
         1   31      1913         1    44798    104960    42.68
         2   55       284         1    49278    101760    48.43

SQL> exit
```

- 9) Run the `/home/oracle/labs/less_13/cleanup_13.sh` script.

```
$ /home/oracle/labs/less_13/cleanup_13.sh
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
```

- 10) Exit all windows when finished.

Practices for Lesson 14

This practice is designed to show you how to discover performance problems in your RAC environment. In this practice, you identify performance issues by using Enterprise Manager, and fix issues in three different steps. At each step, you will generate the same workload to make sure that you are making progress in your resolution.

Practice 14-1: ADDM and RAC Part I

The goal of this lab is to show you how to manually discover performance issues by using the Enterprise Manager performance pages as well as ADDM. This first part generates a workload that uses a bad RAC application design.

Note that all the necessary scripts for this lab are located in the /home/oracle/labs/seq directory on your first cluster node.

- 1) Before you start this lab, make sure that you have the necessary TNS entries in the tnsnames.ora file located in your ORACLE_HOME for the orcl database. You can execute the following script to create those entries: add_tnsinstances.sh.

```
$ cd /home/oracle/labs/seq
$ ./add_tnsinstances.sh
# tnsnames.ora... Network Configuration File:
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/tnsnames
.ora...
# Generated by Oracle configuration tools.

ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = ...example.com) (PORT =
1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )

orcl1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = <host01>) (PORT =
1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = orcl)
      (INSTANCE_NAME = orcl1)
    )
  )

orcl2 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = <host02>) (PORT =
1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = orcl)
      (INSTANCE_NAME = orcl2)
```


Practice 14-1: ADDM and RAC Part I (continued)

```
    )
  )

orcl3 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = <host03>) (PORT =
1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = orcl)
      (INSTANCE_NAME = orcl3)
    )
  )

$
```

- 2) Execute the `setupseq1.sh` script to set up the necessary configuration for this lab.

```
$ ./setupseq1.sh

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

drop user jfv cascade
*
ERROR at line 1:
ORA-01918: user 'JFV' does not exist

drop tablespace seq including contents and datafiles
*
ERROR at line 1:
ORA-00959: tablespace 'SEQ' does not exist

Tablespace created.

User created.

Grant succeeded.

drop sequence s
*
ERROR at line 1:
```

Practice 14-1: ADDM and RAC Part I (continued)

```
ORA-02289: sequence does not exist

drop table s purge
      *
ERROR at line 1:
ORA-00942: table or view does not exist

drop table t purge
      *
ERROR at line 1:
ORA-00942: table or view does not exist

Table created.

Table created.

Index created.

1 row created.

Commit complete.

PL/SQL procedure successfully completed.

$
```

- 3) Using Database Control, and connected as the SYS user, navigate to the Performance page of your Cluster Database.
 - a) Click the Performance tab from the Cluster Database Home page.
 - b) On the Cluster Database Performance page, make sure that Real Time: 15 Seconds Refresh is selected from the View Data drop-down list.
- 4) Use PL/SQL to create a new AWR snapshot.

```
$ ./create_snapshot.sh

PL/SQL procedure successfully completed.

$
```

Practice 14-1: ADDM and RAC Part I (continued)

- 5) Execute the `startseq1.sh` script to generate a workload on all instances of your cluster. Do not wait; proceed with the next step.

```
$ ./startseq1.sh
$ old 7: insert into t values(v,'&1');
new 7: insert into t values(v,'orcl2');
old 7: insert into t values(v,'&1');
new 7: insert into t values(v,'orcl1');
old 7: insert into t values(v,'&1');
new 7: insert into t values(v,'orcl3');

... Do not wait after this point and go to the next step.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

$
```

- 6) Using Database Control, determine the list of blocking locks in your database.
- Still on the Performance page, click the Database Locks link in the Additional Monitoring Links section of the page.
 - On the Database Locks page, make sure that Blocking Locks is selected from the View drop-down list.
 - If you do not see any locks, refresh the page by clicking Refresh. Perform this until you see locks. When you see a session lock, you should also see that the other session is waiting for that same lock. By clicking Refresh several times, you must see that all sessions are alternatively waiting for the other to release the exclusive lock held on table S.
- 7) While the scripts are still executing, look at the Average Active Sessions graphic. Then, drill down to the Cluster wait class for the first node. What are your conclusions?
- By using the drill-down method of Enterprise Manager, you can quickly identify the top waiting SQL statements and the top waiting sessions on both instances. Here it appears that an UPDATE statement on table S is causing most of the waits for the Cluster wait class.
 - Click Cluster Database in the locator link at the top of the page to return to the Cluster Database Performance page.

Practice 14-1: ADDM and RAC Part I (continued)

- c) From there you can now see the Average Active Sessions graph. Make sure that the View Data field is set to Real Time:15 Seconds Refresh. After a few seconds, the graphic must clearly show that the Cluster and Application wait classes are causing most waits. Using the Throughput tabbed page graph underneath the Average Active Sessions graph, you should also notice that the transaction rate is about 250 per second.
 - d) In the Average Active Sessions graph, click the Cluster link on the right. This takes you to the Active Sessions By Instance: Cluster page.
 - e) On the Active Sessions By Instance: Cluster page, you will see that the number of active sessions is almost the same on all nodes. Click the first instance's link (instance number 1). This takes you to the Active Sessions Waiting: Cluster page for the corresponding instance.
 - f) On the Active Sessions Waiting: Cluster page, you can see the most important wait events causing most of the waits in the Cluster wait class on the first instance. In the Top SQL: Cluster section, click the SQL identifier that uses most of the resources. This takes you to the SQL Details page for the corresponding statement. You will see that the script running on the first instance is executing a SELECT/UPDATE statement on table S that causes most of the Cluster waits.
- 8) Using Database Control, look at the Cluster Cache Coherency page. What are your conclusions?
- a) On the Cluster Database Home page, click the Performance tab.
 - b) On the Performance page, click the Cluster Cache Coherency link in the Additional Monitoring Links section.
 - c) The Cluster Cache Coherency page clearly shows that there are lots of blocks transferred per second on the system. This represents more than 17% of the total logical reads. This is reflected in both the Global Cache Block Transfer Rate and the Global Cache Block Transfers and Physical Reads (vs. Logical Reads) graphics.
 - d) On the Cluster Cache Coherency page, you can also click Interconnects in the Additional Links section of the page to get more information about your private interconnect.
- 9) While the scripts are still executing, look at the Average Active Sessions graph on the Database Performance page. Then drill down to the Application wait class for the first instance. What are your conclusions?
- a) By using the drill-down method of Enterprise Manager, you can quickly identify the top waiting SQL statements and the top waiting sessions on both instances. Here it appears that a LOCK statement on table S is causing most of the waits for the Application wait class.
 - b) Go back to the Cluster Database Home page by clicking the Database tab located on the top right-end corner. On the Cluster Database Home page, click the Performance tab.

Practice 14-1: ADDM and RAC Part I (continued)

- c) On the Performance page, make sure that the View Data field is set to Real Time: 15 Seconds Refresh. After a few seconds, the graphic should clearly show that the Cluster and Application wait classes are causing most waits. You will also notice that the transaction rate is about 100 per second.
 - d) In the Average Active Sessions graph, click the Application link on the right. This takes you to the Active Sessions By Instance: Application page.
 - e) On the Active Sessions By Instance: Application page, you must see that the number of active sessions is almost the same on all nodes. Click the link for the first instance (number 1) on the Summary Chart graph. This takes you to the Active Sessions Waiting: Application page of the first instance.
 - f) On the Active Sessions Waiting: Application page, you can see the most important wait events causing most of the waits in the Application wait class on the first instance. In the Top SQL: Application section, click the SQL identifier that uses most of the resources. This takes you to the SQL Details page for the corresponding statement. You must see that the script running on the first instance is executing a LOCK statement on table S that causes most of the Application waits.
 - g) After a while, you can see that all scripts are executed by looking at the Average Active Sessions graph as well as the Database Throughput graphics again. You should see the number of transactions per second going down.
- 10) After the workload finishes, use PL/SQL to create a new AWR snapshot.

```
$ ./create_snapshot.sh

PL/SQL procedure successfully completed.

$
```

- 11) Using Database Control, review the latest ADDM run. What are your conclusions?
- a) On the Cluster Database Home page, click the Advisor Central link in the Related Links section.
 - b) On the Advisor Central page, make sure that the Advisory Type field is set to All Types, and that the Advisor Runs field is set to Last Run. Click Go.
 - c) In the Results table, select the latest ADDM run corresponding to Instance All. Then click View Result. This takes you to the Automatic Database Diagnostic Monitor (ADDM) page.
 - d) On the Automatic Database Diagnostic Monitor (ADDM) page, the ADDM Performance Analysis table shows you the consolidation of ADDM reports from all instances running in your cluster. This is your first entry point before drilling down to specific instances. From there, investigate the Top SQL Statements, Table Locks, and Global Cache Messaging findings.

Practice 14-1: ADDM and RAC Part I (continued)

- e) Click the Top SQL Statements finding, which affects all instances, revealing `LOCK TABLE S` and `UPDATE S` commands as a possible problem to investigate. Click the Back button to return to the ADDM report.
- f) Click the Table Locks finding, which affects all instances, revealing that you should investigate your application logic regarding the JFV.S object.
- g) Click the Global Cache Messaging finding revealing again the `UPDATE S` command as responsible for approximately 30% of Cluster waits during the analysis period.
- h) Back to the Automatic Database Diagnostic Monitor (ADDM) page, you now have the possibility to drill down to each instance using the links located in the Affected Instances table. Click the link corresponding to the most affected instance (although all should be equally affected).
- i) On the corresponding ADDM Database Diagnostic Monitor (ADDM) instance page, you should retrieve similar top findings you previously saw at the cluster level.

Practice 14-2: ADDM and RAC Part II

The goal of this lab is to show you how to manually discover performance issues by using the Enterprise Manager performance pages as well as ADDM. In this second part of the practice, you are going to correct the previously found issue by creating a sequence number instead of by using a table.

Note that all the necessary scripts for this lab are located in the /home/oracle/labs/seq directory on your first cluster node.

- 1) Execute the `setupseq2.sh` script to create the necessary objects used for the rest of this practice.

```
$ ./setupseq2.sh

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

User dropped.

Tablespace dropped.

Tablespace created.

User created.

Grant succeeded.

  drop table s purge
          *
ERROR at line 1:
ORA-00942: table or view does not exist

  drop sequence s
          *
ERROR at line 1:
ORA-02289: sequence does not exist

  drop table t purge
          *
ERROR at line 1:
ORA-00942: table or view does not exist
```

Practice 14-2: ADDM and RAC Part II (continued)

```
Table created.
```

```
Index created.
```

```
Sequence created.
```

```
PL/SQL procedure successfully completed.
```

```
$
```

- 2) Using Database Control, and connected as the SYS user, navigate to the Performance page of your Cluster Database.
 - a) Click the Performance tab from the Cluster Database Home page.
 - b) On the Cluster Database Performance page, make sure Real Time: 15 Seconds Refresh is selected from the View Data drop-down list.
- 3) Use PL/SQL to create a new AWR snapshot.

```
$ ./create_snapshot.sh
```

```
PL/SQL procedure successfully completed.
```

```
$
```

- 4) Execute the startseq2.sh script to generate a workload on all instances of your cluster. Do not wait; proceed with the next step.

```
$ ./startseq2.sh
```

```
$ old 3: insert into t values(s.nextval,'&1');
```

```
new 3: insert into t values(s.nextval,'orcl1');
```

```
old 3: insert into t values(s.nextval,'&1');
```

```
new 3: insert into t values(s.nextval,'orcl3');
```

```
old 3: insert into t values(s.nextval,'&1');
```

```
new 3: insert into t values(s.nextval,'orcl2');
```

```
... Do not wait after this point and go to the next step.
```

```
PL/SQL procedure successfully completed.
```

```
PL/SQL procedure successfully completed.
```

```
PL/SQL procedure successfully completed.
```


Practice 14-2: ADDM and RAC Part II (continued)

```
$
```

- 5) While the scripts are still executing, look at the Average Active Sessions graphic. Then drill down to the Cluster wait class for the first node. What are your conclusions?
 - a) By using the drill-down method of Enterprise Manager, you can quickly identify the top waiting SQL statements and the top waiting sessions on both instances. Here it appears that an INSERT statement on table T is causing most of the waits for the Cluster wait class.
 - b) Click Cluster Database in the locator link at the top of the page to return to the Cluster Database Performance page.
 - c) From there you can now see the Average Active Sessions graph. Make sure that the View Data field is set to Real Time:15 Seconds Refresh. After a few seconds, the graphic will clearly show that the Cluster and Application wait classes are causing most waits. Using the Throughput tabbed page graph underneath the Average Active Sessions graph, you should also notice that the transaction rate is about 320 per second (a better rate than in the previous practice).
 - d) In the Average Active Sessions graph, click the Cluster link on the right. This takes you to the Active Sessions By Instance: Cluster page.
 - e) On the Active Sessions By Instance: Cluster page, you must see that the number of active sessions is almost the same on all nodes. Click the first instance's link (instance number 1). This takes you to the Active Sessions Waiting: Cluster page for the corresponding instance.
 - f) On the Active Sessions Waiting: Cluster page, you can see the most important wait events causing most of the waits in the Cluster wait class on the first instance. In the Top SQL: Cluster section, click the SQL identifier that uses most of the resources. This takes you to the SQL Details page for the corresponding statement. You will see that the script running on the first instance is executing an INSERT statement on table T that causes most of the Cluster waits.
 - g) After a while you can see that all are executed by looking at the Average Active Sessions graphic again. The Database Throughput graphic tells you that this time, the number of transactions per second was a bit higher than in the previous lab for the same workload. Using the sequence number was a bit better in this case.
- 6) After the workload finishes, use PL/SQL to create a new AWR snapshot.

```
$ ./create_snapshot.sh  
  
PL/SQL procedure successfully completed.  
  
$
```

- 7) Using Database Control, review the latest ADDM run. What are your conclusions?
 - a) On the Cluster Database Home page, click the Advisor Central link.

Practice 14-2: ADDM and RAC Part II (continued)

- b) On the Advisor Central page, make sure that the Advisory Type field is set to All Types, and that the Advisor Runs field is set to Last Run. Click Go.
 - c) In the Results table, select the latest ADDM run corresponding to Instance All. Then click View Result. This takes you to the Automatic Database Diagnostic Monitor (ADDM) page.
 - d) On the Automatic Database Diagnostic Monitor (ADDM) page, the ADDM Performance Analysis table shows you the consolidation of ADDM reports from all instances running in your cluster. This is your first entry point before drilling down to specific instances. From there, investigate the Top SQL Statements, Sequence Usage, and Unusual “Concurrency” Wait Event findings.
 - e) The Top SQL Statements should reveal an `INSERT INTO T` command using sequence S as a possible problem to investigate.
 - f) The Sequence Usage finding reveals that you should use larger cache size for your hot sequences.
 - g) The Unusual “Concurrency” Wait Event finding asks you to investigate the cause for high “row cache lock” waits. Refer to the *Oracle Database Reference* for the description of this wait event.
 - h) Back to the Automatic Database Diagnostic Monitor (ADDM) page, you now have the possibility to drill down to each instance using the links located in the Affected Instances table. Click the link corresponding to the most affected instance (although all should be equally affected).
- 8) On the corresponding ADDM Database Diagnostic Monitor (ADDM) instance page, you should retrieve top findings similar to those you previously saw at the cluster level.

Practice 14-3: ADDM and RAC Part III

The goal of this lab is to show you how to manually discover performance issues by using the Enterprise Manager performance pages as well as ADDM. This last part generates the same workload as in the previous lab but uses more cache entries for sequence number S.

Note that all the necessary scripts for this lab are located in the /home/oracle/labs/seq directory on your first cluster node.

- 1) Execute the `setupseq3.sh` script to create the necessary objects used for the rest of this practice.

```
$ ./setupseq3.sh

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

User dropped.

Tablespace dropped.

Tablespace created.

User created.

Grant succeeded.

  drop table s purge
          *
ERROR at line 1:
ORA-00942: table or view does not exist

  drop sequence s
              *
ERROR at line 1:
ORA-02289: sequence does not exist

  drop table t purge
          *
ERROR at line 1:
ORA-00942: table or view does not exist
```

Practice 14-3: ADDM and RAC Part III (continued)

```
Table created.

Index created.

Sequence created.

PL/SQL procedure successfully completed.

$
```

- 2) Using Database Control, and connected as the SYS user, navigate to the Performance page of your Cluster Database.
 - a) Click the Performance tab from the Cluster Database Home page.
 - b) On the Cluster Database Performance page, make sure that Real Time: 15 Seconds Refresh is selected from the View Data drop-down list.
- 3) Use PL/SQL to create a new AWR snapshot.

```
$ ./create_snapshot.sh

PL/SQL procedure successfully completed.

$
```

- 4) Execute the startseq2.sh script to generate the same workload on both instances of your cluster as for the previous lab. Do not wait, and proceed with the next step.

```
$ ./startseq2.sh
$ old 3: insert into t values(s.nextval,'&1');
new 3: insert into t values(s.nextval,'orcl3');
old 3: insert into t values(s.nextval,'&1');
new 3: insert into t values(s.nextval,'orcl2');
old 3: insert into t values(s.nextval,'&1');
new 3: insert into t values(s.nextval,'orcl1');

... Do not wait after this point and go to the next step.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

$
```

Practice 14-3: ADDM and RAC Part III (continued)

- 5) Until the scripts are executed, look at the Sessions: Waiting and Working graphic. What are your conclusions?
 - a) This time, looking at the Sessions: Waiting and Working graphic, it is clear that there are no significant waits. The sequence has a big enough cache value to avoid the most significant waits.
 - b) Click Cluster Database in the locator link at the top of the page to return to the Cluster Database Performance page.
 - c) On the Performance page, make sure that the View Data field is set to Real Time:15 Seconds Refresh. After all the scripts have finished their execution, the Average Active Sessions graph will clearly show that there are no significant waits on your cluster. You must also notice that the transaction rate is now around 2400 per second.
- 6) After the workload finishes, use PL/SQL to create a new AWR snapshot.

```
$ ./create_snapshot.sh  
  
PL/SQL procedure successfully completed.  
  
$
```

- 7) Using Database Control, review the latest ADDM run. What are your conclusions?
 - a) On the Cluster Database Home page, click the Advisor Central link.
 - b) On the Advisor Central page, make sure that the Advisory Type field is set to All Types, and that the Advisor Runs field is set to Last Run. Click Go.
 - c) In the Results table, select the latest ADDM run corresponding to Instance All. Then click View Result. This takes you to the Automatic Database Diagnostic Monitor (ADDM) page.
 - d) On the Automatic Database Diagnostic Monitor (ADDM) page, the ADDM Performance Analysis table shows you the consolidation of ADDM reports from all instances running in your cluster. This is your first entry point before drilling down to specific instances. From there, investigate the Buffer Busy – Hot Block, Buffer Busy – Hot Objects, and Global Cache Busy findings. You should no longer see the Sequence Usage, nor specific instances impacted.
 - e) The Buffer Busy – Hot Block finding should not reveal any particular object.
 - f) The Buffer Busy – Hot Objects finding should not reveal any particular object.
 - g) The Global Cache Busy finding should not reveal anything special.

Practices for Lesson 15

In these practices, you will create, manage, and monitor services.

Practice 15-1: Working with Services

In this practice, you will use Enterprise Manager to create one service called PROD1. You then observe what happens to your service when you terminate one of the instances on which it is running.

- 1) Use Enterprise Manager to create the PROD1 service. Make sure that you define your first and second instances (ORCL1 and ORCL2) as preferred, and the third instance (ORCL3) as available.

- a) Enter your EM address in a browser. It will look something like this:

https://your_host_name:1158/em

- b) Log in using SYS credentials as SYSDBA.
 - c) Click the Availability folder tab.
 - d) Click the Cluster Managed Database Services link under the Services section.
 - e) On the Cluster Managed Database Services: Cluster and Database Login page, provide the login credentials for the operating system user (oracle/oracle) and the SYSDBA credentials for the database (sys/oracle_4U) and click Continue.
 - f) Click the Create Service button on the Cluster Managed Database Services page.
 - g) On the Create Service page, enter PROD1 for the service name. Verify that the “Start service after creation” check box is selected, and select the “Update local naming” check box. Under the High Availability Configuration section, set the service policy for orcl1 and orcl2 to Preferred and ORCL3 to Available. Leave the remaining fields with their default values and click the OK button.
 - h) After the service has been created, you will be returned to the Cluster Managed Database Services page. Check the Running Instances column for PROD1, it should indicate the service running on orcl1 and orcl2. Select PROD1 from the Services list and click the Test Connection button. It should test successfully. Click the Show All TNS Strings button and inspect the new entry to the tnsnames.ora file. It should look like this:

```
PROD1 = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)
(HOST = cluster01-scan.cluster01.example.com)
(PORT = 1521)) (LOAD_BALANCE = YES) (CONNECT_DATA =
(SERVER = DEDICATED) (SERVICE_NAME = PROD1)))
```

- i) Click the Return button.

- 2) Use the `srvctl` command to check the status of the new service.

```
$ srvctl status service -d ORCL -s PROD1
Service PROD1 is running on instance(s) orcl1,orcl2

$
```

- 3) Use the `crsctl` command to view server pool relationships with the new service.

Practice 15-1: Working with Services (continued)

```
$ /u01/app/11.2.0/grid/bin/crsctl status serverpool -p

NAME=Free
IMPORTANCE=0
MIN_SIZE=0
MAX_SIZE=-1
SERVER_NAMES=
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:grid:rw,pgroup:oinstall:rw,other::r-x

NAME=Generic
IMPORTANCE=0
MIN_SIZE=0
MAX_SIZE=-1
SERVER_NAMES=host01 host02 host03
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:grid:r-x,pgroup:oinstall:r-x,other::r-x

NAME=ora.orcl
IMPORTANCE=1
MIN_SIZE=0
MAX_SIZE=-1
SERVER_NAMES=host01 host02 host03
PARENT_POOLS=Generic
EXCLUSIVE_POOLS=
ACL=owner:oracle:rw,pgroup:oinstall:rw,other::r--

NAME=ora.orcl_PROD1
IMPORTANCE=0
MIN_SIZE=0
MAX_SIZE=-1
SERVER_NAMES=host01 host02 host03
PARENT_POOLS=ora.orcl
EXCLUSIVE_POOLS=
ACL=owner:oracle:rw,pgroup:oinstall:rw,other::r--

$
```

- 4) Connect to the service and look at the current value of the `SERVICE_NAMES` initialization parameter, and verify that it is set correctly. Query `V$INSTANCE` and determine what instance you are connected to.

```
$ /u01/app/oracle/product/11.2.0/dbhome_1/bin/sqlplus
sys/oracle_4U@PROD1 as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Fri Sep 4 11:10:29
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.
```


Practice 15-1: Working with Services (continued)

```
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options
```

```
SQL> show parameter service
```

NAME	TYPE	VALUE
service_names	string	PROD1

```
SQL> select instance_name from v$instance;
```

```
INSTANCE_NAME
-----
orcl1
```

```
SQL> exit
```

- 5) From a terminal session as the `oracle` user, crash the instance on the first node. Find and kill the `ora_pmon_orcl` process. Use the `pkill -9 -f pmon_orcl` command to crash the database instance. The `orcl1` instance will crash and the clusterware services will restart it very quickly

```
$ pkill -9 -f pmon_orcl1
```

- 6) Use the `srvctl status service -d ORCL -s PROD1` command to check the status of the PROD1 service.

```
$ srvctl status service -d ORCL -s PROD1
Service PROD1 is running on instance(s) orcl2,orcl3

$
```

- 7) Return to Enterprise Manager. Click the Availability folder tab. In the instance list under the Instances section, you should be able to verify that the first instance is indeed down.
- 8) Click the Cluster Managed Database Services link. On the Cluster Managed Database Service page, you can see `orcl2` and `orcl3` in the running instances column for PROD1. Select Manage from the Actions drop-down list and click Go.
- 9) Under the instances section, find the host that `orcl3` is running on and select the option in the Select column for that host. Click the Relocate button.

Practice 15-1: Working with Services (continued)

- 10) On the Relocate Service from Instance: orcl03 page, select the host name that orcl1 is running on and click OK.
- 11) You should see a message indicating that the service was relocated successfully. Under the Instances section of the page, you should see the service running on orcl1 and orcl2 and stopped on orcl3.
Note: The instance status shown in EM may still show that the instance is down. Click the browser Refresh button to see the actual status of the instance.

Practice 15-2: Monitoring Services

In this practice, you will use Database Control to determine the amount of resources used by sessions executing under a particular service.

- 1) As the oracle user, open a terminal session to your first node. Execute the `/home/oracle/labs/less_15/createuser.sh` script. This script creates a new user called FOO identified by the password `foo`. The default tablespace of this user is `USERS`, and its temporary tablespace is `TEMP`. This new user has the `CONNECT`, `RESOURCE`, and `DBA` roles.

```
$ cat /home/oracle/labs/less_15/createuser.sh

export ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
export ORACLE_SID=orcl1
/u01/app/oracle/product/11.2.0/dbhome_1/bin/sqlplus -s /NOLOG
<<EOF

connect / as sysdba
drop user FOO cascade;
create user FOO identified by foo default tablespace users
temporary tablespace temp;
grant connect, resource, dba to FOO;

EOF
$ /home/oracle/labs/less_15/createuser.sh
drop user FOO cascade
      *
ERROR at line 1:
ORA-01918: user 'FOO' does not exist

User created.

Grant succeeded.

$
```

- 2) Using `SQL*Plus`, connect to `PROD1` as `FOO`. When connected, determine the instance on which your session is currently running. Then execute the following query:
`select count(*) from dba_objects,dba_objects,dba_objects`
Do not wait; instead, proceed with the next step.

```
$ sqlplus foo/foo@PROD1
SQL> select instance_name from v$instance;

INSTANCE_NAME
-----
orcl1
```

Practice 15-2: Monitoring Services (continued)

```
SQL> select count(*) from dba_objects,dba_objects,dba_objects;
```

- 3) After a few moments, go to the Database Control Top Consumers page from the Cluster Database page. Connect as user SYS. Then check that PROD1 is using more and more resources.
 - a. From the Cluster Database Home page, click the Performance tab.
 - b. On the Performance page, click the Top Consumers link in the Additional Monitoring Links section.
 - c. This takes you to the Top Consumers page with the Overview tab selected.
 - d. On the Overview page, you can see the Top Services pie chart.
 - e. Make sure that the View Data drop-down list is set to Real Time: 15 Second Refresh. Wait for the page to be refreshed a couple of times. Little by little, PROD1 is consuming almost all the resources (up to 100%).
 - f. To have more details, click Top Services tab on the Top Consumers page.
 - g. Make sure that the View Data drop-down list is set to Real Time: 15 Second Refresh, and View drop-down list is set to Active Services. You can click the “+” icon on the left of the PROD1 link to expand the service. This shows you the list of instances currently running the service. You can also click the PROD1 link itself to look at the detailed Statistics of the corresponding service.
- 4) In another terminal window as the oracle user, check statistics on your service with `gv$sqlservice_stats` from a SQL*Plus session connected as SYSDBA.

```
$ /u01/app/oracle/product/11.2.0/dbhome_1/bin/sqlplus
sys/oracle_4U@orcl as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Fri Sep 4 16:16:48
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> select stat_name, sum(value) from gv$sqlservice_stats where
service_name = 'PROD1' group by stat_name;

STAT_NAME                                                    SUM(VALUE)
-----
user calls                                                    43
DB CPU                                                        1368469523
redo size                                                     1564
```

Practice 15-2: Monitoring Services (continued)

db block changes	8
DB time	1473281835
user rollbacks	0
gc cr blocks received	2
gc cr block receive time	0
gc current blocks received	2
opened cursors cumulative	99
workarea executions - multipass	0
STAT_NAME	SUM (VALUE)
-----	-----
session cursor cache hits	45
user I/O wait time	3540
parse count (total)	71
physical reads	4
gc current block receive time	0
workarea executions - optimal	22
concurrency wait time	17361
parse time elapsed	110704
physical writes	0
workarea executions - onepass	0
execute count	96
STAT_NAME	SUM (VALUE)
-----	-----
session logical reads	3825
cluster wait time	2161
application wait time	20622
logons cumulative	2
sql execute elapsed time	1473090354
user commits	0
28 rows selected.	
SQL>	

Practice 15-3: Services and Alert Thresholds

In this practice, you will set thresholds for service PROD1, and use Database Control to monitor the response time metric for this service. In this practice, you will set the Elapsed Time in seconds warning threshold at 4 and the critical threshold at 1. Preferred instances should be orcl1 and orcl2, and orcl3 should be available.

- 1) Set alert thresholds for your service PROD1 using Database Control.
 - a) Log in as `sys` with `SYSDBA` privileges.
 - b) On the Database Home page, click the Availability folder tab. Then click the Cluster Managed Service link.
 - c) Select PROD1 from the Services list, select Edit Properties from the Actions drop-down list and click Go.
 - d) Under the High Availability section, set the Service Policy for orcl1 to Preferred and Available for orcl2 and orcl3. Then click OK.
 - e) Return to the Cluster Database Home page, click the link corresponding to your first instance in the Instances table. This is the instance currently running PROD1.
 - f) On the Database Instance page, click Metric and Policy settings in the Related Links section at the bottom of the page.
 - g) On the Metric and Policy Settings page, select All metrics from the View drop-down list.
 - h) Scroll down the Metric and Policy Settings page until you find the Service Response Time (per user call) (microseconds) metric.
 - i) On the same line, click the corresponding multi-pens icon in the last column (Edit column).
 - j) On the Edit Advanced Settings: Service Response Time (per user call) (microseconds) page, click Add.
 - k) The Monitored Objects table should now show two entries.
 - l) Enter PROD1 in the Service Name field, 40000000 in the Warning Threshold field, and 100000000 in the Critical Threshold field. Make sure that the corresponding line is selected, and click Continue.
 - m) On the Metric and Policy Settings page, you should see an Information warning explaining that your settings have been modified but not saved. Click OK to save the new settings.
 - n) On the Confirmation page, you can see an Update succeeded message. Click OK.
 - o) This takes you back to the Database Instance page.
- 2) Use Database Control to view the Service Response Time Metric Value graphic for PROD1.

Practice 15-3: Services and Alert Thresholds (continued)

- a) From the Database Instance page, click All Metrics in the Related Links section at the bottom of the page.
 - b) On the All Metrics page, expand the Database Services link. On the All Metrics page, click the Service Response Time (per user call) (microseconds) link.
 - c) On the Service Response Time (per user call) (microseconds) page, click the PROD1 link in the Service Name column.
 - d) On the Service Response Time (per user call) (microseconds): Service Name PROD1: Last 24 hours page, select Real Time: 30 Second Refresh from the View Data drop-down list.
 - e) You should now see the Service Response Time (per user call) (microseconds): Service Name PROD1 page with your warning and critical thresholds set correctly.
- 3) Execute the `serv_wkload.sh` script to generate workload on your database. Looking at the Service Response time graphic for PROD1, what do you observe?

```
$ cd /home/oracle/labs/less_15
$ ./serv_wkload.sh
```

- a) Still looking at the Service Response Time (per user call) (microseconds): Service Name PROD1 page on your first session, you should see the graphic crossing the warning threshold after few minutes. This will trigger a warning alert soon after the warning threshold is crossed.
- b) You can see this alert propagated to your Database Instance Home page, and Cluster Database Home page.
- c) To go back to your Database Instance Home page, click the Database Instance locator link on the Service Response Time page.
- d) You should see the warning raised in the Alerts section of the Database Instance page.
- e) On the Database Instance page, click the Cluster Database locator link of the page.
- f) You should see the warning alert in the Problem Services line in the High Availability section of the page. Clicking this link takes you to the Cluster Home page. From there you can click the PROD1 link to directly go to the Cluster Managed Database Services: PROD1 page after you clicked Continue on the Login page. The PROD1 page shows you the alert with its details.
- g) Soon after the script finishes its execution, you should not see the corresponding alert on your Cluster Database Home page anymore. You can go to the Alert History page on the first instance to look at the alert history for your services. You can go to the Database Instance Home page using the locator links at the top of any pages. From the Database Instance Home page, scroll down to the bottom of the page, and click Alert History in the Related Links section.

Practice 15-3: Services and Alert Thresholds (continued)

- 4) Use Database Control to remove the thresholds that you specified during this practice.
 - a) From the Cluster Database Home page, click the link corresponding to the first instance of your cluster in the Instances section at the bottom of the page.
 - b) On the Database Instance page, scroll down to the bottom of the page. Click Metric and Policy Settings in the Related Links section.
 - c) On the Metric and Policy Settings page, scroll down the page until you see PROD1 in the Metric Thresholds table.
 - d) On the line corresponding to the PROD1 entry, remove both the Warning Threshold and Critical Threshold values.
 - e) Click OK.
 - f) On the Confirmation page, you should see an Update succeeded message. Click OK.

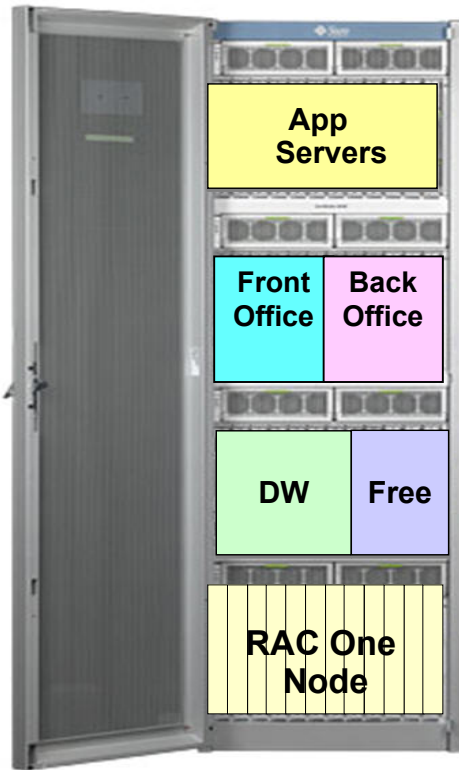
Oracle RAC One Node



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle RAC One Node



- Allows you to standardize Oracle database deployments across the enterprise
- Allows you to consolidate many databases into a single cluster with minimal overhead
- Supports live migration of instances across servers
- Simplifies rolling patches for single instance databases
- Employs built-in cluster failover for high availability
- Is supported on all platforms where Oracle Real Application Clusters is certified
- Complements virtual and physical servers

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle RAC One Node

Oracle RAC One Node allows you to standardize all Oracle Database deployments across the enterprise. Oracle RAC One Node is supported on all platforms where Oracle Real Application Clusters (RAC) is certified. Many databases can be consolidated into a single cluster with minimal overhead while providing the high availability benefits of failover protection, online rolling patch application, as well as rolling upgrades for the operating system and Oracle Clusterware.

Oracle Clusterware provides failover protection to Oracle RAC One Node. If the node fails, Oracle Clusterware will automatically restart the Oracle RAC One Node instance on another server in the cluster.

The Omotion Utility

- Omotion migrates an Oracle RAC One Node instance to another cluster node without application down time.
- Once the instance has been migrated, the node can be patched or upgraded.
- When the maintenance is complete, the Oracle RAC One Node instance can be migrated back to its original home.
- Omotion provides the same load balancing benefits of virtual machines (VMs) by allowing a migration of a database from a busy server to a server with spare capacity.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

The Omotion Utility

The Omotion utility is provided to migrate an Oracle RAC One Node instance to another node in the cluster without down time to the application. Once the instance has been migrated, the node can be patched or upgraded. Once the maintenance is complete, the Oracle RAC One Node can be migrated back to its original home.

The Omotion feature provides the same load balancing benefits of VMs by allowing a migration of a database from a busy server to a server with spare capacity. Omotion leverages the ability of Oracle Real Application Clusters to simultaneously run multiple instances servicing a single database. In the figure above, the DB2 RAC One Node database on Server A is migrated to Server B. Oracle RAC One Node starts up a second DB2 instance on server B, and for a short period of time runs in an active-active configuration. As connections complete their transactions on server A, they are migrated to the instance on server B. Once all the connections have migrated, the instance on server A is shut down and the migration is complete.

Omotion does not require quiescing the environment even when the system is running at peak capacity. VMs generally require the environment to be quiesced in order for medium to heavy database workloads to be moved from one server to another. This requirement does not apply for light workloads or demos.

Oracle RAC One Node and OVM

- Using Oracle RAC One Node with Oracle VM increases the scalability and high availability benefits of Oracle VM.
- If a VM is sized too small, you can migrate the Oracle RAC One instance to another Oracle VM node in your cluster and resize the Oracle VM.
- Usually, migrating VMs requires quiescing the source VM, mirroring the memory, and then switching to the migrated VM.
- Using Omotion, highly loaded instances can be migrated as the work is split between the two servers during the migration.
- Omotion provides the ability to move between servers of different processor generations.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Oracle RAC One Node and OVM

Oracle VM is a free server virtualization and management solution that makes enterprise applications easier to deploy, manage, and support. Using Oracle RAC One Node with Oracle VM increases the benefit of Oracle VM with the high availability and scalability of Oracle RAC. If your VM is sized too small, then you can migrate the Oracle RAC One instance to another Oracle VM node in your cluster using Omotion, and then resize the Oracle VM. When you move the instance back to the newly resized Oracle VM node, you can dynamically increase any limits programmed with Resource Manager Instance Caging.

When migrating a VM, the VM must mirror its complete memory state across a network to the target host, recreating the state of that machine. If the database in question is highly loaded and is actively changing blocks in its database cache, it is very difficult for the memory mirroring function to keep up with the rate of changes. It becomes likely that the only way to successfully mirror the memory is to quiesce the source VM, mirror the memory, and then switch to the migrated VM. With Omotion, highly loaded instances pose no problem, as the work is actually split between two servers during the migration. Oracle RAC One Node can, therefore, easily migrate even heavy workloads to another server. VM migration normally requires that the processors be identical. Both processors must have exactly the same instruction set. Omotion provides the ability to move between servers of different processor generations. Omotion supports migration to new processors, or even between different vendors like Intel and AMD.

High Availability of Connections

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Configure client-side, connect-time load balancing
- Configure client-side, connect-time failover
- Configure server-side, connect-time load balancing
- Use the Load Balancing Advisory (LBA)
- Describe the benefits of Fast Application Notification (FAN)
- Configure server-side callouts
- Configure Transparent Application Failover (TAF)

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Objectives

For more information see

<http://www.oracle.com/technology/products/database/clustering/pdf/awmrac11g.pdf>

Note: Much of this appendix is geared toward Oracle Database 11g Release 1 connections.

Types of Workload Distribution

- Connection balancing is rendered possible by configuring multiple listeners on multiple nodes:
 - Client-side, connect-time load balancing
 - Client-side, connect-time failover
 - Server-side, connect-time load balancing
- Run-time connection load balancing is rendered possible by using connection pools:
 - Work requests automatically balanced across the pool of connections
 - Native feature of the JDBC implicit connection cache and ODP.NET connection pool

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Types of Workload Distribution

With RAC, multiple listeners on multiple nodes can be configured to handle client connection requests for the same database service.

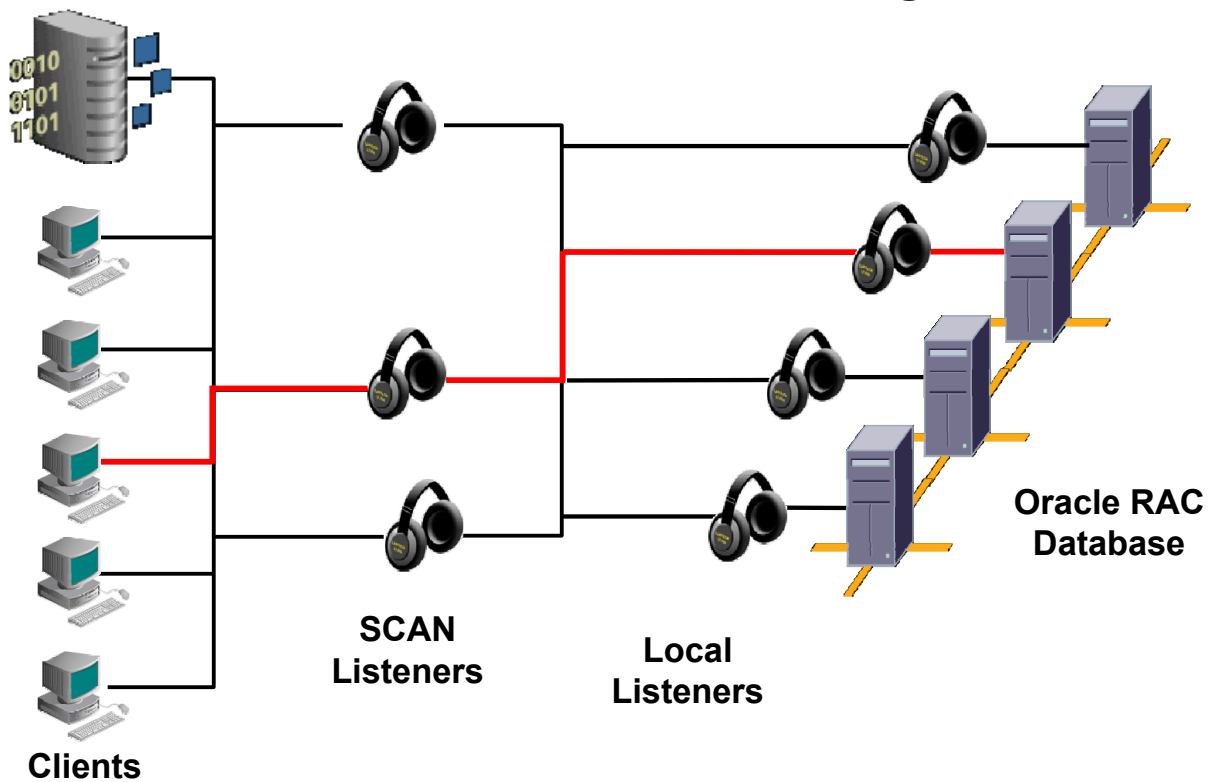
A multiple-listener configuration enables you to leverage the following failover and load-balancing features:

- Client-side, connect-time load balancing
- Client-side, connect-time failover
- Server-side, connect-time load balancing

These features can be implemented either one by one, or in combination with each other.

Moreover, if you are using connection pools, you can benefit from readily available run-time connection load balancing to distribute the client work requests across the pool of connections established by the middle tier. This possibility is offered by the Oracle JDBC implicit connection cache feature as well as Oracle Data Provider for .NET (ODP.NET) connection pool.

Client-Side Load Balancing



ORACLE

Copyright © 2010, Oracle. All rights reserved.

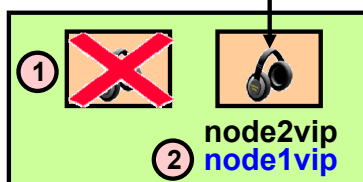
Client-Side, Connect-Time Load Balancing

Client-side load balancing is defined in your client connection definition by setting the parameter `LOAD_BALANCE=ON`. When you set this parameter to `ON`, Oracle Database randomly selects an address in the address list, and connects to that node's listener. This balances client connections across the available SCAN listeners in the cluster.

The SCAN listener redirects the connection request to the local listener of the instance that is least loaded and provides the requested service. When the listener receives the connection request, the listener connects the user to an instance that the listener knows provides the requested service. When using SCAN, Oracle Net automatically load balances client connection requests across the three IP addresses you defined for the SCAN, except when using EZConnect. To see what services a listener supports, run the `lsnrctl services` command.

Client-Side, Connect-Time Failover

```
ERP =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (LOAD_BALANCE=ON)  
      (FAILOVER=ON) ③  
      (ADDRESS= (PROTOCOL=TCP) (HOST=node1vip) (PORT=1521))  
      (ADDRESS= (PROTOCOL=TCP) (HOST=node2vip) (PORT=1521))  
    ) ④  
    (CONNECT_DATA= (SERVICE_NAME=ERP)) )
```



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Client-Side, Connect-Time Failover

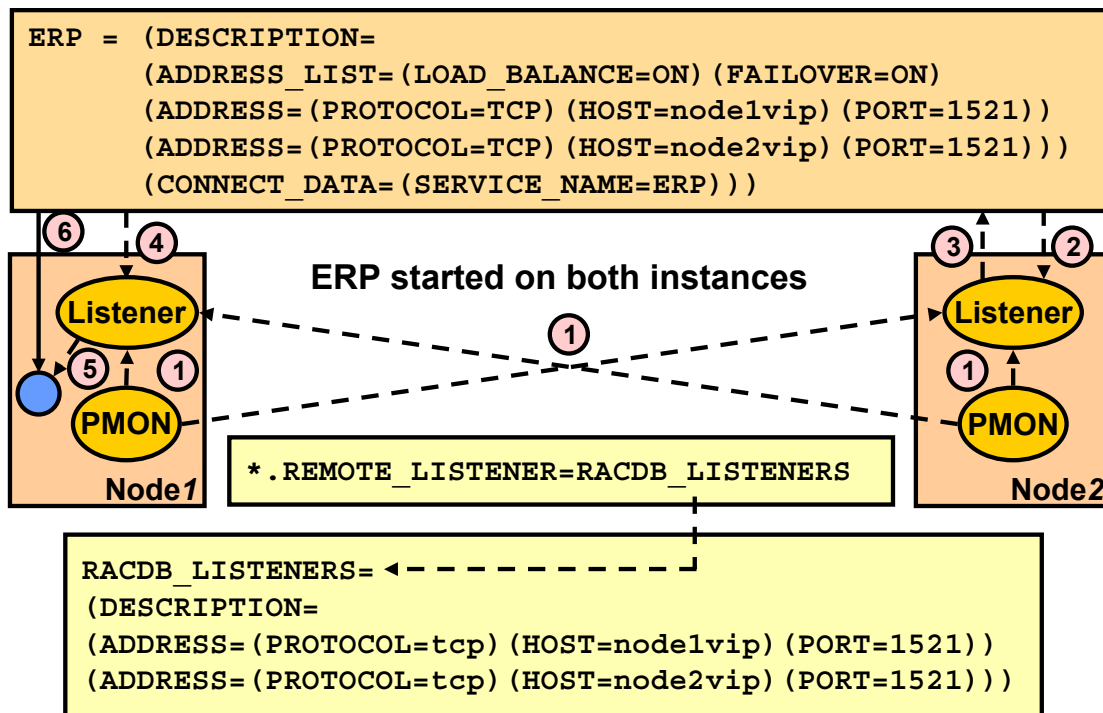
This feature enables clients to connect to another listener if the initial connection to the first listener fails. The number of listener protocol addresses in the connect descriptor determines how many listeners are tried. Without client-side, connect-time failover, Oracle Net attempts a connection with only one listener. As shown in the example in the slide, client-side, connect-time failover is enabled by setting the `FAILOVER=ON` clause in the corresponding client-side TNS entry.

In the example, you expect the client to randomly attempt connections to either `NODE1VIP` or `NODE2VIP`, because `LOAD_BALANCE` is set to `ON`. In the case where one of the nodes is down, the client cannot know this. If a connection attempt is made to a down node, the client needs to wait until it receives the notification that the node is not accessible, before an alternate address in the `ADDRESS_LIST` is tried.

Therefore, using virtual host names in the `ADDRESS_LIST` of your connect descriptors is highly recommended. If a failure of a node occurs (1), the virtual IP address assigned to that node is failed over and brought online on another node in the cluster (2). Thus, all client connection attempts are still able to get a response from the IP address, without the need to wait for the operating system TCP/IP timeout (3). Therefore, clients get an immediate acknowledgement from the IP address, and are notified that the service on that node is not available. The next address in the `ADDRESS_LIST` can then be tried immediately with no delay (4).

Note: If you use connect-time failover, do not set `GLOBAL_DBNAME` in your `listener.ora` file.

Server-Side, Connect-Time Load Balancing



Server-Side, Connect-Time Load Balancing

The slide shows you how listeners distribute service connection requests across a RAC cluster. Here, the client application connects to the ERP service. On the server side, the database is using the dynamic service registration feature. This allows the PMON process of each instance in the cluster to register service performance information with each listener in the cluster (1). Each listener is then aware of which instance has a particular service started, as well as how that service is performing on each instance.

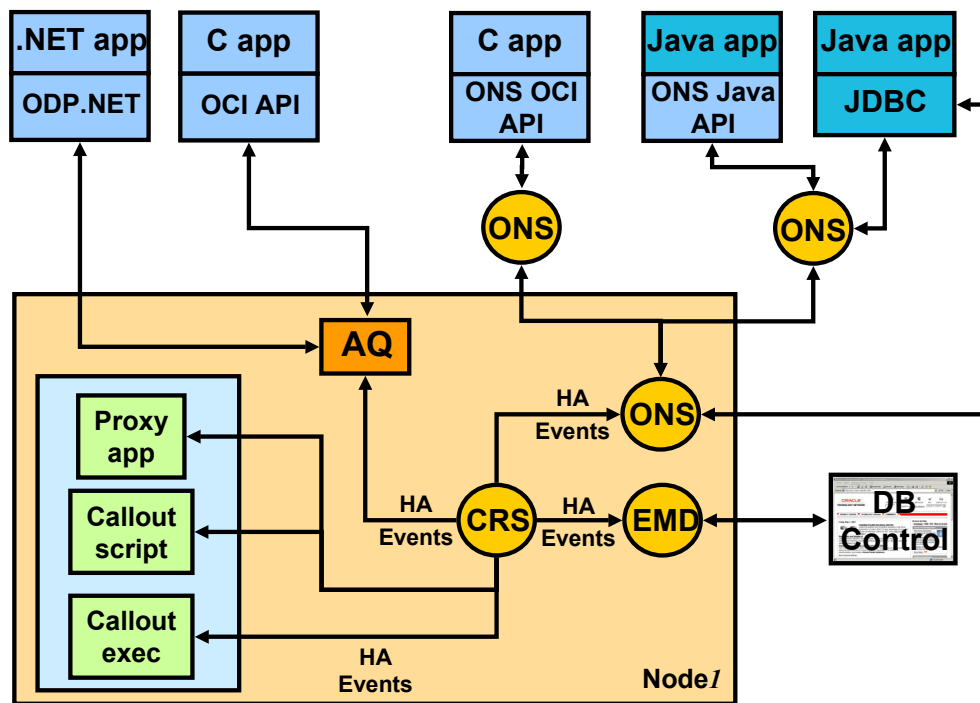
You configure this feature by setting the `REMOTE_LISTENER` initialization parameter of each instance to a TNS name that describes the list of all available listeners. The slide shows the shared entry in the `SPFILE` as well as its corresponding server-side TNS entry.

Depending on the load information, as computed by the Load Balancing Advisory, and sent by each PMON process, a listener redirects the incoming connection request (2) to the listener of the node where the corresponding service is performing the best (3).

In the example, the listener on `NODE2` is tried first. Based on workload information dynamically updated by PMON processes, the listener determines that the best instance is the one residing on `NODE1`. The listener redirects the connection request to the listener on `NODE1` (4). That listener then starts a dedicated server process (5), and the connection is made to that process (6).

Note: For more information, refer to the *Net Services Administrator's Guide*.

Fast Application Notification: Overview



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Fast Application Notification: Overview

Fast Application Notification (FAN) enables end-to-end, lights-out recovery of applications and load balancing based on real transaction performance in a RAC environment. With FAN, the continuous service built into Oracle Real Application Clusters is extended to applications and mid-tier servers. When the state of a database service changes, (for example, up, down, or not restarting), the new status is posted to interested subscribers through FAN events. Applications use these events to achieve very fast detection of failures, and rebalancing of connection pools following failures and recovery. The easiest way to receive all the benefits of FAN, with no effort, is to use a client that is integrated with FAN:

- Oracle Database JDBC
- Server-side callouts
- Connection Manager (CMAN)
- Listeners
- Oracle Notification Service (ONS) API
- OCI Connection Pool or Session Pool
- Transparent Application Failover (TAF)
- ODP.NET Connection Pool

Note: The integrated Oracle clients must be Oracle Database 10g Release 2 or later to take advantage of the load balancing advisory FAN events.

Fast Application Notification: Benefits

- No need for connections to rely on connection timeouts
- Used by Load Balancing Advisory to propagate load information
- Designed for enterprise application and management console integration
- Reliable distributed system that:
 - Detects high-availability event occurrences in a timely manner
 - Pushes notification directly to your applications
- Tightly integrated with:
 - Oracle JDBC applications using connection pools
 - Enterprise Manager
 - Data Guard Broker

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Fast Application Notification: Benefits

Traditionally, client or mid-tier applications connected to the database have relied on connection timeouts, out-of-band polling mechanisms, or other custom solutions to realize that a system component has failed. This approach has huge implications in application availability, because down times are extended and more noticeable.

With FAN, important high-availability events are pushed as soon as they are detected, which results in a more efficient use of existing computing resources, and a better integration with your enterprise applications, including mid-tier connection managers, or IT management consoles, including trouble ticket loggers and email/paging servers.

FAN is a distributed system that is enabled on each participating node. This makes it very reliable and fault tolerant because the failure of one component is detected by another. Therefore, event notification can be detected and pushed by any of the participating nodes.

FAN events are tightly integrated with Oracle Data Guard Broker, Oracle JDBC implicit connection cache, ODP.NET, TAF, and Enterprise Manager. For example, Oracle Database JDBC applications managing connection pools do not need custom code development. They are automatically integrated with the ONS if implicit connection cache and fast connection failover are enabled.

FAN-Supported Event Types

Event type	Description
SERVICE	Primary application service
SRV_PRECONNECT	Shadow application service event (mid-tiers and TAF using primary and secondary instances)
SERVICEMEMBER	Application service on a specific instance
DATABASE	Oracle database
INSTANCE	Oracle instance
ASM	Oracle ASM instance
NODE	Oracle cluster node
SERVICE_METRICS	Load Balancing Advisory

ORACLE

Copyright © 2010, Oracle. All rights reserved.

FAN-Supported Event Types

FAN delivers events pertaining to the list of managed cluster resources shown in the slide. The table describes each of the resources.

Note: SRV_PRECONNECT and SERVICE_METRICS are discussed later in this lesson.

FAN Event Status

Event status	Description
up	Managed resource comes up.
down	Managed resource goes down.
preconn_up	Shadow application service comes up.
preconn_down	Shadow application service goes down.
nodedown	Managed node goes down.
not_restarting	Managed resource cannot fail over to a remote node.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

FAN Event Status

This table describes the event status for each of the managed cluster resources seen previously.

FAN Event Reasons

Event Reason	Description
user	User-initiated commands, such as <code>srvctl</code> and <code>sqlplus</code>
failure	Managed resource polling checks for and detects a failure.
dependency	Dependency of another managed resource that triggered a failure condition
autostart	Initial cluster boot: Managed resource has profile attribute <code>AUTO_START=1</code> , and was offline before the last Oracle Clusterware shutdown.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

FAN Event Reasons

The event status for each managed resource is associated with an event reason. The reason further describes what triggered the event. The table in the slide gives you the list of possible reasons with a corresponding description.

FAN Event Format

```
<Event_Type>
VERSION=<n.n>
[service=<serviceName.dbDomainName>]
[database=<dbName>] [instance=<sid>]
[host=<hostname>]
status=<Event_Status>
reason=<Event_Reason>
[card=<n>]
timestamp=<eventDate> <eventTime>
```

```
SERVICE VERSION=1.0 service=ERP.oracle.com
database=ORCL status=up reason=user card=4
timestamp=21-Jul-2009 15:24:19
```

```
NODE VERSION=1.0 host=host01
status=nodedown timestamp=21-Jul-2009 12:41:02
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

FAN Event Format

In addition to its type, status, and reason, a FAN event has other payload fields to further describe the unique cluster resource whose status is being monitored and published:

- The event payload version, which is currently 1.0
- The name of the primary or shadow application service. This name is excluded from NODE events.
- The name of the RAC database, which is also excluded from NODE events
- The name of the RAC instance, which is excluded from SERVICE, DATABASE, and NODE events
- The name of the cluster host machine, which is excluded from SERVICE and DATABASE events
- The service cardinality, which is excluded from all events except for SERVICE status=up events
- The server-side date and time when the event is detected

The general FAN event format is described in the slide along with possible FAN event examples. Note the differences in event payload for each FAN event type.

Load Balancing Advisory: FAN Event

Parameter	Description
Version	Version of the event record
Event type	SERVICE, SERVICE_MEMBER, DATABASE, INSTANCE, NODE, ASM, SRV_PRECONNECT
Service	Matches the service in DBA_SERVICES
Database unique name	Unique DB name supporting the service
Time stamp	Date and time stamp (local time zone)
Instance	Instance name supporting the service
Percent	Percentage of work to send to this database and instance
Flag	GOOD, VIOLATING, NO DATA, BLOCKED

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Load Balancing Advisory: FAN Event

The Load Balancing Advisory FAN event is described in the slide. Basically, it contains a calculated percentage of work requests that should be sent to each instance. The flag indicates the behavior of the service on the corresponding instance relating to the thresholds set on that instance for the service.

Use the following example to monitor load balancing advisory events:

```
SET PAGES 60 COLSEP '|' LINES 132 NUM 8 VERIFY OFF FEEDBACK OFF
COLUMN user_data HEADING "AQ Service Metrics" FORMAT A60 WRAP
BREAK ON service_name SKIP 1
```

```
SELECT TO_CHAR(enq_time, 'HH:MI:SS') Enq_time, user_data
FROM sys.sys$service_metrics_tab
ORDER BY 1 ;
```

Implementation of Server-Side Callouts

- The callout directory:
 - `<GRID_Home>/racg/usrco`
 - Can store more than one callout
 - Grants execution on callouts and the callout directory to the Oracle Clusterware user
- The order in which callouts are executed is nondeterministic.
- Writing callouts involves:
 1. Parsing callout arguments: The event payload
 2. Filtering incoming FAN events
 3. Executing event-handling programs

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Implementation of Server-Side Callouts

Each database event detected by the RAC High Availability (HA) framework results in the execution of each executable script or program deployed in the standard Oracle Clusterware callout directory. On UNIX, it is `GRID_Home/racg/usrco`. You must deploy each new callout on each RAC node.

The order in which these callouts are executed is nondeterministic. However, RAC guarantees that all callouts are invoked once for each recognized event in an asynchronous fashion. Thus, merging callouts whose executions need to be in a particular order is recommended.

You can install as many callout scripts or programs as your business requires, provided each callout does not incur expensive operations that delay the propagation of HA events. If many callouts are going to be written to perform different operations based on the event received, it might be more efficient to write a single callout program that merges each single callout.

Writing server-side callouts involves the steps shown in the slide. In order for your callout to identify an event, it must parse the event payload sent by the RAC HA framework to your callout. After the sent event is identified, your callout can filter it to avoid execution on each event notification. Then, your callout needs to implement a corresponding event handler that depends on the event itself and the recovery process required by your business.

Note: As a security measure, make sure that the callout directory and its contained callouts have write permissions only to the system user who installed Oracle Clusterware.

Server-Side Callout Parse: Example

```
#!/bin/sh
NOTIFY_EVENTTYPE=$1
for ARGS in $*; do
    PROPERTY=`echo $ARGS | $AWK -F=" " '{print $1}'`
    VALUE=`echo $ARGS | $AWK -F=" " '{print $2}'`
    case $PROPERTY in
        VERSION|version)    NOTIFY_VERSION=$VALUE ;;
        SERVICE|service)    NOTIFY_SERVICE=$VALUE ;;
        DATABASE|database)  NOTIFY_DATABASE=$VALUE ;;
        INSTANCE|instance)  NOTIFY_INSTANCE=$VALUE ;;
        HOST|host)          NOTIFY_HOST=$VALUE ;;
        STATUS|status)      NOTIFY_STATUS=$VALUE ;;
        REASON|reason)      NOTIFY_REASON=$VALUE ;;
        CARD|card)          NOTIFY_CARDINALITY=$VALUE ;;
        TIMESTAMP|timestamp) NOTIFY_LOGDATE=$VALUE ;;
        ??:?:?:??)          NOTIFY_LOGTIME=$PROPERTY ;;
    esac
done
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Server-Side Callout Parse: Example

Unless you want your callouts to be executed on each event notification, you must first identify the event parameters that are passed automatically to your callout during its execution. The example in the slide shows you how to parse these arguments by using a sample Bourne shell script.

The first argument that is passed to your callout is the type of event that is detected. Then, depending on the event type, a set of PROPERTY=VALUE strings are passed to identify exactly the event itself.

The script given in the slide identifies the event type and each pair of PROPERTY=VALUE string. The data is then dispatched into a set of variables that can be used later in the callout for filtering purposes.

As mentioned in the previous slide, it might be better to have a single callout that parses the event payload, and then executes a function or another program on the basis of information in the event, as opposed to having to filter information in each callout. This becomes necessary only if many callouts are required.

Note: Make sure that executable permissions are set correctly on the callout script.

Server-Side Callout Filter: Example

```
if ((( [ $NOTIFY_EVENTTYPE = "SERVICE"      ] ||
        [ $NOTIFY_EVENTTYPE = "DATABASE"      ] ||
        [ $NOTIFY_EVENTTYPE = "NODE"          ]
    ) &&
    ( [ $NOTIFY_STATUS = "not_restarting" ]
    )) &&
    ( [ $NOTIFY_DATABASE = "PROD"          ] ||
      [ $NOTIFY_SERVICE = "ERP"           ]
    ))
then
    /usr/local/bin/logTicket $NOTIFY_LOGDATE \
                           $NOTIFY_LOGTIME \
                           $NOTIFY_SERVICE \
                           $NOTIFY_DBNAME \
                           $NOTIFY_HOST
fi
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Server-Side Callout Filter: Example

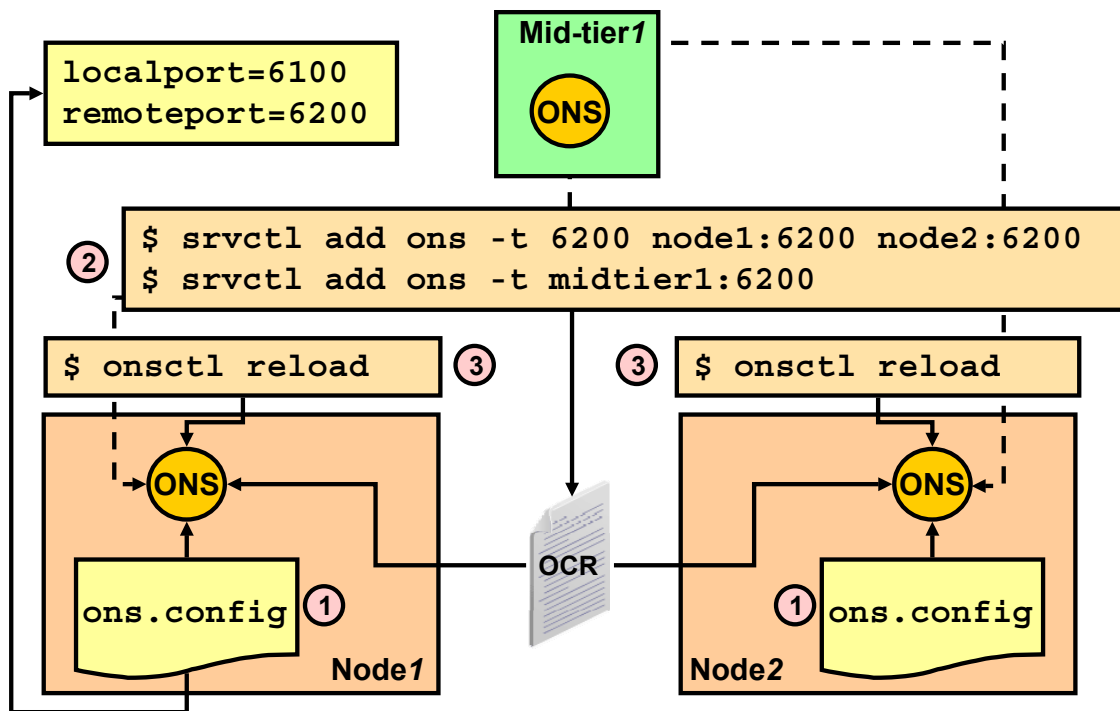
The example in the slide shows you a way to filter FAN events from a callout script. This example is based on the example in the previous slide.

Now that the event characteristics are identified, this script triggers the execution of the trouble-logging program `/usr/local/bin/logTicket` only when the RAC HA framework posts a SERVICE, DATABASE, or NODE event type, with a status set to `not_restarting`, and only for the production PROD RAC database or the ERP service.

It is assumed that the `logTicket` program is already created and that it takes the arguments shown in the slide.

It is also assumed that a ticket is logged only for `not_restarting` events, because they are the ones that exceeded internally monitored timeouts and seriously need human intervention for full resolution.

Configuring the Server-Side ONS



ORACLE

Copyright © 2010, Oracle. All rights reserved.

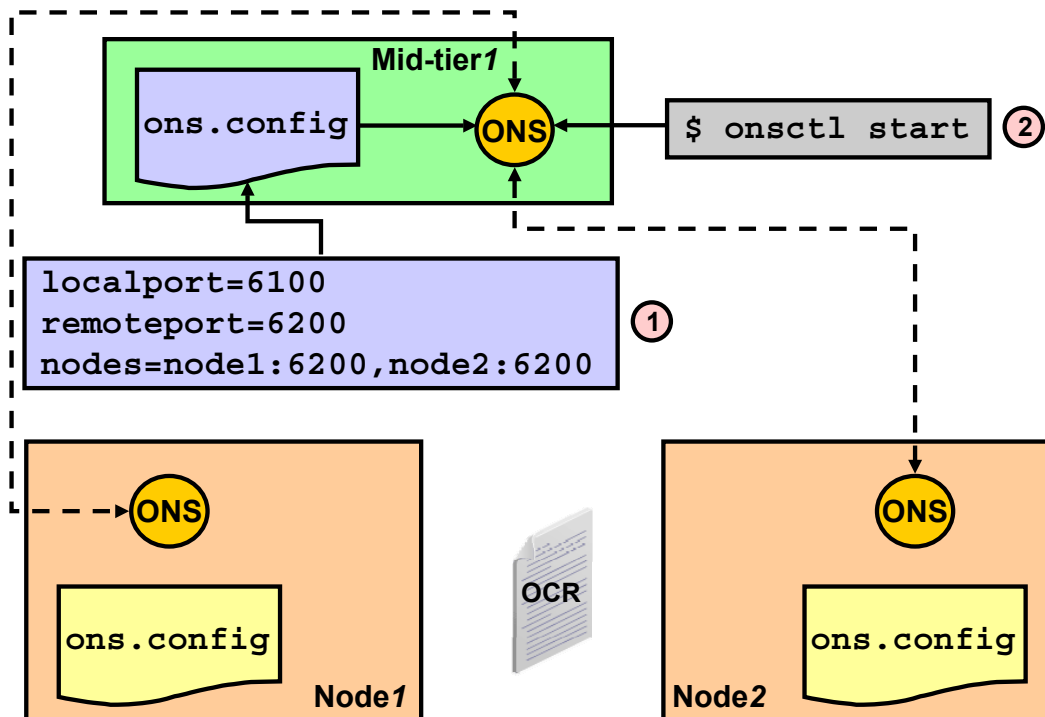
Configuring the Server-Side ONS

The ONS configuration is controlled by the `<GRID Home>/opmn/conf/ons.config` configuration file. This file is automatically created during installation. There are three important parameters that should always be configured for each ONS:

- The first is `localport`, the port that ONS uses to talk to local clients.
- The second is `remoteport`, the port that ONS uses to talk to other ONS daemons.
- The third parameter is called `nodes`. It specifies the list of other ONS daemons to talk to. This list should include all RAC ONS daemons, and all mid-tier ONS daemons. Node values are given as either host names or IP addresses followed by its `remoteport`.

In the slide, it is assumed that ONS daemons are already started on each cluster node. This should be the default situation after a correct RAC installation. However, if you want to use OCR, you should edit the `ons.config` file on each node, and then add the configuration to OCR before reloading it on each cluster node. This is illustrated in the slide.

Optionally Configure the Client-Side ONS



Copyright © 2010, Oracle. All rights reserved.

Optionally Configure the Client-Side ONS

Oracle Database FAN uses Oracle Notification Service (ONS) on the mid-tier to receive FAN events when you are using the Java Database Connectivity (JDBC) connection cache. To use ONS on the mid-tier, you need to install ONS on each host where you have client applications that need to be integrated with FAN. Most of the time, these hosts play the role of a mid-tier application server. Therefore, on the client side, you must configure all the RAC nodes in the ONS configuration file. A sample configuration file might look like the one shown in the slide.

After configuring ONS, you start the ONS daemon with the `onsctl start` command. It is your responsibility to make sure that an ONS daemon is running at all times. You can check that the ONS daemon is active by executing the `onsctl ping` command.

Oracle Internal & Oracle Academy Use Only



JDBC Fast Connection Failover: Overview

Therefore, if ICC and FCF are enabled, your Java program automatically becomes an ONS subscriber without having to manage FAN events directly.

Whenever a service up event is received by the mid-tier ONS, the event handler recycles some unused connections, and reconnects them using the event service name. The number of recycled connections is automatically determined by the connection cache. Because the listeners perform connection load balancing, this automatically rebalances the connections across the preferred instances of the service without waiting for application connection requests or retries.

Note: Similarly, ODP.NET also allows you to use FCF using AQ for FAN notifications.

Using Oracle Streams Advanced Queuing for FAN

- Use AQ to publish FAN to ODP.NET and OCI.
- Turn on FAN notification to alert queue.

```
exec DBMS_SERVICE.MODIFY_SERVICE (  
    service_name => 'SELF-SERVICE', aq_ha_notification => TRUE);
```

- View published FAN events:

```
SQL> select object_name,reason  
2 from dba_outstanding_alerts;
```

OBJECT_NAME	REASON
-------------	--------

xwkE	Database xwkE (domain) up as of time 2005-12-30 11:57:29.000000000 -05:00; reason code: user
JWSERV	Composite service xwkE up as of time 2006-01-02 05:27:46.000000000 -05:00; reason code: user

ORACLE

Copyright © 2010, Oracle. All rights reserved.

Using Oracle Streams Advanced Queuing for FAN

RAC publishes FAN events to a system alert queue in the database by using Oracle Streams Advanced Queuing (AQ). ODP.NET and OCI client integration uses this method to subscribe to FAN events.

To have FAN events for a service posted to that alert queue, the notification must be turned on for the service by using either the DBMS_SERVICE PL/SQL package as shown in the slide, or by using the Enterprise Manager interface.

To view FAN events that are published, you can use the DBA_OUTSTANDING_ALERTS or DBA_ALERT_HISTORY views. An example using DBA_OUTSTANDING_ALERTS is shown in the slide.

JDBC/ODP.NET FCF Benefits

- Database connections are balanced across preferred instances according to LBA.
- Database work requests are balanced across preferred instances according to LBA.
- Database connections are anticipated.
- Database connection failures are immediately detected and stopped.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

JDBC/ODP.NET FCF Benefits

By enabling FCF, your existing Java applications connecting through Oracle JDBC Universal Connection Pool (UCP) and application services, or your .NET applications using ODP.NET connection pools and application services benefit from the following:

- All database connections are balanced across all RAC instances that support the new service name, instead of having the first batch of sessions routed to the first RAC instance. This is done according to the Load Balancing Advisory algorithm you use (see the next slide). Connection pools are rebalanced upon service, instance, or node up events.
- The connection cache immediately starts placing connections to a particular RAC instance when a new service is started on that instance.
- The connection cache immediately shuts down stale connections to RAC instances where the service is stopped, or whose node goes down.
- Your application automatically becomes a FAN subscriber without having to manage FAN events directly by just setting up flags in your connection descriptors..

Note: For more information about how to subscribe to FAN events, refer to the *Oracle Database JDBC Developer's Guide* and *Oracle Data Provider for .NET Developer's Guide*.

Load Balancing Advisory

- The Load Balancing Advisory (LBA) is an advisory for sending work across RAC instances.
- LBA advice is available to all applications that send work:
 - JDBC and ODP connection pools
 - Connection load balancing
- There are two types of service-level goals:
 - **SERVICE_TIME**: Directs work requests to instances according to response time.

```
$ srvctl modify service -d PROD -s OE -B SERVICE_TIME -j SHORT
```

- **THROUGHPUT**: Directs requests based on the rate that work is completed in the service plus available bandwidth.

```
$ srvctl modify service -d PROD -s BATCH -B THROUGHPUT -j LONG
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

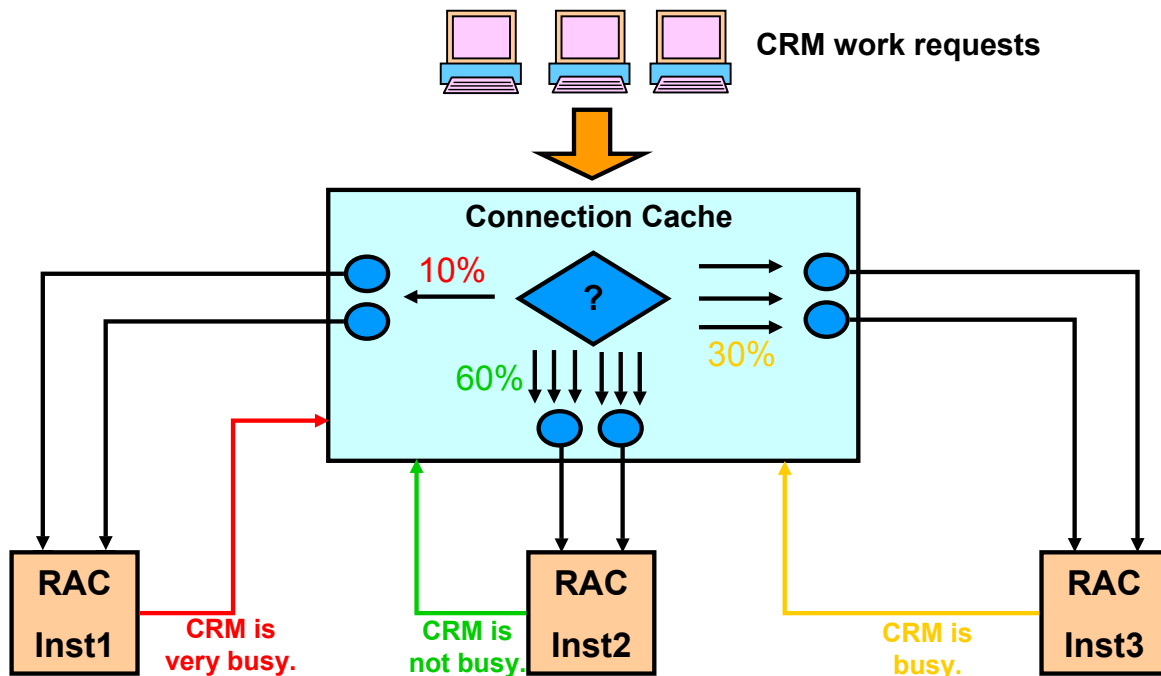
Load Balancing Advisory

Load balancing distributes work across all available database instances. The LBA provides advice about how to direct incoming work to the instances that provide the optimal quality of service for that work, minimizing the need to relocate the work later. By using the **SERVICE_TIME** or **THROUGHPUT** goals, feedback is built into the system.

- **SERVICE_TIME**: Attempts to direct work requests to instances according to response time. Load balancing advisory data is based on elapsed time for work done in the service plus available bandwidth to the service. An example for the use of **SERVICE_TIME** is for workloads such as Internet shopping where the rate of demand changes
- **THROUGHPUT**: Attempts to direct work requests according to throughput. The load balancing advisory is based on the rate that work is completed in the service plus available bandwidth to the service. An example for the use of **THROUGHPUT** is for workloads such as batch processes, where the next job starts when the last job completes:

Work is routed to provide the best service times globally, and routing responds gracefully to changing system conditions. In a steady state, the system approaches equilibrium with improved throughput across all of the Oracle RAC instances. The load balancing advisory is deployed with key Oracle clients, such as a listener, the JDBC universal connection pool, and the ODP.NET Connection Pool. The load balancing advisory is also open for third-party subscription by way of the JDBC and Oracle RAC FAN API or through callbacks with the Oracle Call Interface.

JDBC/ODP.NET Runtime Connection Load Balancing: Overview



Copyright © 2010, Oracle. All rights reserved.

JDBC/ODP.NET Runtime Connection Load Balancing: Overview

Without using the Load Balancing Advisory, work requests to RAC instances are assigned on a random basis, which is suitable when each instance is performing equally well. However, if one of the instances becomes more burdened than the others because of the amount of work resulting from each connection assignment, the random model does not perform optimally.

The Runtime Connection Load Balancing feature provides assignment of connections based on feedback from the instances in the RAC cluster. The Connection Cache assigns connections to clients on the basis of a relative number indicating what percentage of work requests each instance should handle.

In the diagram in the slide, the feedback indicates that the CRM service on Inst1 is so busy that it should service only 10% of the CRM work requests; Inst2 is so lightly loaded that it should service 60%; and Inst3 is somewhere in the middle, servicing 30% of requests. Note that these percentages apply to, and the decision is made on, a per-service basis. In this example, CRM is the service in question.

Monitor LBA FAN Events

```
SQL> SELECT TO_CHAR(enq_time, 'HH:MI:SS') Enq_time, user_data
2   FROM sys.sys$service_metrics_tab
3   ORDER BY 1 ;

ENQ_TIME USER_DATA
-----
...
04:19:46 SYS$RLBTYP('JFSERV', 'VERSION=1.0 database=xwke
        service=JFSERV { {instance=xwke2 percent=50
        flag=UNKNOWN}{instance=xwke1 percent=50 flag=UNKNOWN}
        } timestamp=2009-01-02 06:19:46')
04:20:16 SYS$RLBTYP('JFSERV', 'VERSION=1.0 database=xwke
        service=JFSERV { {instance=xwke2 percent=80
        flag=UNKNOWN}{instance=xwke1 percent=20 flag=UNKNOWN}
        } timestamp=2009-01-02 06:20:16')
SQL>
```

ORACLE

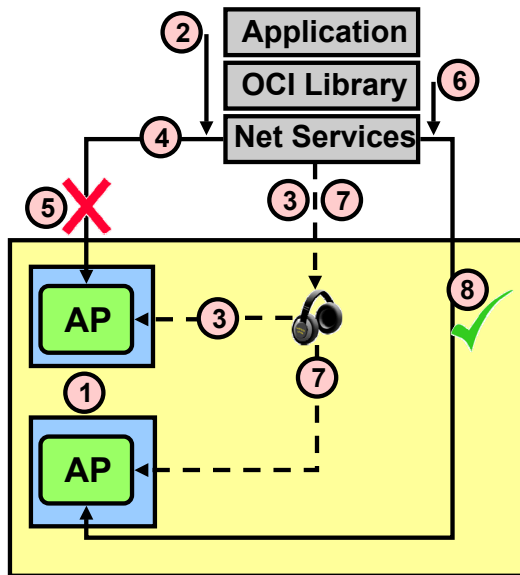
Copyright © 2010, Oracle. All rights reserved.

Monitor LBA FAN Events

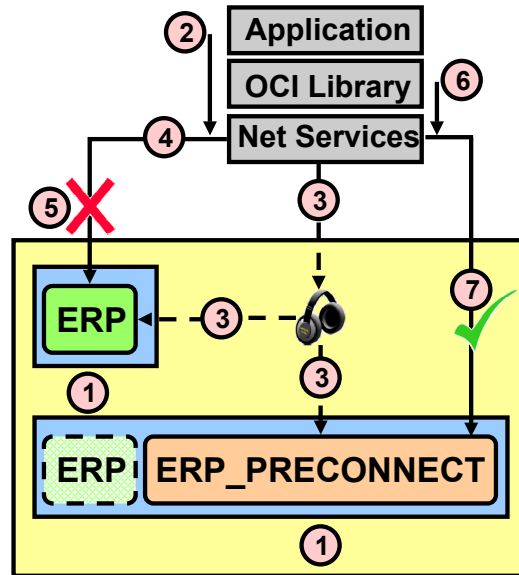
You can use the SQL query shown in the slide to monitor the Load Balancing Advisory FAN events for each of your services.

Transparent Application Failover: Overview

TAF Basic



TAF Preconnect



ORACLE

Copyright © 2010, Oracle. All rights reserved.

Transparent Application Failover (TAF): Overview

TAF is a run-time feature of the OCI driver. It enables your application to automatically reconnect to the service if the initial connection fails. During the reconnection, although your active transactions are rolled back, TAF can optionally resume the execution of a `SELECT` statement that was in progress. TAF supports two failover methods:

- With the **BASIC** method, the reconnection is established at failover time. After the service has been started on the nodes (1), the initial connection (2) is made. The listener establishes the connection (3), and your application accesses the database (4) until the connection fails (5) for any reason. Your application then receives an error the next time it tries to access the database (6). Then, the OCI driver reconnects to the same service (7), and the next time your application tries to access the database, it transparently uses the newly created connection (8). TAF can be enabled to receive FAN events for faster down events detection and failover.
- The **PRECONNECT** method is similar to the **BASIC** method except that it is during the initial connection that a shadow connection is also created to anticipate the failover. TAF guarantees that the shadow connection is always created on the available instances of your service by using an automatically created and maintained shadow service.

Note: Optionally, you can register TAF callbacks with the OCI layer. These callback functions are automatically invoked at failover detection and allow you to have some control of the failover process. For more information, refer to the *Oracle Call Interface Programmer's Guide*.

TAF Basic Configuration Without FAN: Example

```
$ srvctl add service -d RACDB -s AP -r I1,I2 \  
> -P BASIC  
$ srvctl start service -d RACDB -s AP
```

```
AP =  
(DESCRIPTION = (FAILOVER=ON) (LOAD_BALANCE=ON)  
  (ADDRESS= (PROTOCOL=TCP) (HOST=cluster01-scan) (PORT=1521))  
    (CONNECT_DATA =  
      (SERVICE_NAME = AP)  
      (FAILOVER_MODE =  
        (TYPE=SELECT)  
        (METHOD=BASIC)  
        (RETRIES=180)  
        (DELAY=5) ) ) )
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

TAF Basic Configuration Without FAN: Example

Before using TAF, it is recommended that you create and start a service that is used during connections. By doing so, you benefit from the integration of TAF and services. When you want to use BASIC TAF with a service, you should have the `-P BASIC` option when creating the service. After the service is created, you simply start it on your database.

Then, your application needs to connect to the service by using a connection descriptor similar to the one shown in the slide. The `FAILOVER_MODE` parameter must be included in the `CONNECT_DATA` section of your connection descriptor:

- `TYPE` specifies the type of failover. The `SELECT` value means that not only the user session is reauthenticated on the server side, but also the open cursors in the OCI can continue fetching. This implies that the client-side logic maintains the fetch-state of each open cursor. A `SELECT` statement is reexecuted by using the same snapshot, discarding those rows already fetched, and retrieving those rows that were not fetched initially. TAF verifies that the discarded rows are those that were returned initially, or it returns an error message.
- `METHOD=BASIC` is used to reconnect at failover time.
- `RETRIES` specifies the number of times to attempt to connect after a failover.
- `DELAY` specifies the amount of time in seconds to wait between connect attempts.

TAF Basic Configuration with FAN: Example

```
$ srvctl add service -d RACDB -s AP -r I1,I2
```

```
$ srvctl start service -d RACDB -s AP
```

```
srvctl modify service -d RACDB -s AP -q TRUE -P BASIC \  
-e SELECT -z 180 -w 5 -j LONG
```

```
AP =  
(DESCRIPTION =  
  (ADDRESS= (PROTOCOL=TCP) (HOST=cluster01-scan) (PORT=1521))  
  (LOAD_BALANCE = YES)  
  (CONNECT_DATA =  
    (SERVER = DEDICATED)  
    (SERVICE_NAME = TEST)  
    (FAILOVER_MODE =  
      (TYPE = SELECT) (METHOD = BASIC) (RETRIES = 180) (DELAY = 5))))
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

TAF Basic Configuration with FAN: Example

Oracle Database 11g Release 2 supports server-side TAF with FAN. To use server-side TAF, create and start your service using SRVCTL, and then configure TAF in the RDBMS by using the `srvctl` command as shown in the slide. When done, make sure that you define a TNS entry for it in your `tnsnames.ora` file. Note that this TNS name does not need to specify TAF parameters as in the previous slide.

TAF Preconnect Configuration: Example

```
ERP =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = cluster01-scan) (PORT = 1521))
    (LOAD_BALANCE = YES)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ERP)
      (FAILOVER_MODE =
        (BACKUP = ERP_PRECONNECT)
        (TYPE = SELECT) (METHOD = PRECONNECT) (RETRIES = 180) (DELAY = 5))))

ERP_PRECONNECT =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = cluster01-scan) (PORT = 1521))
    (LOAD_BALANCE = YES)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ERP_PRECONNECT)
      (FAILOVER_MODE =
        (TYPE = SELECT) (METHOD = BASIC) (RETRIES = 180) (DELAY = 5)))
```

ORACLE

Copyright © 2010, Oracle. All rights reserved.

TAF Preconnect Configuration: Example

In order for the shadow service to be created and managed automatically by Oracle Clusterware, you must define the service with the `-P PRECONNECT` option. The shadow service is always named using the format `<service_name>_PRECONNECT`.

The main differences with the previous example are that `METHOD` is set to `PRECONNECT` and an additional parameter is added. This parameter is called `BACKUP` and must be set to another entry in your `tnsnames.ora` file that points to the shadow service.

Note: In all cases where TAF cannot use the `PRECONNECT` method, TAF falls back to the `BASIC` method automatically.

TAF Verification

```
SELECT  machine, failover_method, failover_type,
        failed_over, service_name, COUNT(*)
FROM    v$session
GROUP BY machine, failover_method, failover_type,
        failed_over, service_name;
```

**First
node**

MACHINE	FAILOVER_M	FAILOVER_T	FAI	SERVICE_N	COUNT(*)
node1	BASIC	SESSION	NO	AP	1
node1	PRECONNECT	SESSION	NO	ERP	1

**Second
node**

MACHINE	FAILOVER_M	FAILOVER_T	FAI	SERVICE_N	COUNT(*)
node2	NONE	NONE	NO	ERP_PRECO	1

**Second
node
after**

MACHINE	FAILOVER_M	FAILOVER_T	FAI	SERVICE_N	COUNT(*)
node2	BASIC	SESSION	YES	AP	1
node2	PRECONNECT	SESSION	YES	ERP_PRECO	1

ORACLE

Copyright © 2010, Oracle. All rights reserved.

TAF Verification

To determine whether TAF is correctly configured and that connections are associated with a failover option, you can examine the V\$SESSION view. To obtain information about the connected clients and their TAF status, examine the FAILOVER_TYPE, FAILOVER_METHOD, FAILED_OVER, and SERVICE_NAME columns. The example includes one query that you could execute to verify that you have correctly configured TAF.

This example is based on the previously configured AP and ERP services, and their corresponding connection descriptors.

The first output in the slide is the result of the execution of the query on the first node after two SQL*Plus sessions from the first node have connected to the AP and ERP services, respectively. The output shows that the AP connection ended up on the first instance. Because of the load-balancing algorithm, it can end up on the second instance. Alternatively, the ERP connection must end up on the first instance because it is the only preferred one.

The second output is the result of the execution of the query on the second node before any connection failure. Note that there is currently one unused connection established under the ERP_PROCONNECT service that is automatically started on the available ERP instance.

The third output is the one corresponding to the execution of the query on the second node after the failure of the first instance. A second connection has been created automatically for the AP service connection, and the original ERP connection now uses the preconnected connection.

FAN Connection Pools and TAF Considerations

- Both techniques are integrated with services and provide service connection load balancing.
- Do not use FCF when working with TAF, and vice versa.
- Connection pools that use FAN are always preconnected.
- TAF may rely on operating system (OS) timeouts to detect failures.
- FAN never relies on OS timeouts to detect failures.

ORACLE

Copyright © 2010, Oracle. All rights reserved.

FAN Connection Pools and TAF Considerations

Because the connection load balancing is a listener functionality, both FCF and TAF automatically benefit from connection load balancing for services.

When you use FCF, there is no need to use TAF. Moreover, FCF and TAF cannot work together.

For example, you do not need to preconnect if you use FAN in conjunction with connection pools. The connection pool is always preconnected.

With both techniques, you automatically benefit from VIPs at connection time. This means that your application does not rely on lengthy operating system connection timeouts at connect time, or when issuing a SQL statement. However, when in the SQL stack, and the application is blocked on a read/write call, the application needs to be integrated with FAN in order to receive an interrupt if a node goes down. In a similar case, TAF may rely on OS timeouts to detect the failure. This takes much more time to fail over the connection than when using FAN.

Summary

In this lesson, you should have learned how to:

- Configure client-side, connect-time load balancing
- Configure client-side, connect-time failover
- Configure server-side, connect-time load balancing
- Use the Load Balancing Advisory
- Describe the benefits of Fast Application Notification
- Configure server-side callouts
- Configure Transparent Application Failover

ORACLE

Copyright © 2010, Oracle. All rights reserved.

