Oracle Database 11*g*: RAC Administration

Volume II • Student Guide

D50311GC11
Edition 1.1
September 2008
D56242

Author

James Womack

Technical Contributors and Reviewers

Christopher Andrews
Troy Anthony
Lothar Auert
Bruce Carter
Michael Cebulla
Carol Colrain
Jonathan Creighton
Joel Goodman
Arturo Gutierrez
Lutz Hartmann
Pete Jones
David Kirby
Roland Knapp
Miroslav Lorenc
Barb Lundhild
Roderick Manalac
Sabiha Miri
Philip Newlan
Roman Niehoff
Erik Peterson
Stefan Pommerenk
Marshall Presser
Rick Pulliam
Srinivas Putrevu
Roy Rossebo
Ira Singer
Linda Smalley
Ranbir Singh
Harald van Breederode
Michael Zoll

Editors

Raj Kumar
Richard Wallis
Nita Pavitran
Amitha Narayan

Graphic Designer

Satish Bettegowda

Publishers

Nita Brozowski
Veena Narasimhan

Contents

7 Services

11

Node Addition and Removal

Objectives

After completing this lesson, you should be able to:

- Add a new node to your cluster database
- Remove a node from your cluster database

Add and Delete Nodes and Instances: Overview

Three main methods:

- **Silent cloning procedures**
- **Enterprise Manager Grid Control cloning and adding instance**
- **`addNode.sh/rootdeletenode.sh` and DBCA**

Add and Delete Nodes and Instances: Overview

This lesson describes how to add and delete nodes and instances in Oracle Real Application Clusters (RAC) databases. There are mainly three methods you can use to add and delete nodes in a RAC environment:

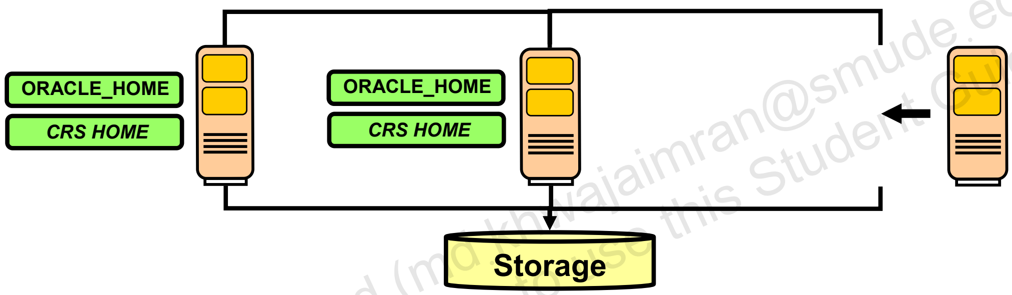
- Silent cloning procedures. Cloning enables you to copy images of Oracle Clusterware and RAC software onto the other nodes that have identical hardware and software.
- Enterprise Manager Grid Control. This is basically a GUI interface to cloning procedures.
- Interactive or silent procedures using `addNode.sh/rootdeletenode.sh` and the Database Configuration Assistant (DBCA)

The preferred method to add multiple nodes and instances to RAC databases is to use the cloning procedures. This is especially relevant when you are massively deploying software across your enterprise. Refer to the *Oracle Real Application Clusters Administration and Deployment Guide* as well as the *Oracle Clusterware Administration and Deployment Guide* for more information about cloning procedures.

However, in this lesson you are going to see how you can directly use Oracle Universal Installer (OUI) and DBCA to add one node to and delete one node from your cluster.

Main Steps to Add a Node to a RAC Cluster

1. Install and configure OS and hardware for new node.
2. Add Oracle Clusterware to the new node.
3. Configure ONS for the new node.
[4. Add ASM home to the new node.]
5. Add RAC home to the new node.
6. Add a listener to the new node.
7. Add a database instance to the new node.



Main Steps to Add a Node to a RAC Cluster

The slide lists the main steps you need to follow to add a new node to your RAC cluster. Basically, you are going to use OUI to copy the Oracle Clusterware software as well as the RAC software to the new node. For each main step, you have to do some manual configurations.

Note: For all the add node and delete node procedures for UNIX-based systems, temporary directories such as `/tmp`, `$TEMP`, or `$TMP` should not be shared directories. If your temporary directories are shared, set your temporary environment variable, such as `$TEMP`, to a nonshared location on a local node. In addition, use a directory that exists on all the nodes.

Check Prerequisites Before Oracle Clusterware Installation

```
Terminal
File  Edit  View  Terminal  Tabs  Help
bash-3.00$ /u01/crs11g/bin/cluvfy stage -pre crsinst -n vx0306,vx0313 -r 11gR1

Performing pre-checks for cluster services setup

Checking node reachability...
Node reachability check passed from node "vx0306".


Checking user equivalence...
User equivalence check passed for user "oracle".

Checking administrative privileges...
User existence check passed for "oracle".
Group existence check passed for "oinstall".
Membership check for user "oracle" in group "oinstall" [as Primary] passed.

Administrative privileges check passed.

Checking node connectivity...

Node connectivity check passed for subnet "10.216.4.0" with node(s) vx0306,vx0313.
```

ORACLE

Check Prerequisites Before Oracle Clusterware Installation

Before you can proceed with the Oracle Clusterware installation on the node you want to add to your RAC cluster, you must make sure that all operating system and hardware prerequisites are met. Because installation and configuration of your operating system is not the scope of this lesson, refer to the first lessons of this course for more information.

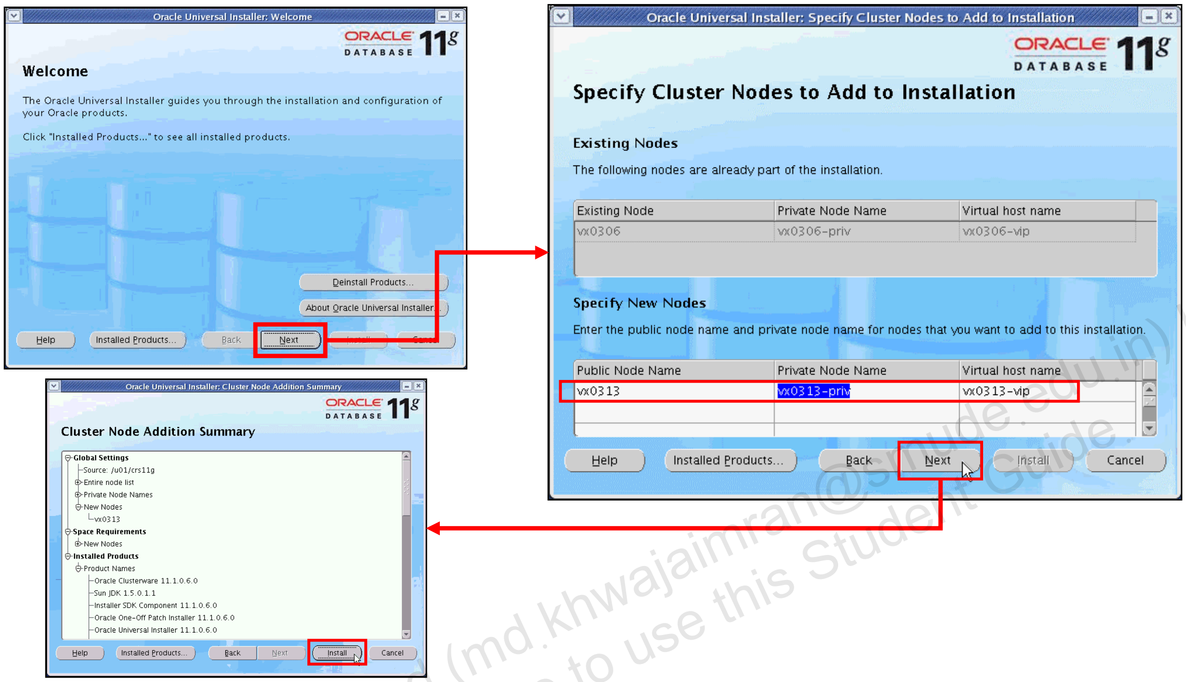
After this is done, you can verify that the system has been configured properly for Oracle Clusterware by using the following Cluster Verify command from one of the nodes that is already part of your cluster: \

`cluvfy stage -pre crsinst -n` <*list of all nodes*> `-r 11gR1`

The example shown in the slide assumes that you have only one node currently as part of your cluster, and you want to add a new one called VX0313. If any errors are reported during the preceding verification, fix them before proceeding to the next step.

Add Oracle Clusterware to the New Node

Execute <*Oracle Clusterware home*>`/oui/bin/addNode.sh`**.**



ORACLE

Add Oracle Clusterware to the New Node

Log in as the `oracle` user and execute the `addNode.sh` script located in your Oracle Clusterware home directory on the first node. This script runs the Oracle Universal Installer.

On the Welcome screen, click Next.

On the Specify Cluster Nodes to Add to Installation screen, OUI recognizes the existing nodes and asks you to enter the “short” public node name of the host you want to add to your cluster. That should automatically populate the corresponding Private Node Name and “Virtual host name” fields. Make sure that those three names are correct and click Next.

You next see the Cluster Node Addition Summary screen, where you can review the list of products to be installed. Click Install.

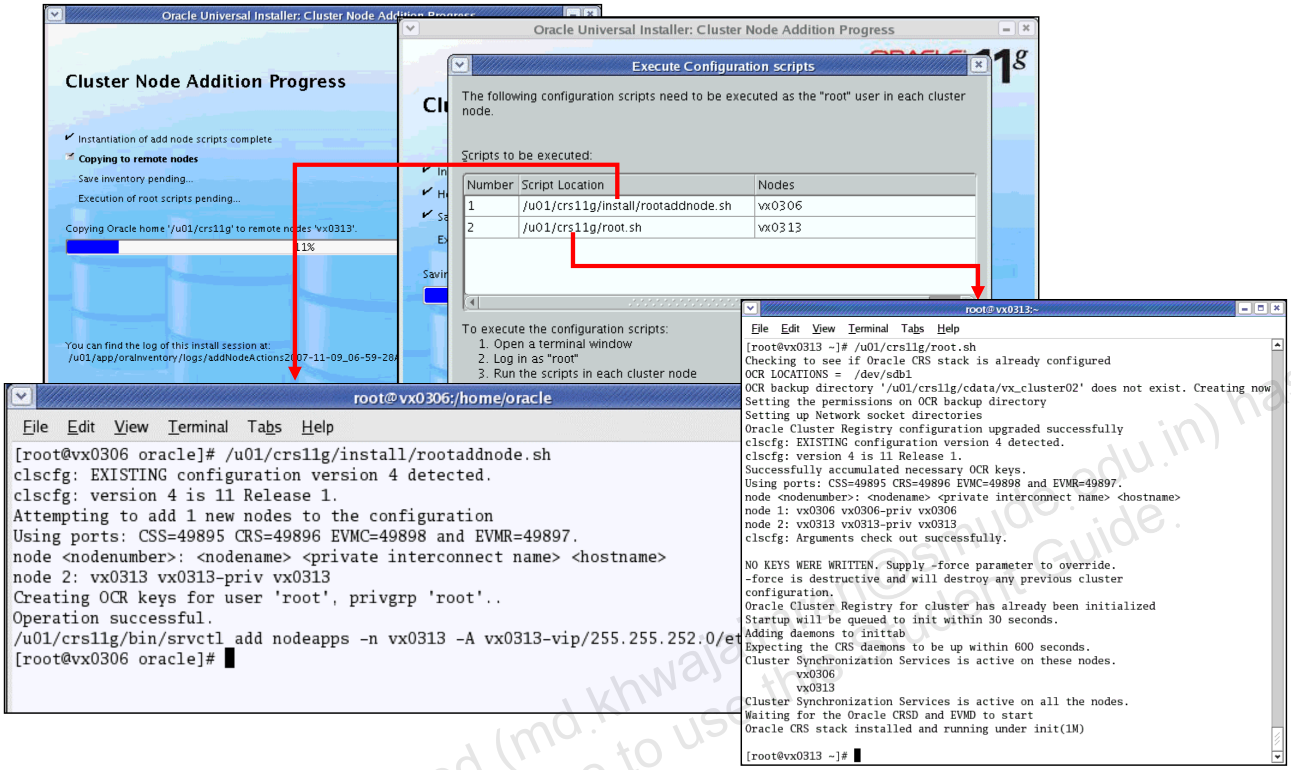
Note: It is also possible to execute OUI in silent mode. Here is a possible example where you want to add a new node called `newnode`:

```
addNode.sh -silent -responseFile myinstallresponsefile
```

Here, `myinstallresponsefile` contains the following:

```
CLUSTER_NEW_NODES = {"newnode"}
CLUSTER_NEW_PRIVATE_NODE_NAMES = {"newnode-priv"} CLUSTER_NEW_VIRTUAL_HOSTNAMES =
{"newnode-vip"}
```

Add Oracle Clusterware to the New Node



ORACLE

Add Oracle Clusterware to the New Node (continued)

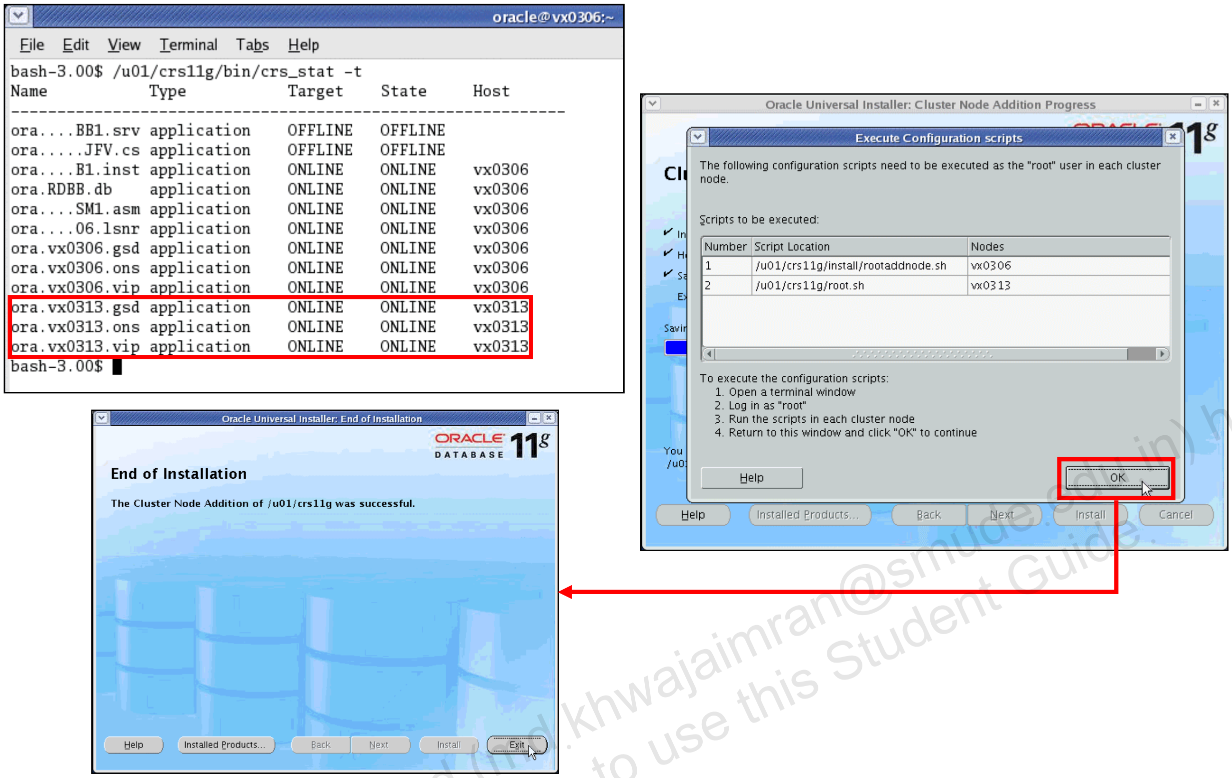
You can now follow the installation progression from the Cluster Node Addition Progress screen.

The OUI copies the Oracle Clusterware software to the new node, and then asks you to run few scripts as the `root` user on both nodes. Make sure that you run the scripts on the correct node as specified one after another.

You have to execute the `rootaddnode.sh` script on the first node. Basically, this script adds the `nodeapps` of the new node to the OCR configuration.

After this is done, you have to execute the `root.sh` script from the new node. This script starts the Oracle Clusterware stack on the new node and then uses VIPCA (Virtual IP Configuration Assistant) in silent mode for configuring `nodeapps`.

Add Oracle Clusterware to the New Node



Add Oracle Clusterware to the New Node (continued)

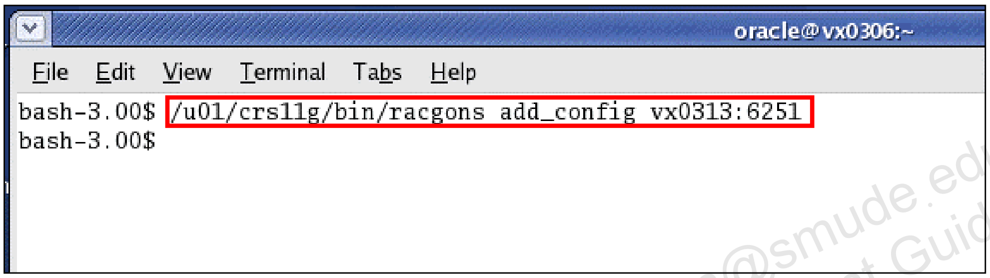
After both scripts are executed successfully, you can check your Oracle Cluster Registry (OCR) configuration as shown in the slide. At this point, the `crs_stat` command reports three new resources on the new node. These resources correspond to `nodeapps`.

Click OK on the Execute Configuration scripts screen to reach the end of the Oracle Clusterware installation.

On the End of Installation screen, click Exit.

Configure the New ONS

Use the `racgons add_config` command to add new node ONS configuration information to OCR.

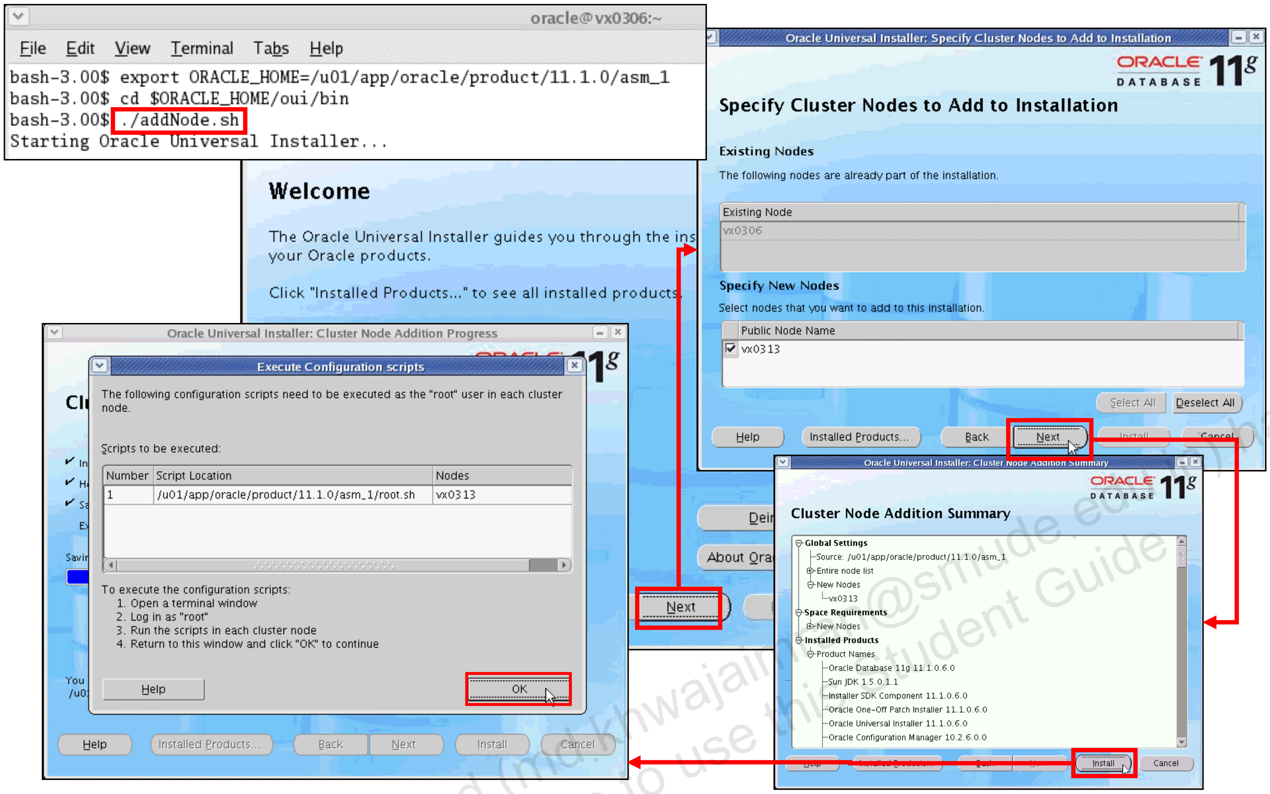


ORACLE

Configure the New ONS

You now need to add the new node ONS (Oracle Notification Server) configuration information to the shared ONS configuration information stored in OCR. From the first node, and looking at the `ons.config` file located in the `<Oracle Clusterware home>/opmn/conf` directory, you can determine the ONS remote port to be used (6251 in the slide). You need to use this port in the `racgons add_config` command as shown in the slide to make sure that the ONS on the first node can communicate with the ONS on the new node.

Add ASM Home to the New Node

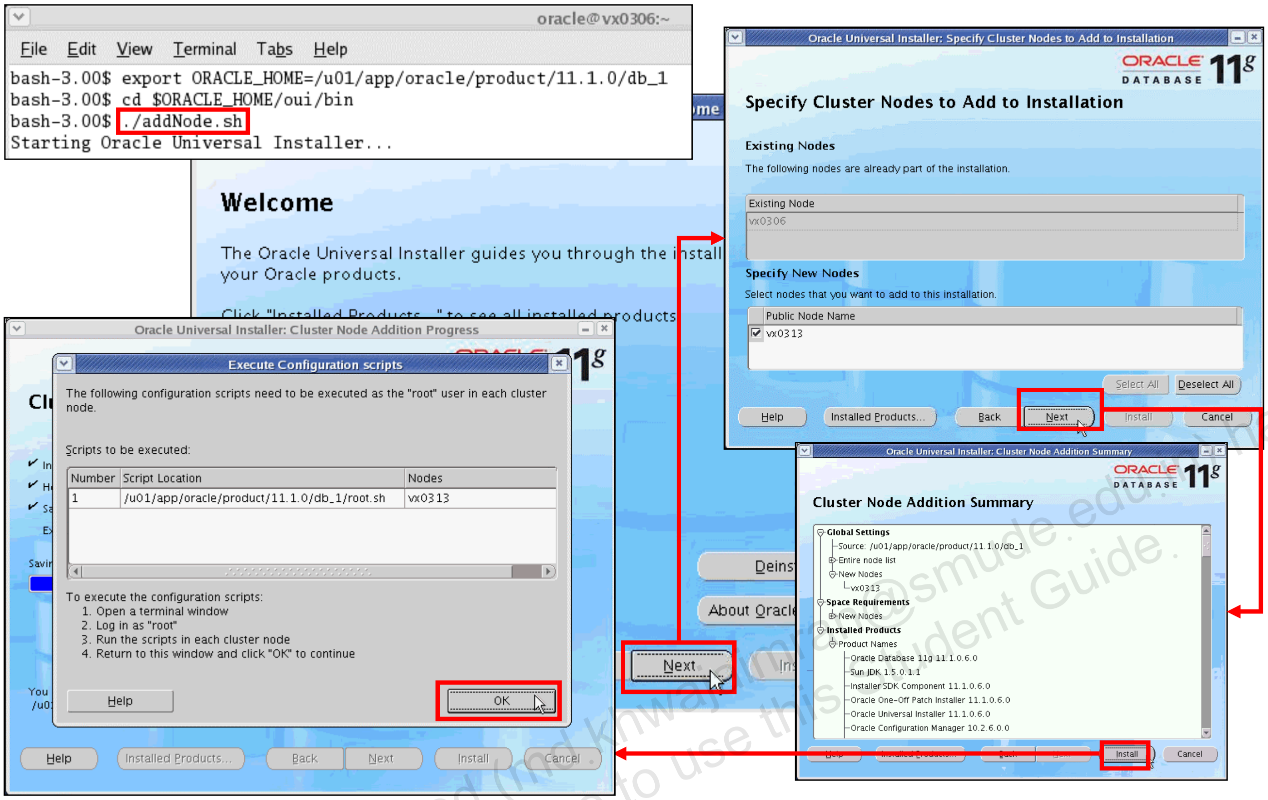


Add ASM Home to the New Node

This step is needed only if you use a specific home directory to host ASM. If you run ASM and your RAC database out of the same Oracle Home, you can skip this step.

From the first node, you need to execute the `addNode.sh` script from the ASM home directory as shown in the slide. The scenario is identical to the one shown for the Oracle Clusterware installation. However, in the case of an Oracle Home, you just need to select the name of the node you want to add on the Specify Cluster Nodes to Add to Installation screen, and then run the `root.sh` script from the new node after OUI has copied the database software.

Add RAC Home to the New Node

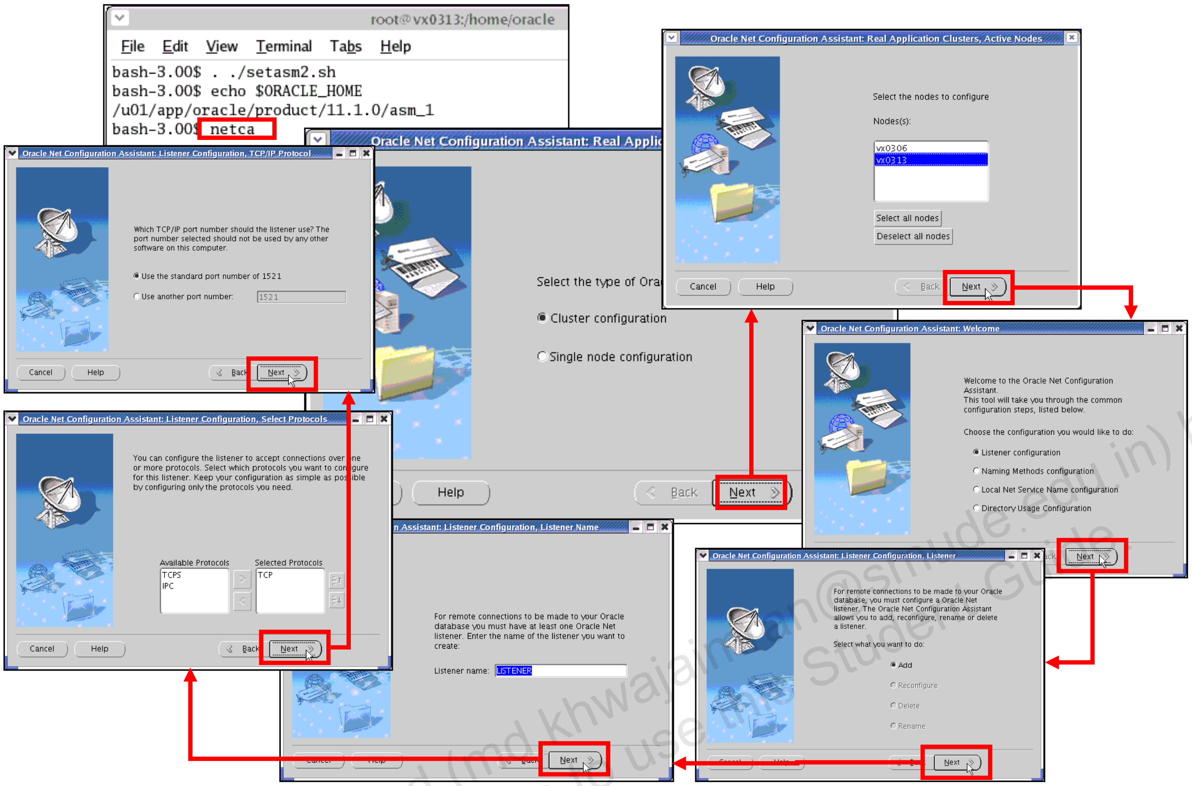


ORACLE

Add RAC Home to the New Node

From the first node, you need to execute the `addNode.sh` script from the RAC home directory as shown in the slide. The scenario is identical to the one shown for the ASM home installation.

Add a Listener to the New Node



Add a Listener to the New Node

From the new node, you need to add a listener. In this example, you are adding a listener from the ASM Home. You need to use NETCA (NETwork Configuration Assistant) for that.

On the Configuration screen, select "Cluster configuration" and click Next.

On the Active Nodes screen, select the name of the new node and click Next.

On the Welcome screen, select "Listener configuration" and click Next.

On the Listener screen, select Add and click Next.

One the Listener Name screen, enter `LISTENER` in the "Listener name" field.

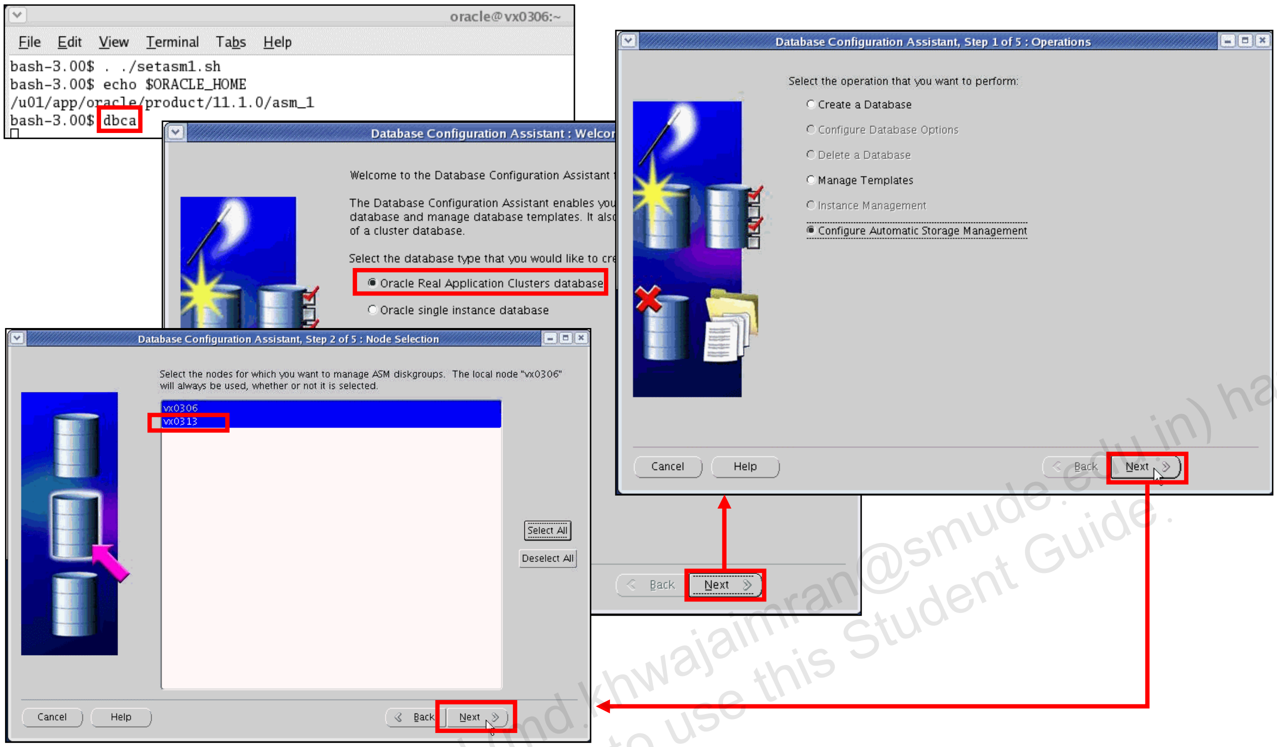
On the Select Protocols screen, select TCP and click Next.

On the TCP/IP Protocol screen, select "Use the standard port number of 1521," and click Next.

Continue to click Next until you exit from NETCA.

The steps above add a listener on the new node with the name `LISTENER_<New node name>`.

Add an ASM Instance to the New Node



Add an ASM Instance to the New Node

Before you can add your database instance to the new node, you need to add an ASM instance to the new node. To do so, use DBCA from your ASM home as shown on the above slide.

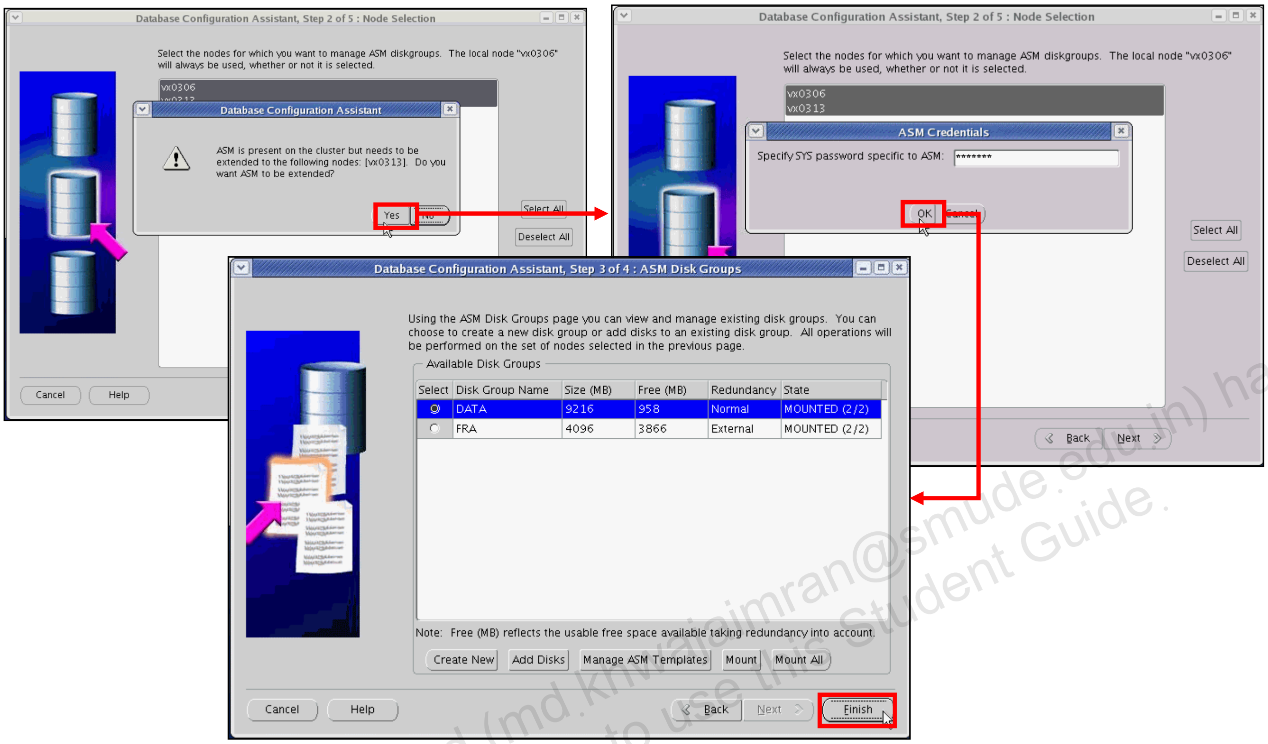
On the Welcome screen, click Next.

On the Operations screen, select Configure Automatic Storage Management and click Next.

On the Node Selection screen, select the node you want to add, and click Next.

Note: By default if your ASM instance is running out of the same oracle home, then DBCA automatically extends ASM to the new node when you use DBCA to extend your database instance to the new node.

Add an ASM Instance to the New Node



Add an ASM Instance to the New Node (continued)

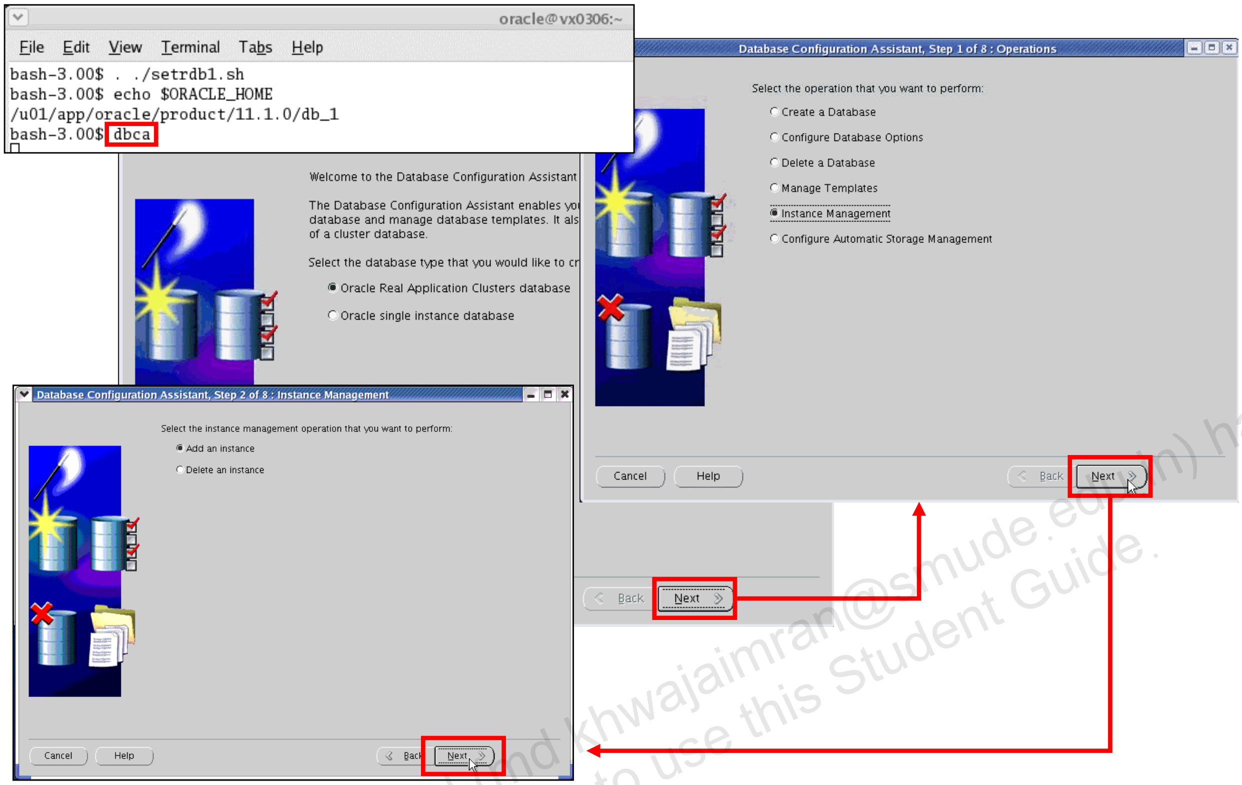
After a while, DBCA prompts you for ASM instance addition on your second node. Click Yes.

DBCA then asks you to enter the password for the ASM administrator. Enter your password and click OK.

On the ASM Disk Groups screen, you should see all your disk groups mounted on all nodes of your cluster.

Click Finish.

Add a Database Instance to the New Node



Add a Database Instance to the New Node

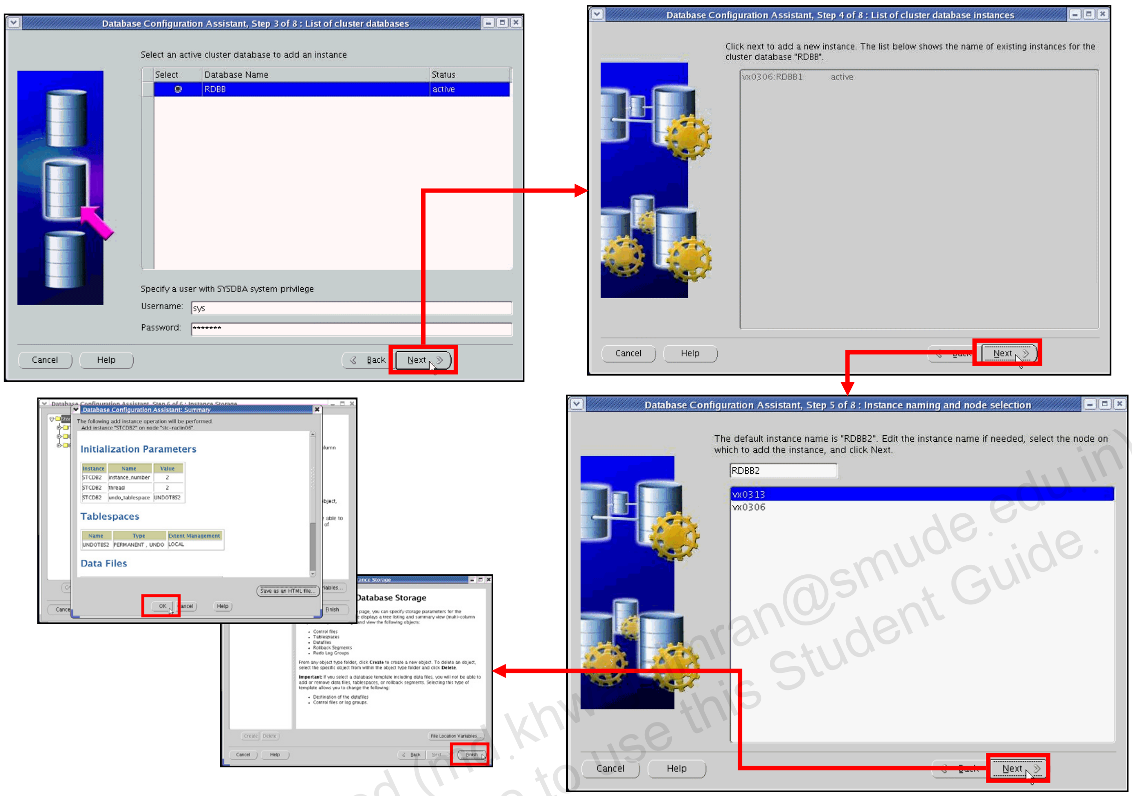
You now need to add a database instance to your RAC database. You can do so by using the DBCA from the first node.

On the Welcome screen, select "Oracle Real Application Clusters database" and click Next.

On the Operations screen, select Instance Management and click Next.

On the Instance Management screen, select "Add an instance" and click Next.

Add a Database Instance to the New Node



Add a Database Instance to the New Node (continued)

On the "List of cluster databases" screen, select your RAC database and enter `SYS` credentials. Then, click Next.

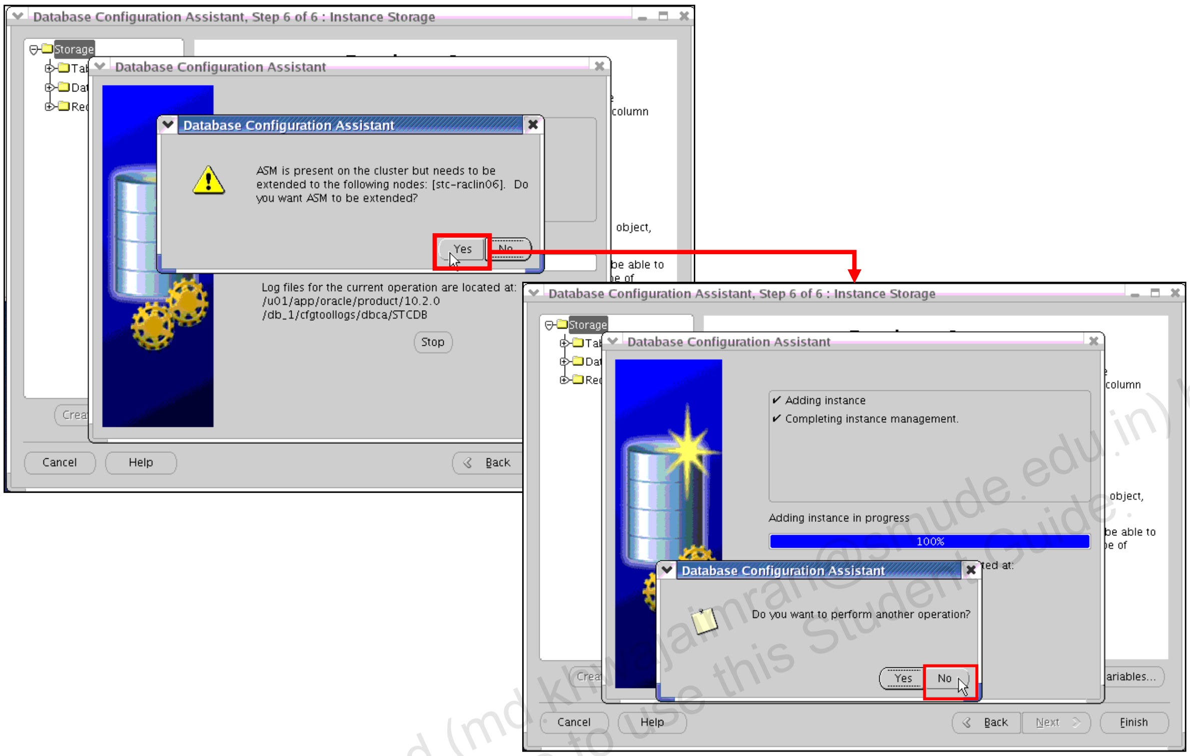
On the "List of cluster database instances" screen, click Next.

On the "Instance naming and node selection" screen, select the node name on which you want to add the instance, and specify the name of that instance. When done, click Next.

On the Instance Storage screen, click Finish.

On the Summary screen, check the various parameters and click OK.

Add a Database Instance to the New Node

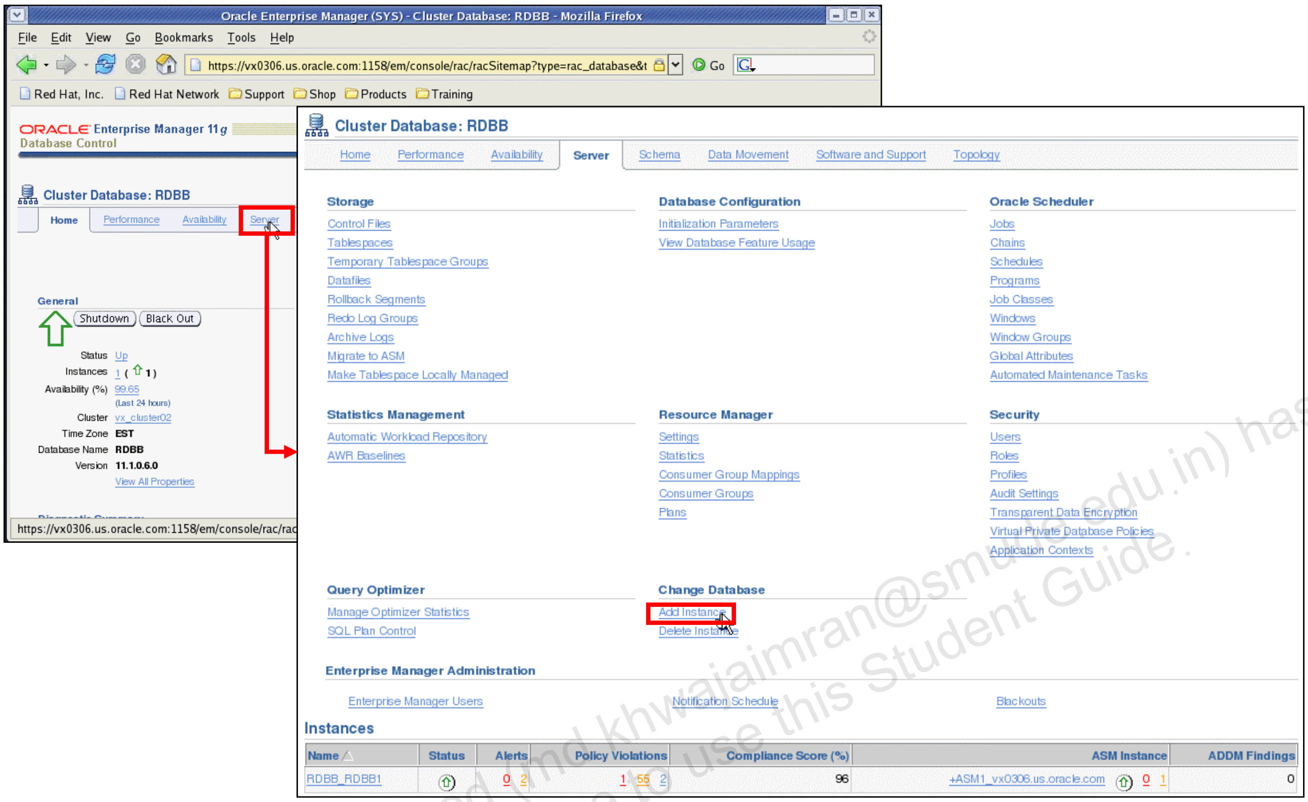


Add a Database Instance to the New Node (continued)

At this point, if you are using ASM for your database storage, and there is currently no running ASM instance on your new node, DBCA detects the need for an ASM instance creation on the new node. This must be done before the DBCA can create the database instance on that node. Click Yes.

The assistant is now adding your instance to your RAC database on the new node. It will also start that instance at the end of the operation.

Add an Instance to Your RAC Database Using EM

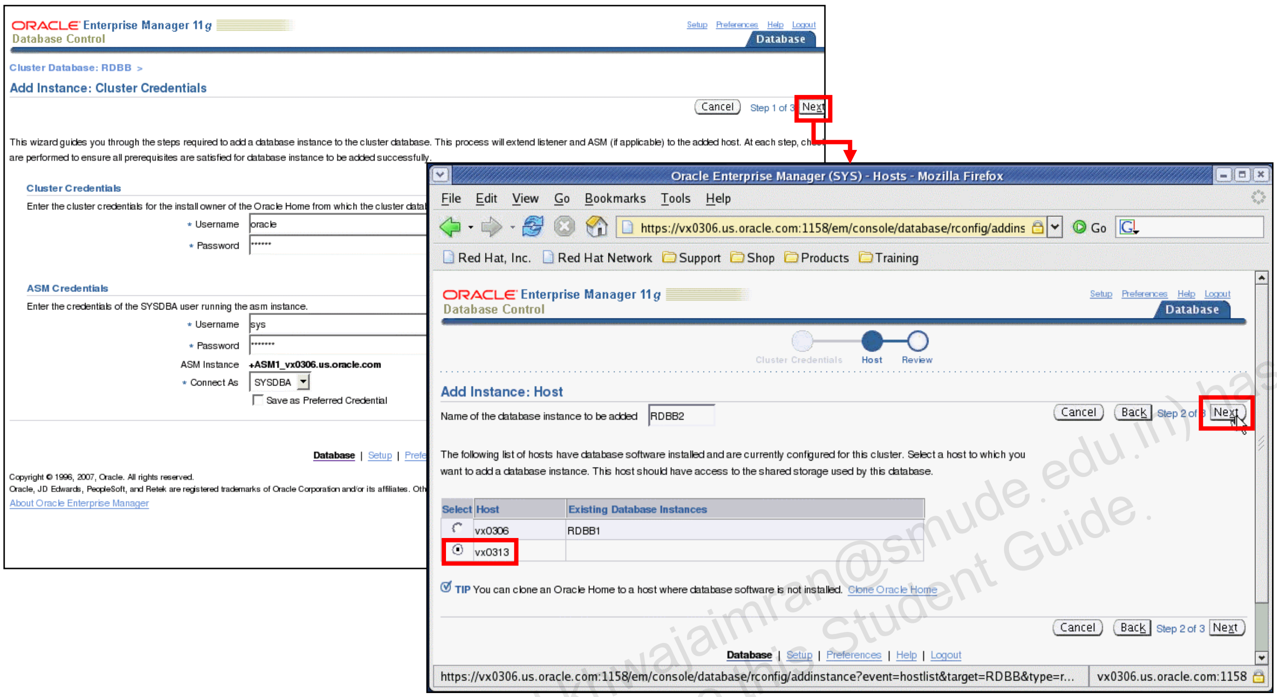


ORACLE

Add an Instance to Your RAC Database Using EM

You can also add a new instance to your RAC database by using the Add Instance Wizard. From the Cluster Database page, click the Server tab. On the Server tabbed page, click Add Instance in the Change Database section of the page.

Add an Instance to Your RAC Database Using EM

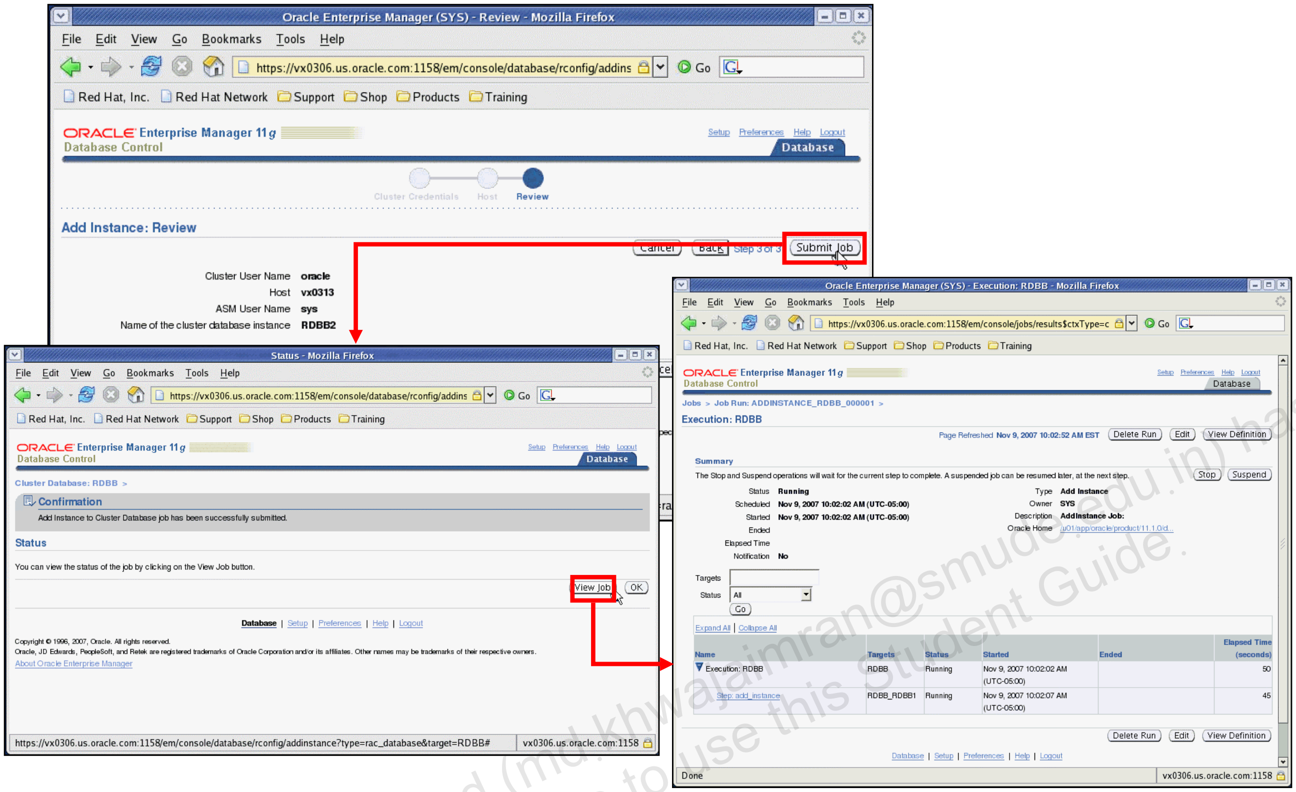


Add an Instance to Your RAC Database Using EM (continued)

You are now on the Cluster Credentials page where you specify the cluster and ASM credentials. The wizard automatically adds the ASM instance before adding the database instance if it is not already created.

When done, click Next to go to the Host page where you specify on which host you want to add the database instance. Select the node in question and click Next.

Add an Instance to Your RAC Database Using EM



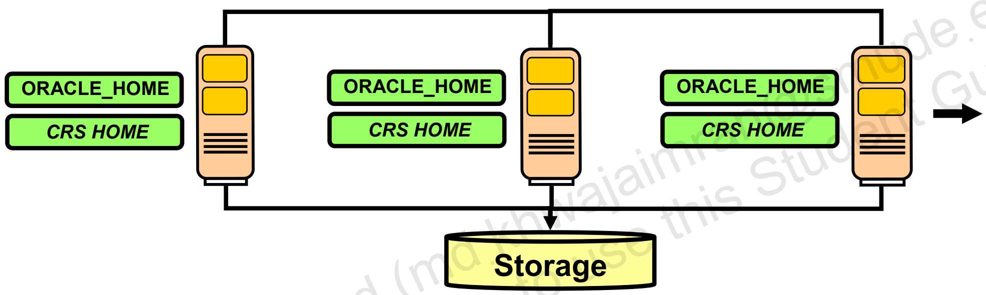
Add an Instance to Your RAC Database Using EM (continued)

On the Review page, click Submit to start the job's execution.

On the Confirmation page, click "View job" to see the job's log. After some refreshes of that page, you should get a succeeded status.

Main Steps to Delete a Node from a RAC Cluster

1. Delete the instance on the node to be deleted.
2. Clean up the ASM instance.
3. Remove the listener from the node to be deleted.
4. Remove the node from the database.
[5. Remove the node from ASM.]
6. Remove ONS configuration from the node to be deleted.
7. Remove the node from the clusterware.

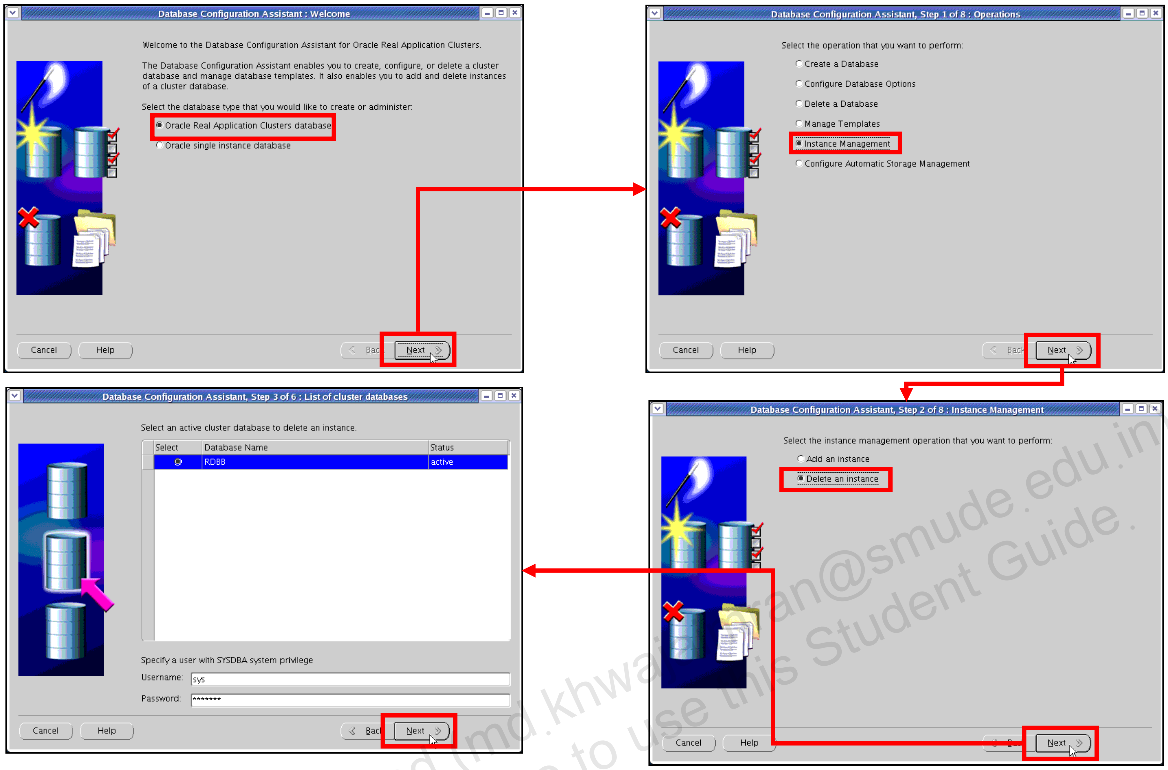


ORACLE

Main Steps to Delete a Node from a RAC Cluster

Note: For all of the add node and delete node procedures for UNIX-based systems, temporary directories such as `/tmp`, `$TEMP`, or `$TMP`, should not be shared directories. If your temporary directories are shared, then set your temporary environment variable, such as `$TEMP`, to a nonshared location on a local node. In addition, use a directory that exists on all the nodes.

Delete the Instance on the Node to Be Deleted



Delete the Instance on the Node to Be Deleted

The first step is to remove the database instance from the node that you want to delete. For that, you use the DBCA from the node you want to delete.

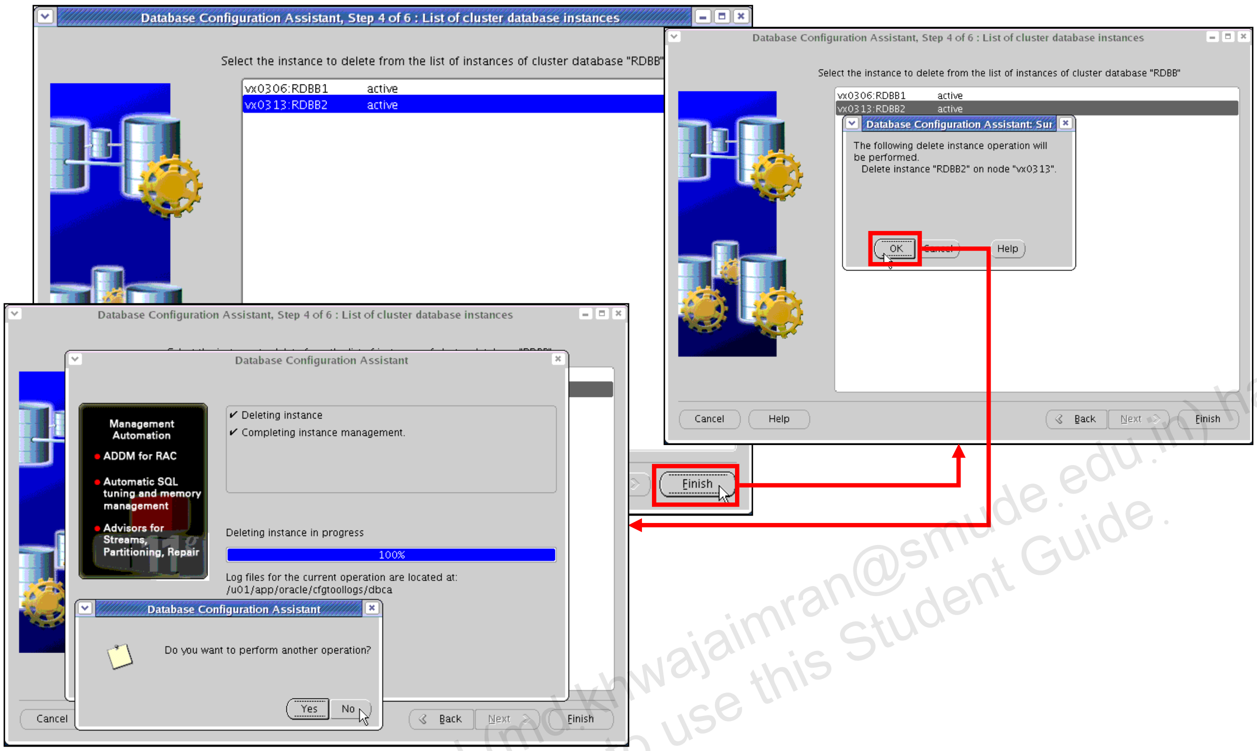
On the Welcome screen, select “Oracle Real Application Clusters database” and click Next.

On the Operations screen, select Instance Management and click Next.

On the Instance Management screen, select “Delete an instance” and click Next.

On the “List of cluster databases” screen, select the RAC database from which you want to delete an instance, enter SYS and its password, and click Next.

Delete the Instance on the Node to Be Deleted



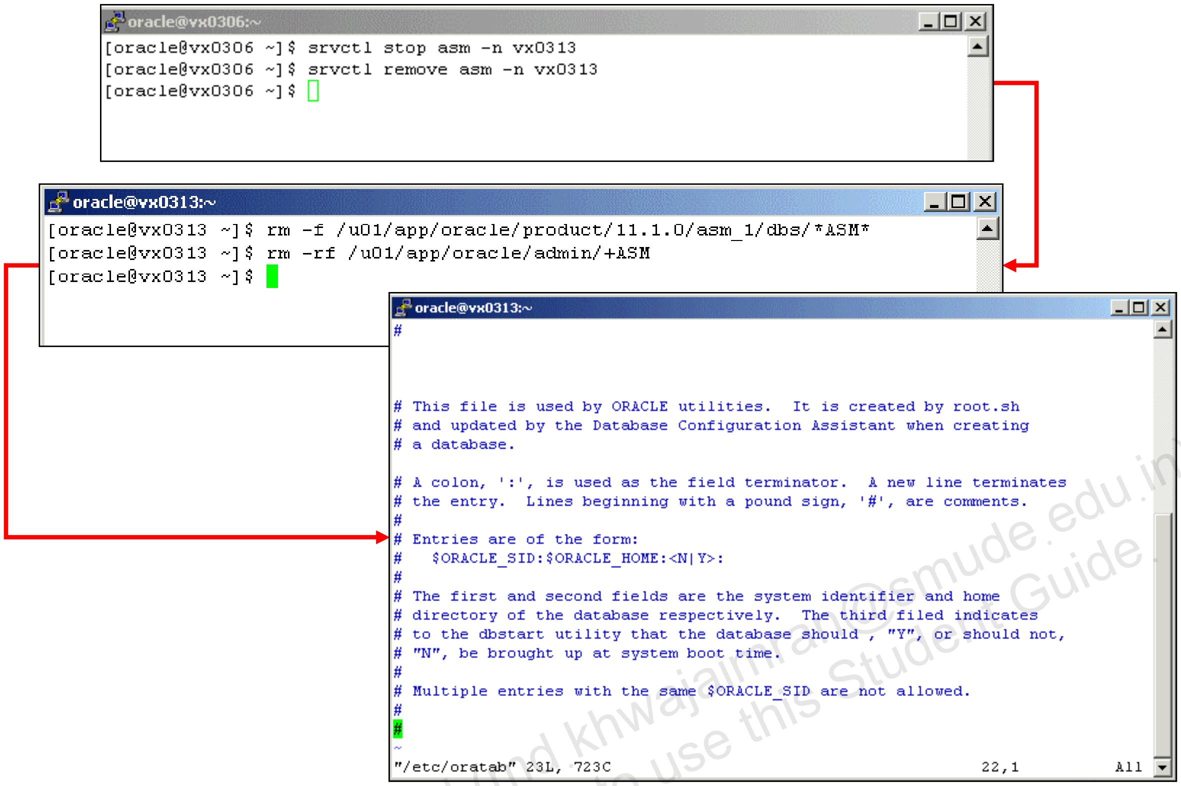
Delete the Instance on the Node to Be Deleted (continued)

On the "List of cluster database instances" screen, select the instance that you want to delete and click Finish.

In the Database Configuration Assistant dialog box, click OK to validate your choice.

This triggers the remove instance process. When completed, your instance is removed from your cluster database.

Clean Up the ASM Instance



Clean Up the ASM Instance

After your database instance is removed from the node, you can clean up the corresponding ASM instance.

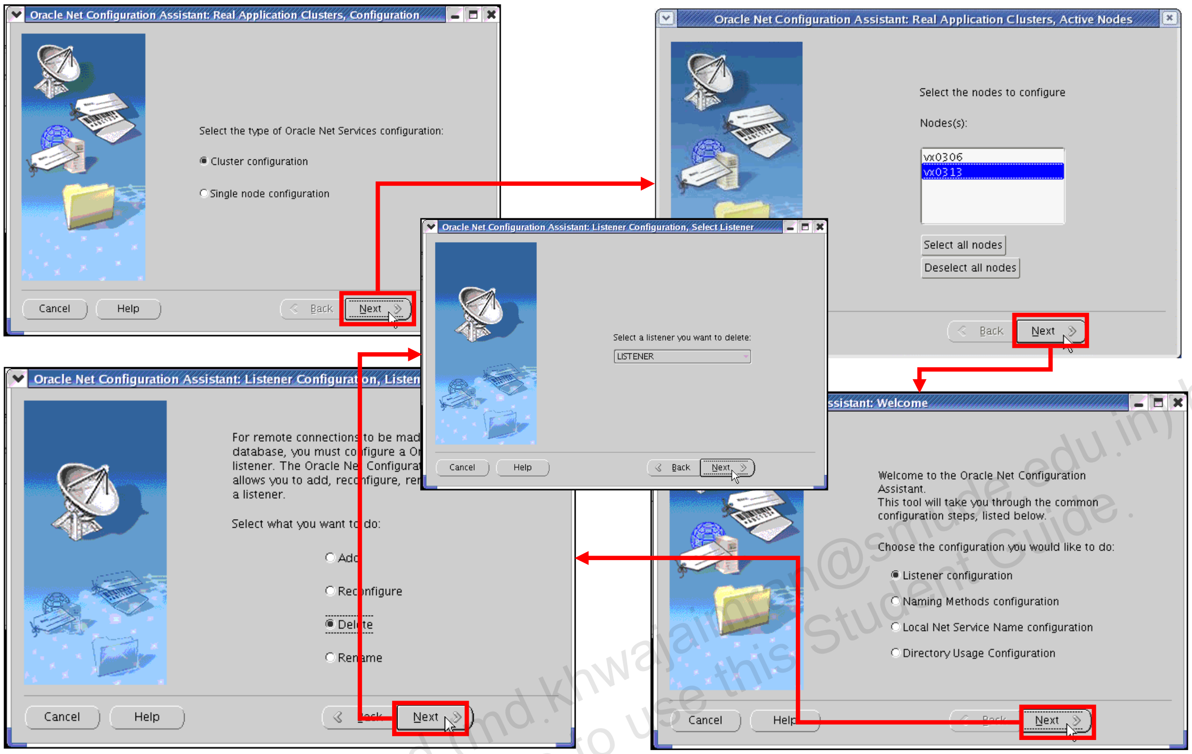
To do this, you need to use `SRVCTL` to first stop the ASM instance currently running on the node that you want to remove, and then remove that ASM instance from the same node. The two commands are illustrated in the screenshot shown in the slide.

Then, you need to manually remove the initialization parameter file of that ASM instance. As shown in the slide, you can remove files containing the ASM string from the `<ASM home>/dbs` directory.

After this is done, you can also remove all the log files of that ASM instance. These files are generally located in the `$ORACLE_BASE/admin` directory.

The last thing you can do is to remove the associated ASM entry from the `/etc/oratab` file.

Remove the Listener from the Node to Be Deleted



Remove the Listener from the Node to Be Deleted

You can now remove the listener from the node that you want to delete. This listener can be from either the ASM home or the database home depending on when it was created.

To remove the listener, you can use NETCA as shown in the slide.

On the Configuration screen, select “Cluster configuration” and click Next.

On the Active Nodes screen, select the node from which you want to remove the listener and click Next.

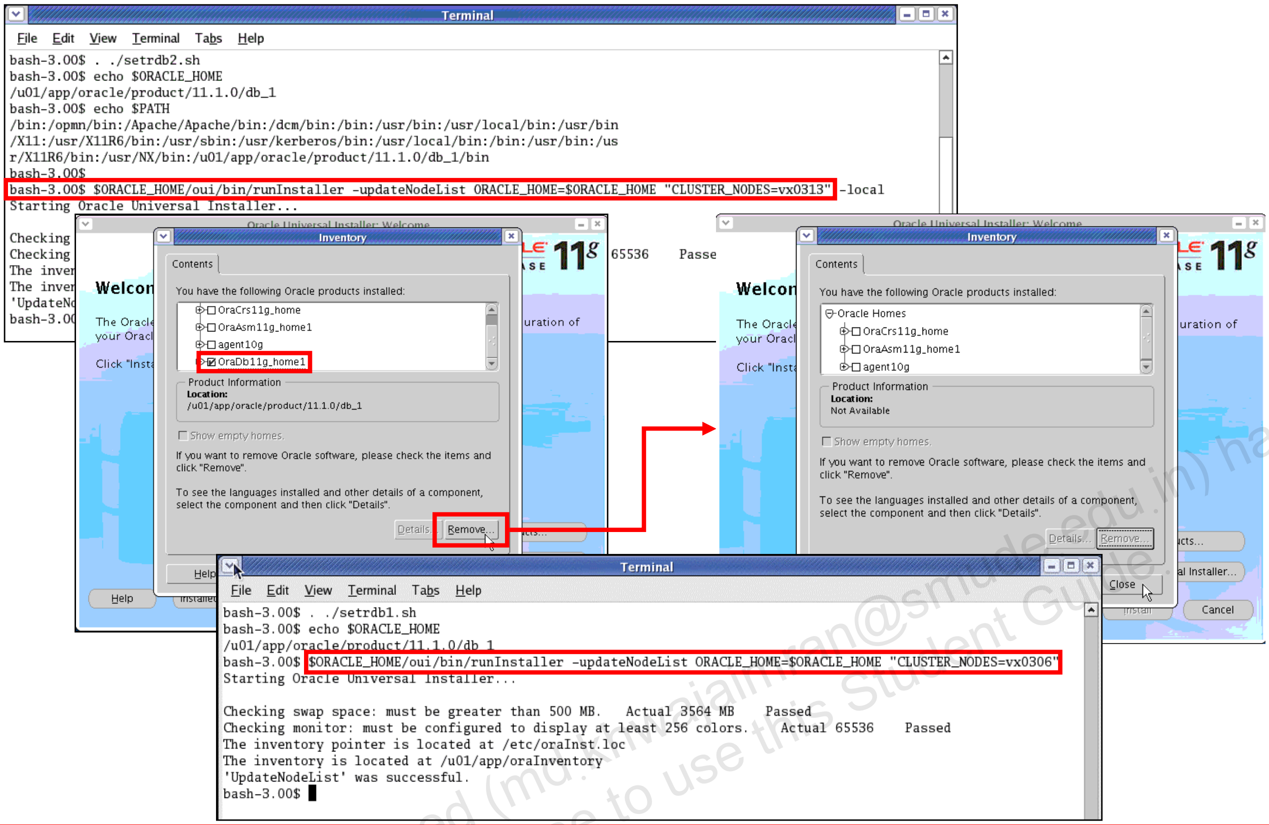
On the Welcome screen, select “Listener configuration” and click Next.

On the Listener screen, select Delete and click Next.

On the “Select listener” screen, select the corresponding listener, normally called LISTENER, and click Next.

Follow the rest of the screens until the listener is removed from the node.

Remove the Node from the Database



ORACLE

Remove the Node from the Database

Before you can use the Oracle Universal Installer to remove the database software installation, you need to update the inventory on the node to be deleted by executing the following command (also shown in the slide):

`./runInstaller -updateNodeList ORACLE_HOME=`*<Database home>* `"CLUSTER_NODES=`*<node to be removed>*`" -local`

You need to execute this command from the `oui/bin` subdirectory in the database home.

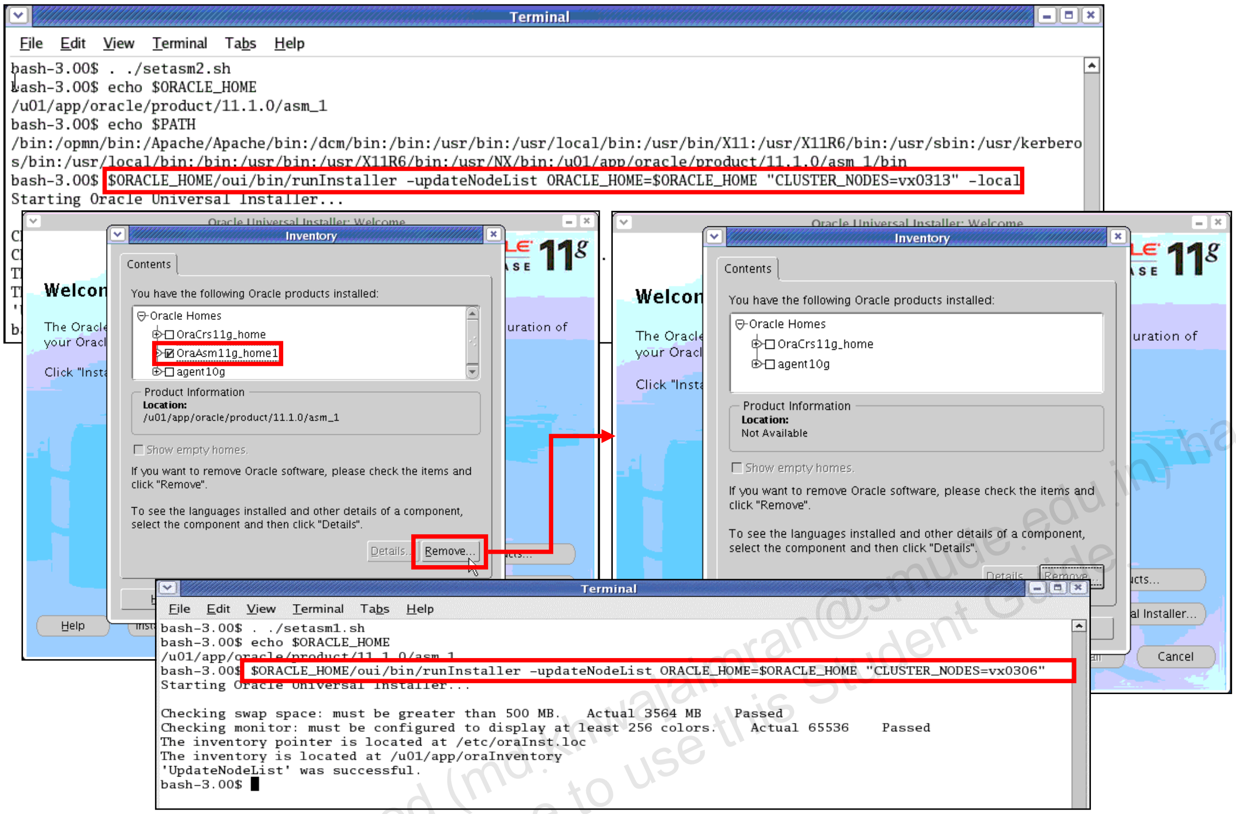
After this command is executed, you can start OUI from the same directory, and click “Deinstall products” on the Welcome screen. Then, select the database home and click Remove. This will remove the database home from the node to be deleted.

You now need to update the corresponding inventory on the remaining nodes. You can use the following command from the first node:

`./runInstaller -updateNodeList ORACLE_HOME=`*<Database home>* `"CLUSTER_NODES=`*<remaining nodes>*`"`

This command needs to be executed from the `oui/bin` subdirectory of the database home.

Remove the Node from ASM



Remove the Node from ASM

Before you can use the Oracle Universal Installer to remove the ASM software installation, you need to update the inventory on the node to be deleted by executing the following command (also shown in the slide):

`./runInstaller -updateNodeList ORACLE_HOME=`<*ASM home*> `"CLUSTER_NODES=`<*node to be removed*>`" -local`

You need to execute this command from the `oui/bin` subdirectory in the ASM home.

After this command is executed, you can start OUI from the same directory, and click “Deinstall products” on the Welcome screen. Then, select the ASM home and click Remove. This will remove the ASM home from the node to be deleted.

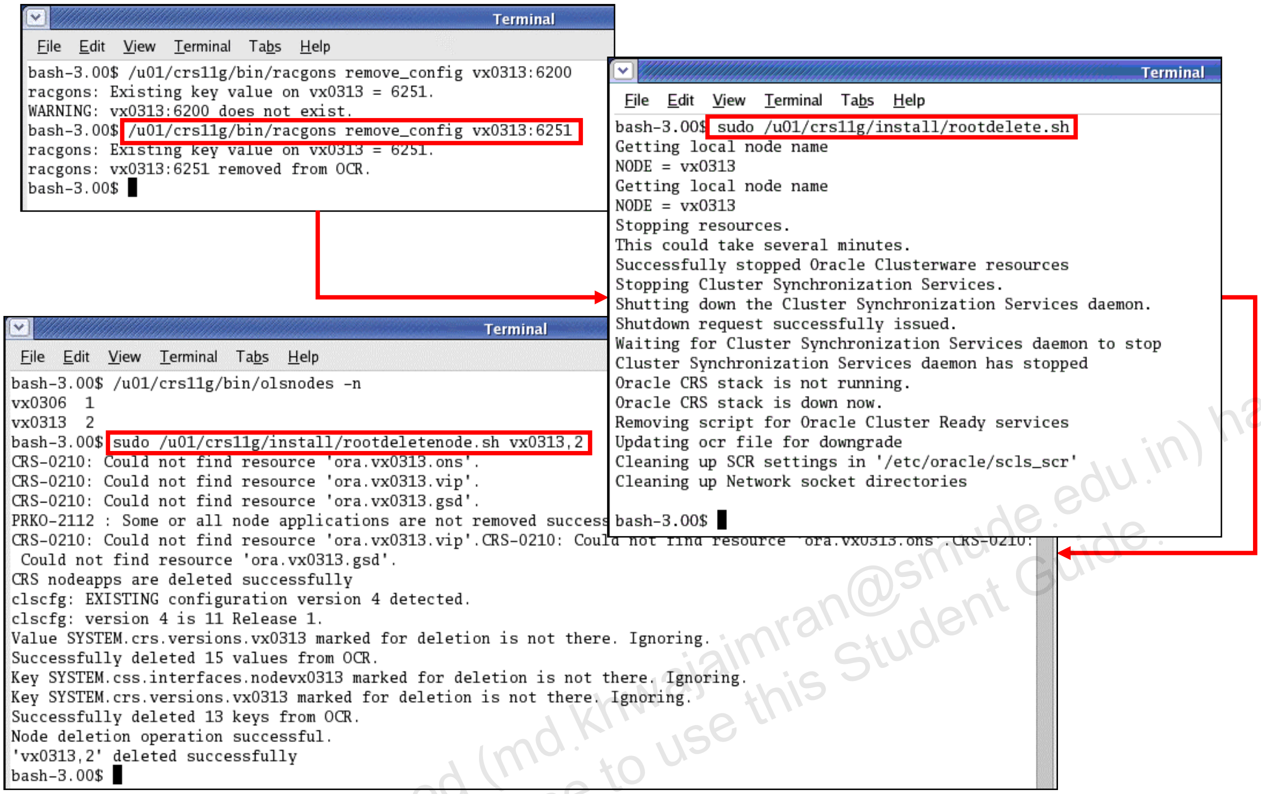
You now need to update the corresponding inventory on the remaining nodes. You can use the following command from the first node:

`./runInstaller -updateNodeList ORACLE_HOME=`<*ASM home*> `"CLUSTER_NODES=`<*remaining nodes*>`"`

This command needs to be executed from the `oui/bin` subdirectory of the ASM home.

Note: This step is not needed if you are not using a separate home directory for ASM.

Remove the Node from the Oracle Clusterware



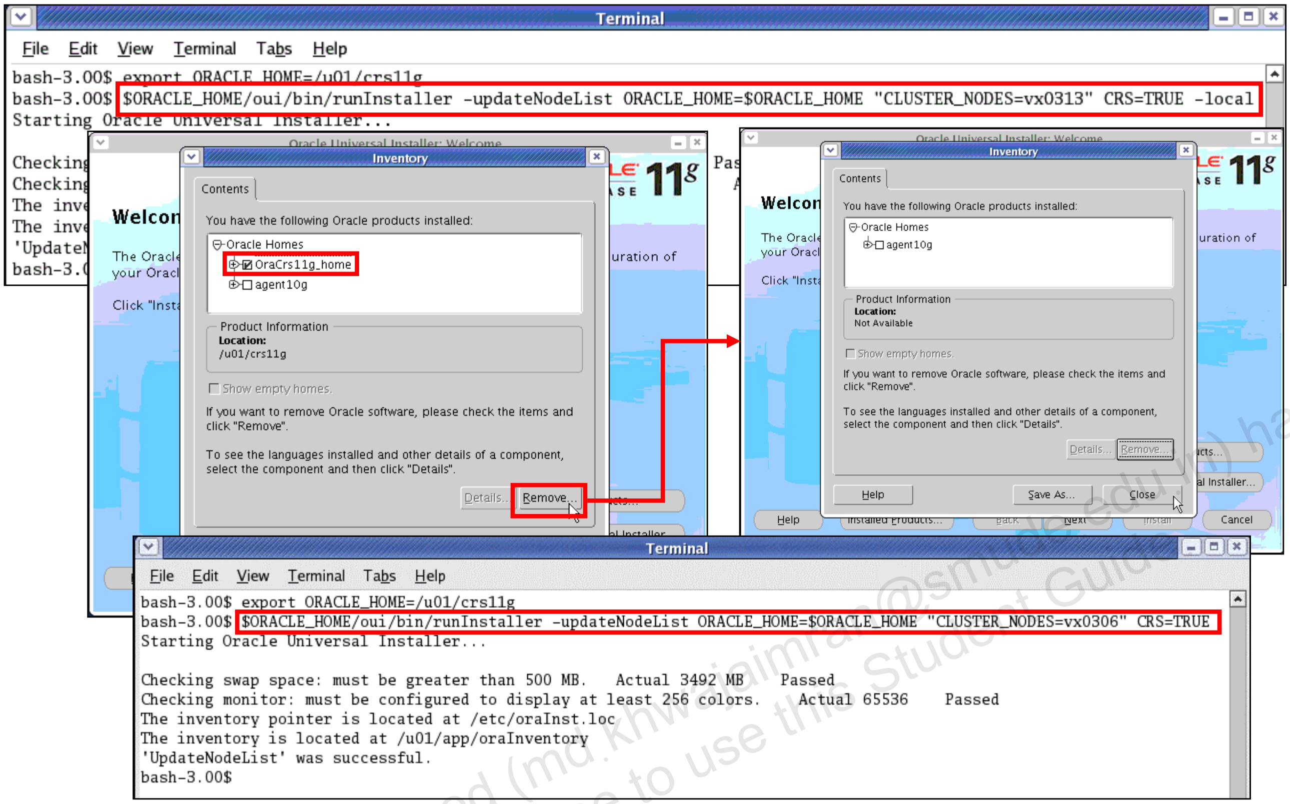
Remove the Node from the Oracle Clusterware

Before you can use OUI to remove the Oracle Clusterware software installation from the node to be deleted, you need to perform the following commands:

- The following needs to be done from the first node: *<Oracle Clusterware home>*`/bin/racgons remove_config` *<Node to be removed>*`:6251` (Replace port 6251 in the command above with the port number that you get in the remoteport section of the `ons.config` file found in the *<Oracle Clusterware home>*`/opmn/conf` directory.)
- The following needs to be done from the node to be removed as the `root` user: *<Oracle Clusterware home>*`/install/rootdelete.sh`
- The following needs to be done from the first node as the `root` user: Determine the node number to be deleted using *<Oracle Clusterware home>*`/bin/olsnodes -n`. Then execute *<Oracle Clusterware home>*`/install/rootdeletenode.sh` *<node name to be deleted>*`,`*<node number to be deleted>*.

These three steps are illustrated in the slide.

Remove the Node from the Oracle Clusterware



Remove the Node from the Oracle Clusterware (continued)

You now need to update the inventory from the node to be deleted by executing: `<`*Oracle Clusterware home*`>/oui/bin/runInstaller -updateNodeList ORACLE_HOME=<`*Oracle Clusterware home*`> "CLUSTER_NODES=<`*Node to be deleted*`>" CRS=TRUE -local`

When done, run OUI from the same directory and choose “Deinstall products” and remove the Oracle Clusterware installation on the node to be deleted as illustrated in the slide.

You can now update the inventory from the first node by executing the following command: `<`*Oracle Clusterware home*`>/oui/bin/runInstaller -updateNodeList ORACLE_HOME=<`*Oracle Clusterware home*`> "CLUSTER_NODES=<`*Remaining nodes*`>" CRS=TRUE`

To verify the removal of the node from the cluster, run the following commands from the first node:

- `srvctl status nodeapps -n <`*Deleted node*`>` should get a message saying Invalid node.
- `crs_stat | grep -i <`*Deleted node*`>` should not get any output.
- `olsnodes –n` should get all the present nodes list without the deleted node.

Note: You should also remove all corresponding oracle homes after this step.

Node Addition and Deletion and the SYSAUX Tablespace

- The SYSAUX tablespace combines the storage needs for the following tablespaces:
  - DRSYS
  - CWMLITE
  - XDB
  - ODM
  - OEM-REPO
- Use this formula to size the SYSAUX tablespace:

```
300M + (250M * number_of_nodes)
```


Node Addition and Deletion and the SYSAUX Tablespace

A new auxiliary, system-managed tablespace called SYSAUX contains performance data and combines content that was stored in different tablespaces (some of which are no longer required) in earlier releases of the Oracle database. This is a required tablespace for which you must plan disk space. The SYSAUX system tablespace now contains the DRSYS (contains data for OracleText), CWMLITE (contains the OLAP schemas), XDB (for XML features), ODM (for Oracle Data Mining), and OEM-REPO tablespaces.

If you add nodes to your RAC database environment, then you may need to increase the size of the SYSAUX tablespace. Conversely, if you remove nodes from your cluster database, then you may be able to reduce the size of your SYSAUX tablespace and thus save valuable disk space. The following is a formula that you can use to properly size the SYSAUX tablespace:

```
300 megabytes + (250 megabytes * number_of_nodes)
```

If you apply this formula to a four-node cluster, then you find that the SYSAUX tablespace is sized around 1,300 megabytes as shown below:

```
300 + (250 * 4) = 1300
```

Summary

In this lesson, you should have learned how to:
- Add a new node to your cluster database
- Remove a node from your cluster database

Practice 11: Overview

This practice covers the following topics:

- Removing the second node of your cluster
- Adding it back again

Important Note

You are strongly advised to follow directly the solution appendix for this lab.

12

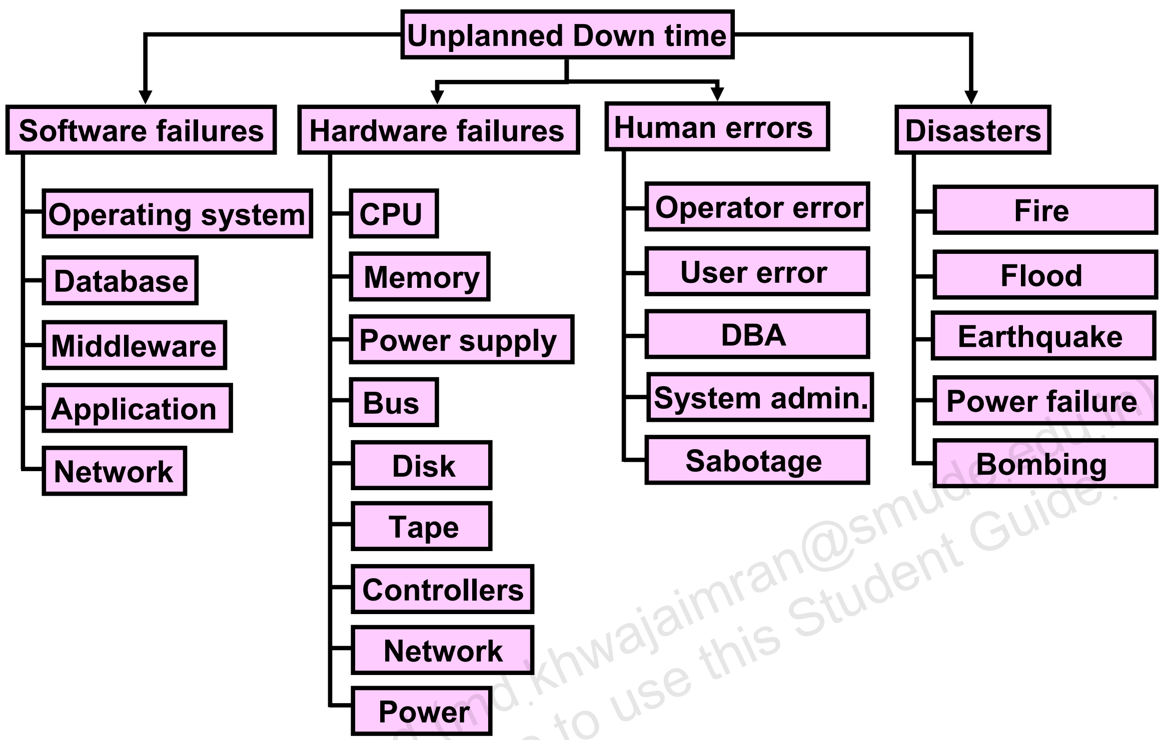
Design for High Availability

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
- Decide on the best ASM configuration to use
- Patch your RAC system in a rolling fashion

Causes of Unplanned Down Time



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 schema 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.

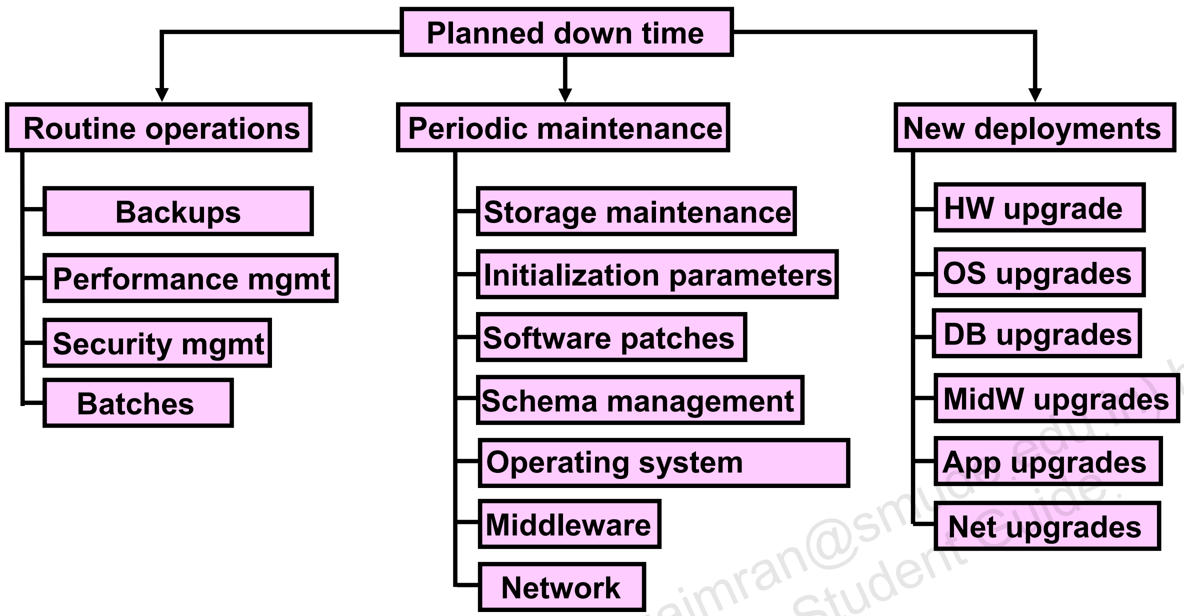
Software failures include operating system, database, middleware, application, and network failures. A failure of any one of these components can cause a system fault.

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

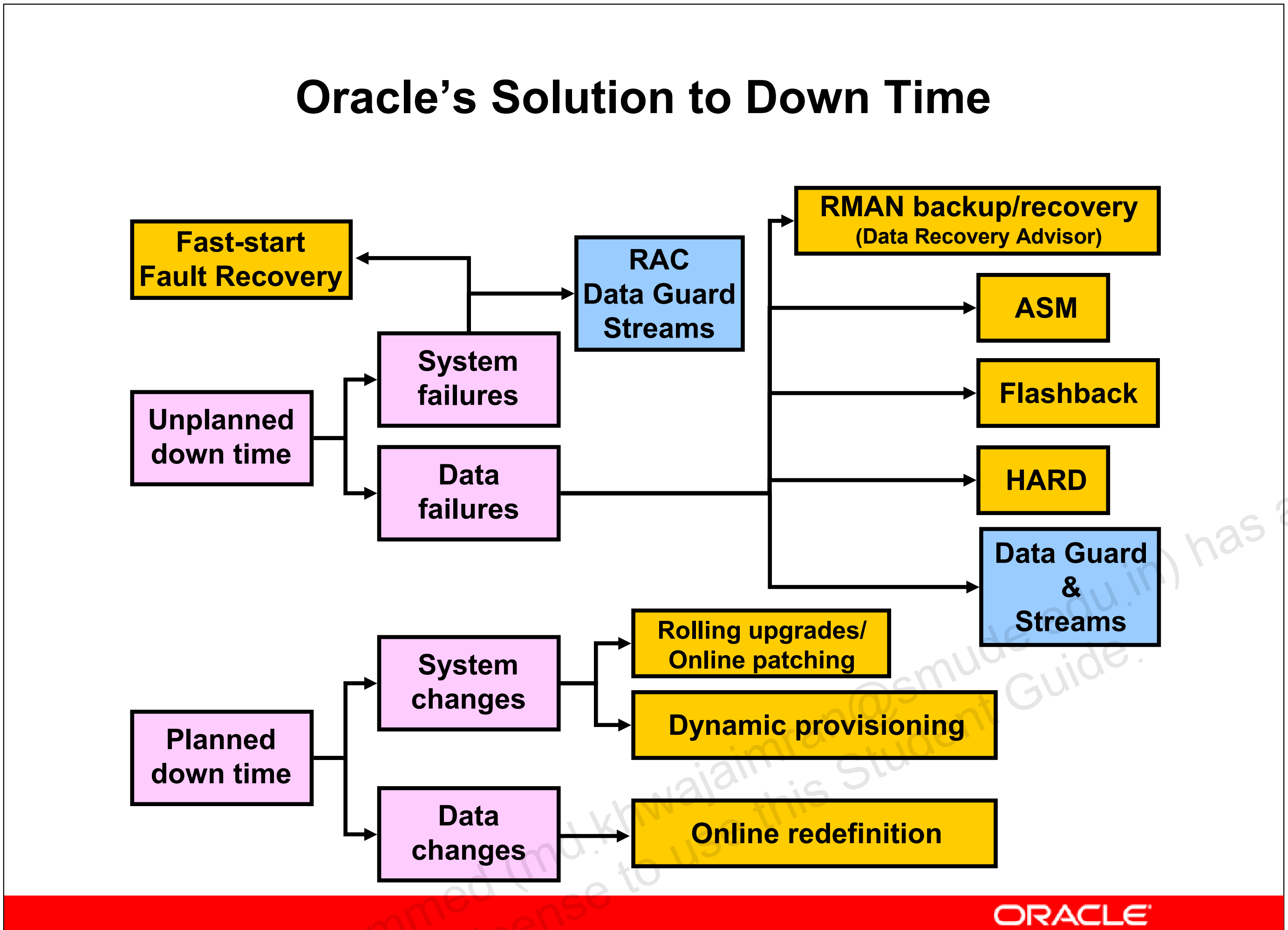


Causes of Planned Down Time

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 schema 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.

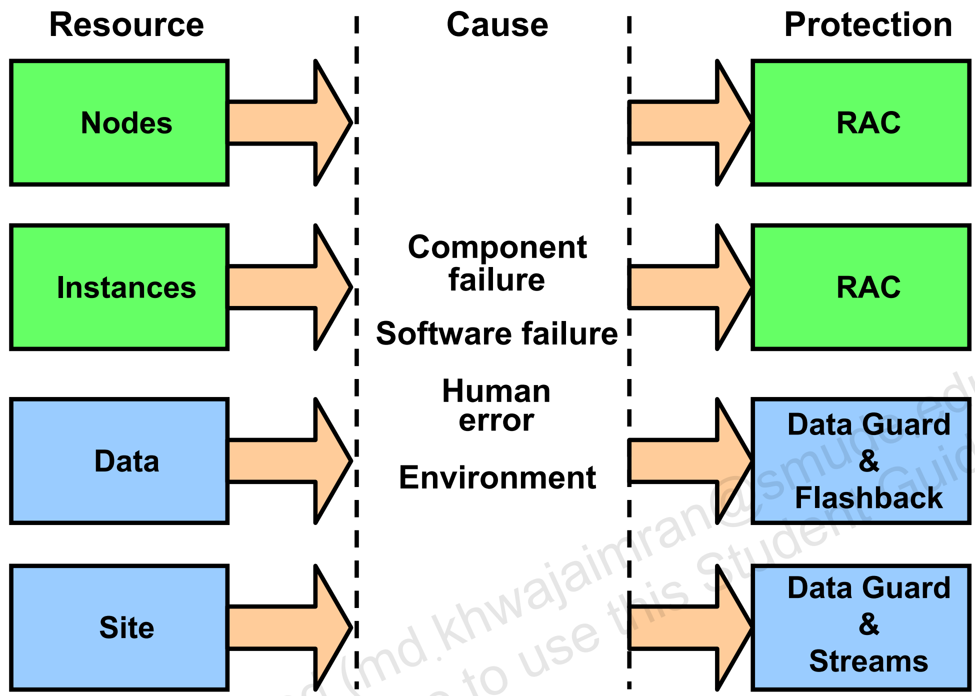


Oracle's Solution to Down Time

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 database 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 database 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 investment 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.
- The Oracle database continues to broaden support for dynamic reconfiguration, enabling it to adapt to changes in demand and hardware with no disruption of service.
- The 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

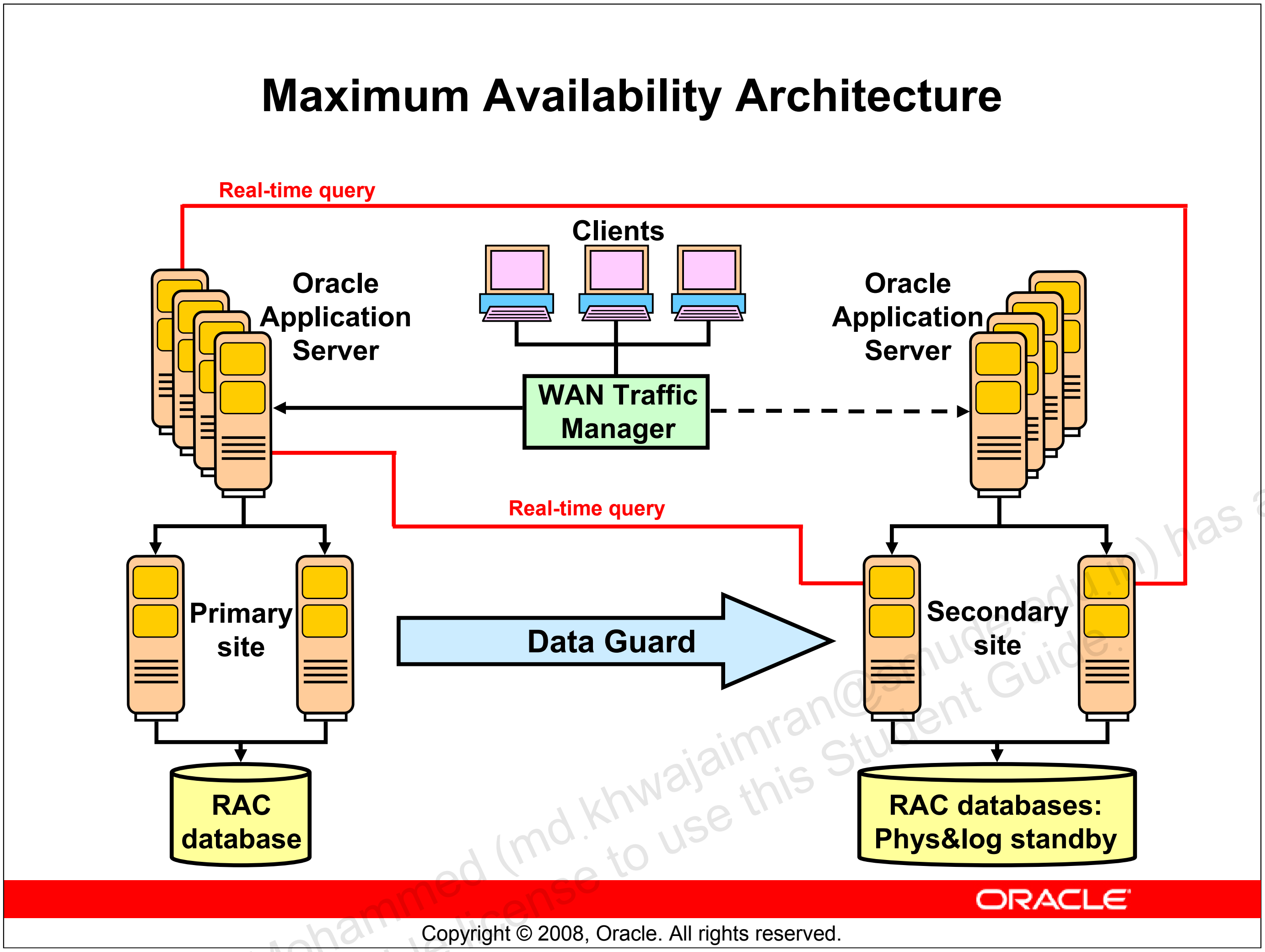


RAC and Data Guard Complementarity

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 (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

RAC and Data Guard Topologies

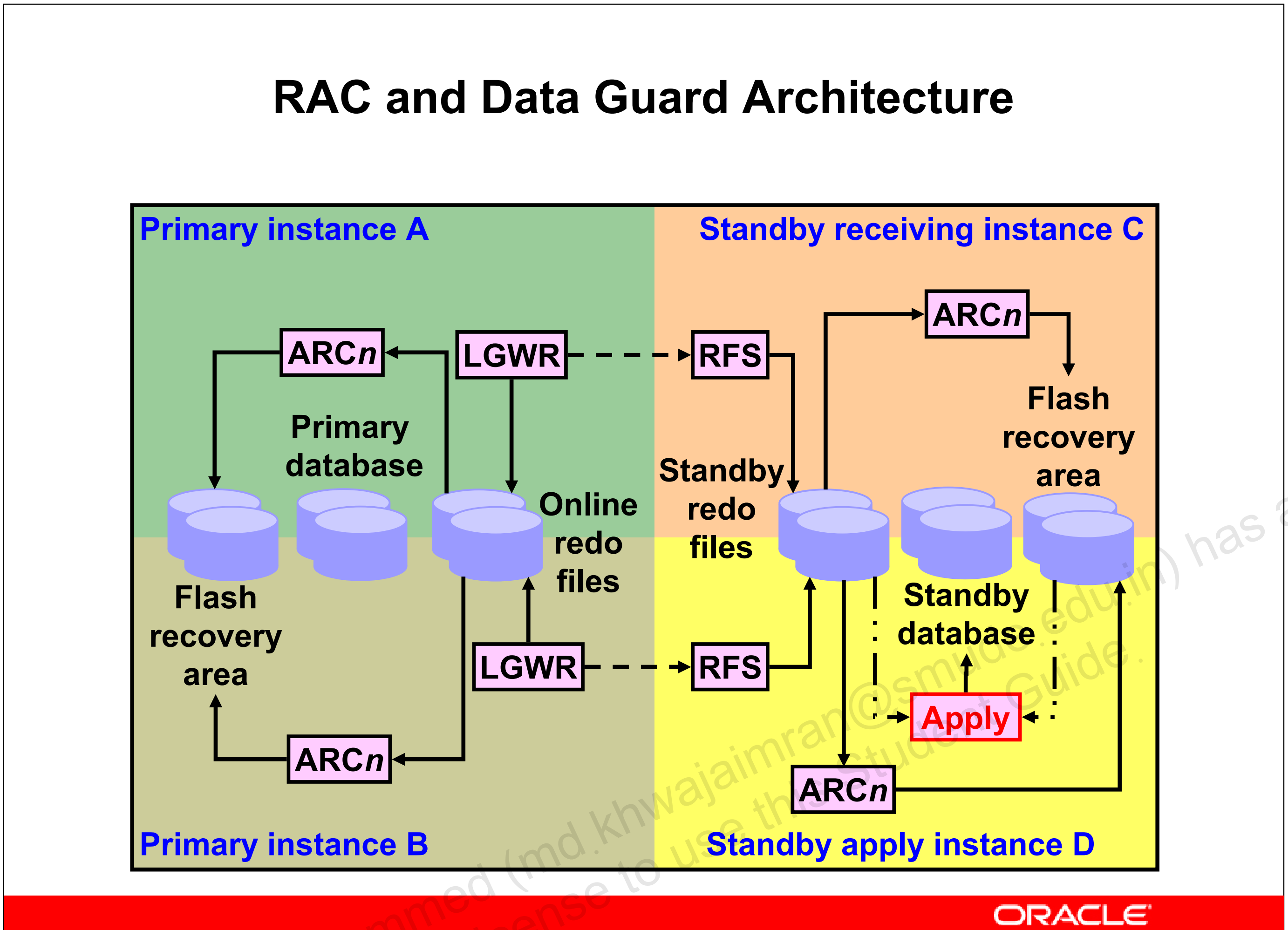
You can configure a standby database to protect a primary database in a RAC environment. Basically, all kinds of combinations are supported. For example, it is possible to have your primary database running under RAC and your standby database running as a single-instance database. It is also possible 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 release 11*g*, 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). For the latest capabilities and restrictions, see Metalink note 413484.1 at https://metalink.oracle.com/.



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.**

Data Guard Broker (DGB) and Oracle Clusterware (OC) 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

Fast-Start Failover: Overview

Fast-Start Failover is an Oracle Data Guard 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 10*g* Grid Control. There are three essential participants in a Fast-Start Failover configuration:

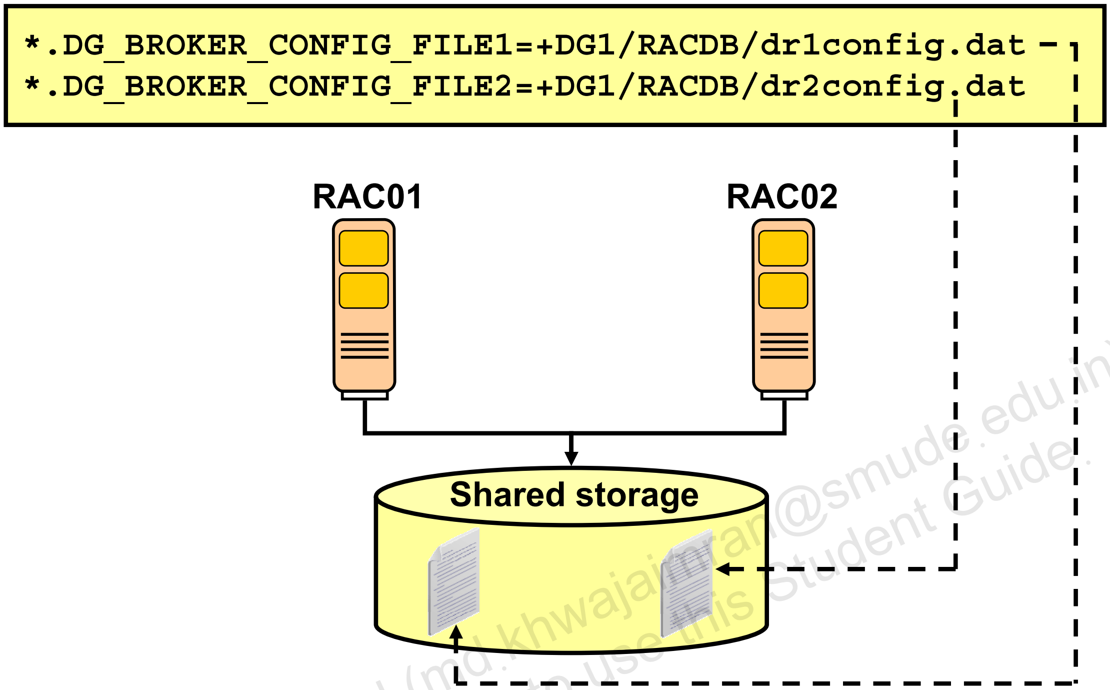
- The primary database, which can be a RAC database
- A target standby database, which becomes the new primary database following a fast-start failover.
- The Fast-Start Failover Observer, which is a separate process incorporated into the DGMGRL client that continuously monitors the primary database 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



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.

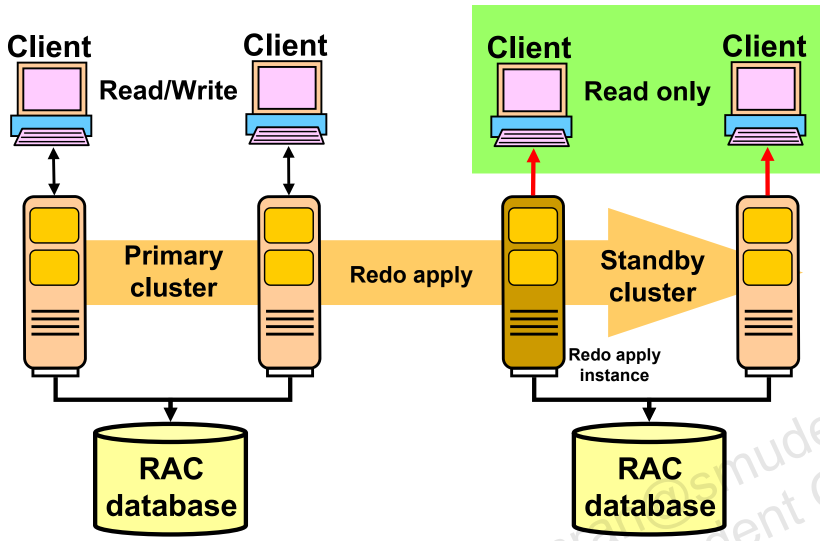
When using a RAC environment, the DGB configuration files must be shared by all instances of the same database. You can override the default path name and file name by setting the following initialization parameters for that database: `DG_BROKER_CONFIG_FILE1`, `DG_BROKER_CONFIG_FILE2`.

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 `RACDB` in `DG1`.

Real-Time Query Physical Standby Database



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 11*g*, 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.
- Enables queries to return 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 if the primary database fails

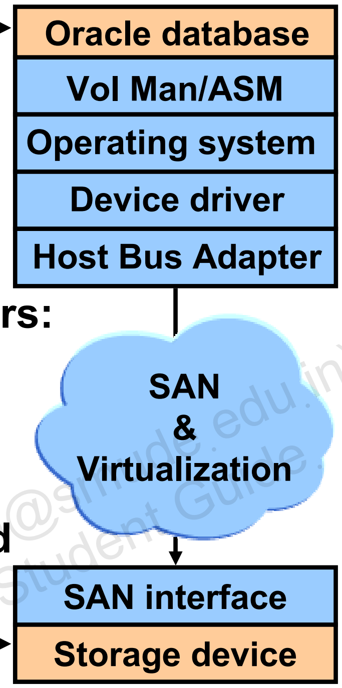
Note: The `COMPATIBLE` parameter must be set to 11.0.0 on both the primary and physical standby databases.

Hardware Assisted Resilient Data

Blocks validated and protection information added to blocks
`[DB_BLOCK_CHECKSUM=TRUE]`

- **Prevents corruption introduced in I/O path**
- **Is supported by major storage vendors:**
  - **EMC, Fujitsu, Hitachi, HP, NEC,**
  - **Network Appliance**
  - **Sun Microsystems**
- **All file types and block sizes checked**

Protection information validated by storage device when enabled
`symchksum -type Oracle enable`



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 can no longer provide any checking that the block being written is still correct.

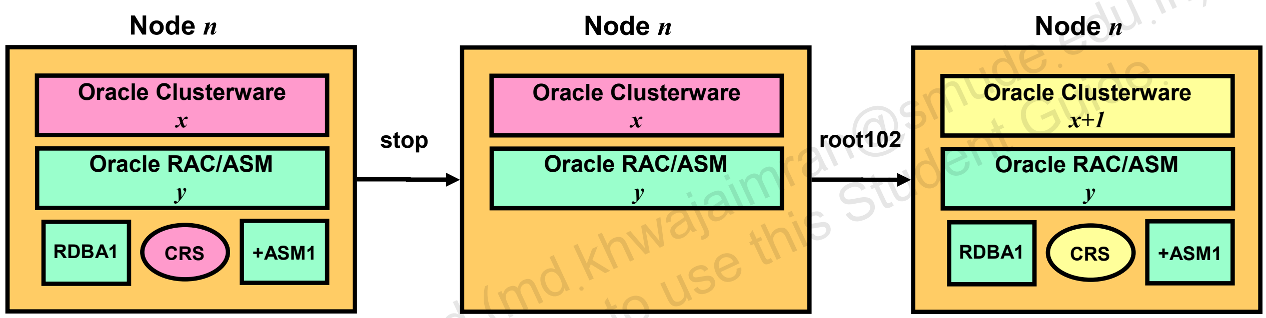
With disk technologies expanding in complexity, and with configurations such as Storage Area Networks (SANs) becoming more popular, the number of layers between the host processor and the physical spindle continues to increase. With more layers, the chance of any problem increases. With the HARD initiative, 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.

Oracle Clusterware Rolling Upgrade

1. `unzip p….zip`
2. `runInstaller` from `Disk1` directory:
   – Choose your Oracle Clusterware home installation.
   – Choose all nodes.
   – Install.
3. Repeat on each node, *one after the other*:
   – `crsctl stop crs`
   – <crs home>/install/root….sh



Oracle Clusterware Rolling Upgrade

All upgrades to Oracle Clusterware can be performed in a rolling fashion. The slide illustrates how to do this.

Once you installed the new software, you need to execute the two commands described in step 3 (in sequence, one node at a time). The diagram shows you the configuration evolution after stopping and running the corresponding `root….sh` script.

Clustered ASM Rolling Upgrade

1. Upgrade Oracle Clusterware
2. From one ASM instance:

```
ALTER SYSTEM START ROLLING MIGRATION TO 11.2.0.0.0;
```

3. Repeat on each node, one after the other:
   a. Shut down database instance.
   b. Shut down ASM instance.
   c. Upgrade ASM software using OUI.
   d. Restart Oracle Clusterware and all corresponding resources.
4. From one ASM instance:

```
ALTER SYSTEM STOP ROLLING MIGRATION;
```


Clustered ASM Rolling Upgrade

All ASM upgrades starting with Oracle Database 11*g* (and later releases) can be performed in a rolling fashion. ASM rolling upgrades enable you to independently upgrade or patch clustered ASM nodes without affecting database availability. The slide describes how to perform a hypothetical ASM rolling upgrade from 11.1.0.6 to 11.2.0.0.

To perform a rolling upgrade, your environment must be prepared. If you are using Oracle Clusterware, your Oracle Clusterware must be fully upgraded to the next patch or release version before you start the ASM rolling upgrade.

Before you patch or upgrade the ASM software on a node, you must place the ASM cluster into rolling upgrade mode. This enables you to begin an upgrade and operate your environment in multiversion software mode. This is illustrated by step 2. You can run this statement from any ASM instance part of your cluster. The instance from which you run this statement verifies whether the value that you specified for the target version is compatible with the current installed version of your software. When the upgrade begins, the behavior of the clustered ASM environment changes, and only the following operations are permitted on the ASM instance:

- Disk group mount and dismount
- Database file open, close, resize, and delete
- Limited access to fixed views and fixed packages

Oracle disables all global views when a clustered ASM environment is in rolling upgrade mode.

Clustered ASM Rolling Upgrade (continued)

You can use the following SQL function to query the state of a clustered ASM environment:

```
SELECT SYS_CONTEXT('sys_cluster_properties', 'cluster_state') FROM DUAL;
```

After the rolling upgrade has started, you can do the following on each node, one after the other:

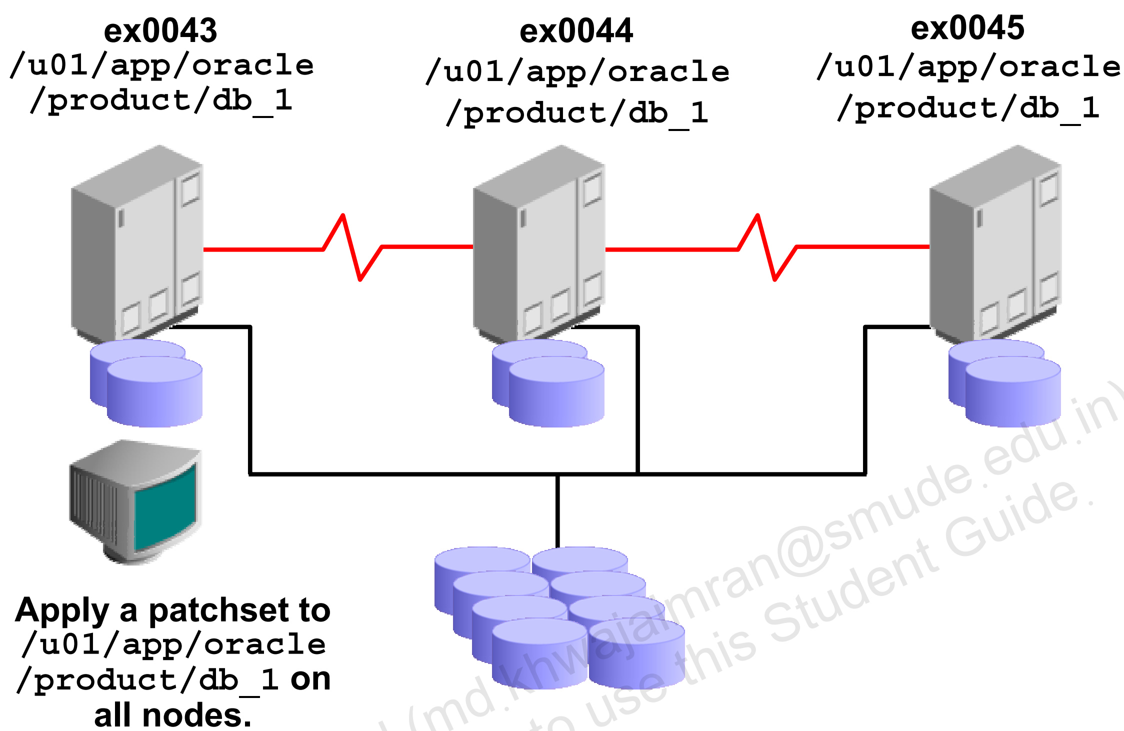
- Shut down each ASM instance.
- Perform the software upgrade using OUI.
- On node start up, the updated ASM instance can rejoin the cluster.

When you have migrated all of the nodes in your clustered ASM environment to the latest software version, you can end the rolling upgrade mode from one ASM instance as shown in step 4 on the slide.

If all of the instances in a clustered ASM environment stop running, the restarted instance will not be in rolling upgrade mode when any of the ASM instances restart. To perform the upgrade after your instances restart, you must rerun the commands to restart the rolling upgrade operation.

Note: If a disk goes offline when the ASM instance is in rolling upgrade mode, the disk remains offline until the rolling upgrade ends. Also, the timer for dropping the disk is stopped until the ASM cluster is out of rolling upgrade mode.

Patches and the RAC Environment



Patches and the RAC Environment

Applying patches to your RAC installation is a simple process with the OUI. The OUI can keep track of multiple `ORACLE_HOME` deployments. This intelligence prevents potentially destructive or conflicting patchsets from being applied.

In the example in the slide, a patchset is applied to the `/u01/app/oracle/product/db_1` Oracle Home on all the three nodes of your cluster database. Although you execute the installation on `ex0043`, you can choose any of the nodes to perform this task. The steps that you must perform to add a patchset through the OUI are essentially the same as those to install a new release. You must change directory to `$ORACLE_HOME/bin`. After starting the OUI, perform the following steps:

1. Select "Installation from a stage location," and enter the appropriate patchset source on the Welcome screen.
2. Select the nodes on the Node Selection screen, where you need to add the patch, and ensure that they are all available. In this example, this should be all three of the nodes because `/u01/app/oracle/product/db_1` is installed on all of them.
3. Check the Summary screen to confirm that space requirements are met for each node.
4. Continue with the installation and monitor the progress as usual.

The OUI automatically manages the installation progress, including the copying of files to remote nodes, just as it does with the Oracle Clusterware and database binary installations.

Inventory List Locks

- The OUI employs a timed lock on the inventory list stored on a node.
- The lock prevents an installation from changing a list being used concurrently by another installation.
- If a conflict is detected, the second installation is suspended and the following message appears:

```
"Unable to acquire a writer lock on nodes ex0044.
Restart the install after verifying that there is
no OUI session on any of the selected nodes."
```


Inventory List Locks

One of the improvements in the OUI is that it prevents potentially destructive concurrent installations. The mechanism involves a timed lock on the inventory list stored on a node. When you start multiple concurrent installations, the OUI displays an error message that is similar to the one shown in the slide. You must cancel the installation and wait until the conflicting installation completes, before retrying it.

Although this mechanism works with all types of installations, see how it can function if you attempt concurrent patchset installations in the sample cluster. Use the same configuration as in the previous scenario for your starting point.

Assume that you start a patchset installation on `ex0043` to update `ORACLE_HOME2` on nodes `ex0044` and `ex0045`. While this is still running, you start another patchset installation on `ex0044` to update `ORACLE_HOME3` on that node. Will these installations succeed? As long as there are no other problems, such as a down node or interconnect, these processes have no conflicts with each other and should succeed. However, what if you start your patchset installation on `ex0044` to update `ORACLE_HOME3` and then start a concurrent patchset installation for `ORACLE_HOME2` (using either `ex0044` or `ex0043`) on all nodes where this Oracle Home is installed? In this case, the second installation should fail with the error shown because the inventory on `ex0044` is already locked by the patchset installation for `ORACLE_HOME3`.

OPatch Support for RAC: Overview

- OPatch supports four different methods:
  - All-node patch: `Stop all/Patch all/Start all`
  - Minimize down time: `Stop/Patch all but one, Stop last, Start all down, Patch last/Start last`
  - Rolling patch: `Stop/Patch/Start` one at a time
  - Local patch: `Stop/Patch/Start` only one
- How does OPatch select which method to use:

```
If (users specify -local | -local_node)
      patching mechanism = Local
else if (users specify -minimize_downtime)
     patching mechanism = Min. Downtime
     else if (patch is a rolling patch)
          patching mechanism = Rolling
          else  patching mechanism = All-node
```


Patch Upgrade Using RAC: Overview

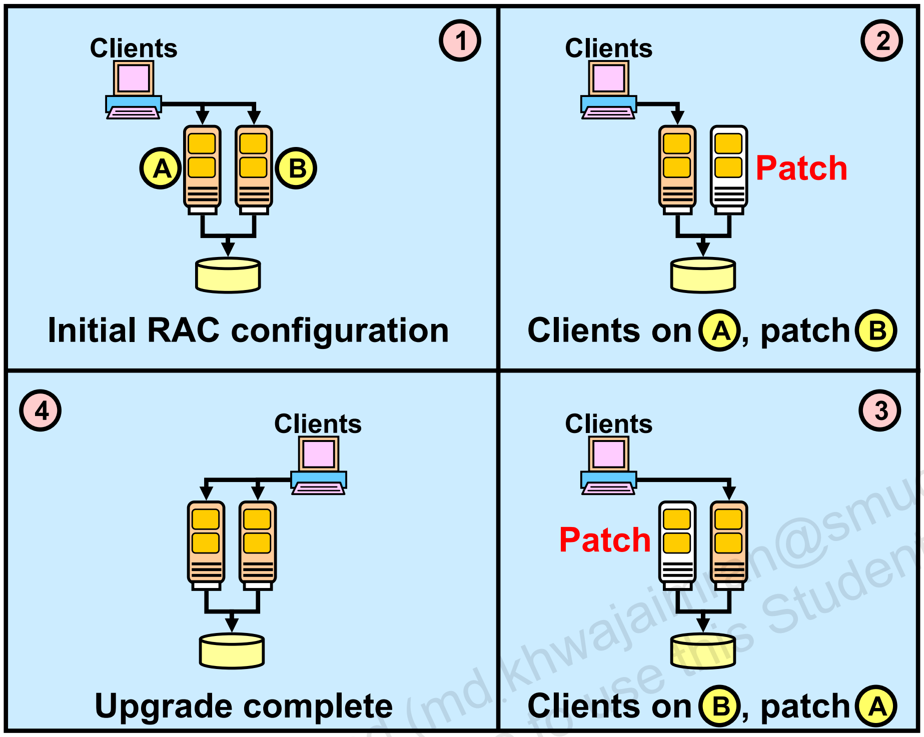
OPatch supports four different patch methods on a RAC environment:

- Patching RAC as a single instance (all-node patch): In this mode, OPatch applies the patch to the local node first, then propagates the patch to all other nodes, and finally updates the inventory. All instances are down during the entire patching process.
- Patching RAC using a minimum down-time strategy (min. downtime patch): In this mode, OPatch patches the local node, asks users for a subset of nodes, which will be the first nodes to be patched. After the initial subset of nodes are patched, OPatch propagates the patch to the other nodes and finally updates the inventory. The down time happens between the shutdown of the second subset of nodes and the startup of the initial subset of nodes patched.
- Patching RAC using a rolling strategy (rolling patch): With this method, there is no down time. Each node is patched and brought up while all the other nodes are up and running, resulting in no disruption of the system.
- The OPatch strategies discussed above presume that all nodes are patched at the same time. Additionally, each node can be patched individually, at different times, using the `-local`, or `-local_node` key words, which patch only the local node, or the remote node.

When executing the `opatch apply` command, the slide shows you which method is used.

Note: Currently, OPatch treats a shared file system as a single-instance patch. This means that OPatch cannot take advantage of a rolling patch in this case because all nodes must be down.

Rolling Patch Upgrade Using RAC



Oracle patch upgrades

Operating system upgrades

Hardware upgrades

Rolling Patch Upgrade Using RAC

This is supported *only* for single patches that are marked as rolling upgrade compatible.

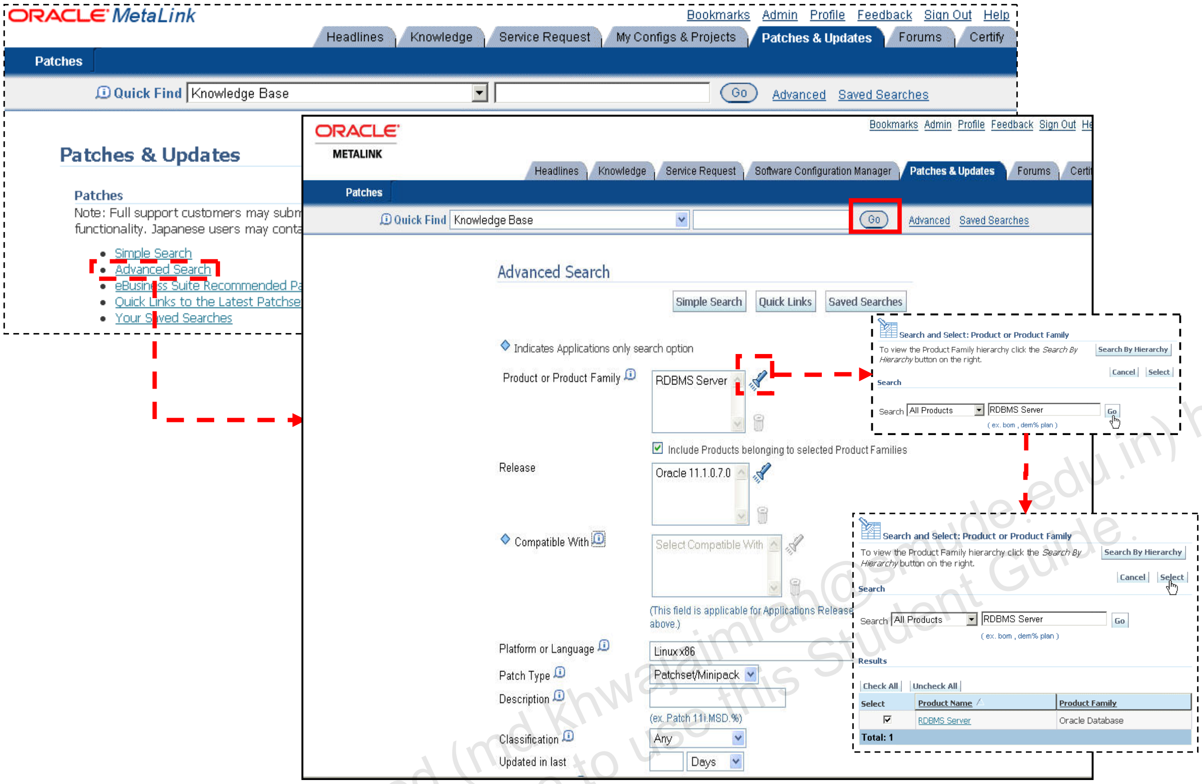
Rolling RAC patching allows the interoperation of a patched node and an unpatched node simultaneously. This means only one node is out of commission while it is patched.

Using the `OPATCH` tool to apply a rolling RAC patch, you are prompted to stop the instances on the node to be patched. First, the local node is patched, and then you are asked for the next node to patch from a list. As each node is patched, the user is prompted when it is safe to restart the patched node. The cycle of prompting for a node, of stopping the instances on the node, of patching the node, and of restarting the instances continues until you stop the cycle or until all nodes are patched. After you download the patch to your node, you need to unzip it before you can apply it. You can determine whether the patch is flagged as rolling upgradable by checking the *Patch_number*`/etc/config/inventory` file. Near the end of that file, you must see the following mark: <online_rac_installable>true</online_rac_installable>

It is important to stress that although rolling patch upgrade allows you to test the patch before propagating it to the other nodes, it is preferable to test patches in a test environment rather than directly on your production system.

Note: Some components cannot be changed one node at a time. The classic example is the data dictionary. Because there is only a single data dictionary, all instances need to be shut down. In these cases, Oracle Data Guard and physical standby databases are the recommended solutions. Of course, using Online Patching is the recommended solution for avoiding downtime when an online patch is available.

Download and Install Patch Updates



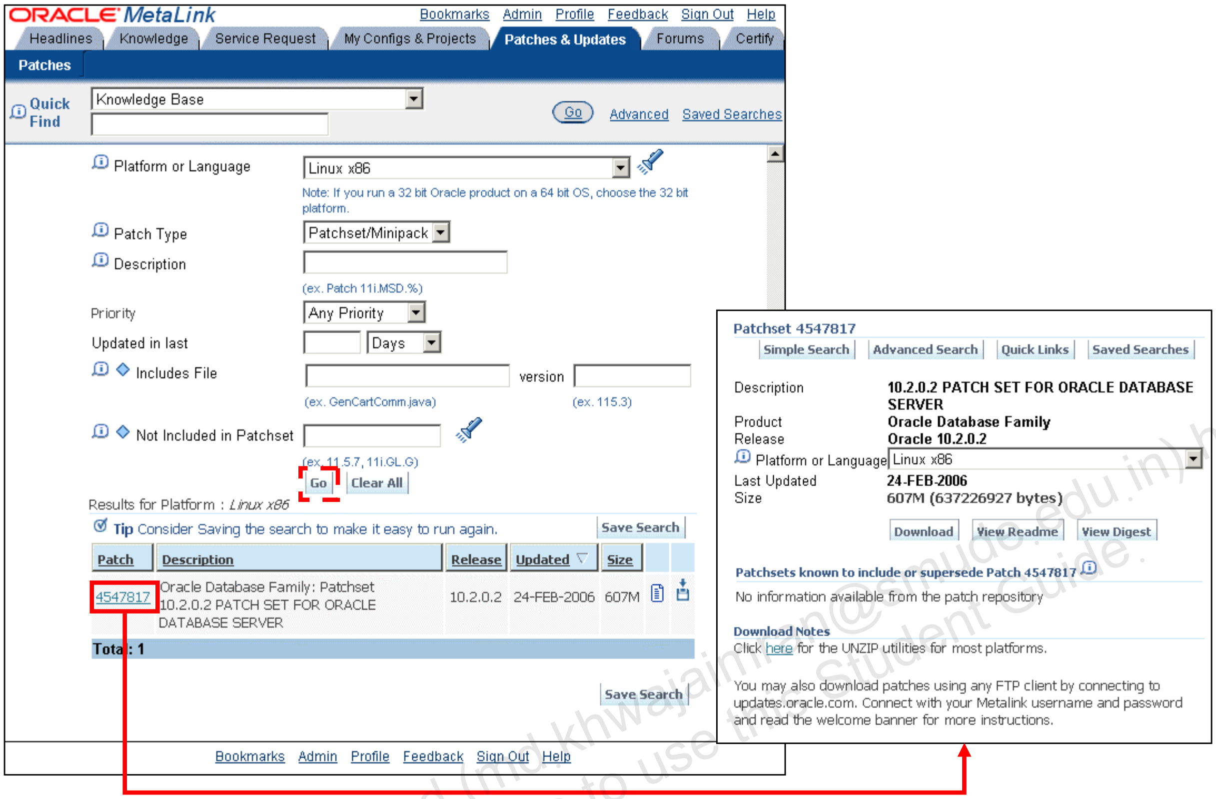
Download and Install Patch Updates

Refer to the OracleMetaLink Web site for required patch updates for your installation.

To download the required patch updates:

1. Use a Web browser to log in to the OracleMetaLink Web site: *http://metalink.oracle.com*.
2. On the main OracleMetaLink page, click Patches and Updates.
3. On the Patches & Updates page, click Advanced Search.
4. On the Advanced Search page, click the search icon next to the Product or Product Family field.
5. In the Search field, enter `RDBMS Server` and click Go. Select RDBMS Server under the Results heading and click Select. `RDBMS Server` appears in the Product or Product Family field. The current release appears in the Release field.

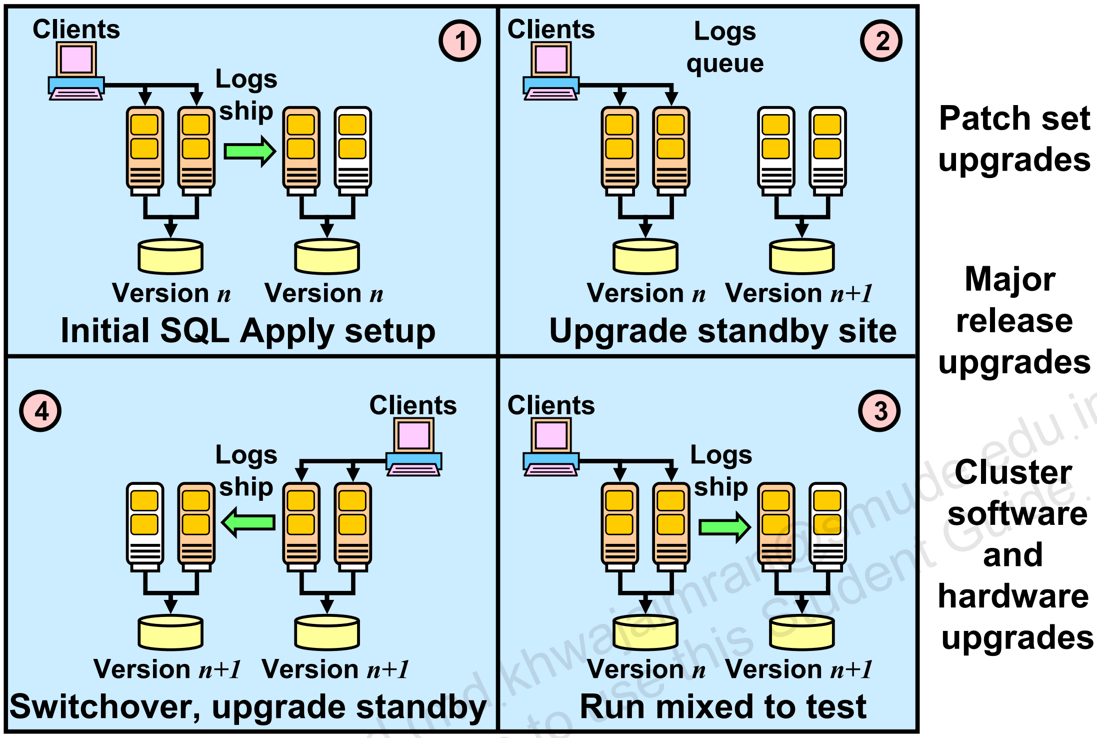
Download and Install Patch Updates



Download and Install Patch Updates (continued)

6. Select your platform from the list in the Platform or Language field, and click Go. Any available patch updates appear under the Results heading.
7. Click the number of the patch that you want to download.
8. On the Patchset page, click View README and read the page that appears. The README page contains information about the patchset and how to apply the patches to your installation.
9. Return to the Patchset page, click Download, and save the file on your system.
10. Use the `unzip` utility provided with Oracle Database 11*g* to uncompress the Oracle patch updates that you downloaded from OracleMetaLink. The unzip utility is located in the `$ORACLE_HOME/bin` directory.

Rolling Release Upgrade Using SQL Apply



Rolling Release Upgrade Using SQL Apply

Beginning with Oracle Database 10.1.0.3 and continuing with later versions, a logical standby database can operate at a later Oracle release than on the primary when using SQL Apply to execute a rolling database upgrade. For example, using SQL Apply and logical standby databases, you are able to upgrade the Oracle database software from patchset release 10.2.0.*n* to the next database 10.2.0.(*n*+1) patchset release. Step 1 in the slide shows the Data Guard configuration before the upgrade begins, with the primary and logical standby databases both running the same Oracle software version. In step 2, you stop SQL Apply and upgrade the Oracle database software on the logical standby database to version *n*+1. During the upgrade, redo data accumulates on the primary system. In step 3, you restart SQL Apply, and the redo data that was accumulating on the primary system is automatically transmitted and applied on the newly upgraded logical standby database. The Data Guard configuration can run the mixed versions for an arbitrary period. In step 4, you perform a switchover. Then you activate the user applications and services on the new primary database. Before you can enable SQL Apply again, you need to upgrade the new standby site. This is because the new standby site does not understand new redo information. Finally, you raise the compatibility level on each database.

Note: SQL Apply does not support all data types. This can prevent you from using this method. A possible alternative is to use Oracle Streams in this case.

Database High Availability: Best Practices

Use SPFILE.	Create two or more control files.	Set CONTROL_FILE_RECORD_KEEP_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.

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.

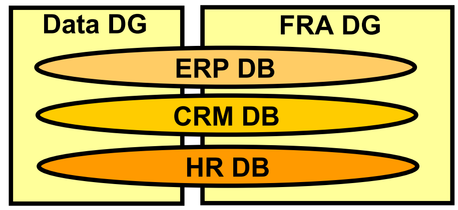
These practices affect the performance, availability, and mean time to recover (MTTR) of your system. Some of these practices may reduce performance, but they are necessary to reduce or avoid outages. The minimal performance impact is outweighed by the reduced risk of corruption or the performance improvement 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

How Many ASM Disk Groups per Database

Most of the time, only two disk groups are enough to share the storage between multiple databases.

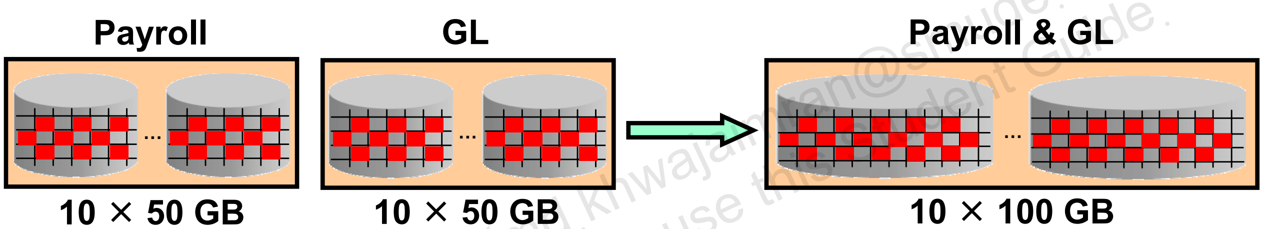
That way you can maximize the number of Logical Unit Numbers (LUNs) used as ASM disks, which gives you the best performance, especially if these LUNs are carved on the outer edge of your disks.

Using a second disk group allows you to have a backup of your data by using it as your common flash 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.

Database Storage Consolidation

- Shared storage across several databases:
  - RAC and single-instance databases can use the same ASM instance.
- Benefits:
  - Simplified and centralized management
  - Higher storage utilization
  - Higher performance



Database Storage Consolidation

In Oracle Database 10*g* Release 2 and later, Oracle Clusterware does not require an Oracle Real Application Clusters license. Oracle Clusterware is now available with ASM and single-instance Oracle Database 11*g* allowing support for a shared clustered pool of storage for RAC and single-instance Oracle databases.

This allows you to optimize your storage utilization by eliminating wasted, over-provisioned storage. This is illustrated in the slide, where instead of having various pools of disks used for different databases, you consolidate all that in one single pool shared by all your databases.

By doing this, you can reduce the number of LUNs to manage by increasing their sizes, which gives you a higher storage utilization as well as a higher performance.

Note: RAC and single-instance databases could not be managed by the same ASM instance in Oracle Database 10*g* Release 1.

Which RAID Configuration for Best 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)

ASM leverages hardware RAID.

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, it 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 RAID 1 or RAID 5?

RAID 1 (Mirroring)
- Recommended by Oracle
- Most demanding applications

Pros:
- Best redundancy
- Best performance
- Low recovery overhead

Cons:
- Requires higher capacity

RAID 5 (Parity)
- DSS and moderate OLTP

Pros:
- Requires less capacity

Cons:
- Less redundancy
- Less performance
- High recovery overhead

Should You Use RAID 1 or RAID 5?

RAID 1 is a mirroring technique. Mirroring involves taking all writes issued to a given disk and duplicating the write to another disk. In this way, if there is a failure of the first disk, the second disk, or mirror, can take over without any data loss.

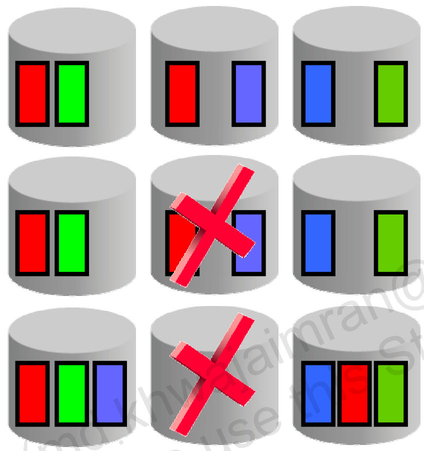
The goal of the RAID-5 design is to provide a reliable, high-performance array of disks with the minimum amount of redundant hardware. RAID 5 is based on the use of parity protection across several drives in order to provide protection against disk failure. The RAID-5 configuration is essentially a striped configuration, like RAID 0, with an additional disk added to cater to the additional storage needed for the parity information. With the data striped across the drives in this way, the read performance of RAID 5 is comparable to that of RAID 0. RAID-5 writes, on the other hand, are almost legendary for their poor performance.

The slide lists the pros and cons of using both techniques, and although Oracle recommends using RAID 1, you need to take into account that you have to double the number of your disks to store the same amount of data. The general rule of thumb is to deploy RAID 5 where cost of storage is critical and performance is not the primary goal, and for applications with primary read operations such as data warehouse applications. The Flash Recovery Area disk group can be another good use of RAID 5, where the storage capacity requirement is the highest and predominantly sequential I/O.

Note: The ORION tools (http://www.oracle.com/technology/software/index.html#util) can be used to test and determine the pros and cons of storage arrays for your application.

Should You Use ASM Mirroring Protection?

- Best choice for low-cost storage
- Enables extended clustering solutions
- No hardware mirroring



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.

Note: For more information about the Oracle Resilient Low-cost Storage Initiative, see the Web site at: http://www.oracle.com/technology/deploy/availability/htdocs/lowcoststorage.html.

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.

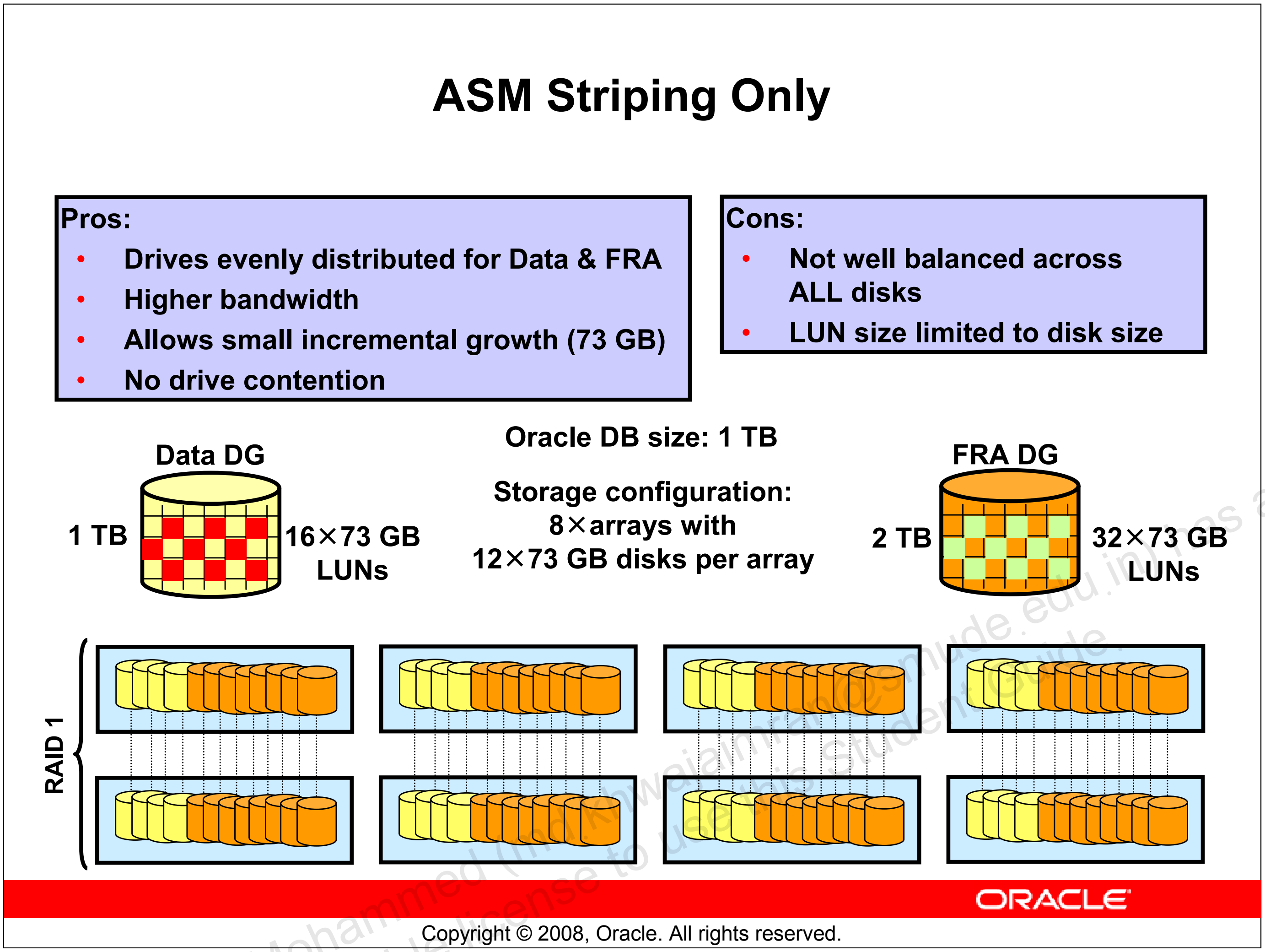
What Type of Striping Works Best?

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 it is recommended to not use any striping at all.



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 Flash Recovery Area disk group is assigned 32 LUNs of 73 GB each.

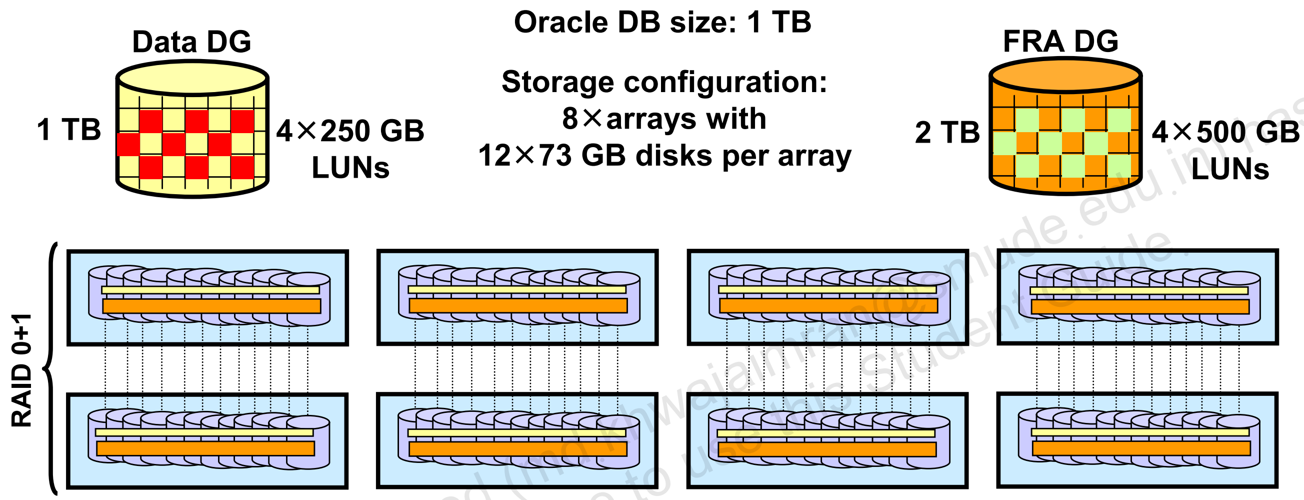
This configuration enables you to evenly distribute disks for your data and backups, achieving good performance and allowing you to manage your storage in small incremental chunks.

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”



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 disk. 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.

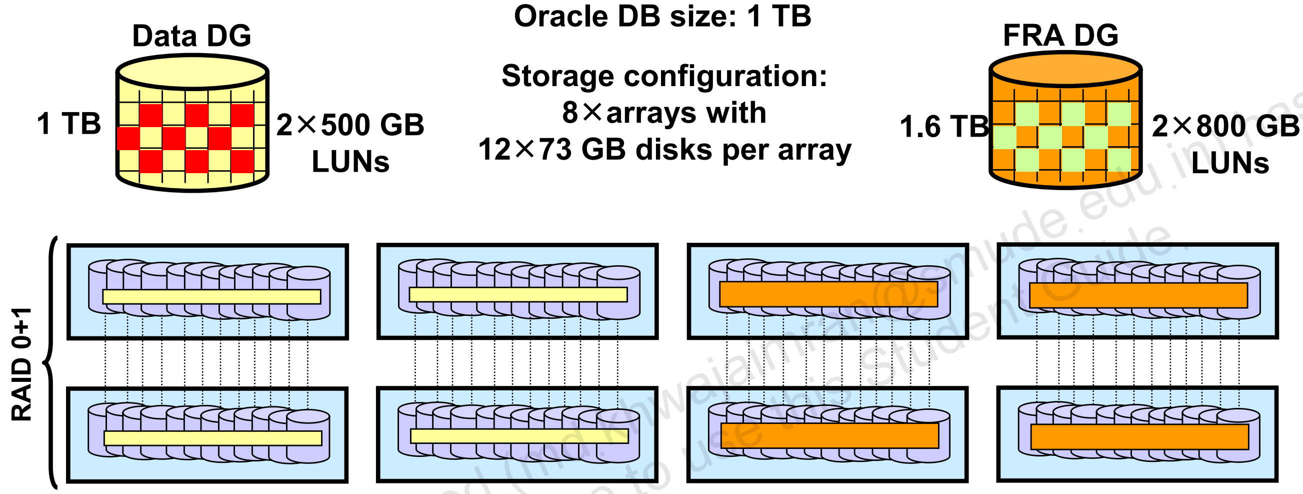
However, you must manipulate your storage in much larger chunks than in the previous configuration.

Note: The hardware stripe size you choose is also very important because you want 1 MB alignment as much as possible to keep in synch 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



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 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.

Compared to the previous slide, you use bigger LUNs for both the Data disk group and the Flash 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.

It Is Real Simple

- 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 Database 11*g* and ASM do the rest!

ORACLE

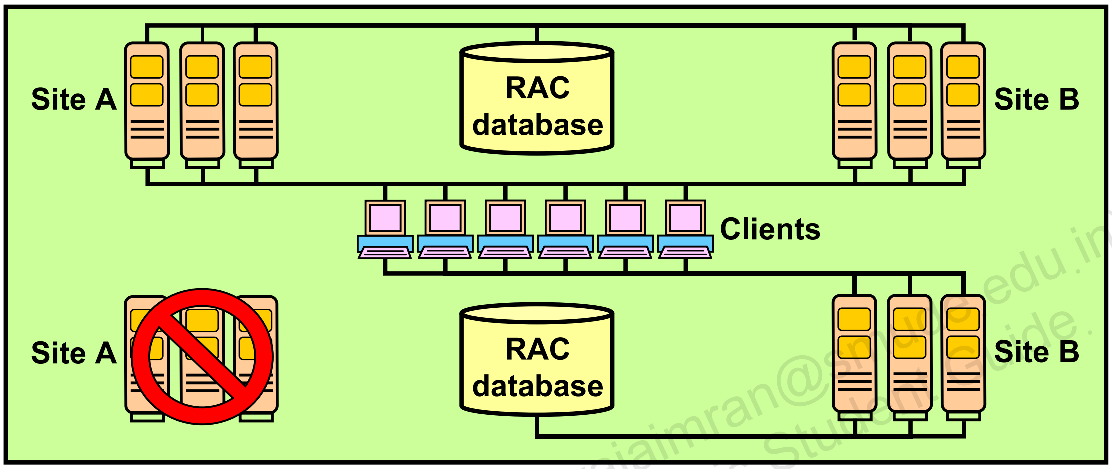
It Is Real Simple

Use Oracle Database 11*g* 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**

Extended RAC: Overview

Typically, RAC databases share a single set of storage and are located on servers in the same data center.

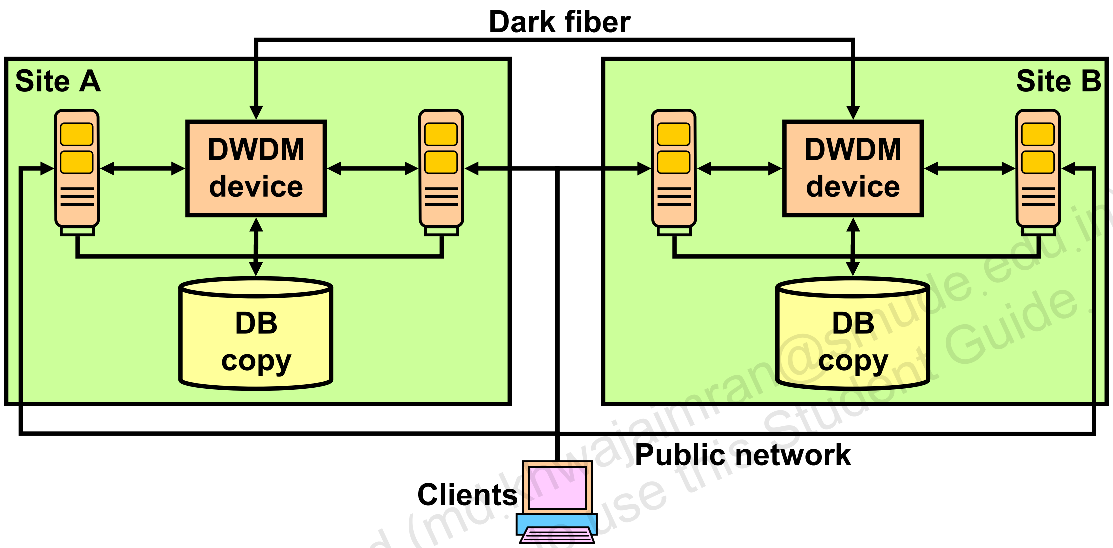
With extended RAC, you can use disk mirroring and Dense Wavelength Division Multiplexing (DWDM) equipment to extend the reach of the cluster. This configuration allows two data centers, separated by up to 100 kilometers, to share the same RAC database with multiple RAC instances spread across the two 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.



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.

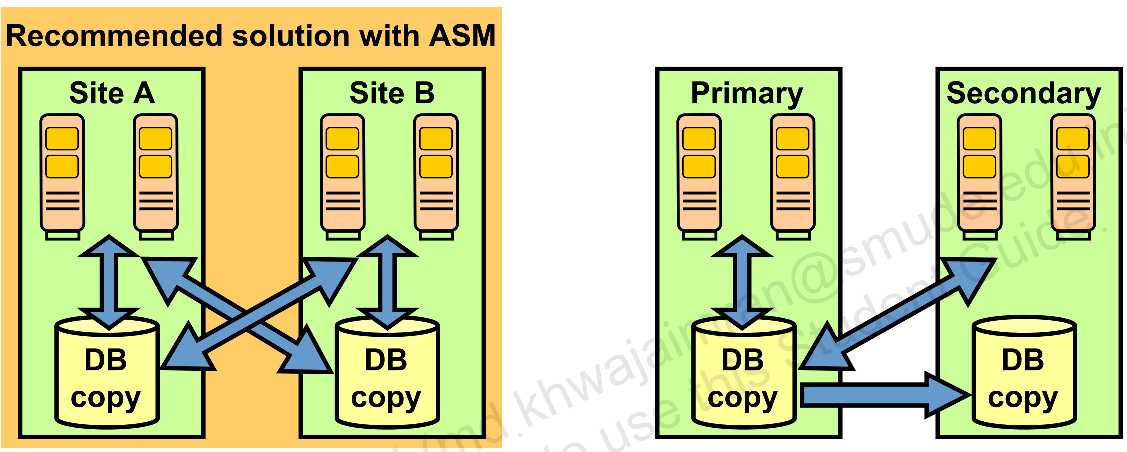
As shown in the slide, each site should have its own DWDM device connected together by a dark fiber optical strand. All traffic between the two sites is sent through the DWDM and carried on dark fiber. This includes mirrored disk writes, network and heartbeat traffic, and memory-to-memory data passage. 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



Extended RAC Disk Mirroring

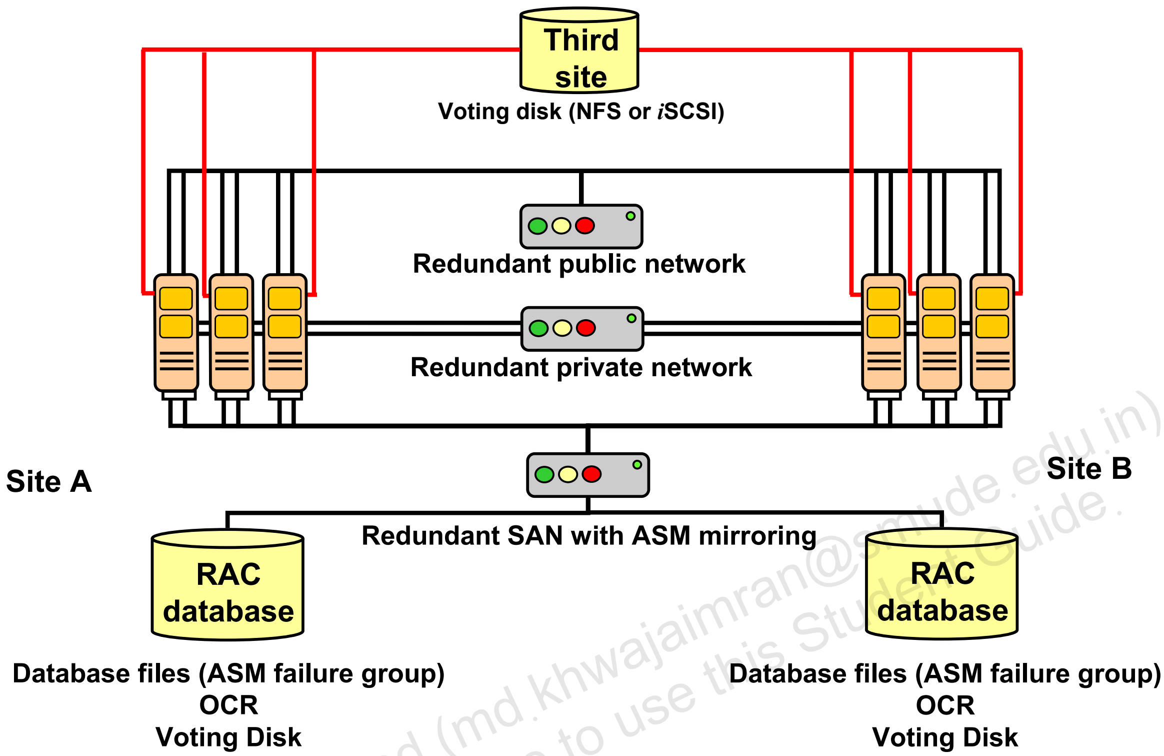
Although there is only one RAC database, each data center has its own set of storage that is synchronously mirrored using either a cluster-aware, host-based Logical Volume Manager (LVM) solution, such as SLVM with MirrorDiskUX, or an array-based mirroring solution, such as EMC SRDF.

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, it is critical to design the cluster in a manner that ensures that the cluster can achieve quorum after a site failure. For more information, refer to the *Oracle Technology Network* site.

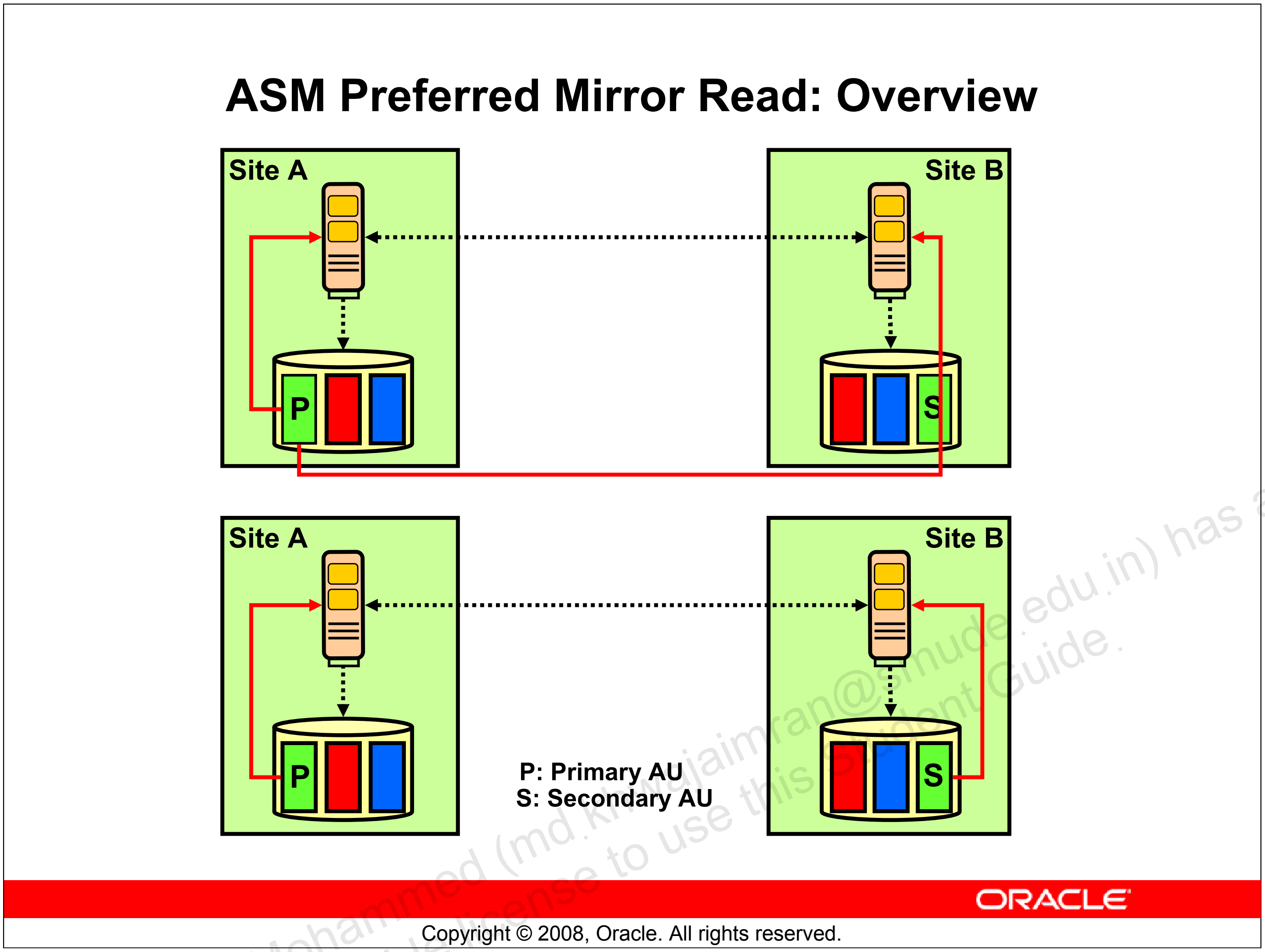
Achieving Quorum with Extended RAC



Achieving Quorum with Extended RAC

A node must be able to access strictly more than half of the voting disks at any time; otherwise, 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 that is visible from both sites. This situation is illustrated in the slide. The goal is for each site to 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.



ASM Preferred Mirror Read: Overview

When you configure ASM failure groups in Oracle Database 11*g*, ASM always reads the primary copy of a mirrored extent. It may be more efficient for a node to read from a failure group extent that is closest to the node, even if it is a secondary extent. This is especially true in extended cluster configurations (when nodes are spread across several sites) where reading from a local copy of an extent provides improved performance.

With Oracle Database 11*g*, you can do this by configuring the preferred mirror read by using the new `ASM_PREFERRED_READ_FAILURE_GROUPS` initialization parameter to specify a list of preferred mirror read names. The disks in those failure groups become the preferred read disks. Thus, every node can read from its local disks. This results in higher efficiency and performance as well as reduced network traffic. The setting for this parameter is instance specific.

ASM Preferred Mirror Read: Setup

Setup

```
ASM_PREFERRED_READ_FAILURE_GROUPS=DATA.SITEA
```
On first instance

```
ASM_PREFERRED_READ_FAILURE_GROUPS=DATA.SITEB
```
On second instance

Monitor

```
SELECT preferred_read FROM v$asm_disk;
```

```
SELECT * FROM v$asm_disk_iostat;
```

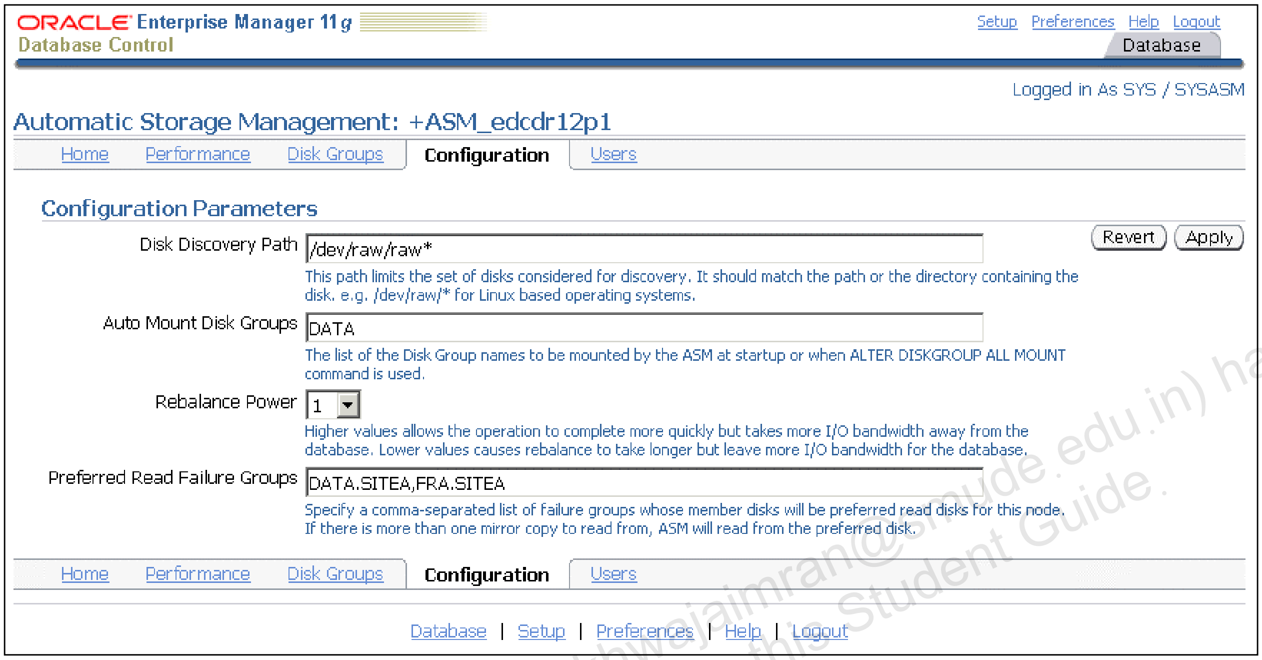

ASM Preferred Mirror Read: Setup

To configure this feature, set the new ASM_PREFERRED_READ_FAILURE_GROUPS initialization parameter. This parameter is a multivalued parameter and should contain a string with a list of failure group names separated by commas. Each failure group name specified should be prefixed with its disk group name and a "." character. This parameter is dynamic and can be modified using the ALTER SYSTEM command at any time. An example is shown in the slide. However, this initialization parameter is valid only for ASM instances. With the extended cluster, the failure groups specified in this parameter should contain only those disks that are local to the corresponding instance.

The new column PREFERRED_READ has been added to the V$ASM_DISK view. Its format is a single character. If the disk group to which the disk belongs pertains to a preferred read failure group, the value of this column is Y.

To identify specific performance issues with the ASM preferred read failure groups, use the V$ASM_DISK_IOSTAT view. This view displays the disk input/output (I/O) statistics for each ASM client. If this view is queried from a database instance, only the rows for this instance are shown.

Enterprise Manager ASM Configuration Page



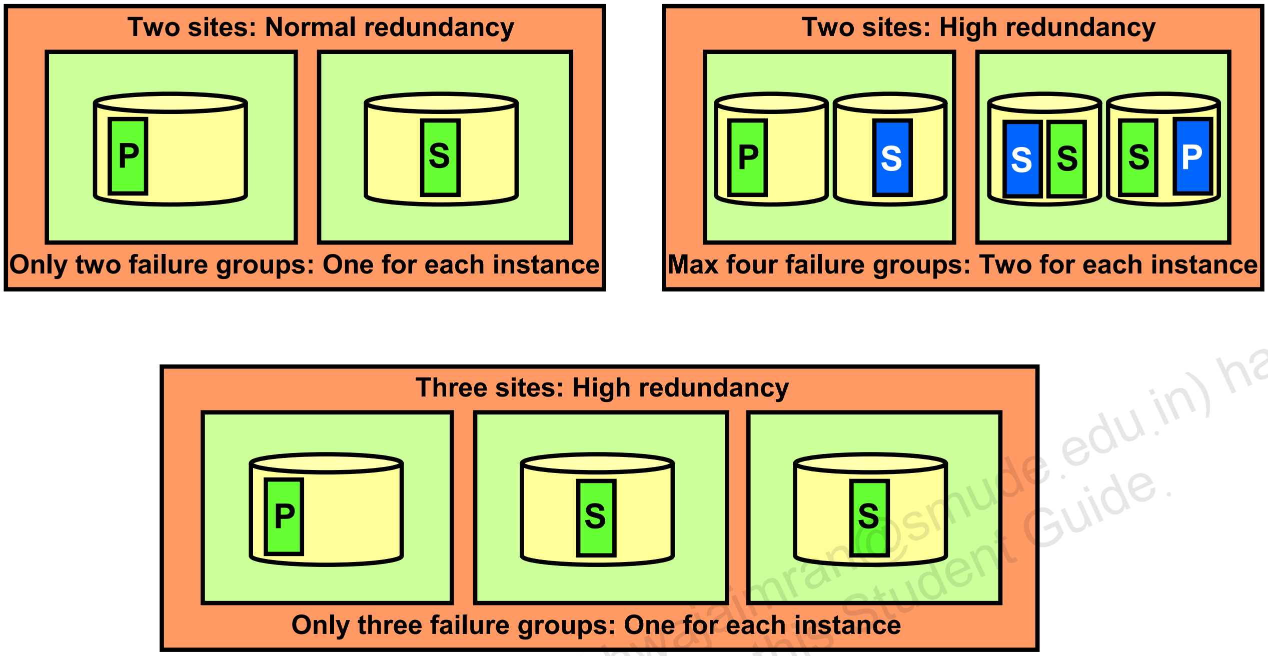
ORACLE

Enterprise Manager ASM Configuration Page

You can specify a set of disks as preferred disks for each ASM instance by using Enterprise Manager. The preferred read attributes are instance specific. In Oracle Database 11*g*, the Preferred Read Failure Groups field (`asm_preferred_read_failure_group`) is added to the configuration page.

This parameter takes effect only before the disk group is mounted or when the disk group is created. It applies only to newly opened files or to a newly loaded extent map for a file.

ASM Preferred Mirror Read: Best Practice



ASM Preferred Mirror Read: Best Practice

In practice, there are only a limited number of good disk group configurations in an extended cluster. A good configuration takes into account both performance and availability of a disk group in an extended cluster. Here are some possible examples:

For a two-site extended cluster, a normal redundancy disk group should have only two failure groups; all disks local to one site should belong to the same failure group. Also, no more than one failure group should be specified as a preferred read failure group by each instance. If there are more than two failure groups, ASM may not mirror a virtual extent across both sites. Furthermore, if the site with more than two failure groups were to go down, it would take the disk group down as well. If the disk group to be created is a high-redundancy disk group, at most two failure groups should be created on each site with its local disks, with both local failure groups specified as preferred read failure groups for the local instance.

For a three-site extended cluster, a high redundancy disk group with three failure groups should be used. In this way, ASM can guarantee that each virtual extent has a mirror copy local to each site and that the disk group is protected against a catastrophic disaster on any of the three sites.

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

Additional Data Guard Benefits

Data Guard provides a greater disaster protection:

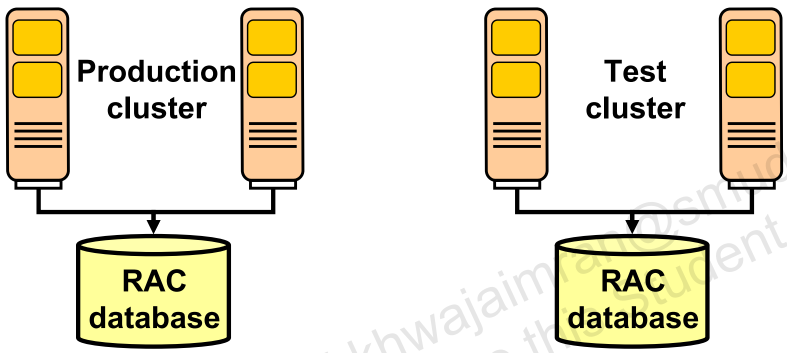
- Distance over 100 kilometers without performance hit
- Additional protection against corruptions because it uses a separate database
- Optional delay to protect against user errors

Data Guard also provides better planned maintenance capabilities by supporting full rolling upgrades.

Also, if you cannot handle the costs of a DWDM network, Data Guard still works over inexpensive, 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.



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.

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
- Decide on the best ASM configuration to use
- Patch your RAC system in a rolling fashion

Note

There is no practice for this lesson.

Appendix A
Practices and Solutions

Table of Contents

Practices for Lesson 1

Practice 1-1: Oracle Clusterware Installation

For most of these practice exercises, you use various Oracle graphical tools (OUI, DBCA, EMCA, and so on), and telnet/ssh sessions. Your instructor will give you details about your node names and accounts, as well as your database and your cluster name.

Notes:

- When working on these practices, you *must* use the provided values to prevent interference with other students or other classes.
- The solutions provided are not necessarily based on your assigned account. You should thus substitute your designated user, database, instance, host names, and related information as appropriate.

In this practice, you set up user equivalence for the oracle user employing Secure Shell (ssh). The second step entails checking the readiness of the cluster for a Clusterware installation. This is done using the CLUVFY utility. The third and final step of this practice is the actual installation of the Oracle Clusterware software. Install the software in the `/u01/crs11g` directory as the oracle user. The install group should be `oinstall`.

1) Using a telnet session, connect as user `oracle` to your first RAC node. You need to configure secure shell (`ssh`) on both nodes. To do this, navigate to `$HOME/solutions/less01` and execute the `ssh_setup.sh` script. When finished, test the setup by using ssh to log in to your second node. You should *not* be prompted for a password.

```
[oracle@vx0301 lesson1]$ ./ssh_setup.sh
The authenticity of host 'vx0302 (10.216.4.13)' can't be
established.
RSA key fingerprint is
ab:33:1b:a6:89:53:ba:24:77:76:90:e2:7a:1c:79:11.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'vx0302,10.216.4.13' (RSA) to the
list of known hosts.
oracle@vx0302's password:
oracle@vx0302's password:
[oracle@vx0301 lesson1]$ ssh vx0302
Last login: Fri Nov 16 14:35:30 2007 from vx0301.us.oracle.com
[oracle@vx0302 ~]$ exit
[oracle@vx0301 lesson1]$
```

2) Run CLUVFY from `/stage/db` to make sure that the Clusterware minimum requirements are met on both nodes before beginning the installation:

```
[oracle@vx0301 lesson1]$ /stage/db/runcluvfy.sh stage -pre
crsinst -n vx0301,vx0302
```

Practice 1-1: Oracle Clusterware Installation (continued)

```
Performing pre-checks for cluster services setup

Checking node reachability...
Node reachability check passed from node "vx0301".


Checking user equivalence...
User equivalence check passed for user "oracle".

Checking administrative privileges...
User existence check passed for "oracle".
Group existence check passed for "oinstall".
Membership check for user "oracle" in group "oinstall" [as
Primary] passed.

Administrative privileges check passed.

Checking node connectivity...

Node connectivity check passed for subnet "10.216.4.0" with
node(s) vx0302,vx0301.

WARNING:
Make sure IP address "10.216.102.142" is up and is a valid IP
address on node "vx0301".
Node connectivity check passed for subnet "10.196.28.0" with
node(s) vx0302,vx0301.

Interfaces found on subnet "10.216.4.0" that are likely
candidates for VIP:
vx0302 eth0:10.216.4.13
vx0301 eth0:10.216.4.12

Interfaces found on subnet "10.196.28.0" that are likely
candidates for a private interconnect:
vx0302 eth2:10.196.30.23
vx0301 eth2:10.196.30.22


Checking system requirements for 'crs'...
Total memory check passed.
Free disk space check passed.
Swap space check passed.
System architecture check passed.
Kernel version check passed.
Package existence check passed for "make-3.80".
Package existence check passed for "binutils-2.15.92.0.2".
Package existence check passed for "gcc-3.4.5".
Package existence check passed for "libaio-0.3.105".
Package existence check passed for "libaio-devel-0.3.105".
Package existence check passed for "libstdc++-3.4.5".
```

```
Package existence check passed for "elfutils-libelf-devel-
0.97".
Package existence check passed for "sysstat-5.0.5".
Package existence check passed for "libgcc-3.4.5".
Package existence check passed for "libstdc++-devel-3.4.5".
Package existence check passed for "unixODBC-2.2.11".
Package existence check passed for "unixODBC-devel-2.2.11".
Package existence check passed for "glibc-2.3.4-2.19".
Group existence check passed for "dba".
Group existence check passed for "oinstall".
User existence check passed for "nobody".

System requirement passed for 'crs'

Pre-check for cluster services setup was successful on all the
nodes.
```

3) Execute `runInstaller` from the `/stage/db` directory and install Oracle Clusterware.

   a) From a graphical terminal session on your first node, set the value of `ORACLE_BASE=/u01/app/oracle` and execute `/stage/db/runInstaller`.

```
[oracle@vx0301 lesson1]$ export ORACLE_BASE=/u01/app/oracle
[oracle@vx0301 lesson1]$ /stage/db/runInstaller
```

   b) On the **Select a Product to Instal**l screen, choose **Oracle Clusterware** and click **Next**.

   c) On the **Specify Inventory directory and credentials** page, ensure the inventory directory path is `/u01/app/oraInventory` and the operating system group name is `oinstall`. Click **Next** to continue.

   d) On the **Specify Home Details** page, the Name should be `OraCrs11g_home`. Do not accept the default location for the Path field. Enter `/u01/crs11g` in the Path field. Click **Next**.

   e) The **Product-Specific Prerequisite** Page should check the minimum requirements and report no errors. Click **Next** to continue.

Practice 1-1: Oracle Clusterware Installation (continued)

f) Enter the Cluster Name assigned to you by your instructor in the Cluster Name field on the **Specify Cluster Configuration** page. The only node appearing in the Cluster Nodes window is the installing (local) node. Click the **Add** button to add the second node in your cluster. Enter the Public Node Name, the Private Node Name, and the Virtual Host name, all in fully qualified format. Click **OK** to return to the **Specify Cluster Configuration** page. Review the name information for the second node and click **Next**.

g) On the **Specify Network Interface Usage** page, all network interfaces will be marked private. The proper interface type designations are: **eth0-Public**, **eth1-Do Not Use**, and **eth2-Private**. Select **eth0** and click **Edit**. Change the interface type to **Public** and click **OK**. Next, select **eth1** (the storage network), and click **Edit**. Change the interface type to **Do Not Use** and click **OK**. Leave eth2 as the Private interface and click **Next** to continue.

h) On the **Specify OCR Location** page, select **External Redundancy** and enter **`/dev/sdb1`** for the OCR location. Click **Next** when finished.

i) Select **External Redundancy** on the **Specify Voting Disk Location** page and enter **`/dev/sdb5`** for the Voting Disk location and click **Next**.

j) Review the information on the **Summary** page and click **Install**. Monitor the progress on the **Install** page.

k) When the Execute Configuration scripts dialog box appears, open a terminal window to your local node and execute the `/u01/app/oraInventory/orainstRoot.sh` script as root using the `sudo` command. When finished, ssh to the second node and run the `/u01/app/oraInventory/orainstRoot.sh` as root. When finished, exit back to your first node in your terminal window.

```
[oracle@vx0301 ~]$ sudo /u01/app/oraInventory/orainstRoot.sh
Changing permissions of /u01/app/oraInventory to 770.
Changing groupname of /u01/app/oraInventory to oinstall.
The execution of the script is complete
[oracle@vx0301 ~]$
[oracle@vx0301 ~]$ ssh vx0302
Last login: Mon Nov 19 10:03:42 2007 from vx0301.us.oracle.com
[oracle@vx0302 ~]$ sudo /u01/app/oraInventory/orainstRoot.sh
Changing permissions of /u01/app/oraInventory to 770.
Changing groupname of /u01/app/oraInventory to oinstall.
The execution of the script is complete
[oracle@vx0302 ~]$ exit
logout
Connection to vx0302 closed.
[oracle@vx0301 ~]$
```

l) Next, run the `/u01/crs11g/root.sh` script as root using `sudo` from the terminal window on your FIRST node. When it has finished executing, ssh to your second node and execute the `/u01/crs11g/root.sh` script as root.

```
[oracle@vx0301 u01]$ sudo /u01/crs11g/root.sh
```

Practice 1-1: Oracle Clusterware Installation (continued)

```
Checking to see if Oracle CRS stack is already configured
/etc/oracle does not exist. Creating it now.

Setting the permissions on OCR backup directory
Setting up Network socket directories
Oracle Cluster Registry configuration upgraded successfully
Successfully accumulated necessary OCR keys.
Using ports: CSS=49895 CRS=49896 EVMC=49898 and EVMR=49897.
node <nodenumber>: <nodename> <private interconnect name>
<hostname>
node 1: vx0301 vx0301-priv vx0301
node 2: vx0302 vx0302-priv vx0302
Creating OCR keys for user 'root', privgrp 'root'..
Operation successful.
Now formatting voting device: /dev/sdb5
Format of 1 voting devices complete.
Startup will be queued to init within 30 seconds.
Adding daemons to inittab
Expecting the CRS daemons to be up within 600 seconds.
Cluster Synchronization Services is active on these nodes.
        vx0301
Cluster Synchronization Services is inactive on these nodes.
        vx0302
Local node checking complete. Run root.sh on remaining nodes
to start CRS daemons.
[oracle@vx0301 u01]$
[oracle@vx0301 u01]$ ssh vx0302
Last login: Mon Nov 19 10:56:21 2007 from vx0301.us.oracle.com
[oracle@vx0302 ~]$ sudo /u01/crs11g/root.sh
Checking to see if Oracle CRS stack is already configured
/etc/oracle does not exist. Creating it now.

Setting the permissions on OCR backup directory
Setting up Network socket directories
Oracle Cluster Registry configuration upgraded successfully
clscfg: EXISTING configuration version 4 detected.
clscfg: version 4 is 11 Release 1.
Successfully accumulated necessary OCR keys.
Using ports: CSS=49895 CRS=49896 EVMC=49898 and EVMR=49897.
node <nodenumber>: <nodename> <private interconnect name>
<hostname>
node 1: vx0301 vx0301-priv vx0301
node 2: vx0302 vx0302-priv vx0302
clscfg: Arguments check out successfully.

NO KEYS WERE WRITTEN. Supply -force parameter to override.
-force is destructive and will destroy any previous cluster
configuration.
Oracle Cluster Registry for cluster has already been
initialized
Startup will be queued to init within 30 seconds.
Adding daemons to inittab
```

```
Expecting the CRS daemons to be up within 600 seconds.
Cluster Synchronization Services is active on these nodes.
        vx0301
        vx0302
Cluster Synchronization Services is active on all the nodes.
Waiting for the Oracle CRSD and EVMD to start
Oracle CRS stack installed and running under init(1M)
Running vipca(silent) for configuring nodeapps

Creating VIP application resource on (2) nodes...
Creating GSD application resource on (2) nodes...
Creating ONS application resource on (2) nodes...
Starting VIP application resource on (2) nodes...
Starting GSD application resource on (2) nodes...
Starting ONS application resource on (2) nodes...


Done.
[oracle@vx0302 ~]$
```

m) When both `orainstRoot.sh` and `root.sh` scripts have been run on both nodes, click **OK** on the **Execute Configuration scripts** dialog box.

n) Click the **Exit** button on the **End of Installation** page to exit the installer. Click **Yes** on the dialog box to finish.

o) From a terminal session, execute `/u01/crs11g/bin/crs_stat –t` to check the viability of the nodeapps on both nodes.

```
[oracle@vx0301 u01]$ /u01/crs11g/bin/crs_stat -t
Name           Type           Target    State     Host
------------------------------------------------------------
ora.vx0301.gsd application    ONLINE    ONLINE    vx0301
ora.vx0301.ons application    ONLINE    ONLINE    vx0301
ora.vx0301.vip application    ONLINE    ONLINE    vx0301
ora.vx0302.gsd application    ONLINE    ONLINE    vx0302
ora.vx0302.ons application    ONLINE    ONLINE    vx0302
ora.vx0302.vip application    ONLINE    ONLINE    vx0302
```

Practices for Lesson 2

Practice 2-1: ASM Installation and Configuration

In this practice, you install and configure ASM

1) Start the Oracle Universal Installer from /stage/db and create a non-shared ORACLE HOME for ASM.

   a) Execute `/stage/db/runInstaller` to start OUI.

   b) Select **Oracle Database 11g** from the **Select a Product to Install** page and click **Next**.

   c) Select Enterprise Edition from the **Select Installation Type** page and click **Next**.

   d) On the Install Location page, make sure ORACLE_BASE is **`/u01/app/oracle`**. Under **Software Location**, change the Name to **`OraASM11g_home1`**. Change the Path to **`/u01/app/oracle/product/11.1.0/asm_1`**.
   DO NOT ACCEPT THE DEFAULT LOCATION!! Click **Next** to continue.

   e) On the **Specify Hardware Cluster Installation Mode** page, make sure **Cluster Installation** is selected. Click the **Select All** button and then click **Next** to continue.

   f) The **Product-Specific Prerequisite** page should check the minimum requirements and report no errors. Click **Next** to continue.

   g) Select **Install Software Only** on the **Select Configuration Option** page and click **Next**.

   h) On the **Privileged Operating System** page, select **`dba`** for the OSDBA, OSOPER, and OSASM groups. Click **Next** to continue.

   i) Start the installation by clicking the **Install** button on the **Summary** page. Monitor the progress on the **Install** page.

   j) When the **Execute Configuration** scripts dialog box appears, open a terminal window to your local node and execute the `/u01/app/oracle/product/11.1.0/asm_1/root.sh` script as root using the `sudo` command. When finished, ssh to the second node and run the `/u01/app/oracle/product/11.1.0/asm_1/root.sh` as root. When finished, exit back to your first node in your terminal window.

```
[oracle@vx0301 ~]$ sudo
/u01/app/oracle/product/11.1.0/asm_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/11.1.0/asm_1

Enter the full pathname of the local bin directory:
[/usr/local/bin]:
   Copying dbhome to /usr/local/bin ...
   Copying oraenv to /usr/local/bin ...
```

```
   Copying coraenv to /usr/local/bin ...


Creating /etc/oratab file...
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.
[oracle@vx0301 ~]$
[oracle@vx0301 ~]$ ssh vx0302
Last login: Mon Nov 19 11:04:10 2007 from vx0301.us.oracle.com
[oracle@vx0302 ~]$ sudo
/u01/app/oracle/product/11.1.0/asm_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/11.1.0/asm_1

Enter the full pathname of the local bin directory:
[/usr/local/bin]:
   Copying dbhome to /usr/local/bin ...
   Copying oraenv to /usr/local/bin ...
   Copying coraenv to /usr/local/bin ...


Creating /etc/oratab file...
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.
[oracle@vx0302 ~]$ exit
logout
Connection to vx0302 closed.
[oracle@vx0301 ~]$
```

k) When the `root.sh` script has been run on both nodes, click **OK** on the **Execute Configuration scripts** dialog box.

l) Click the **Exit** button on the **End of Installation** page to exit the installer. Click **Yes** on the dialog box to finish.

2) Use DBCA to configure two disk groups, DATA and FRA using block devices `/dev/sd[c-f]`.

a) Set `ORACLE_HOME=/u01/app/oracle/product/11.1.0/asm_1` and execute dbca from `/u01/app/oracle/product/11.1.0/asm_1/bin`.

```
[oracle@vx0301 ~]$ export
ORACLE_HOME=/u01/app/oracle/product/11.1.0/asm_1
```

```
[oracle@vx0301 ~]$
/u01/app/oracle/product/11.1.0/asm_1/bin/dbca
```

b) On the **Welcome** screen, select Oracle Real Application Cluster Database and click **Next**.

c) On the **Operations** screen, select **Configure Automatic Storage Management** and click **Next**.

d) On the **Node Selection** screen, click the **Select All** button and click **Next**.

e) On the **Create ASM Instance** screen enter `oracle1` as the SYS password for the ASM instances. You must enter it twice. You must also choose the type of parameter file to use for the ASM instances. Select **Create Initialization Parameter File (IFILE)** and accept the default file name and location. Click **Next** to continue.

f) A Database Configuration Assistant dialog box appears informing that the ASM instances will be started. Click **OK** to continue.

g) Another Database Configuration Assistant dialog box appears informing you that no listeners exist and asking if listeners should be started on your nodes. Click **YES** to continue.

h) On the **ASM Disk Groups** screen, click the **Create New** button.

i) On the **Create Disk Group** screen, enter **DATA** for the disk group name. **Select Normal Redundancy**. Click the **Change Disk Discovery Path** button. Enter **`/dev/sd[c-f]`** in the popup window and click **OK**. In the Show Members area of the window, select **`/dev/sdc`** and **`/dev/sdd`** for the DATA disk group and click **OK**.

j) You are returned to the ASM Disk Groups page. DATA should now be present, mounted by both instances. Click the **Create New** button.

k) On the **Create Disk Group** page, enter **FRA** for the Disk Group name and select **Normal Redundancy**. Select the remaining devices, **/dev/sde** and **/dev/sdf** and click **OK**.

l) You are returned to the ASM Disk Groups page. Both DATAand FRA disk groups should now be present, mounted by both instances. Click the **Finish** button then click **No** when prompted to perform another operation.

Practice 2-2: Database Software Installation

In this practice, you install the Oracle Database 11g software.

1) Start the Oracle Universal Installer from `/stage/db` and create a non-shared ORACLE HOME for the database.

   a) Execute `/stage/db/runInstaller` to start OUI.

   b) Select **Oracle Database 11g** from the **Select a Product to Install** page and click **Next**.

   c) Select Enterprise Edition from the **Select Installation Type** page and click **Next**.

   d) On the Install Location page, make sure ORACLE_BASE is **`/u01/app/oracle`**. Under **Software Location**, change the Name to **`OraDb11g_home1`**. Make sure the Path is **`/u01/app/oracle/product/11.1.0/db_1`**. Click **Next** to continue.

   e) On the **Specify Hardware Cluster Installation Mode** page, make sure **Cluster Installation** is selected. Click the **Select All** button and then click **Next** to continue.

   f) The **Product-Specific Prerequisite** Page should check the minimum requirements and report no errors. Click **Next** to continue.

   g) Select **Install Software Only** on the **Select Configuration Option** page and click **Next**.

   h) On the **Privileged Operating System** page, select **`dba`** for the OSDBA, OSOPER, and OSASM groups. Click **Next** to continue.

   i) Start the installation by clicking the **Install** button on the **Summary** page. Monitor the progress on the **Install** page.

   j) When the **Execute Configuration scripts** dialog box appears, open a terminal window to your local node and execute the `/u01/app/oracle/product/11.1.0/db_1/root.sh` script as root using the `sudo` command. When finished, ssh to the second node and run the `/u01/app/oracle/product/11.1.0/db_1/root.sh` as root. When finished, exit back to your first node in your terminal window.

```
[oracle@vx0301 ~]$ sudo
/u01/app/oracle/product/11.1.0/db_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/11.1.0/db_1

Enter the full pathname of the local bin directory:
[/usr/local/bin]:
   Copying dbhome to /usr/local/bin ...
   Copying oraenv to /usr/local/bin ...
```

```
   Copying coraenv to /usr/local/bin ...


Creating /etc/oratab file...
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.
[oracle@vx0301 ~]$
[oracle@vx0301 ~]$ ssh vx0302
Last login: Mon Nov 19 11:04:10 2007 from vx0301.us.oracle.com
[oracle@vx0302 ~]$ sudo
/u01/app/oracle/product/11.1.0/db_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/11.1.0/db_1

Enter the full pathname of the local bin directory:
[/usr/local/bin]:
   Copying dbhome to /usr/local/bin ...
   Copying oraenv to /usr/local/bin ...
   Copying coraenv to /usr/local/bin ...


Creating /etc/oratab file...
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.
[oracle@vx0302 ~]$ exit
logout
Connection to vx0302 closed.
[oracle@vx0301 ~]$
```

k) When the `root.sh` script has been run on both nodes, click **OK** on the **Execute Configuration scripts** dialog box.

l) Click the **Exit** button on the **End of Installation** page to exit the installer. Click **Yes** on the dialog box to finish.

Practices for Lesson 3

Practice 3-1: Create a Cluster Database

In this practice you create a cluster database using DBCA.

1) Create a cluster database using the Global Database Name provided by your instructor. When the database has been successfully created, log in to Enterprise Manager to test the configuration.

   a) Export ORACLE_HOME as `/u01/app/oracle/product/11.1.0/db_1` and execute `dbca` from `ORACLE_HOME/bin`.

```
[oracle@vx0301 ~]$ export
ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1
[oracle@vx0301 ~]$ $ORACLE_HOME/bin/dbca
```

   b) On the **Welcome** screen, select **Oracle Real Application Cluster Database** and click **Next**.

   c) On the **Operations** screen, select **Create a Database** and click **Next**.

   d) On the Node Selection screen, click the Select All button and click **Next**.

   e) On the **Database Templates** page, select **General Purpose or Transaction Processing** and click **Next** to continue.

   f) On the Database Identification page, enter the Global Database Name assigned to you by your instructor (the SID field will autofill) and click **Next.**

   g) Click the **Configure Enterprise Manager** check box on the **Management Options** page. Select **Configure Database Control for local management** and click **Next**.

   h) On the **Database Credentials** page, select **Use the Same Administrative Password For All Accounts.** Enter **`oracle1`** for the password and click **Next**.

   i) Select **Automatic Storage Management (ASM)** from the **Storage Options** page and click **Next** to continue.

   j) On the **ASM Disk Groups** page, select the **DATA** disk group for the database storage and click **Next**.

   k) Select **Oracle Managed Files** on the **Database File Locations** page and enter **+DATA** in the **Database Area** field. Click **Next** to continue.

   l) On the **Recovery Configuration** page, click the **Specify Flash Recovery Area** check box and enter **+FRA** for the **Flash Recovery Area**. Accept the default value of **2048** for the **Flash Recovery Area Size** and click **Next**.

   m) Click the **Sample Schemas** check box on the **Database Content** page and click **Next**.

   n) On the **Initialization Parameters** page click the **Character Sets** folder tab. In the Database Character Set area select **Use Unicode (AL32UTF8)** and click **Next** to continue.

   o) Click the **Keep the enhanced 11g security settings …** check box on the **Security Settings** page and click **Next.**

Practice 3-1: Create a Cluster Database (continued)

p) On the **Automatic Maintenance Tasks** page make sure the **Enable automatic maintenance tasks** check box is checked and click **Next**.

q) Click **Next** on the **Database Storage** page.

r) On the **Database Creation** page, click the **Create Database** check box and click the **Finish** button.

s) When the **Summary** window appears, click **OK**.

t) A Database Configuration Assistant window appears informing you that the database creation has completed. Click the **Exit** button. The Start Cluster Database dialog box informs you that the instances are being started.

u) Test the Enterprise Manager configuration by opening up a web browser and entering the following address: **https://*your_hostname*:1158**

Practices for Lesson 4

Practice 4-1: Add/Remove Redo Log Groups in a RAC Environment

The goal of this practice is to show you how to add and remove redo log groups in a RAC database environment.

1) Use Database Control to create two new redo log groups in your database. The two groups must pertain to the thread number three, and each group must have only one 51200 KB member called redo05.log and redo06.log, respectively.

   a) From Database Control Home page click the Server tab. In this example we use RDB as your database, but make sure you are using the one you were assigned to.

   b) On the Cluster Database Server page, click Redo Log Groups in the Storage section.

   c) On the Redo Log Groups page, click Create.

   d) On the Create Redo Log Group page, leave the current value of the Group# field as is it is (5). Make sure that the File size field is set to 51200 KB. Set the Thread# field to 3. Make sure that the File Directory field in the Redo Log Members section is set to +DATA. When you are finished, click OK.

   e) This takes you back to the Redo Log Groups page from where you should see a Confirmation message giving you successful status on the new object creation.

   f) On the Redo Log Groups page click Create.

   g) On the Create Redo Log Group page, leave the current value of the Group# field as is it is (6). Make sure that the File size field is set to 51200 KB. Set the Thread# field to 3. See that the File Directory field in the Redo Log Members section is set to +DATA. When you are finished, click OK.

   h) This takes you back to the Redo Log Groups page from where you should see a Confirmation message giving you successful status on the new object creation.

2) Use Database Control to set the RDB2.THREAD initialization parameter to 3 in SPFILE only.

   a) From Database Control Home page click the Server tab.

   b) On the Server page, click the Initialization Parameters link in the Database Configuration section.

   c) On the Initialization Parameters page, click the SPFile tab.

   d) On the SPFile page, click the Show All button.

   e) Make sure that the "Apply changes in SPFile mode to the current running instance(s). For static parameters, you must restart the database" check box is deselected.

   f) Scroll down the page to the end, and enter 3 in the Value field for the RDB2.thread field.

Practice 4-1: Add/Remove Redo Log Groups in a RAC Environment (continued)

g) Click the Apply button at the bottom of the page.

h) This takes you back to the Initialization Parameters page on which you should see an Update Message giving you successful status on your changes.

3) Use the SRVCTL control utility to stop the RDB2 instance, and start it up again. What happens and why? If necessary, fix the situation.

   a) Although it is possible to stop the RDB2 instance, it is not possible to start it up again because the redo thread number three is not enabled yet. To enable the redo thread number three, you can connect to the first instance and issue the ALTER DATABASE ENABLE THREAD 3 command. Then you can start up the RDB2 instance again.

```
[oracle@vx0306 less04]$ cat sol_04_01_03.sh
#!/bin/bash
#
# sol_04_01_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/srvctl stop instance -d $DBNAME -i $I2NAME

/u01/crs11g/bin/srvctl start instance -d $DBNAME -i $I2NAME

/u01/crs11g/bin/crs_stat -t

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1
export ORACLE_SID=$I1NAME

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
 set echo on
 ALTER DATABASE ENABLE THREAD 3;

EOF
```

Practice 4-1: Add/Remove Redo Log Groups in a RAC Environment (continued)

```
/u01/crs11g/bin/srvctl start instance -d $DBNAME -i $I2NAME

/u01/crs11g/bin/crs_stat -t
[oracle@vx0306 less04]$


[oracle@vx0306 less04]$ ./sol_04_01_03.sh
PRKP-1001 : Error starting instance RDB2 on node vx0313
CRS-0215: Could not start resource 'ora.RDB.RDB2.inst'.
Name           Type           Target    State     Host
------------------------------------------------------------
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora....B2.inst application    ONLINE    OFFLINE
ora.RDB.db     application    ONLINE    ONLINE    vx0313
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
ora....SM2.asm application    ONLINE    ONLINE    vx0313
ora....13.lsnr application    ONLINE    ONLINE    vx0313
ora.vx0313.gsd application    ONLINE    ONLINE    vx0313
ora.vx0313.ons application    ONLINE    ONLINE    vx0313
ora.vx0313.vip application    ONLINE    ONLINE    vx0313

Database altered.

Name           Type           Target    State     Host
------------------------------------------------------------
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora....B2.inst application    ONLINE    ONLINE    vx0313
ora.RDB.db     application    ONLINE    ONLINE    vx0313
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
ora....SM2.asm application    ONLINE    ONLINE    vx0313
ora....13.lsnr application    ONLINE    ONLINE    vx0313
ora.vx0313.gsd application    ONLINE    ONLINE    vx0313
ora.vx0313.ons application    ONLINE    ONLINE    vx0313
ora.vx0313.vip application    ONLINE    ONLINE    vx0313
[oracle@vx0306 less04]$
```

4) Revert to the original situation where RDB2 was using the redo thread two, and destroy redo thread number three. Make sure that in the end both instances are up and running and managed by Oracle Clusterware.

```
[oracle@vx0306 less04]$ cat sol_04_01_04.sh
#!/bin/bash
#
# sol_04_01_04.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1
export ORACLE_SID=$I1NAME

echo "Reset thread to 2 for second instance ..."

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
 ALTER SYSTEM SET thread = 2 SCOPE=SPFILE SID='$I2NAME';

EOF

echo "Stop second instance ..."

/u01/crs11g/bin/srvctl stop instance -d $DBNAME -i $I2NAME

echo "Restart second instance ..."

/u01/crs11g/bin/srvctl start instance -d $DBNAME -i $I2NAME

echo "Removing thread 3 from database ..."

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
 alter database disable thread 3;
 alter database drop logfile group 5;
 alter database drop logfile group 6;

EOF
```

Practice 4-1: Add/Remove Redo Log Groups in a RAC Environment (continued)

```
/u01/crs11g/bin/crs_stat -t
[oracle@vx0306 less04]$


[oracle@vx0306 less04]$ ./sol_04_01_04.sh
Reset thread to 2 for second instance ...

System altered.

Stop second instance ...
Restart second instance ...
Removing thread 3 from database ...

Database altered.


Database altered.


Database altered.

Name           Type           Target    State     Host
------------------------------------------------------------
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora....B2.inst application    ONLINE    ONLINE    vx0313
ora.RDB.db     application    ONLINE    ONLINE    vx0313
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
ora....SM2.asm application    ONLINE    ONLINE    vx0313
ora....13.lsnr application    ONLINE    ONLINE    vx0313
ora.vx0313.gsd application    ONLINE    ONLINE    vx0313
ora.vx0313.ons application    ONLINE    ONLINE    vx0313
ora.vx0313.vip application    ONLINE    ONLINE    vx0313
[oracle@vx0306 less04]$
```

Practices for Lesson 5

Practice 5-1: Backup and Recovery

The following short exercises allow you to set backup and recovery options, configure a Recovery Manager Repository, schedule a backup, and review RMAN reports using Enterprise Manager.

1) Use Enterprise Manager to place the database in ARCHIVELOG mode. Enable Flashback Recovery and set `LOG_ARCHIVE_DEST_1` to the Flash Recovery Diskgroup (FRA). Adjust the Flash Recovery Area to 3 gigabytes and enable Flashback Database.

   a) Log in to Enterprise Manager as `sys/oracle1` and connect as `SYSDBA`.

   b) From your database home page, click the **Availability** folder tab then click **Recovery Settings** under Backup/Recovery. Click the **ARCHIVELOG Mode** check box. Add your database name to the beginning of the archive log file name format string, %t_%s_%r.dbf , for example `RDB%t_%s_%r.dbf`. Enter **+FRA** for the value of LOG_ARCHIVE_DEST_1.

   c) Scroll down to the Flash Recovery section and click the **Enable Flashback Database**… button, and set the **Flash Recovery Area size** to **3 GB**. Click the **Apply** button.

   d) The **Confirmation** page asks if it is OK to restart the database. Click **Yes**.

   e) On the **Startup/Shutdown** page supply the username and password to log in to the cluster nodes, **`oracle/oracle`** and the database user, **`sys/oracle1`** connecting as **SYSDBA**. Click **Continue**.

   f) After a few moments, click the **Refresh** button on the **Request in Process** page to check on the progress of the database restart.

   g) Log back into Enterprise Manager. When all instances have been restarted, navigate back to **Recovery Settings** and verify that your changes have been processed.

2) Using Enterprise Manager, configure backup settings in support of RAC. Set disk parallelism to two, the disk backup location to the Flash Recovery Area, and the backup type to compressed. The backup policy should include autobackups of the control file and spfile for every backup, unchanged files should be skipped and block change tracking should be enabled. Set the point-in-time window to be 31 days.

   a) Return to the **Availability** folder and click on **Backup Settings**. Under Backup Settings, click on the **Device** folder tab. Under Disk Settings, change the Parallelism value to 2, the Disk Backup location to +FRA, and click the **Compressed Backup Set** button under Disk Backup Type. Include the host login, **`oracle/oracle`** under **Host Credentials**.

Practice 5-1: Backup and Recovery (continued)

b) Next, click on the **Policy** folder tab. Under Backup Policy, select **Automatically backup the control file…** . Enable skipping of unchanged files and block change tracking. Under Retention Policy, select the **Retain at least the specified number of full backups for each datafile** option, specify the **Recovery Window** value to be **31 days**, and set the Redundancy value to **2**. When finished click the **OK** button. Return to **Backup Settings** and verify the changes.

3) Configure an RMAN recovery catalog and set persistent RMAN configuration parameters that are RAC friendly. When the parameters have been saved, initiate a full cluster database backup.

a) Execute the `sol_05_01_03a.sh` script on the first node to create a recovery catalog and owner.

```
$ $HOME/solutions/less05/sol_05_01_03a.sh
SQL> CREATE USER rman IDENTIFIED BY rman
  TEMPORARY TABLESPACE temp
  DEFAULT TABLESPACE USERS
  QUOTA UNLIMITED ON USERS;
 GRANT recovery_catalog_owner TO rman;  2    3    4
User created.

SQL> exit
  2  exit
  3  ;

SQL> GRANT recovery_catalog_owner TO rman;

Grant succeeded.


Recovery Manager: Release 11.1.0.6.0 - Production on Tue Nov
20 15:46:45 2007

Copyright (c) 1982, 2007, Oracle.  All rights reserved.

CONNECT CATALOG rman/rman

connected to recovery catalog database

CREATE CATALOG;

recovery catalog created

RMAN> exit

Recovery Manager complete.
```

b) Return to the **Availability** folder and click **Recovery Catalog Settings** under **Backup/Recovery**.

c) Next, click the **Add Recovery Catalog** button.

d) On the **Add a Recovery Catalog** page, enter the hostname of your first node in the HOST field and 1521 for PORT. Enter the SID on the specified host (RDB1 for example) Enter `rman/rman` for the recovery catalog user and password. Click **Next**.

e) Click **Finish** on the **Review** page.

f) Return to the **Recovery Catalog Settings** page. Select **Use Recovery Catalog** option and select your database from the pull down list. Provide the host credentials, `oracle/oracle`. Click the **OK** button when finished.

g) The next screen will inform you that your database has been successfully registered with the recovery catalog.

4) Using Enterprise Manager, perform a one-time, full database backup. When the backup is finished, use the RMAN reporting functionality in Enterprise Manager to view the backup details.

a) Go to the **Availability** folder tab. Click on **Schedule Backup** under **Manage**.

b) Select **Whole Database** under **Customized Backup**. Make sure you have provided the host login credentials (`oracle/oracle`) then click the **Schedule Customized Backup** button.

c) Select **Full Backup** under **Backup Type**, select **Online Backup** for **Backup Mode**, and select also back up all archived logs on disk under **Advanced**. Click **Next** when finished.

d) On the next page, make sure that your backup destination is **Disk**. Click **Next** to continue.

e) Accept the default information; Job Name, Job Description, and One Time (Immediately) on the **Schedule** page and click **Next**.

f) On the **Review** page, click **Submit Job**.

g) The next page will inform you that your job has been successfully submitted. To monitor the backup progress you can click the **View Job** button and occasionally refreshing your browser.

h) When the backup is finished, return to the **Availability** folder and click **Manage Current Backups** under **Manage**. Click the backup name components that correspond to your recently completed database backup.

Practices for Lesson 6

This practice shows you how to discover performance problems in your RAC environment. In this practice, you identify performance issues using Enterprise Manager, and you fix those issues in three different steps. At each step, you generate the same workload to make sure you are making progress in your resolution.

Practice 6-1: ADDM and RAC, Part I

The goal of this practice is to show you how to manually discover performance issues by using the Enterprise Manager performance pages as well as ADDM. Part I generates a workload that uses a bad RAC application design.

1) Execute the setupseq1.sh script to set up the necessary configuration for this practice.

```
[oracle@vx0306 less06]$ cat setupseq1.sh
#!/bin/bash
#
# setupseq1.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1
export ORACLE_SID=$I1NAME

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
 exec
dbms_advisor.set_default_task_parameter('ADDM','DB_ACTIVITY_MI
N',30);
 exec
dbms_workload_repository.modify_snapshot_settings(interval=>60
0);
 drop user jfv cascade;
 drop tablespace seq including contents and datafiles;
 create tablespace seq extent management local autoallocate
segment space management auto;
 create user jfv identified by jfv default tablespace seq
temporary tablespace temp;
 grant connect,resource,dba to jfv;
 connect jfv/jfv
 drop sequence s;
 drop table s purge;
 drop table t purge;
 create table s(sn number);
 create table t(c number,d varchar2(20));
```

```
 create index it on t(c);
 insert into s values(1);
 commit;
 begin
  for i in 1..90000 loop
  insert into t values (i,'initial');
  end loop;
 end;
 /

EOF
[oracle@vx0306 less06]$ ./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:
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.

[oracle@vx0306 less06]$
```

2) Using Database Control, and connected as user SYS, navigate to the Performance page of your Cluster Database.
   a) Click the Performance tab on 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 dropdown list.
3) Use PL/SQL to create a new AWR snapshot.

```
[oracle@vx0306 less06]$ cat sol_06_01_03.sh
#!/bin/bash
#
# sol_06_01_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"
```

```
I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1
export ORACLE_SID=$I1NAME

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
 exec dbms_workload_repository.create_snapshot

EOF
[oracle@vx0306 less06]$

[oracle@vx0306 less06]$ ./sol_06_01_03.sh

PL/SQL procedure successfully completed.

[oracle@vx0306 less06]$
```

4) Execute the startseq1.sh script to generate a workload on both instances of your cluster. Do not wait; instead, proceed with the next step.

```
[oracle@vx0306 less06]$ cat startseq1.sh
#!/bin/bash
#
# startseq1.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /nolog @runseq1 $I1NAME &

$ORACLE_HOME/bin/sqlplus -s /nolog @runseq1 $I2NAME &
[oracle@vx0306 less06]$

[oracle@vx0306 less06]$ cat runseq1.sql
```

Practice 6-1: ADDM and RAC, Part I (continued)

```
set echo on

connect jfv/jfv@&1

declare
 v number;
begin
 for i in 1..40000 loop
  lock table s in exclusive mode;
  select sn into v from s;
  insert into t values(v,'&1');
  update s set sn=sn+1;
  commit;
 end loop;
end;
/

exit;
[oracle@vx0306 less06]$

[oracle@vx0306 less06]$ ./startseq1.sh
[oracle@vx0306 less06]$ old   7:   insert into t
values(v,'&1');
new   7:   insert into t values(v,'RDB1');
old   7:   insert into t values(v,'&1');
new   7:   insert into t values(v,'RDB2');

… Do not wait after this point.

PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


[oracle@vx0306 less06]$
```

5) Using Database Control, determine the list of blocking locks in your database.

   a) Still on the Performance page, click the Database Locks link in the Additional Monitoring Links section of the page.

   b) On the Database Locks page, make sure that Blocking Locks is selected from the View dropdown list.

   c) 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 should see that both sessions are alternately waiting for the other to release the exclusive lock held on table S.

6) 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 drilldown 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.

   b) Click the Cluster Database 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 should 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 300 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 should see that the number of active sessions is almost the same on both 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 should 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.

7) 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.

Practice 6-1: ADDM and RAC, Part I (continued)

8) While the scripts are still executing, look at the Average Active Sessions graph. Then drill down to the Application wait class for the first instance. What are your conclusions?

   a) By using the drilldown 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.

   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 should 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 should see that the number of active sessions is almost the same on both 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 should 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 both 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.

9) After the workload finishes, use PL/SQL to create a new AWR snapshot.

```
[oracle@vx0306 less06]$ cat sol_06_01_09.sh
#!/bin/bash
#
# sol_06_01_09.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`
```

Practice 6-1: ADDM and RAC, Part I (continued)

```
DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1
export ORACLE_SID=$I1NAME

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
 exec dbms_workload_repository.create_snapshot

EOF
[oracle@vx0306 less06]$

[oracle@vx0306 less06]$ ./sol_06_01_09.sh

PL/SQL procedure successfully completed.

[oracle@vx0306 less06]$
```

10) 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 Top SQL by DB Time, Table Locks, and Top SQL by Cluster Wait findings.

e) The Top SQL by DB Time should reveal a LOCK TABLE S command as a possible problem to investigate.

f) The Table Lock finding, which affects both instances, reveals that you should investigate your application logic regarding the JFV.S object.

g) The Top SQL by Cluster Wait finding again reveals the LOCK TABLE S command embedded in a PL/SQL block.

Practice 6-1: ADDM and RAC, Part I (continued)

h) Back on the Automatic Database Diagnostic Monitor (ADDM) page, you now have the possibility to drill down to each instance by using the links located in the Affected Instances table. Click the link corresponding to the most affected instance (although both should be equally affected).

i) On the corresponding ADDM Database Diagnostic Monitor (ADDM) instance page, you should retrieve exactly the same top findings that you previously saw at the cluster level.

Practice 6-2: ADDM and RAC, Part II

The goal of Practice 6 is to show you how to manually discover performance issues by using the Enterprise Manager performance pages as well as ADDM. In Part II of the practice, you correct the previously found issue by creating a sequence number instead of using a table.

1) Execute the setupseq2.sh script to create the necessary objects used for the rest of this practice.

```
[oracle@vx0306 less06]$ cat setupseq2.sh
#!/bin/bash
#
# setupseq2.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1
export ORACLE_SID=$I1NAME

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
 exec
dbms_advisor.set_default_task_parameter('ADDM','DB_ACTIVITY_MI
N',30);
 exec
dbms_workload_repository.modify_snapshot_settings(interval=>60
0);
 drop user jfv cascade;
 drop tablespace seq including contents and datafiles;
 create tablespace seq extent management local autoallocate
segment space management auto;
 create user jfv identified by jfv default tablespace seq
temporary tablespace temp;
 grant connect,resource,dba to jfv;
 connect jfv/jfv
 drop table s purge;
 drop sequence s;
 drop table t purge;
```

```
 create table t(c number,d varchar2(20));
 create index it on t(c);
 create sequence s start with 1 increment by 1 nomaxvalue
nocache;

 begin
  for i in 1..90000 loop
   insert into t values (i,'initial');
  end loop;
 end;
 /

EOF
[oracle@vx0306 less06]$

[oracle@vx0306 less06]$ ./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



Table created.


Index created.


Sequence created.


PL/SQL procedure successfully completed.

[oracle@vx0306 less06]$
```

2) Using Database Control, and connected as user SYS, navigate to the Performance page of your Cluster Database.

   a) Click the Performance tab on 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 dropdown list.

3) Use PL/SQL to create a new AWR snapshot.

```
[oracle@vx0306 less06]$ cat sol_06_02_03.sh
#!/bin/bash
#
# sol_06_02_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1
export ORACLE_SID=$I1NAME

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
 exec dbms_workload_repository.create_snapshot
```

```
EOF
[oracle@vx0306 less06]$

[oracle@vx0306 less06]$ ./sol_06_02_03.sh

PL/SQL procedure successfully completed.

[oracle@vx0306 less06]$
```

4) Execute the startseq2.sh script to generate a workload on both instances of your cluster. Do not wait; instead, proceed with the next step.

```
[oracle@vx0306 less06]$ cat startseq2.sh
#!/bin/bash
#
# startseq2.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /nolog @runseq1 $I1NAME &

$ORACLE_HOME/bin/sqlplus -s /nolog @runseq1 $I2NAME &
[oracle@vx0306 less06]$

[oracle@vx0306 less06]$ cat runseq2.sql

set echo on

connect jfv/jfv@&1

declare
 v number;
begin
 for i in 1..40000 loop
  lock table s in exclusive mode;
```

```
  select sn into v from s;
  insert into t values(v,'&1');
  update s set sn=sn+1;
  commit;
 end loop;
end;
/

exit;
[oracle@vx0306 less06]$

[oracle@vx0306 less06]$ ./startseq2.sh
[oracle@vx0306 less06]$ old   7:   insert into t
values(v,'&1');
new   7:   insert into t values(v,'RDB1');
old   7:   insert into t values(v,'&1');
new   7:   insert into t values(v,'RDB2');

… Do not wait after this point.

PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


[oracle@vx0306 less06]$
```

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 drilldown 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 the Cluster Database 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 should clearly show that the Cluster and Application wait classes are causing the most waits. In the Throughput tabbed page graph underneath the Average Active Sessions graph, you should also notice that the transaction rate is about 400 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.

Practice 6-2: ADDM and RAC, Part II (continued)

e) On the Active Sessions By Instance: Cluster page, you should see that the number of active sessions is almost the same on both 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 that cause 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 should 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 both scripts are executed by looking at the Average Active Sessions graphic again. The Database Throughput graphic tells you that the number of transactions per second is now a bit higher than in the previous practice 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.

```
[oracle@vx0306 less06]$ cat sol_06_02_06.sh
#!/bin/bash
#
# sol_06_02_06.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1
export ORACLE_SID=$I1NAME

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
 exec dbms_workload_repository.create_snapshot

EOF
[oracle@vx0306 less06]$

[oracle@vx0306 less06]$ ./sol_06_02_06.sh
```

Practice 6-2: ADDM and RAC, Part II (continued)

```
PL/SQL procedure successfully completed.

[oracle@vx0306 less06]$
```

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 Top SQL by DB Time, Top SQL by Cluster Wait, and Sequence Usage findings.
   e) The Top SQL by DB Time should reveal an INSERT INTO T command using sequence S as a possible problem to investigate.
   f) The Top SQL by Cluster Wait finding reveals the same statement as above: an INSERT INTO T command using sequence S.
   g) The Sequence Usage finding, which affects both instances, reveals a high frequency of sequence cache misses.
   h) On 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 both should be equally affected).
8) On the corresponding ADDM Database Diagnostic Monitor (ADDM) instance page, you should retrieve exactly the same top findings that you previously saw at the cluster level.

Practice 6-3: ADDM and RAC, Part III

The goal of Practice 6 is to show you how to manually discover performance issues by using the Enterprise Manager performance pages as well as ADDM. Part III generates the same workload as in Part II but uses more cache entries for sequence number S.

1) Execute the setupseq3.sh script to create the necessary objects used for the rest of this practice.

```
[oracle@vx0306 less06]$ cat setupseq3.sh
#!/bin/bash
#
# setupseq3.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1
export ORACLE_SID=$I1NAME

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
 exec
dbms_advisor.set_default_task_parameter('ADDM','DB_ACTIVITY_MI
N',30);
 exec
dbms_workload_repository.modify_snapshot_settings(interval=>60
0);
 drop user jfv cascade;
 drop tablespace seq including contents and datafiles;
 create tablespace seq extent management local autoallocate
segment space management auto;
 create user jfv identified by jfv default tablespace seq
temporary tablespace temp;
 grant connect,resource,dba to jfv;
 connect jfv/jfv
 drop table s purge;
 drop sequence s;
 drop table t purge;
 create table t(c number,d varchar2(20));
```

Practice 6-3: ADDM and RAC, Part III (continued)

```
 create index it on t(c);
 create sequence s start with 1 increment by 1 nomaxvalue
cache 90000;

 begin
  for i in 1..90000 loop
   insert into t values (i,'initial');
  end loop;
 end;
 /

EOF
[oracle@vx0306 less06]$

[oracle@vx0306 less06]$ ./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
```

```
Table created.


Index created.


Sequence created.


PL/SQL procedure successfully completed.

[oracle@vx0306 less06]$
```

2) Using Database Control, and connected as user SYS, navigate to the Performance page of your Cluster Database.
   a) Click the Performance tab on 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 dropdown list.
3) Use PL/SQL to create a new AWR snapshot.

```
[oracle@vx0306 less06]$ cat sol_06_03_03.sh
#!/bin/bash
#
# sol_06_03_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1
export ORACLE_SID=$I1NAME

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
 exec dbms_workload_repository.create_snapshot
```

```
EOF
[oracle@vx0306 less06]$

[oracle@vx0306 less06]$ ./sol_06_03_03.sh

PL/SQL procedure successfully completed.

[oracle@vx0306 less06]$
```

4) Execute the startseq2.sh script to generate the same workload on both instances of your cluster as for the previous practice. Do not wait; instead, proceed with the next step.

```
[oracle@vx0306 less06]$ cat startseq2.sh
#!/bin/bash
#
# startseq2.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /nolog @runseq1 $I1NAME &

$ORACLE_HOME/bin/sqlplus -s /nolog @runseq1 $I2NAME &
[oracle@vx0306 less06]$

[oracle@vx0306 less06]$ cat runseq2.sql

set echo on

connect jfv/jfv@&1

declare
 v number;
begin
 for i in 1..40000 loop
  lock table s in exclusive mode;
```

```
  select sn into v from s;
  insert into t values(v,'&1');
  update s set sn=sn+1;
  commit;
 end loop;
end;
/

exit;
[oracle@vx0306 less06]$

[oracle@vx0306 less06]$ ./startseq2.sh
[oracle@vx0306 less06]$ old   7:   insert into t
values(v,'&1');
new   7:   insert into t values(v,'RDB1');
old   7:   insert into t values(v,'&1');
new   7:   insert into t values(v,'RDB2');

… Do not wait after this point.

PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


[oracle@vx0306 less06]$
```

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 the Cluster Database 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 both scripts finish execution, the Average Active Sessions graph should clearly show that there are no significant waits on your cluster. You should also notice that the transaction rate is now around 1,000 per second.

6) After the workload finishes, use PL/SQL to create a new AWR snapshot.

```
[oracle@vx0306 less06]$ cat sol_06_03_06.sh
#!/bin/bash
#
# sol_06_03_06.sh
#
# Must be executed on NODE1 !!!!!!!!!!
```

```
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1
export ORACLE_SID=$I1NAME

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
 exec dbms_workload_repository.create_snapshot

EOF
[oracle@vx0306 less06]$

[oracle@vx0306 less06]$ ./sol_06_03_06.sh

PL/SQL procedure successfully completed.

[oracle@vx0306 less06]$
```

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 “Top SQL by DB Time” and “Top SQL by Cluster Wait” findings. You should no longer see the Sequence Usage or the specific instances affected.
   e) The “Top SQL by DB Time” finding should reveal an INSERT INTO T command using sequence S as a possible problem to investigate.
   f) The “Top SQL by Cluster Wait” finding reveals the same statement as above: an INSERT INTO T command using sequence S.

Practice 6-3: ADDM and RAC, Part III (continued)

g) On the Automatic Database Diagnostic Monitor (ADDM) page, you now have the ability to drill down to each instance by using the links located in the Affected Instances table. Click the link corresponding to the most affected instance.

8) On the corresponding ADDM Database Diagnostic Monitor (ADDM) instance page, you should retrieve exactly the same top findings that you previously saw at the cluster level.

Practices for Lesson 7

In this practice, you create a service, manipulate it, and monitor it.

Practice 7-1: Manage Services

In this practice, you use Database Control to create one service called SERV1. You then observe what happens to your service when you terminate the instance on which it is running.

1) Use Database Control to create the SERV1 service. Make sure that you define your first instance (RDB1 in this example) as preferred, and the second instance (RDB2 in this example) as available.

   a) Connect as user SYS in Database Control.

   b) On the Cluster Home page, click the Availability tab.

   c) On the Availability page, click the Cluster Managed Database Services link in the Services section.

   d) On the Cluster and Database Login page, enter oracle in both Username and Password fields in the Cluster Credentials section. In the Database Credentials section, enter `sys` in the Username field, and `oracle1` in the Password field. Make sure you select the "Save as preferred credential" check box. When you are finished, click Continue.

   e) On the Cluster Managed Database Services page, click the Create Service button.

   f) On the Create Service page, enter SERV1 in the Service Name field. Make sure the Start service after creation check box is checked. Set Service Policy to Preferred for RDB1, and Available for RDB2. Leave all other fields to their defaults.

   g) When you are finished, click OK.

2) After you have created SERV1, make sure it is taken into account by Oracle Clusterware. Use the crs_stat command from one of the nodes, and then use the SRVCTL command.

```
[oracle@vx0306 less07]$ cat sol_07_01_02.sh
#!/bin/ksh
#
# sol_07_01_02.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"
```

Practice 7-1: Manage Services (continued)

```
/u01/crs11g/bin/crs_stat

/u01/crs11g/bin/srvctl status service -d $DBNAME -s SERV1
[oracle@vx0306 less07]$

[oracle@vx0306 less07]$ ./sol_07_01_02.sh
NAME=ora.RDB.RDB1.inst
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0306

NAME=ora.RDB.RDB2.inst
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0313

NAME=ora.RDB.SERV1.RDB1.srv
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0306

NAME=ora.RDB.SERV1.cs
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0306

NAME=ora.RDB.db
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0313

NAME=ora.vx0306.ASM1.asm
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0306

NAME=ora.vx0306.LISTENER_VX0306.lsnr
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0306

NAME=ora.vx0306.gsd
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0306

NAME=ora.vx0306.ons
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0306
```

Practice 7-1: Manage Services (continued)

```
NAME=ora.vx0306.vip
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0306

NAME=ora.vx0313.ASM2.asm
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0313

NAME=ora.vx0313.LISTENER_VX0313.lsnr
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0313

NAME=ora.vx0313.gsd
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0313

NAME=ora.vx0313.ons
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0313

NAME=ora.vx0313.vip
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0313

Service SERV1 is running on instance(s) RDB1
[oracle@vx0306 less07]$
```


3) Add SERV1 to the tnsnames.ora files on both nodes to both oracle homes. When you are finished, make sure listeners are aware of its existence.

```
[oracle@vx0306 less07]$ cat sol_07_01_03.sh
#!/bin/ksh
#
# sol_07_01_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`
```

Practice 7-1: Manage Services (continued)

```
DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

sed 's/NODE1/'$y'/'
/home/oracle/solutions/less07/wrong_tnsserv1.ora >
/home/oracle/solutions/less07/w1tnsserv1.ora

sed 's/NODE2/'$z'/'
/home/oracle/solutions/less07/w1tnsserv1.ora >
/home/oracle/solutions/less07/tnsserv1.ora

cp
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora
.bak1
cat /home/oracle/solutions/less07/tnsserv1.ora >>
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora

cp
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a.bak2
cat /home/oracle/solutions/less07/tnsserv1.ora >>
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a

ssh $z cp
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora
.bak1

ssh $z cp
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a.bak2

scp /home/oracle/solutions/less07/tnsserv1.ora
$z:/home/oracle/solutions/less07

ssh $z "cat /home/oracle/solutions/less07/tnsserv1.ora >>
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora
"

ssh $z "cat /home/oracle/solutions/less07/tnsserv1.ora >>
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/asm_1
```

Practice 7-1: Manage Services (continued)

```
/u01/app/oracle/product/11.1.0/asm_1/bin/lsnrctl services |
grep SERV1

[oracle@vx0306 less07]$


[oracle@vx0306 less07]$ cat wrong_tnsserv1.ora

SERV1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = NODE1-vip)(PORT =
1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = NODE2-vip)(PORT =
1521))
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = SERV1)
    )
  )

[oracle@vx0306 less07]$


[oracle@vx0306 less07]$ ./sol_07_01_03.sh
tnsserv1.ora                                   100%  268
0.3KB/s   00:00
Service "SERV1" has 1 instance(s).

[oracle@vx0306 less07]$
```


4) Connect as SYSTEM under each instance and look at the current value of the SERVICE_NAMES initialization parameter, and check that it is set correctly.

```
 [oracle@vx0306 less07]$ cat sol_07_01_04.sh
#!/bin/ksh
#
# sol_07_01_04.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`
```

Practice 7-1: Manage Services (continued)

```
I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 set echo on
 connect system/oracle1@$I1NAME
 select instance_name from v\$instance;
 show parameter service
 connect system/oracle1@$I2NAME
 select instance_name from v\$instance;
 show parameter service

EOF
[oracle@vx0306 less07]$

[oracle@vx0306 less07]$ ./sol_07_01_04.sh

INSTANCE_NAME
----------------
RDB1


NAME                                 TYPE        VALUE
------------------------------------ ----------- -------------
-----------------
service_names                        string      SERV1, RDB

INSTANCE_NAME
----------------
RDB2


NAME                                 TYPE        VALUE
------------------------------------ ----------- -------------
-----------------
service_names                        string      RDB
[oracle@vx0306 less07]$
```

5) Using Grid Control, how can you check that SERV1 service is currently running as expected?

   a) If you are still connected to your previous Enterprise Manager session, you should see it from the Cluster Managed Database Services page.

Practice 7-1: Manage Services (continued)

b) If not, connect as user SYS in Database Control. On the Cluster Home page, click the Availability tab. On the Availability page, click the Cluster Managed Database Services link in the Services section. On the Cluster and Database Login page, click Continue.

c) On the Cluster Managed Database Services page, you should see SERV1 up and running on the first instance only.

d) On the Cluster Managed Database Services page, click the SERV1 link.

e) On the SERV1 page, you should see that SERV1 is running on the first instance and is down on the second instance.

6) Using a telnet session connected as user `oracle` to the first node, execute the sol_07_01_06_a.sh script. This script monitors events happening inside Oracle Clusterware. From a second terminal window as user `oracle`, kill the SMON process of the first instance (RDB1 in this example). Observe the sequence of events in the first session. When the first instance is back, look at the SERVICE_NAMES initialization parameter values on both instances. What do you observe? When you are finished, enter Ctrl + C in the first window to stop event monitoring.

   a) Immediately after the SMON process is killed, Oracle Clusterware automatically moves the SERV1 service to its available instance RDB2 (in this example). Although Oracle Clusterware also restarts RDB1, SERV1 does not fall back to RDB1.

   b) Here is what you should see in the first session:

```
[oracle@vx0306 less07]$ cat sol_07_01_06_a.sh
#!/bin/ksh
#
# sol_07_01_06_a.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


/u01/crs11g/bin/evmwatch -A -t "@timestamp @@"
[oracle@vx0306 less07]$

[oracle@vx0306 less07]$ ./sol_07_01_06_a.sh
"14-Nov-2007 09:41:42 RAC: ora.vx0306.ASM1.asm: imcheck: "
"14-Nov-2007 09:41:42 RAC: ora.vx0306.ASM1.asm: imup: "
"14-Nov-2007 09:43:19 RAC: ora.RDB.RDB1.inst: imcheck: "
"14-Nov-2007 09:43:19 RAC: ora.RDB.RDB1.inst: imup: "
"14-Nov-2007 09:44:59 CRS is requested to perform action fail
on resource ora.RDB.RDB1.inst by Instance Monitor"
"14-Nov-2007 09:44:59 CRS ora.RDB.RDB1.inst is transitioning
from state ONLINE to state OFFLINE on member vx0306"
"14-Nov-2007 09:44:59 RAC: ora.RDB.RDB1.inst: down: "
"14-Nov-2007 09:44:59 CRS is requested to perform action fail
on resource ora.RDB.SERV1.RDB1.srv by ora.RDB.RDB1.inst"
```

Practice 7-1: Manage Services (continued)

```
"14-Nov-2007 09:44:59 CRS ora.RDB.SERV1.RDB1.srv is
transitioning from state ONLINE to state OFFLINE on member
vx0306"
"14-Nov-2007 09:44:59 RAC: ora.RDB.SERV1.RDB1.srv: down: "
"14-Nov-2007 09:44:59 CRS ora.RDB.SERV1.RDB1.srv stopped"
"14-Nov-2007 09:44:59 CRS ora.RDB.SERV1.RDB1.srv is
transitioning from state OFFLINE to state ONLINE on member
vx0313"
"14-Nov-2007 09:44:05 RAC: ora.RDB.SERV1.RDB1.srv: up: "
"14-Nov-2007 09:45:01 CRS ora.RDB.SERV1.RDB1.srv started on
member vx0313"
"14-Nov-2007 09:44:05 CRS ora.RDB.SERV1.RDB1.srv was modified"
"14-Nov-2007 09:45:06 RAC: ora.RDB.RDB1.inst: imstop: "
"14-Nov-2007 09:45:06 CRS ora.RDB.RDB1.inst stopped"
"14-Nov-2007 09:45:06 CRS ora.RDB.RDB1.inst is transitioning
from state OFFLINE to state ONLINE on member vx0306"
"14-Nov-2007 09:45:06 CRS is requested to perform action check
on resource ora.RDB.db by ora.RDB.RDB1.inst"
"14-Nov-2007 09:45:14 CRS is requested to perform action check
on resource ora.RDB.SERV1.cs by ora.RDB.SERV1.RDB1.srv"
"14-Nov-2007 09:45:14 CRS ora.RDB.SERV1.RDB1.srv was modified"
"14-Nov-2007 09:44:47 RAC: ora.RDB.RDB2.inst: imcheck: "
"14-Nov-2007 09:44:47 RAC: ora.RDB.RDB2.inst: imup: "
"14-Nov-2007 09:45:45 RAC: ora.RDB.RDB1.inst: up: "
"14-Nov-2007 09:45:45 RAC: ora.RDB.RDB1.inst: imstop: "
"14-Nov-2007 09:45:45 RAC: ora.RDB.RDB1.inst: imcheck: "
"14-Nov-2007 09:45:45 CRS ora.RDB.SERV1.RDB1.srv was modified"
"14-Nov-2007 09:45:45 CRS ora.RDB.RDB1.inst started on member
vx0306"
"14-Nov-2007 09:45:08 RAC: ora.vx0313.ASM2.asm: imcheck: "
"14-Nov-2007 09:45:08 RAC: ora.vx0313.ASM2.asm: imup: "
"14-Nov-2007 09:46:42 RAC: ora.vx0306.ASM1.asm: imcheck: "
"14-Nov-2007 09:46:42 RAC: ora.vx0306.ASM1.asm: imup: "
```

c) Here is what you should see in the second session:

```
[oracle@vx0306 less07]$ cat sol_07_01_06_b.sh
#!/bin/ksh
#
# sol_07_01_06_b.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`
```

Practice 7-1: Manage Services (continued)

```
I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

/u01/crs11g/bin/srvctl status database -d $DBNAME

ps -ef | grep "ora_smon_$I1NAME" | grep -v grep | grep -v
killsmon | awk '{print "kill -9 " $2 }' >
/home/oracle/solutions/less07/z.sh

chmod 777 /home/oracle/solutions/less07/z.sh


echo "Killing smon..."

/home/oracle/solutions/less07/z.sh

echo "waiting for instance back ..."

sleep 120

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect system/oracle1@$I1NAME
 select instance_name from v\$instance;
 show parameter service
 connect system/oracle1@$I2NAME
 select instance_name from v\$instance;
 show parameter service

EOF
[oracle@vx0306 less07]$

[oracle@vx0306 less07]$ ./sol_07_01_06_b.sh
Instance RDB1 is running on node vx0306
Instance RDB2 is running on node vx0313
Killing smon...
waiting for instance back ...

INSTANCE_NAME
----------------
RDB1


NAME                                 TYPE        VALUE
```

```
----------------------------------- ----------- -------------
-----------------
service_names                       string      RDB

INSTANCE_NAME
----------------
RDB2


NAME                                TYPE        VALUE
----------------------------------- ----------- -------------
-----------------
service_names                       string      SERV1, RDB
[oracle@vx0306 less07]$
```

7) Using Grid Control, check that SERV1 is running on RDB2.

   a) Connect as user SYS in Database Control. On the Cluster Home page, click the Availability tab. On the Availability page, click the Cluster Managed Database Services link in the Services section. On the Cluster and Database Login page, click Continue.

   b) On the Cluster Managed Database Services page, you should see SERV1 up and running on the second instance only with a warning.

   c) On Cluster Managed Database Services page, click the SERV1 link.

   d) On the SERV1 page, you should see that SERV1 is stopped on the preferred instance and up on the available instance.

Practice 7-2: Monitor Services

The goal of this practice is to use Database Control to determine the amount of resources used by sessions executing under a particular service. You also use Database Control to relocate a service to another instance.

1) Execute the createjfv.sh script. This script creates a new user called JFV identified by the password JFV. The default tablespace of this user is USERS, and its temporary tablespace is TEMP. This new user has the CONNECT, RESOURCE, and DBA roles.

```
[oracle@vx0306 less07]$ cat createjfv.sh
#!/bin/ksh
#
# createjfv.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
 drop user JFV cascade;
 create user JFV identified by jfv default tablespace users
temporary tablespace temp;
 grant connect, resource, dba to JFV;

EOF
[oracle@vx0306 less07]$

[oracle@vx0306 less07]$ ./createjfv.sh

User dropped.


User created.
```

Practice 7-2: Monitor Services (continued)

```
Grant succeeded.

[oracle@vx0306 less07]$
```

2) From a terminal session connected to node1, using SQL*Plus, connect to SERV1 with user JFV. 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.

```
[oracle@vx0306 less07]$ cat sol_07_02_02.sh
#!/bin/ksh
#
# sol_07_02_02.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect jfv/jfv@SERV1
 select instance_name from v\$instance;
 select count(*) from dba_objects,dba_objects,dba_objects;

EOF
[oracle@vx0306 less07]$

[oracle@vx0306 less07]$ ./sol_07_02_02.sh

INSTANCE_NAME
----------------
RDB2

…
```

Practice 7-2: Monitor Services (continued)

3) After a while, go to the Database Control Top Consumers page from the Cluster Database page. Connect as user SYS. Then check that SERV1 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 the View Data dropdown list is set to Real Time: 15 Second Refresh. Wait for the page to be refreshed a couple of times. Little by little, SERV1 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 the View Data dropdown list is set to Real Time: 15 Second Refresh, and View dropdown list is set to Active Services. You can click the "+" icon on the left of the SERV1 link to expand the service. This shows you the list of instances currently running the service. You can also click the SERV1 link itself to look at the detailed Statistics of the corresponding service.

4) Check statistics on your service with gv$service_stats from a SQL*Plus session connected as SYSDBA.

```
[oracle@vx0306 less07]$ cat sol_07_02_04.sh
#!/bin/ksh
#
# sol_07_02_04.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect system/oracle1@$I1NAME
```

```
 select instance_name from v\$instance;
 select stat_name, sum(value) from gv\$service_stats where
service_name = 'SERV1' group by stat_name;

EOF
[oracle@vx0306 less07]$

[oracle@vx0306 less07]$ ./sol_07_02_04.sh

INSTANCE_NAME
----------------
RDB1


STAT_NAME
SUM(VALUE)
----------------------------------------------------------------
-- ----------
user calls
21
DB CPU
212910000
redo size
1076
db block changes
6
DB time
216690394
user rollbacks
0
gc cr blocks received
8
gc cr block receive time
0
gc current blocks received
6
opened cursors cumulative
42
workarea executions - multipass
0

STAT_NAME
SUM(VALUE)
----------------------------------------------------------------
-- ----------
session cursor cache hits
8
user I/O wait time
10798
parse count (total)
41
physical reads                                                 2
```

```
gc current block receive time                                    0
workarea executions - optimal                                   11
concurrency wait time                                         7522
parse time elapsed                                           78594
physical writes                                                  0
workarea executions - onepass                                    0
execute count                                                   41

STAT_NAME                                                SUM(VALUE)
---------------------------------------------------------------
-- ----------
session logical reads                                         3377
cluster wait time                                             7896
application wait time                                            0
logons cumulative                                                1
sql execute elapsed time                                 216621181
user commits                                                     0

28 rows selected.

[oracle@vx0306 less07]$
```

5) Using Database Control, relocate SERV1 to its preferred instance.

   a) Click the Cluster Database link at the top of your current Database Control page.

   b) This takes you back to the Performance page for your database. Click the Availability tab.

   c) On the Availability page, click Cluster Managed Database Services.

   d) On the Cluster Managed Database Services: Cluster and Database Login page, click Continue.

   e) On the Cluster Managed Database Services page, click SERV1.

   f) On the Cluster Managed Database Services: SERV1 page, select the Available instance (RDB2 in this example) and click Relocate.

   g) On the Relocate Service from instance page, make sure the Preferred instance (RDB1 in this example) is selected, and click OK.

   h) On the Cluster Managed Database Service: SERV1 page, you should see a successful Service Relocate status at the top of the page. Also, the Instances table should indicate that SERV1 is now running on its preferred instance.

6) What happens to your already connected SERV1 session running on the second instance?

   a) You can see that although the service has been switched to RDB1, your session is still executing under SERV1 on RDB2. So, if you want to manually relocate a service, you should make sure that no session is currently connected. From a terminal window, execute the following.

```
[oracle@vx0306 less07]$ cat sol_07_02_06.sh
#!/bin/ksh
```

```
#
# sol_07_02_06.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect system/oracle1@$I1NAME
 select inst_id, username, service_name from gv\$session where
username = 'JFV';

EOF
[oracle@vx0306 less07]$ ./sol_07_02_06.sh

   INST_ID USERNAME
---------- ------------------------------
SERVICE_NAME
----------------------------------------------------------------
--
         2 JFV
SERV1


[oracle@vx0306 less07]$
```

b) If your first session is still executing the query, stop its execution by pressing [Ctrl] + [C].

Practice 7-3: Alert Thresholds and Services

The goal of this practice is to set thresholds to service SERV1, and use Database Control to monitor the response time metric for this service. In this practice, you create the following configuration:

Service Name	Usage	Preferred Instances	Available Instances	Response Time (sec)– Warning/Critical	
SERV1	Client service	RDBA1	RDBA2	0.4	1.0

1) Set alert thresholds for your service SERV1 by using Database Control. Specify the values defined above.
   a) From the Cluster Database Home page, click the link corresponding to your first instance in the Instances table. This is the instance currently running SERV1.
   b) On the Database Instance page, click Metric and Policy settings in the Related Links section at the bottom of the page.
   c) On the Metric and Policy Settings page, select All metrics from the View dropdown list.
   d) Scroll down the Metric and Policy Settings page until you find the Service Response Time (per user call) (microseconds) metric.
   e) On the same line, click the corresponding multi-pens icon in the last column (Edit column).
   f) On the Edit Advanced Settings: Service Response Time (per user call) (microseconds) page, click Add.
   g) The Monitored Objects table should now show two entries.
   h) Enter SERV1 in the Service Name field, 40000000 in the Warning Threshold field, and 100000000 in the Critical Threshold field. Make sure the corresponding line is selected, and click Continue.
   i) 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.
   j) On the Confirmation page, you can see an Update succeeded message. Click OK.
   k) This takes you back to the Database Instance page.
2) Use Database Control to print the Service Response Time Metric Value graphic for SERV1.
   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 SERV1 link in the Service Name column.

d) On the Service Response Time (per user call) (microseconds): Service Name SERV1: 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 SERV1 page with your warning and critical thresholds set correctly.

3) Execute the sol_07_03_03.sh script to generate workload on your database. Looking at the Service Response time graphic for SERV1, what do you observe?

```
[oracle@vx0306 less07]$ cat sol_07_03_03.sh
#!/bin/ksh
#
# sol_07_03_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect jfv/jfv@SERV1
 DECLARE
  t number;
 BEGIN
  for i in 1..10000 loop
   select count(*) into t from dba_objects;
  end loop;
 END;
 /

EOF
[oracle@vx0306 less07]$

[oracle@vx0306 less07]$ ./sol_07_03_03.sh
```

```
… Do not wait!

PL/SQL procedure successfully completed.

[oracle@vx0306 less07]$
```

a) Still looking at the Service Response Time (per user call) (microseconds): Service Name SERV1 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 SERV1 link to directly go to the Cluster Managed Database Services: SERV1 page after you clicked Continue on the Login page. The SERV1 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.

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 SERV1 in the Metric Thresholds table.

d) On the line corresponding to the SERV1 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.

Practice 7-3: Alert Thresholds and Services (continued)

5) Execute sol_07_03_05.sh to cleanup your environment.

```
[oracle@vx0306 less07]$ cat sol_07_03_05.sh
#!/bin/ksh
#
# sol_07_03_05.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/srvctl stop service -d $DBNAME -s SERV1 -f
/u01/crs11g/bin/srvctl remove service -d $DBNAME -s SERV1

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect system/oracle1
 drop user jfv cascade;
 execute dbms_service.delete_service('SERV1');

EOF
[oracle@vx0306 less07]$

[oracle@vx0306 less07]$ ./sol_07_03_05.sh
SERV1 PREF: RDB1 AVAIL: RDB2
Remove service SERV1 from the database RDB? (y/[n]) y
…

PL/SQL procedure successfully completed.

[oracle@vx0306 less07]$
```

Practices for Lesson 8

Practice 8-1: Create a Server-Side Callout

The goal of this practice is to create a server-side callout program to trap various database events.

1) Write a shell script that is able to trap FAN events generated by CRS. The events that need to be trapped must be for your database (RDB in this example), and the event types that need to be trapped are SERVICEMEMBER (up/down), SERVICE (up/down), and INSTANCE (up/down). The callout script should create a file that logs the events that are trapped. The log file must be located in the $ORA_CRS_HOME/racg/log directory. Make sure that the callout script is deployed on both nodes.

```
[oracle@vx0306 less08]$ cat sol_08_01_01.sh
#!/bin/ksh
#
# sol_08_01_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

sed 's/RACDATABASE/'$DBNAME'/'
/home/oracle/solutions/less08/wrong_callout1.sh >
/home/oracle/solutions/less08/callout1.sh

cat /home/oracle/solutions/less08/callout1.sh

cp /home/oracle/solutions/less08/callout1.sh
/u01/crs11g/racg/usrco/

chmod 777 /u01/crs11g/racg/usrco/callout1.sh

scp /u01/crs11g/racg/usrco/callout1.sh
$z:/u01/crs11g/racg/usrco/

ssh $z chmod 777 /u01/crs11g/racg/usrco/callout1.sh
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ cat wrong_callout1.sh
#!/bin/sh

NOTIFY_EVENTTYPE=$1
AWK=/usr/bin/awk
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

if ((( [ $NOTIFY_EVENTTYPE = "SERVICEMEMBER" ] ||
       [ $NOTIFY_EVENTTYPE = "SERVICE"       ] ||
       [ $NOTIFY_EVENTTYPE = "INSTANCE"      ]      \
     ) &&                                           \
     ( [ $NOTIFY_STATUS    = "up"            ] ||   \
       [ $NOTIFY_STATUS    = "down"          ]      \
     )) &&                                          \
     ( [ $NOTIFY_DATABASE  = "RACDATABASE"          ]      \
     ))
then
     echo $* >> /u01/crs11g/racg/log/crsevtco.log
fi
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ ./sol_08_01_01.sh
#!/bin/sh

NOTIFY_EVENTTYPE=$1
AWK=/usr/bin/awk
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

if ((( [ $NOTIFY_EVENTTYPE = "SERVICEMEMBER" ] ||
```

```
        [ $NOTIFY_EVENTTYPE = "SERVICE"       ] ||
        [ $NOTIFY_EVENTTYPE = "INSTANCE"      ]     \
     ) &&                                           \
     ( [ $NOTIFY_STATUS    = "up"             ] ||  \
        [ $NOTIFY_STATUS    = "down"          ]     \
     )) &&                                          \
     ( [ $NOTIFY_DATABASE  = "RDB"           ]     \
     ))
then
     echo $* >> /u01/crs11g/racg/log/crsevtco.log
fi
callout1.sh
100% 1210     1.2KB/s   00:00
[oracle@vx0306 less08]$
```

2) Create two new services. The first one is called SCO1 and has RDB1 as its preferred instance, and RDB2 as its available instance. The second is called SCO2 and has RDB2 as its preferred instance, and RDB1 as its available instance.

```
[oracle@vx0306 less08]$ cat sol_08_01_02.sh
#!/bin/ksh
#
# sol_08_01_02.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/srvctl add service -d $DBNAME -s SCO1 -r
$I1NAME -a $I2NAME

/u01/crs11g/bin/srvctl add service -d $DBNAME -s SCO2 -r
$I2NAME -a $I1NAME
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ ./sol_08_01_02.sh
[oracle@vx0306 less08]$
```

3) Run the following command on one terminal window on the first node. We call that window TW1N1: evmwatch -A -t "@timestamp @@"
Do not wait; proceed with the next step.

```
[oracle@vx0306 less08]$ cat sol_08_01_03.sh
#!/bin/ksh
#
# sol_08_01_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


/u01/crs11g/bin/evmwatch -A -t "@timestamp @@"
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ ./sol_08_01_03.sh
```

4) Run the following command on one terminal window on the second node. We call that window TW1N2: evmwatch -A -t "@timestamp @@"
Do not wait; proceed with the next step.

```
[oracle@vx0313 less08]$ cat sol_08_01_04.sh
#!/bin/ksh
#
# sol_08_01_04.sh
#
# Must be executed on NODE2 !!!!!!!!!!
#


/u01/crs11g/bin/evmwatch -A -t "@timestamp @@"
[oracle@vx0313 less08]$

[oracle@vx0313 less08]$ ./sol_08_01_04.sh
```


5) From another terminal window connected to node1 (TW2N1), start the SCO1 service using srvctl. When finished, look at terminal windows TW1N1 and TW1N2, and then look at the generated log files in the /u01/crs1020/racg/log directory by using TW2N1. What do you observe?

   a) You should see that only the log on the first instance contains the start of SCO1.

   b) On TW2N1 you can see:

```
[oracle@vx0306 less08]$ cat sol_08_01_05.sh
#!/bin/ksh
#
# sol_08_01_05.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
```

```
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

/u01/crs11g/bin/srvctl start service -d $DBNAME -s SCO1

sleep 10

cat /u01/crs11g/racg/log/crsevtco.log

ssh $z cat /u01/crs11g/racg/log/crsevtco.log
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ ./sol_08_01_05.sh
SERVICE VERSION=1.0 service=SCO1 database=RDB instance=
host=vx0306 status=up reason=user timestamp=15-Nov-2007
03:11:27
SERVICEMEMBER VERSION=1.0 service=SCO1 database=RDB
instance=RDB1 host=vx0306 status=up reason=user card=1
timestamp=15-Nov-2007 03:11:27
cat: /u01/crs11g/racg/log/crsevtco.log: No such file or
directory
[oracle@vx0306 less08]$
```

c) On TW1N1 you can see:

```
[oracle@vx0306 less08]$ ./sol_08_01_03.sh
"15-Nov-2007 03:11:27 CRS ora.RDB.SCO1.cs is transitioning
from state OFFLINE to state ONLINE on member vx0306"
"15-Nov-2007 03:11:27 CRS ora.RDB.SCO1.RDB1.srv is
transitioning from state OFFLINE to state ONLINE on member
vx0306"
"15-Nov-2007 03:11:27 RAC: ora.RDB.SCO1.RDB1.srv: up: "
"15-Nov-2007 03:11:27 RAC: ora.RDB.SCO1.cs: up: "
"15-Nov-2007 03:11:27 CRS ora.RDB.SCO1.RDB1.srv started on
member vx0306"
"15-Nov-2007 03:11:27 CRS ora.RDB.SCO1.RDB1.srv was modified"
"15-Nov-2007 03:11:28 CRS ora.RDB.SCO1.cs started on member
vx0306"
"15-Nov-2007 03:10:56 RAC: ora.RDB.RDB2.inst: imcheck: "
"15-Nov-2007 03:10:56 RAC: ora.RDB.RDB2.inst: imup: "
"15-Nov-2007 03:11:58 RAC: ora.RDB.RDB1.inst: imcheck: "
"15-Nov-2007 03:11:58 RAC: ora.RDB.RDB1.inst: imup: "
```

d) On TW1N2 you can see:

```
[oracle@vx0313 less08]$ ./sol_08_01_04.sh
"15-Nov-2007 03:11:27 CRS ora.RDB.SCO1.cs is transitioning
from state OFFLINE to state ONLINE on member vx0306"
```

```
"15-Nov-2007 03:11:27 CRS ora.RDB.SCO1.RDB1.srv is
transitioning from state OFFLINE to state ONLINE on member
vx0306"
"15-Nov-2007 03:11:27 RAC: ora.RDB.SCO1.RDB1.srv: up: "
"15-Nov-2007 03:11:27 RAC: ora.RDB.SCO1.cs: up: "
"15-Nov-2007 03:11:27 CRS ora.RDB.SCO1.RDB1.srv started on
member vx0306"
"15-Nov-2007 03:11:27 CRS ora.RDB.SCO1.RDB1.srv was modified"
"15-Nov-2007 03:11:28 CRS ora.RDB.SCO1.cs started on member
vx0306"
"15-Nov-2007 03:10:56 RAC: ora.RDB.RDB2.inst: imcheck: "
"15-Nov-2007 03:10:56 RAC: ora.RDB.RDB2.inst: imup: "
"15-Nov-2007 03:11:58 RAC: ora.RDB.RDB1.inst: imcheck: "
"15-Nov-2007 03:11:58 RAC: ora.RDB.RDB1.inst: imup: "
"15-Nov-2007 03:11:16 RAC: ora.vx0313.ASM2.asm: imcheck: "
"15-Nov-2007 03:11:16 RAC: ora.vx0313.ASM2.asm: imup: "
```

6) From TW2N1, start the SCO2 service using srvctl. When finished, look at terminal windows TW1N1 and TW1N2, and then look at the generated log files in the /u01/crs1020/racg/log directory by using TW2N1. What do you observe?

   a) You should see that only the log on the second instance contains the start of SCO2.

   b) On TW2N1 you can see:

```
[oracle@vx0306 less08]$ cat sol_08_01_06.sh
#!/bin/ksh
#
# sol_08_01_06.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

/u01/crs11g/bin/srvctl start service -d $DBNAME -s SCO2

sleep 10

cat /u01/crs11g/racg/log/crsevtco.log

ssh $z cat /u01/crs11g/racg/log/crsevtco.log
[oracle@vx0306 less08]$
```

Practice 8-1: Create a Server-Side Callout (continued)

```
[oracle@vx0306 less08]$ ./sol_08_01_06.sh
SERVICE VERSION=1.0 service=SCO1 database=RDB instance=
host=vx0306 status=up reason=user timestamp=15-Nov-2007
03:11:27
SERVICEMEMBER VERSION=1.0 service=SCO1 database=RDB
instance=RDB1 host=vx0306 status=up reason=user card=1
timestamp=15-Nov-2007 03:11:27
SERVICE VERSION=1.0 service=SCO2 database=RDB instance=
host=vx0313 status=up reason=user timestamp=15-Nov-2007
03:13:52
SERVICEMEMBER VERSION=1.0 service=SCO2 database=RDB
instance=RDB2 host=vx0313 status=up reason=user card=1
timestamp=15-Nov-2007 03:13:52
[oracle@vx0306 less08]$
```

c) On TW1N1 you can see:

```
"15-Nov-2007 03:14:50 CRS ora.RDB.SCO2.cs is transitioning
from state OFFLINE to state ONLINE on member vx0313"
"15-Nov-2007 03:13:52 CRS ora.RDB.SCO2.RDB2.srv is
transitioning from state OFFLINE to state ONLINE on member
vx0313"
"15-Nov-2007 03:13:52 RAC: ora.RDB.SCO2.RDB2.srv: up: "
"15-Nov-2007 03:13:52 RAC: ora.RDB.SCO2.cs: up: "
"15-Nov-2007 03:14:50 CRS ora.RDB.SCO2.cs started on member
vx0313"
"15-Nov-2007 03:13:52 CRS ora.RDB.SCO2.RDB2.srv started on
member vx0313"
"15-Nov-2007 03:13:53 CRS ora.RDB.SCO2.RDB2.srv was modified"
```

d) On TW1N2 you can see:

```
"15-Nov-2007 03:14:50 CRS ora.RDB.SCO2.cs is transitioning
from state OFFLINE to state ONLINE on member vx0313"
"15-Nov-2007 03:13:52 CRS ora.RDB.SCO2.RDB2.srv is
transitioning from state OFFLINE to state ONLINE on member
vx0313"
"15-Nov-2007 03:13:52 RAC: ora.RDB.SCO2.RDB2.srv: up: "
"15-Nov-2007 03:13:52 RAC: ora.RDB.SCO2.cs: up: "
"15-Nov-2007 03:14:50 CRS ora.RDB.SCO2.cs started on member
vx0313"
"15-Nov-2007 03:13:52 CRS ora.RDB.SCO2.RDB2.srv started on
member vx0313"
"15-Nov-2007 03:13:53 CRS ora.RDB.SCO2.RDB2.srv was modified"
```

7) From TW2N1, stop both services using srvctl, and then remove both services. When you are finished, remove callout1.sh script from the /u01/crs11g/racg/usrco directory as well as /u01/crs11g/racg/log/crsevtco.log from both nodes. When you are finished, stop both evmwatch sessions with Ctrl-C.

```
[oracle@vx0306 less08]$ cat sol_08_01_07.sh
```

```
#!/bin/ksh
#
# sol_08_01_07.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/srvctl stop service -d $DBNAME -s SCO1

/u01/crs11g/bin/srvctl stop service -d $DBNAME -s SCO2

/u01/crs11g/bin/srvctl remove service -d $DBNAME -s SCO1

/u01/crs11g/bin/srvctl remove service -d $DBNAME -s SCO2

rm /u01/crs11g/racg/usrco/callout1.sh

rm /u01/crs11g/racg/log/crsevtco.log

ssh $z rm /u01/crs11g/racg/usrco/callout1.sh

ssh $z rm /u01/crs11g/racg/log/crsevtco.log

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
 exec DBMS_SERVICE.DELETE_SERVICE('SCO1');
 exec DBMS_SERVICE.DELETE_SERVICE('SCO2');

EOF
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ ./sol_08_01_07.sh
SCO1 PREF: RDB1 AVAIL: RDB2
Remove service SCO1 from the database RDB? (y/[n]) y
SCO2 PREF: RDB2 AVAIL: RDB1
Remove service SCO2 from the database RDB? (y/[n]) y
```

Practice 8-1: Create a Server-Side Callout (continued)

```
PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.

[oracle@vx0306 less08]$
```

Practice 8-2: Use Load Balancing Advisory

The goal of this practice is to test the Load Balancing Advisory on the listener side only to determine how connections are spread across your instances while running an OLTP-type of workload.

1) Create two new services using Database Control:
   First one is called SNOLBA and should be defined with the following parameters: goal => DBMS_SERVICE.GOAL_NONE, clb_goal => DBMS_SERVICE.CLB_GOAL_LONG.
   Second one is called SLBA and should be defined with the following parameters: goal => DBMS_SERVICE.GOAL_SERVICE_TIME, clb_goal => DBMS_SERVICE.CLB_GOAL_SHORT.
   Basically, SNOLBA does not use the Load Balancing Advisory, while SLBA uses it. Make sure that both services have both instances as preferred, and start them both.

   a) Connect to your Database Control console as user SYS.

   b) On the Cluster Database Home page, click the Availability tab.

   c) On the Availability page, click the Cluster Managed Database Services link in the Services section.

   d) On the Cluster and Database Login page, enter all necessary credentials if not done already, and click Continue.

   e) On the Cluster Managed Database Services page, click the Create Service button.

   f) On the Create Service page, enter `SNOLBA` in the Service Name field, and select Long as the Connection Load Balancing Goal option. Then click OK.

   g) You should now see a Create Service: Successful status on the Cluster Managed Database Services. On this page, you should also see that SNOLBA is started on both instances.

   h) On the Cluster Managed Database Services page, click Create Service.

   i) On the Create Service page, enter `SLBA` in the Service Name field, and select Short as the Connection Load Balancing Goal option.

   j) In the Notification Properties section, check Enable Load Balancing Advisory, and make sure Service Time option is selected. Then click OK.

   k) You should now see a Create Service: Successful status on the Cluster Managed Database Services. On this page, you should also see that SLBA is started on both instances.

2) Add corresponding service names to your tnsnames.ora files on both nodes. We recommend that you add them to both tnsnames.ora files: the one residing in your ASM home and the one residing in your Database home.

```
[oracle@vx0306 less08]$ cat sol_08_02_02.sh
#!/bin/ksh
#
# sol_08_02_02.sh
#
```

```
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

sed 's/NODE1/'$y'/'
/home/oracle/solutions/less08/wrong_tnslba.ora >
/home/oracle/solutions/less08/w1tnslba.ora

sed 's/NODE2/'$z'/' /home/oracle/solutions/less08/w1tnslba.ora
> /home/oracle/solutions/less08/tnslba.ora

cp
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora
.bak1
cat /home/oracle/solutions/less08/tnslba.ora >>
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora

cp
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a.bak2
cat /home/oracle/solutions/less08/tnslba.ora >>
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a

ssh $z cp
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora
.bak1

ssh $z cp
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a.bak2

scp /home/oracle/solutions/less08/tnslba.ora
$z:/home/oracle/solutions/less08

ssh $z "cat /home/oracle/solutions/less08/tnslba.ora >>
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora
"
```

```
ssh $z "cat /home/oracle/solutions/less08/tnslba.ora >>
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a"
[oracle@vx0306 less08]$


[oracle@vx0306 less08]$ cat wrong_tnslba.ora

SNOLBA =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = NODE1-vip)(PORT =
1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = NODE2-vip)(PORT =
1521))
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = SNOLBA)
    )
  )

SLBA =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = NODE1-vip)(PORT =
1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = NODE2-vip)(PORT =
1521))
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = SLBA)
    )
  )
[oracle@vx0306 less08]$


[oracle@vx0306 less08]$ ./sol_08_02_02.sh
tnslba.ora                                    100%  534
0.5KB/s   00:00
[oracle@vx0306 less08]$
```

3) From a terminal window, execute the createfan.sh script. This script creates a simple table used by the following scripts in this practice. This script is located in your $HOME/solutions/less08 directory.

```
[oracle@vx0306 less08]$ cat createfan.sh
#!/bin/ksh
#
# createfan.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#
```

```
y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
 drop user JFV cascade;
 create user JFV identified by jfv default tablespace users
temporary tablespace temp;
 grant connect, resource, dba to JFV;
 connect jfv/jfv
 create table fan(c number);

EOF
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ ./createfan.sh

User dropped.


User created.


Grant succeeded.


Table created.

[oracle@vx0306 less08]$
```

4) Make sure you keep your terminal window (referred to as TW1) open. You use it later.

5) Create a new terminal window from where you will start the workload. This terminal window is referred to as TW2.

Practice 8-2: Use Load Balancing Advisory (continued)

6) Create a new terminal window from where you will execute the following SQL statement as SYSDBA under SQL*Plus: `select inst_id,count(*) from gv$session where username='JFV' group by inst_id order by inst_id;`
This statement count the number of sessions connected as user JFV on both instances. This terminal window is referred to as TW3.

```
[oracle@vx0306 less08]$ cat sol_08_02_06.sh
#!/bin/ksh
#
# sol_08_02_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus /NOLOG @sol_08_02_06_00.sql



[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ cat sol_08_02_06_00.sql
connect / as sysdba
host echo "execute sol_08_02_06.sql from here"
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ cat sol_08_02_06_00.sql
connect / as sysdba
host echo "execute sol_08_02_06.sql from here"
[oracle@vx0306 less08]$ cat sol_08_02_06.sql

select inst_id,count(*) from gv$session where username='JFV'
group by inst_id order by inst_id;
[oracle@vx0306 less08]$


[oracle@vx0306 less08]$ ./sol_08_02_06.sh
```

```
SQL*Plus: Release 11.1.0.6.0 - Production on Thu Nov 15
05:02:47 2007

Copyright (c) 1982, 2007, Oracle.  All rights reserved.

Connected.
execute sol_08_02_06.sql from here

SQL> @sol_08_02_06

no rows selected

SQL>
```

7) Create two additional terminal windows from where you will run the primes executable to generate CPU load on your first node. These terminal windows are called TW4 and TW5, respectively.

8) In TW2, execute the startfanload.sh script using SNOLBA as the first argument to the command.

```
[oracle@vx0306 less08]$ cat startfanload.sh
/home/oracle/solutions/less08/fan.sh $1 &
/home/oracle/solutions/less08/fan.sh $1 &
/home/oracle/solutions/less08/fan.sh $1 &
/home/oracle/solutions/less08/fan.sh $1 &
/home/oracle/solutions/less08/fan.sh $1 &
/home/oracle/solutions/less08/fan.sh $1 &
/home/oracle/solutions/less08/fan.sh $1 &
/home/oracle/solutions/less08/fan.sh $1 &
/home/oracle/solutions/less08/fan.sh $1 &
/home/oracle/solutions/less08/fan.sh $1 &
/home/oracle/solutions/less08/fan.sh $1 &
/home/oracle/solutions/less08/fan.sh $1 &
/home/oracle/solutions/less08/fan.sh $1 &
/home/oracle/solutions/less08/fan.sh $1 &
/home/oracle/solutions/less08/fan.sh $1 &
/home/oracle/solutions/less08/fan.sh $1 &
/home/oracle/solutions/less08/fan.sh $1 &
/home/oracle/solutions/less08/fan.sh $1 &
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ cat fan.sh
#!/bin/ksh
#
# fan.sh
#

users=120
x=1
y=$users
```

```
UNPW="jfv/jfv@"$1

ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1
export ORACLE_HOME

while [ $x -le $y ]
do
    /u01/app/oracle/product/11.1.0/db_1/bin/sqlplus -s $UNPW
@fan.sql
done
[oracle@vx0306 less08]$
[oracle@vx0306 less08]$ cat fan.sql
begin
for i in 1..1000 loop
insert into fan select sid from v$mystat where rownum<2;
commit;
delete fan where c in (select sid from v$mystat where
rownum<2);
commit;
end loop;
end;
/
exit;
[oracle@vx0306 less08]$



[oracle@vx0306 less08]$ ./startfanload.sh SNOLBA
[oracle@vx0306 less08]$ begin
*
ERROR at line 1:
ORA-00060: deadlock detected while waiting for resource
ORA-06512: at line 5


begin
*
ERROR at line 1:
ORA-00060: deadlock detected while waiting for resource
ORA-06512: at line 5


begin
*
ERROR at line 1:
ORA-00060: deadlock detected while waiting for resource
ORA-06512: at line 5


begin
*
ERROR at line 1:
```

```
ORA-00060: deadlock detected while waiting for resource
ORA-06512: at line 5



PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.

begin
*
ERROR at line 1:
ORA-00060: deadlock detected while waiting for resource
ORA-06512: at line 5


PL/SQL procedure successfully completed.
```

9) Start the primes executable in both TW4 and TW5.

   a) In TW4, you can see:

```
[oracle@vx0306 less08]$ ./primes
DBG: Got primestore ptr is 0xb1ba7008
DBG: Initialized.
next_prime=0xb1ba7010,cur_prime=0xb1ba700c,*next_prime=0,*cur_
prime=3
pp_10K=1700
pp_10K=1422
pp_10K=1353
pp_10K=1312
pp_10K=1275
pp_10K=1258
pp_10K=1218
pp_10K=1222
pp_10K=1199
pp_10K=1203 pp_100K=13162
pp_10K=1178
pp_10K=1165
```

   b) In TW5, you can see:

```
[oracle@vx0306 less08]$ ./primes
DBG: Got primestore ptr is 0xb1bbf008
```

```
DBG: Initialized.
next_prime=0xb1bbf010,cur_prime=0xb1bbf00c,*next_prime=0,*cur_
prime=3
pp_10K=1700
pp_10K=1422
pp_10K=1353
pp_10K=1312
pp_10K=1275
pp_10K=1258
pp_10K=1218
pp_10K=1222
pp_10K=1199
pp_10K=1203 pp_100K=13162
pp_10K=1178
pp_10K=1165
pp_10K=1177
pp_10K=1151
pp_10K=1149
```

10) Monitor TW3 by repeating the execution of the SQL statement. What do you observe?

a) You should see approximately the same number of connections on both nodes—perhaps a bit more on the second node because the first node has much more load than the second. Wait at least three minutes before using figures for your analysis. This is the time necessary before the LBA algorithm takes place.

b) In TW3, you can see:

```
[oracle@vx0306 less08]$ ./sol_08_02_06.sh

SQL*Plus: Release 11.1.0.6.0 - Production on Thu Nov 15
05:02:47 2007

Copyright (c) 1982, 2007, Oracle.  All rights reserved.

Connected.
execute sol_08_02_06.sql from here

SQL> @sol_08_02_06

no rows selected

SQL> /

no rows selected

SQL> /

   INST_ID   COUNT(*)
---------- ----------
         1          9
         2          9
```

```
SQL> /

   INST_ID   COUNT(*)
---------- ----------
         1          8
         2         10

SQL> /

   INST_ID   COUNT(*)
---------- ----------
         1          8
         2         10

SQL> /

   INST_ID   COUNT(*)
---------- ----------
         1          8
         2         10

SQL> /

   INST_ID   COUNT(*)
---------- ----------
         1          9
         2          9

SQL> /

   INST_ID   COUNT(*)
---------- ----------
         1          8
         2         10

SQL> /

   INST_ID   COUNT(*)
---------- ----------
         1          9
         2          9

SQL> /

   INST_ID   COUNT(*)
---------- ----------
         1          8
         2         10

SQL>
```

Practice 8-2: Use Load Balancing Advisory (continued)

11) Stop primes generation by pressing [Ctrl] + [C] in both corresponding terminal windows: TW4 and TW5.

    a) In TW4, you can see

```
pp_10K=786 pp_100K=8091
pp_10K=799
pp_10K=767
pp_10K=792
pp_10K=838
pp_10K=802
pp_10K=806
pp_10K=813
pp_10K=818
pp_10K=796
pp_10K=790 pp_100K=8021
pp_10K=816
pp_10K=819
pp_10K=768
pp_10K=796

[oracle@vx0306 less08]$
```

    b) In TW5, you can see:

```
pp_10K=817
pp_10K=806
pp_10K=818
pp_10K=793
pp_10K=804
pp_10K=805 pp_100K=8042
pp_10K=798
pp_10K=791
pp_10K=803
pp_10K=789
pp_10K=801
pp_10K=833
pp_10K=812
pp_10K=792
pp_10K=783
pp_10K=822 pp_100K=8024
pp_10K=811
pp_10K=817
pp_10K=809
pp_10K=790

[oracle@vx0306 less08]$
```

12) In TW4, execute stopfanload.sh script to stop the workload.

```
[oracle@vx0306 less08]$ cat stopfanload.sh
#!/bin/ksh
#
# stopfanload.sh
```

```
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

ps -ef | grep "fan.sh" | awk '{print "kill -9 " $2 }' >
/home/oracle/solutions/less08/x.sh

chmod 777 /home/oracle/solutions/less08/x.sh

/home/oracle/solutions/less08/x.sh


ps -ef | grep "fan.sql" | awk '{print "kill -9 " $2 }' >
/home/oracle/solutions/less08/x.sh

chmod 777 /home/oracle/solutions/less08/x.sh

/home/oracle/solutions/less08/x.sh

$ORACLE_HOME/bin/sqlplus system/oracle1@$I1NAME
@/home/oracle/solutions/less08/kill_fan.sql

$ORACLE_HOME/bin/sqlplus system/oracle1@$I1NAME
@/tmp/drop_fan.sql

rm /tmp/drop_fan.sql

$ORACLE_HOME/bin/sqlplus system/oracle1@$I2NAME
@/home/oracle/solutions/less08/kill_fan.sql

$ORACLE_HOME/bin/sqlplus system/oracle1@$I2NAME
@/tmp/drop_fan.sql

rm /tmp/drop_fan.sql
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ cat kill_fan.sql
```

```
set head off
set timing off
set feedback off;
set pagesize 0
set verify off

spool /tmp/drop_fan.sql;

select 'alter system kill session ''' || sid || ',' || serial#
|| ''';'
from v$session
where username='JFV';

select 'exit;' from dual;

spool off

exit;
[oracle@vx0306 less08]$


[oracle@vx0306 less08]$ ./stopfanload.sh
/home/oracle/solutions/less08/x.sh: line 19: kill: (17917) -
No such process

SQL*Plus: Release 11.1.0.6.0 - Production on Thu Nov 15
05:11:55 2007

Copyright (c) 1982, 2007, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 -
Production
With the Partitioning, Real Application Clusters, OLAP, Data
Mining
and Real Application Testing options

alter system kill session '96,12499';
alter system kill session '101,17760';
alter system kill session '102,13977';
alter system kill session '103,12859';
alter system kill session '106,4508';
alter system kill session '109,12588';
alter system kill session '111,13128';
alter system kill session '114,27665';
exit;
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.1.0.6.0 - Production
With the Partitioning, Real Application Clusters, OLAP, Data
Mining
and Real Application Testing options
```

```
SQL*Plus: Release 11.1.0.6.0 - Production on Thu Nov 15
05:11:55 2007

Copyright (c) 1982, 2007, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 -
Production
With the Partitioning, Real Application Clusters, OLAP, Data
Mining
and Real Application Testing options


System altered.


System altered.


System altered.


System altered.


System altered.


System altered.

alter system kill session '111,13128'
*
ERROR at line 1:
ORA-00030: User session ID does not exist.


alter system kill session '114,27665'
*
ERROR at line 1:
ORA-00030: User session ID does not exist.



…


Disconnected from Oracle Database 11g Enterprise Edition
Release 11.1.0.6.0 - Production
With the Partitioning, Real Application Clusters, OLAP, Data
Mining
and Real Application Testing options
```

```
[oracle@vx0306 less08]$
```

13) Repeat steps 8-9-10-11-12 using the command "startfanload SLBA" in step 8. What do you observe this time? Wait for at least five minutes before doing your analysis. This leaves enough time for the LBA to take place.

a) You should see that the listener distributes more connections to the second instance.

b) In TW3, you can see:

```
SQL> /

   INST_ID   COUNT(*)
---------- ----------
         1          4
         2         11

SQL> /

   INST_ID   COUNT(*)
---------- ----------
         1         11
         2          7

SQL> /

   INST_ID   COUNT(*)
---------- ----------
         1          9
         2          8

SQL> /

   INST_ID   COUNT(*)
---------- ----------
         1          7
         2         10

SQL> /

   INST_ID   COUNT(*)
---------- ----------
         1          4
         2         14

SQL> /

   INST_ID   COUNT(*)
---------- ----------
         1          4
         2         13
```

```
SQL> /

   INST_ID   COUNT(*)
---------- ----------
         1          4
         2         15

SQL>
```

14) In TW1, look at the generated LBA FAN events that were sent during the second period while using the SLBA service. What do you observe?

   a) You should see that percentages are different for both instances after you started using the SLBA workload.

```
[oracle@vx0306 less08]$ cat sol_08_02_14.sh
#!/bin/ksh
#
# sol_08_02_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus /NOLOG @sol_08_02_14.sql



[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ cat sol_08_02_14.sql
connect / as sysdba

SELECT TO_CHAR(enq_time,'HH:MI:SS') Enq_time, user_data FROM
sys.sys$service_metrics_tab ORDER BY 1 ;

exit;
```

```
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ ./sol_08_02_14.sh

SQL*Plus: Release 11.1.0.6.0 - Production on Thu Nov 15
05:54:24 2007

Copyright (c) 1982, 2007, Oracle.  All rights reserved.

Connected.

ENQ_TIME
--------
USER_DATA(SRV, PAYLOAD)
--------------------------------------------------------------
------------------
02:02:28
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce
nt=50 flag=UNKNOWN aff=FALSE}{instance=RDB1 percent=50
flag=UNKNOWN aff=FALSE} }
 timestamp=2007-11-15 04:02:28')

02:02:58
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce
nt=50 flag=UNKNOWN aff=FALSE}{instance=RDB1 percent=50
flag=UNKNOWN aff=FALSE} }
 timestamp=2007-11-15 04:02:58')


...


ENQ_TIME
--------
USER_DATA(SRV, PAYLOAD)
--------------------------------------------------------------
------------------
nt=50 flag=UNKNOWN aff=FALSE}{instance=RDB1 percent=50
flag=UNKNOWN aff=FALSE} }
 timestamp=2007-11-15 05:21:33')

03:22:04
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce
nt=50 flag=UNKNOWN aff=FALSE}{instance=RDB1 percent=50
flag=UNKNOWN aff=FALSE} }
 timestamp=2007-11-15 05:22:04')

03:22:34
```

Practice 8-2: Use Load Balancing Advisory (continued)

```
ENQ_TIME
--------
USER_DATA(SRV, PAYLOAD)
---------------------------------------------------------------
------------------
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce
nt=50 flag=UNKNOWN aff=FALSE}{instance=RDB1 percent=50
flag=UNKNOWN aff=FALSE} }
 timestamp=2007-11-15 05:22:34')

03:23:04
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce
nt=74 flag=GOOD aff=FALSE}{instance=RDB1 percent=26 flag=GOOD
aff=TRUE} } timest
amp=2007-11-15 05:23:04')


ENQ_TIME
--------
USER_DATA(SRV, PAYLOAD)
---------------------------------------------------------------
------------------
03:23:34
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce
nt=78 flag=GOOD aff=TRUE}{instance=RDB1 percent=22 flag=GOOD
aff=FALSE} } timest
amp=2007-11-15 05:23:34')

03:24:04
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce
nt=70 flag=GOOD aff=FALSE}{instance=RDB1 percent=30 flag=GOOD
aff=TRUE} } timest
amp=2007-11-15 05:24:04')

ENQ_TIME
--------
USER_DATA(SRV, PAYLOAD)
---------------------------------------------------------------
------------------

03:24:35
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce
nt=58 flag=GOOD aff=TRUE}{instance=RDB1 percent=42 flag=GOOD
aff=FALSE} } timest
amp=2007-11-15 05:24:35')

03:25:05
```

Practice 8-2: Use Load Balancing Advisory (continued)

```
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce
nt=47 flag=GOOD aff=FALSE}{instance=RDB1 percent=53 flag=GOOD
aff=TRUE} } timest

ENQ_TIME
--------
USER_DATA(SRV, PAYLOAD)
---------------------------------------------------------------
------------------
amp=2007-11-15 05:25:05')

03:25:35
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce
nt=41 flag=GOOD aff=TRUE}{instance=RDB1 percent=59 flag=GOOD
aff=FALSE} } timest
amp=2007-11-15 05:25:35')

03:26:07
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce

ENQ_TIME
--------
USER_DATA(SRV, PAYLOAD)
---------------------------------------------------------------
------------------
nt=38 flag=GOOD aff=TRUE}{instance=RDB1 percent=62 flag=GOOD
aff=TRUE} } timesta
mp=2007-11-15 05:26:07')

03:26:38
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce
nt=35 flag=GOOD aff=TRUE}{instance=RDB1 percent=65 flag=GOOD
aff=FALSE} } timest
amp=2007-11-15 05:26:38')

03:27:08

ENQ_TIME
--------
USER_DATA(SRV, PAYLOAD)
---------------------------------------------------------------
------------------
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce
nt=43 flag=UNKNOWN aff=FALSE}{instance=RDB1 percent=57
flag=UNKNOWN aff=FALSE} }
 timestamp=2007-11-15 05:27:08')
```

Practice 8-2: Use Load Balancing Advisory (continued)

```
03:27:38
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce
nt=46 flag=UNKNOWN aff=FALSE}{instance=RDB1 percent=54
flag=UNKNOWN aff=FALSE} }
 timestamp=2007-11-15 05:27:38')


ENQ_TIME
--------
USER_DATA(SRV, PAYLOAD)
--------------------------------------------------------------
------------------
03:28:08
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce
nt=48 flag=UNKNOWN aff=FALSE}{instance=RDB1 percent=52
flag=UNKNOWN aff=FALSE} }
 timestamp=2007-11-15 05:28:08')

03:28:38
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce
nt=49 flag=UNKNOWN aff=FALSE}{instance=RDB1 percent=51
flag=UNKNOWN aff=FALSE} }
 timestamp=2007-11-15 05:28:38')


...


ENQ_TIME
--------
USER_DATA(SRV, PAYLOAD)
--------------------------------------------------------------
------------------
03:41:39
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce
nt=50 flag=UNKNOWN aff=FALSE}{instance=RDB1 percent=50
flag=UNKNOWN aff=FALSE} }
 timestamp=2007-11-15 05:41:39')

03:42:09
SYS$RLBTYP('SLBA', 'VERSION=1.0 database=RDB service=SLBA {
{instance=RDB2 perce
nt=50 flag=UNKNOWN aff=FALSE}{instance=RDB1 percent=50
flag=UNKNOWN aff=FALSE} }
 timestamp=2007-11-15 05:42:09')

ENQ_TIME
--------
```

```
USER_DATA(SRV, PAYLOAD)
--------------------------------------------------------------
------------------


200 rows selected.

Disconnected from Oracle Database 11g Enterprise Edition
Release 11.1.0.6.0 - Production
With the Partitioning, Real Application Clusters, OLAP, Data
Mining
and Real Application Testing options
[oracle@vx0306 less08]$
```

15) Stop both services: SNOLBA and SLBA. When you are finished, remove them from your cluster configuration and your database.

```
[oracle@vx0306 less08]$ cat sol_08_02_15.sh
#!/bin/ksh
#
# sol_08_02_15.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/srvctl stop service -d $DBNAME -s SNOLBA

/u01/crs11g/bin/srvctl stop service -d $DBNAME -s SLBA

/u01/crs11g/bin/srvctl remove service -d $DBNAME -s SNOLBA

/u01/crs11g/bin/srvctl remove service -d $DBNAME -s SLBA

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
```

Practice 8-2: Use Load Balancing Advisory (continued)

```
 exec DBMS_SERVICE.DELETE_SERVICE ('SNOLBA');
 exec DBMS_SERVICE.DELETE_SERVICE ('SLBA');

EOF
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ ./sol_08_02_15.sh
SNOLBA PREF: RDB2 RDB1 AVAIL:
Remove service SNOLBA from the database RDB? (y/[n]) y
SLBA PREF: RDB2 RDB1 AVAIL:
Remove service SLBA from the database RDB? (y/[n]) y

PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.

[oracle@vx0306 less08]$
```

Practice 8-3: Use Transparent Application Failover

The goal of this practice is to create a service called TAFB that uses the new FAN propagation system through Advance Queuing to support Transparent Application Failover.

1) Using Database Control and PL/SQL, create a new service called TAFB that is configured to do BASIC SESSION TAF.

   a) Connect to your Database Control console as user SYS.

   b) On the Cluster Database Home page, click the Availability tab.

   c) On the Availability page, click the Cluster Managed Database Services link in the Services section.

   d) On the Cluster and Database Login page, enter all necessary credentials if not done already, and click Continue.

   e) On the Cluster Managed Database Services page, click the Create Service button.

   f) On the Create Service page, enter "TAFB" in the Service Name field.

   g) Select Long as the Connection Load Balancing Goal option.

   h) Click OK.

   i) You should now see a Create Service: Successful status on the Cluster Managed Database Services. You should also see that TAFB is now started on both instances.

   j) When TAFB is created, use the DBMS_SERVICE.MODIFY_SERVICE procedure to set the remaining service attributes:

```
[oracle@vx0306 less08]$ cat sol_08_03_01_j.sh
#!/bin/ksh
#
# sol_08_03_01_12.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/srvctl add service -d $DBNAME -s TAFB -r
$I1NAME,$I2NAME
```

Practice 8-3: Use Transparent Application Failover (continued)

```
/u01/crs11g/bin/srvctl start service -d $DBNAME -s TAFB

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
 execute dbms_service.modify_service (
-
   'TAFB'                                                    ,
-
   aq_ha_notifications => true                               ,
-
   failover_method     => dbms_service.failover_method_basic ,
-
   failover_type       => dbms_service.failover_type_session ,
-
   failover_retries    => 180                                ,
-
   failover_delay      => 5                                  ,
-
   clb_goal            => dbms_service.clb_goal_long);

EOF
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ ./sol_08_03_01_j.sh
PRKP-1083 : The service TAFB already exists.
PRKP-1062 : Service TAFB is already running.

PL/SQL procedure successfully completed.

[oracle@vx0306 less08]$
```

2) To connect to your database by using the TAFB service, what must you do next?

   a) You should add the corresponding TNS entry in your $ORACLE_HOME/network/admin/tnsnames.ora file. We highly recommend that you add that entry in both your ASM and Database homes:

```
[oracle@vx0306 less08]$ cat sol_08_03_02.sh
#!/bin/ksh
#
# sol_08_03_02.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#
```

```
y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

sed 's/NODE1/'$y'/'
/home/oracle/solutions/less08/wrong_tnstafb.ora >
/home/oracle/solutions/less08/w1tnstafb.ora

sed 's/NODE2/'$z'/'
/home/oracle/solutions/less08/w1tnstafb.ora >
/home/oracle/solutions/less08/tnstafb.ora

cp
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora
.bak3
cat /home/oracle/solutions/less08/tnstafb.ora >>
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora

cp
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a.bak4
cat /home/oracle/solutions/less08/tnstafb.ora >>
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a

ssh $z cp
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora
.bak3

ssh $z cp
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a.bak4

scp /home/oracle/solutions/less08/tnstafb.ora
$z:/home/oracle/solutions/less08

ssh $z "cat /home/oracle/solutions/less08/tnstafb.ora >>
/u01/app/oracle/product/11.1.0/db_1/network/admin/tnsnames.ora
"

ssh $z "cat /home/oracle/solutions/less08/tnstafb.ora >>
/u01/app/oracle/product/11.1.0/asm_1/network/admin/tnsnames.or
a"
```

Practice 8-3: Use Transparent Application Failover (continued)

```
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ cat wrong_tnstafb.ora

TAFB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = NODE1-vip)(PORT =
1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = NODE2-vip)(PORT =
1521))
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = TAFB)
    )
  )
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ ./sol_08_03_02.sh
tnstafb.ora                                         100%  265
0.3KB/s   00:00
[oracle@vx0306 less08]$
```

3) Using SQL*Plus from your first node, verify that the TAFB service is started on both instances.

```
[oracle@vx0306 less08]$ cat sol_08_03_03.sh
#!/bin/ksh
#
# sol_08_03_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect system/oracle1@$I1NAME
```

```
 show parameter service
 connect system/oracle1@$I2NAME
 show parameter service

EOF
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ ./sol_08_03_03.sh

NAME                                 TYPE        VALUE
------------------------------------ ----------- -------------
-----------------
service_names                        string      RDB, TAFB

NAME                                 TYPE        VALUE
------------------------------------ ----------- -------------
-----------------
service_names                        string      RDB, TAFB
[oracle@vx0306 less08]$
```

4) Connect to the database by using the TAFB service. Use user JFV to connect. Check which instance you are currently connected to. Also check which TAF policy you are using. We refer to the telnet session that you use for this step as TW1.

```
[oracle@vx0306 less08]$ cat vi sol_08_03_04.sh
cat: vi: No such file or directory
#!/bin/ksh
#
# sol_08_02_04.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus /NOLOG @sol_08_03_04_00.sql

[oracle@vx0306 less08]$
```

```
[oracle@vx0306 less08]$ cat sol_08_03_04_00.sql
connect / as sysdba
host echo "execute sol_08_03_04.sql from here"
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ cat sol_08_03_04_00.sql
connect / as sysdba
host echo "execute sol_08_03_04.sql from here"
[oracle@vx0306 less08]$ cat sol_08_03_04.sql
connect jfv/jfv@TAFB
select instance_name from v$instance;
col service_name format a10
select failover_type,failover_method,failed_over from
v$session where username='JFV';
[oracle@vx0306 less08]$


[oracle@vx0306 less08]$ ./sol_08_03_04.sh

SQL*Plus: Release 11.1.0.6.0 - Production on Thu Nov 15
08:39:30 2007

Copyright (c) 1982, 2007, Oracle.  All rights reserved.

Connected.
execute sol_08_03_04.sql from here

SQL> @sol_08_03_04
Connected.

INSTANCE_NAME
----------------
RDB1


FAILOVER_TYPE FAILOVER_M FAI
------------- ---------- ---
SESSION       BASIC      NO

SQL>
```

5) Create a new terminal window, referred to as TW2, and determine how many sessions are started on each instance to support your connection on the first terminal session?

   a) Only one session is created to support your connection. Note that this session can be started on either RDB1 or RDB2. In this example, it is started on RDB2, and no corresponding session on RDB1 exists

```
[oracle@vx0306 less08]$ cat sol_08_03_05.sh
#!/bin/ksh
#
```

Practice 8-3: Use Transparent Application Failover (continued)

```
# sol_08_03_05.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect system/oracle1@$I1NAME
 select instance_name from v\$instance;
 col service_name format a10
 select failover_type,failover_method,failed_over from
v\$session where username='JFV';
 connect system/oracle1@$I2NAME
 select instance_name from v\$instance;
 col service_name format a10
 select failover_type,failover_method,failed_over from
v\$session where username='JFV';

EOF
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ ./sol_08_03_05.sh

INSTANCE_NAME
----------------
RDB1


FAILOVER_TYPE FAILOVER_M FAI
------------- ---------- ---
SESSION       BASIC      NO


INSTANCE_NAME
----------------
RDB2
```

```
no rows selected

[oracle@vx0306 less08]$
```

6) Still connected as user JFV from your first session, insert a row into the FAN table and commit your modification.

```
[oracle@vx0306 less08]$ cat sol_08_03_06.sql
insert into fan values(1);
commit;
[oracle@vx0306 less08]$


SQL> @sol_08_03_06

1 row created.


Commit complete.

SQL>
```

7) Kill the OS process corresponding to the `oracle` shadow session that is connected using TAFB service. You can use the killtafb.sh script for that.

   a) As an indication, you can use the following SQL statement to help you kill the correct process:

```
select 'kill -9 ' || spid
from v$process
where addr in (select paddr
               from v$session where username='JFV');
```

   You should run this statement on both nodes.

```
[oracle@vx0306 less08]$ cat killtafb.sh
#!/bin/ksh
#
# killtafb.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"
```

```
export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus system/oracle1@$I1NAME
@/home/oracle/solutions/less08/kill_tafb.sql

chmod 777 /tmp/drop_tafb.sh

/tmp/drop_tafb.sh

rm /tmp/drop_tafb.sh

$ORACLE_HOME/bin/sqlplus system/oracle1@$I2NAME
@/home/oracle/solutions/less08/kill_tafb.sql

scp /tmp/drop_tafb.sh $z:/tmp/drop_tafb.sh

ssh $z chmod 777 /tmp/drop_tafb.sh

ssh $z /tmp/drop_tafb.sh

ssh $z rm /tmp/drop_tafb.sh

rm /tmp/drop_tafb.sh
[oracle@vx0306 less08]$


[oracle@vx0306 less08]$ cat kill_tafb.sql

set head off
set timing off
set feedback off;
set pagesize 0
set verify off

spool /tmp/drop_tafb.sh;

select 'kill -9 ' || spid
from v$process
where addr in (select paddr from v$session where
username='JFV');

spool off

exit;
[oracle@vx0306 less08]$


[oracle@vx0306 less08]$ ./killtafb.sh
```

```
SQL*Plus: Release 11.1.0.6.0 - Production on Thu Nov 15
08:43:33 2007

Copyright (c) 1982, 2007, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 -
Production
With the Partitioning, Real Application Clusters, OLAP, Data
Mining
and Real Application Testing options

kill -9 1517
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.1.0.6.0 - Production
With the Partitioning, Real Application Clusters, OLAP, Data
Mining
and Real Application Testing options

SQL*Plus: Release 11.1.0.6.0 - Production on Thu Nov 15
08:43:33 2007

Copyright (c) 1982, 2007, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 -
Production
With the Partitioning, Real Application Clusters, OLAP, Data
Mining
and Real Application Testing options

Disconnected from Oracle Database 11g Enterprise Edition
Release 11.1.0.6.0 - Production
With the Partitioning, Real Application Clusters, OLAP, Data
Mining
and Real Application Testing options
drop_tafb.sh                                  100%    0
0.0KB/s   00:00
[oracle@vx0306 less08]$
```

8) Insert again one row in FAN from your first session. What do you observe?

   a) Because your server process has been killed, you are no longer able to correctly execute a SQL statement from your session.

```
[oracle@vx0306 less08]$ cat sol_08_03_08.sql
insert into fan values(1);
[oracle@vx0306 less08]$

SQL> @sol_08_03_08
insert into fan values(1)
```

Practice 8-3: Use Transparent Application Failover (continued)

```
*
ERROR at line 1:
ORA-03113: end-of-file on communication channel
Process ID: 5805
Session ID: 114 Serial number: 39333


SQL>
```

9) Try again. What happens?

   a) This time it is possible to again execute a SQL command because you have been automatically failed over to an instance that offers the TAFB service. Note that this instance can be either RDB1 or RDB2.

```
[oracle@vx0306 less08]$ cat sol_08_03_09.sql
insert into fan values(1);
commit;
select instance_name from v$instance;
[oracle@vx0306 less08]$

SQL> @sol_08_03_09

1 row created.


Commit complete.


INSTANCE_NAME
----------------
RDB1

SQL>
```

10) Verify that you automatically failed over.

```
[oracle@vx0306 less08]$ cat sol_08_03_10.sql
select instance_name from v$instance;

select failover_type,failover_method,failed_over from
v$session where username='JFV';

exit;
[oracle@vx0306 less08]$


SQL> @sol_08_03_10

INSTANCE_NAME
----------------
RDB1
```

Practice 8-3: Use Transparent Application Failover (continued)

```
FAILOVER_TYPE FAILOVER_M FAI
------------- ---------- ---
SESSION       BASIC      YES

Disconnected from Oracle Database 11g Enterprise Edition
Release 11.1.0.6.0 - Production
With the Partitioning, Real Application Clusters, OLAP, Data
Mining
and Real Application Testing options
[oracle@vx0306 less08]$
```

11) Remove TAFB from both the cluster configuration and the database.

```
[oracle@vx0306 less08]$ cat sol_08_03_11.sh
#!/bin/ksh
#
# sol_08_03_11.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/srvctl stop service -d $DBNAME -s TAFB

/u01/crs11g/bin/srvctl remove service -d $DBNAME -s TAFB

export ORACLE_SID=$I1NAME
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

$ORACLE_HOME/bin/sqlplus -s /NOLOG <<EOF

 connect / as sysdba
 exec dbms_service.delete_service('TAFB');

EOF
[oracle@vx0306 less08]$

[oracle@vx0306 less08]$ ./sol_08_03_11.sh
TAFB PREF: RDB2 RDB1 AVAIL:
```

Practice 8-3: Use Transparent Application Failover (continued)

```
Remove service TAFB from the database RDB? (y/[n]) y

PL/SQL procedure successfully completed.

[oracle@vx0306 less08]$
```

Practices for Lesson 9

Practice 9-1: Mirror the OCR

The goal of this practice is to mirror your OCR file, and replace a corrupted OCR file.

1) Check your OCR configuration.

```
[oracle@vx0306 less09]$ cat sol_09_01_01.sh
#!/bin/bash
#
# sol_09_01_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/ocrcheck

more /etc/oracle/ocr.loc
[oracle@vx0306 less09]$

[oracle@vx0306 less09]$ ./sol_09_01_01.sh
Status of Oracle Cluster Registry is as follows :
         Version                  :          2
         Total space (kbytes)     :     616652
         Used space (kbytes)      :       3832
         Available space (kbytes) :     612820
         ID                       :  992212279
         Device/File Name         :  /dev/sdb1
                                    Device/File integrity
check succeeded

                                    Device/File not configured

         Cluster registry integrity check succeeded

ocrconfig_loc=/dev/sdb1
local_only=FALSE
[oracle@vx0306 less09]$
```

2) Mirror your OCR using /dev/sdb2 as the new mirror.

```
[oracle@vx0306 less09]$ cat sol_09_01_02.sh
#!/bin/bash
#
```

```
# sol_09_01_02.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

sudo /u01/crs11g/bin/ocrconfig -replace ocrmirror /dev/sdb2
[oracle@vx0306 less09]$

[oracle@vx0306 less09]$ ./sol_09_01_02.sh
[oracle@vx0306 less09]$
```

3) Check again your OCR configuration.

```
[oracle@vx0306 less09]$ cat sol_09_01_03.sh
#!/bin/bash
#
# sol_09_01_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/ocrcheck

more /etc/oracle/ocr.loc
[oracle@vx0306 less09]$

[oracle@vx0306 less09]$ ./sol_09_01_03.sh
Status of Oracle Cluster Registry is as follows :
         Version                  :          2
         Total space (kbytes)     :     616652
         Used space (kbytes)      :       3832
```

```
         Available space (kbytes) :     612820
         ID                       :  992212279
         Device/File Name         :  /dev/sdb1
                                    Device/File integrity
check succeeded
         Device/File Name         :  /dev/sdb2
                                    Device/File integrity
check succeeded

         Cluster registry integrity check succeeded

#Device/file  getting replaced by device /dev/sdb2
ocrconfig_loc=/dev/sdb1
ocrmirrorconfig_loc=/dev/sdb2
local_only=false
[oracle@vx0306 less09]$
```

4) Corrupt your OCR mirror using the dd command.

```
[oracle@vx0306 less09]$ cat sol_09_01_04.sh
#!/bin/bash
#
# sol_09_01_04.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

sudo dd if=/dev/zero of=/dev/sdb2 bs=1024k count=200
[oracle@vx0306 less09]$


[oracle@vx0306 less09]$ ./sol_09_01_04.sh
200+0 records in
200+0 records out
[oracle@vx0306 less09]$
```

5) Check the integrity of your OCR configuration again.

```
[oracle@vx0306 less09]$ cat sol_09_01_05.sh
#!/bin/bash
#
# sol_09_01_05.sh
```

```
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/ocrcheck
[oracle@vx0306 less09]$

-- What we get now!
[oracle@vx0308 less09]$ /u01/crs11g/bin/ocrcheck
PROT-602: Failed to retrieve data from the cluster registry
[oracle@vx0308 less09]$


-- What we should be getting!
[oracle@vx0306 less09]$ ./sol_09_01_05.sh
Status of Oracle Cluster Registry is as follows :
         Version                  :          2
         Total space (kbytes)     :     616652
         Used space (kbytes)      :       3832
         Available space (kbytes) :     612820
         ID                       :  992212279
         Device/File Name         :  /dev/sdb1
                                    Device/File integrity
check succeeded
         Device/File Name         :  /dev/sdb2
                                    Device/File needs to be
synchronized with the other device

         Cluster registry integrity check succeeded

[oracle@vx0306 less09]$
```

6) Replace your OCR mirror using /dev/sdb1

```
[oracle@vx0306 less09]$ cat sol_09_01_06.sh
#!/bin/bash
#
# sol_09_01_06.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#
```

Practice 9-1: Mirror the OCR (continued)

```
y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

sudo /u01/crs11g/bin/ocrconfig -replace ocrmirror /dev/sdb2
[oracle@vx0306 less09]$

[oracle@vx0306 less09]$ ./sol_09_01_06.sh
[oracle@vx0306 less09]$
```

7) Check the integrity of your OCR configuration again.

```
[oracle@vx0306 less09]$ cat sol_09_01_07.sh
#!/bin/bash
#
# sol_09_01_07.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/ocrcheck
[oracle@vx0306 less09]$

[oracle@vx0306 less09]$ ./sol_09_01_07.sh
Status of Oracle Cluster Registry is as follows :
         Version                  :          2
         Total space (kbytes)     :     616652
         Used space (kbytes)      :       3832
         Available space (kbytes) :     612820
         ID                       :  992212279
         Device/File Name         :  /dev/sdb1
                                    Device/File integrity
check succeeded
         Device/File Name         :  /dev/sdb2
```

Practice 9-1: Mirror the OCR (continued)

```
                                    Device/File integrity
check succeeded

         Cluster registry integrity check succeeded

[oracle@vx0306 less09]$
```

Practice 9-2: OCR Backup and Restore

The goal of this practice is to back up and restore an OCR file.

1) For security purposes, generate a logical OCR backup file.

```
[oracle@vx0306 less09]$ cat sol_09_02_01.sh
#!/bin/bash
#
# sol_09_02_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

sudo /u01/crs11g/bin/ocrconfig -export
/home/oracle/solutions/less09/logicalocrbak

ls -l /home/oracle/solutions/less09/logicalocrbak
[oracle@vx0306 less09]$

[oracle@vx0306 less09]$ ./sol_09_02_01.sh
-rw-r--r--  1 root root 86550 Nov 16  2007
/home/oracle/solutions/less09/logicalocrbak
[oracle@vx0306 less09]$
```

2) Locate a physical backup of your OCR.

```
[oracle@vx0306 less09]$ cat sol_09_02_02.sh
#!/bin/bash
#
# sol_09_02_02.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"
```

Practice 9-2: OCR Backup and Restore (continued)

```
I2NAME=$DBNAME"2"

HNAME=`/u01/crs11g/bin/ocrconfig -showbackup | grep -m 1 vx |
awk '{ print $1 }'`

BNAME=`/u01/crs11g/bin/ocrconfig -showbackup | grep -m 1 vx |
awk '{ print $4 }'`

echo "backups on $y"
/u01/crs11g/bin/ocrconfig -showbackup

echo "backups on $HNAME"
ssh $HNAME ls -l $BNAME
[oracle@vx0306 less09]$

[oracle@vx0306 less09]$ ./sol_09_02_02.sh
backups on vx0306

vx0313     2007/11/16 02:15:43
/u01/crs11g/cdata/vx_cluster02/backup00.ocr

vx0313     2007/11/15 22:15:43
/u01/crs11g/cdata/vx_cluster02/backup01.ocr

vx0313     2007/11/15 18:15:43
/u01/crs11g/cdata/vx_cluster02/backup02.ocr

vx0306     2007/11/15 01:58:23
/u01/crs11g/cdata/vx_cluster02/day.ocr

vx0306     2007/11/13 05:58:16
/u01/crs11g/cdata/vx_cluster02/week.ocr
backups on vx0313
-rw-r--r--  1 root root 4603904 Nov 16 02:15
/u01/crs11g/cdata/vx_cluster02/backup00.ocr
[oracle@vx0306 less09]$
```

3) Stop CRS resources on both nodes. When you are finished, verify that they are all stopped.

```
[oracle@vx0306 less09]$ cat sol_09_02_03.sh
#!/bin/bash
#
# sol_09_02_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`
```

```
DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/srvctl stop nodeapps -n $z

/u01/crs11g/bin/srvctl stop nodeapps -n $y

/u01/crs11g/bin/srvctl stop database -d $DBNAME

/u01/crs11g/bin/srvctl stop asm -n $y
/u01/crs11g/bin/srvctl stop asm -n $z


/u01/crs11g/bin/crs_stat -t
[oracle@vx0306 less09]$


[oracle@vx0306 less09]$ ./sol_09_02_03.sh
Name           Type           Target    State     Host
------------------------------------------------------------
ora....B1.inst application    OFFLINE   OFFLINE
ora....B2.inst application    OFFLINE   OFFLINE
ora.RDB.db     application    OFFLINE   OFFLINE
ora....SM1.asm application    OFFLINE   OFFLINE
ora....06.lsnr application    OFFLINE   OFFLINE
ora.vx0306.gsd application    OFFLINE   OFFLINE
ora.vx0306.ons application    OFFLINE   OFFLINE
ora.vx0306.vip application    OFFLINE   OFFLINE
ora....SM2.asm application    OFFLINE   OFFLINE
ora....13.lsnr application    OFFLINE   OFFLINE
ora.vx0313.gsd application    OFFLINE   OFFLINE
ora.vx0313.ons application    OFFLINE   OFFLINE
ora.vx0313.vip application    OFFLINE   OFFLINE
[oracle@vx0306 less09]$
```

4) Stop CRS on both nodes.

```
[oracle@vx0306 less09]$ cat sol_09_02_04.sh
#!/bin/bash
#
# sol_09_02_04.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`
```

```
DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

echo "Stopping Oracle Clusterware on $z"
ssh $z sudo /u01/crs11g/bin/crsctl stop crs

sleep 60

echo "Stopping Oracle Clusterware on $y"
sudo /u01/crs11g/bin/crsctl stop crs

sleep 60
[oracle@vx0306 less09]$


[oracle@vx0306 less09]$ ./sol_09_02_04.sh
Stopping Oracle Clusterware on vx0313
Stopping resources.
This could take several minutes.
Successfully stopped Oracle Clusterware resources
Stopping Cluster Synchronization Services.
Shutting down the Cluster Synchronization Services daemon.
Shutdown request successfully issued.
Stopping Oracle Clusterware on vx0306
Stopping resources.
This could take several minutes.
Successfully stopped Oracle Clusterware resources
Stopping Cluster Synchronization Services.
Shutting down the Cluster Synchronization Services daemon.
Shutdown request successfully issued.
[oracle@vx0306 less09]$
```

5) Restore your OCR by using the backup identified in step 2.

```
[oracle@vx0306 less09]$ cat sol_09_02_05.sh
#!/bin/bash
#
# sol_09_02_05.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`
```

```
I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

HNAME=`/u01/crs11g/bin/ocrconfig -showbackup | grep -m 1 vx |
awk '{ print $1 }'`

BNAME=`/u01/crs11g/bin/ocrconfig -showbackup | grep -m 1 vx |
awk '{ print $4 }'`

ssh $HNAME sudo /u01/crs11g/bin/ocrconfig -restore $BNAME
[oracle@vx0306 less09]$

[oracle@vx0306 less09]$ ./sol_09_02_05.sh
[oracle@vx0306 less09]$
```

6) Restart CRS on both nodes.

```
[oracle@vx0306 less09]$ cat sol_09_02_06.sh
#!/bin/bash
#
# sol_09_02_06.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

sudo /u01/crs11g/bin/crsctl start crs

sleep 300

/u01/crs11g/bin/crs_stat -t

ssh $z sudo /u01/crs11g/bin/crsctl start crs

sleep 300

/u01/crs11g/bin/crs_stat -t
[oracle@vx0306 less09]$

[oracle@vx0306 less09]$ ./sol_09_02_06.sh
Attempting to start Oracle Clusterware stack
```

Practice 9-2: OCR Backup and Restore (continued)

```
The CRS stack will be started shortly
Name           Type           Target    State     Host
------------------------------------------------------------
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora....B2.inst application    ONLINE    OFFLINE
ora.RDB.db     application    ONLINE    ONLINE    vx0306
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
ora....SM2.asm application    ONLINE    OFFLINE
ora....13.lsnr application    ONLINE    OFFLINE
ora.vx0313.gsd application    ONLINE    OFFLINE
ora.vx0313.ons application    ONLINE    OFFLINE
ora.vx0313.vip application    ONLINE    ONLINE    vx0306
Attempting to start Oracle Clusterware stack
The CRS stack will be started shortly
Name           Type           Target    State     Host
------------------------------------------------------------
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora....B2.inst application    ONLINE    ONLINE    vx0313
ora.RDB.db     application    ONLINE    ONLINE    vx0306
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
ora....SM2.asm application    ONLINE    ONLINE    vx0313
ora....13.lsnr application    ONLINE    ONLINE    vx0313
ora.vx0313.gsd application    ONLINE    ONLINE    vx0313
ora.vx0313.ons application    ONLINE    ONLINE    vx0313
ora.vx0313.vip application    ONLINE    ONLINE    vx0313
[oracle@vx0306 less09]$
```

7) Check your OCR integrity from both nodes.

```
[oracle@vx0306 less09]$ cat sol_09_02_07.sh
#!/bin/bash
#
# sol_09_02_07.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`
```

```
I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/cluvfy comp ocr -n all
[oracle@vx0306 less09]$

[oracle@vx0306 less09]$ ./sol_09_02_07.sh

Verifying OCR integrity

Checking OCR integrity...

Checking the absence of a non-clustered configuration...
All nodes free of non-clustered, local-only configurations.

Uniqueness check for OCR device passed.

Checking the version of OCR...
OCR of correct Version "2" exists.

Checking data integrity of OCR...
Data integrity check for OCR passed.

OCR integrity check passed.

Verification of OCR integrity was successful.
[oracle@vx0306 less09]$
```

8) Do a manual backup of your OCR.

```
[oracle@vx0306 less09]$ cat sol_09_02_08.sh
#!/bin/bash
#
# sol_09_02_08.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

sudo /u01/crs11g/bin/ocrconfig -manualbackup
```

Practice 9-2: OCR Backup and Restore (continued)

```
/u01/crs11g/bin/ocrconfig -showbackup

echo "and on $z"
ssh $z /u01/crs11g/bin/ocrconfig -showbackup

[oracle@vx0306 less09]$


[oracle@vx0306 less09]$ ./sol_09_02_08.sh

vx0306     2007/11/16 03:58:37
/u01/crs11g/cdata/vx_cluster02/backup_20071116_035837.ocr

vx0313     2007/11/15 22:15:43
/u01/crs11g/cdata/vx_cluster02/backup00.ocr

vx0313     2007/11/15 18:15:43
/u01/crs11g/cdata/vx_cluster02/backup01.ocr

vx0313     2007/11/15 14:15:43
/u01/crs11g/cdata/vx_cluster02/backup02.ocr

vx0306     2007/11/14 01:58:19
/u01/crs11g/cdata/vx_cluster02/day.ocr

vx0306     2007/11/13 05:58:16
/u01/crs11g/cdata/vx_cluster02/week.ocr

vx0306     2007/11/16 03:58:37
/u01/crs11g/cdata/vx_cluster02/backup_20071116_035837.ocr
and on vx0313

vx0313     2007/11/15 22:15:43
/u01/crs11g/cdata/vx_cluster02/backup00.ocr

vx0313     2007/11/15 18:15:43
/u01/crs11g/cdata/vx_cluster02/backup01.ocr

vx0313     2007/11/15 14:15:43
/u01/crs11g/cdata/vx_cluster02/backup02.ocr

vx0306     2007/11/14 01:58:19
/u01/crs11g/cdata/vx_cluster02/day.ocr

vx0306     2007/11/13 05:58:16
/u01/crs11g/cdata/vx_cluster02/week.ocr

vx0306     2007/11/16 03:58:37
/u01/crs11g/cdata/vx_cluster02/backup_20071116_035837.ocr
[oracle@vx0306 less09]$
```

Practice 9-3: Multiplex Your Voting Disk

The goal of this practice is to add two mirrors to your voting disk configuration.

1) Determine your current voting disk configuration.

```
[oracle@vx0306 less09]$ cat sol_09_03_01.sh
#!/bin/bash
#
# sol_09_03_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/crsctl query css votedisk
[oracle@vx0306 less09]$

[oracle@vx0306 less09]$ ./sol_09_03_01.sh
 0.     0    /dev/sdb5
Located 1 voting disk(s).
[oracle@vx0306 less09]$
```

2) Add two new members to your voting disk configuration using both /dev/sdb6, and /dev/sdb7.

```
[oracle@vx0306 less09]$ cat sol_09_03_02.sh
#!/bin/bash
#
# sol_09_03_04.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"
```

Practice 9-3: Multiplex Your Voting Disk (continued)

```
sudo /u01/crs11g/bin/crsctl add css votedisk /dev/sdb6

sudo /u01/crs11g/bin/crsctl add css votedisk /dev/sdb7
[oracle@vx0306 less09]$


[oracle@vx0306 less09]$ ./sol_09_03_02.sh
Now formatting voting disk: /dev/sdb6.
Successful addition of voting disk /dev/sdb6.
Now formatting voting disk: /dev/sdb7.
Successful addition of voting disk /dev/sdb7.
[oracle@vx0306 less09]$
```

3) Determine your current voting disk configuration.

```
[oracle@vx0306 less09]$ cat sol_09_03_03.sh
#!/bin/bash
#
# sol_09_03_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/crsctl query css votedisk
[oracle@vx0306 less09]$


[oracle@vx0306 less09]$ ./sol_09_03_03.sh
 0.     0    /dev/sdb5
 1.     0    /dev/sdb6
 2.     0    /dev/sdb7
Located 3 voting disk(s).
[oracle@vx0306 less09]$
```

Practice 9-4: Protect Xclock with Oracle Clusterware

Important note: Before you start the following practices, make sure that VNC is started on both nodes as user `oracle` on port 5802.
You execute all of your commands for this practice under VNC terminal sessions from the first and second nodes.
The goal of this practice is to protect the Xclock application with Oracle Clusterware so that it is automatically restarted each time you kill it.

1) Make sure that both nodes are added to the xhost access control list. Do this on both nodes.

```
[oracle@vx0306 less09]$ cat sol_09_04_01.sh
#!/bin/ksh
#
# sol_09_04_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

cd /home/oracle/solutions/less09

echo export DISPLAY=:2.0 > displayenv.sh
echo /usr/X11R6/bin/xhost + $y >> displayenv.sh
echo /usr/X11R6/bin/xhost + $z >> displayenv.sh

chmod 777 displayenv.sh

./displayenv.sh

scp displayenv.sh $z:/home/oracle/solutions/less09
ssh $z "chmod 777 /home/oracle/solutions/less09/displayenv.sh"
ssh $z /home/oracle/solutions/less09/displayenv.sh
[oracle@vx0306 less09]$


[oracle@vx0306 less09]$ ./sol_09_04_01.sh
vx0306 being added to access control list
vx0313 being added to access control list
displayenv.sh                                     100%   80
0.1KB/s   00:00
vx0306 being added to access control list
vx0313 being added to access control list
[oracle@vx0306 less09]$
```

2) Create an action script for CRS to monitor the xclock application. You need to make sure that the clock is always displayed on the VNC session attached to your first node. When you are finished, copy that script in the default CRS action scripts location on both nodes. Call your action script crsclock_action.scr.

Practice 9-4: Protect Xclock with Oracle Clusterware (continued)

```
[oracle@vx0306 less09]$ cat sol_09_04_02.sh
#!/bin/ksh
#
# sol_09_04_02.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

cd /home/oracle/solutions/less09

sed 's/NODE1/'$y'/' wrong_crsclock_action.scr >
crsclock_action.scr

sudo cp /home/oracle/solutions/less09/crsclock_action.scr
/u01/crs11g/crs/script/

sudo chmod 777 /u01/crs11g/crs/script/crsclock_action.scr

scp crsclock_action.scr $z:/home/oracle/solutions/less09

ssh $z "sudo cp
/home/oracle/solutions/less09/crsclock_action.scr
/u01/crs11g/crs/script/"

ssh $z "sudo chmod 777
/u01/crs11g/crs/script/crsclock_action.scr"

cat /u01/crs11g/crs/script/crsclock_action.scr

[oracle@vx0306 less09]$


[oracle@vx0306 less09]$ cat wrong_crsclock_action.scr
#!/bin/bash
# start/stop/check script for xclock example
# To test this change BIN_DIR to the directory where xclock is
based
# and set the DISPLAY variable to a server within your
network.


BIN_DIR=/usr/X11R6/bin
LOG_DIR=/tmp
BIN_NAME=xclock
DISPLAY=NODE1:2.0
export DISPLAY

if [ ! -d $BIN_DIR ]
then
```

```
        echo "start failed"
        exit 2
fi

PID1=`ps -ef | grep $BIN_NAME | grep -v grep | grep -v
xclock_app | awk '{ print $2 }'`

case $1 in
'start')
        if [ "$PID1" != "" ]
        then
           status_p1="running"
        else
           if [ -x $BIN_DIR/$BIN_NAME  ]
           then
             umask 002
             ${BIN_DIR}/${BIN_NAME} &
2>${LOG_DIR}/${BIN1_NAME}.log
             status_p1="started"
           else
             echo `basename $0`": $BIN_NAME: Executable not
found"
           fi
        fi

         echo "$BIN_NAME: $status_p1"
        ;;

'stop')
        if [ "${PID1}" != "" ]
        then
           kill -9 ${PID1} && echo "$BIN_NAME daemon killed"
        else
           echo "$BIN_NAME: no running Process!"
        fi
        ;;
'check')
        if [ "$PID1" != "" ]
        then
           echo "running"
           exit 0
        else
           echo "not running"
           exit 1
        fi
        ;;
*)
        echo "Usage: "`basename $0`" {start|stop|check}"
        ;;
esac
[oracle@vx0306 less09]$
```

```
[oracle@vx0306 less09]$ ./sol_09_04_02.sh
crsclock_action.scr                              100% 1339
1.3KB/s   00:00
#!/bin/bash
# start/stop/check script for xclock example
# To test this change BIN_DIR to the directory where xclock is
based
# and set the DISPLAY variable to a server within your
network.


BIN_DIR=/usr/X11R6/bin
LOG_DIR=/tmp
BIN_NAME=xclock
DISPLAY=vx0306:2.0
export DISPLAY

if [ ! -d $BIN_DIR ]
then
        echo "start failed"
        exit 2
fi

PID1=`ps -ef | grep $BIN_NAME | grep -v grep | grep -v
xclock_app | awk '{ print $2 }'`

case $1 in
'start')
        if [ "$PID1" != "" ]
        then
           status_p1="running"
        else
           if [ -x $BIN_DIR/$BIN_NAME  ]
           then
             umask 002
             ${BIN_DIR}/${BIN_NAME} &
2>${LOG_DIR}/${BIN1_NAME}.log
             status_p1="started"
           else
             echo `basename $0`": $BIN_NAME: Executable not
found"
           fi
        fi

         echo "$BIN_NAME: $status_p1"
        ;;

'stop')
        if [ "${PID1}" != "" ]
        then
```

```
            kill -9 ${PID1} && echo "$BIN_NAME daemon killed"
         else
            echo "$BIN_NAME: no running Process!"
         fi
         ;;
'check')
         if [ "$PID1" != "" ]
         then
            echo "running"
            exit 0
         else
            echo "not running"
            exit 1
         fi
         ;;
*)
         echo "Usage: "`basename $0`" {start|stop|check}"
         ;;
esac
[oracle@vx0306 less09]$
```

3) Create a new CRS application resource profile called myClock using the action script you just created. Make sure that the resource attribute *p* is set to *favored*, *h* is set to the list corresponding to your two nodes with the second one first, *ci* is set to *5* and *ra* is set to *2*.

```
[oracle@vx0306 less09]$ cat sol_09_04_03.sh
#!/bin/ksh
#
# sol_09_04_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

sudo /u01/crs11g/bin/crs_profile -create myClock -t
application -a crsclock_action.scr -p favored -h "$z $y" -o
ci=5,ra=2

more /u01/crs11g/crs/profile/myClock.cap
[oracle@vx0306 less09]$


[oracle@vx0306 less09]$ ./sol_09_04_03.sh
NAME=myClock
TYPE=application
ACTION_SCRIPT=crsclock_action.scr
ACTIVE_PLACEMENT=0
AUTO_START=restore
CHECK_INTERVAL=5
```

```
DESCRIPTION=myClock
FAILOVER_DELAY=0
FAILURE_INTERVAL=0
FAILURE_THRESHOLD=0
HOSTING_MEMBERS=vx0313 vx0306
OPTIONAL_RESOURCES=
PLACEMENT=favored
REQUIRED_RESOURCES=
RESTART_ATTEMPTS=2
SCRIPT_TIMEOUT=60
START_TIMEOUT=0
STOP_TIMEOUT=0
UPTIME_THRESHOLD=7d
USR_ORA_ALERT_NAME=
USR_ORA_CHECK_TIMEOUT=0
USR_ORA_CONNECT_STR=/ as sysdba
USR_ORA_DEBUG=0
USR_ORA_DISCONNECT=false
USR_ORA_FLAGS=
USR_ORA_IF=
USR_ORA_INST_NOT_SHUTDOWN=
USR_ORA_LANG=
USR_ORA_NETMASK=
USR_ORA_OPEN_MODE=
USR_ORA_OPI=false
USR_ORA_PFILE=
USR_ORA_PRECONNECT=none
USR_ORA_SRV=
USR_ORA_START_TIMEOUT=0
USR_ORA_STOP_MODE=immediate
USR_ORA_STOP_TIMEOUT=0
USR_ORA_VIP=
[oracle@vx0306 less09]$
```


4) Register the myClock resource with CRS, and make sure user oracle can manage that resource.

```
[oracle@vx0306 less09]$ cat sol_09_04_04.sh
#!/bin/ksh
#
# sol_09_04_04.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

/u01/crs11g/bin/crs_stat myClock

sudo /u01/crs11g/bin/crs_register myClock

sudo /u01/crs11g/bin/crs_setperm myClock -u user:oracle:r-x

/u01/crs11g/bin/crs_stat myClock
```

```
/u01/crs11g/bin/crs_stat -t
[oracle@vx0306 less09]$


[oracle@vx0306 less09]$ ./sol_09_04_04.sh
CRS-0210: Could not find resource 'myClock'.

NAME=myClock
TYPE=application
TARGET=OFFLINE
STATE=OFFLINE

Name           Type           Target    State     Host
------------------------------------------------------------
myClock        application    OFFLINE   OFFLINE
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora....B2.inst application    ONLINE    ONLINE    vx0313
ora.RDB.db     application    ONLINE    ONLINE    vx0306
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
ora....SM2.asm application    ONLINE    ONLINE    vx0313
ora....13.lsnr application    ONLINE    ONLINE    vx0313
ora.vx0313.gsd application    ONLINE    ONLINE    vx0313
ora.vx0313.ons application    ONLINE    ONLINE    vx0313
ora.vx0313.vip application    ONLINE    ONLINE    vx0313
[oracle@vx0306 less09]$
```

5) Make sure you have a VNC session under user oracle opened on your first node terminal window. This is necessary to see the clock. Now start the myClock resource as user `oracle`, and determine the location and status of this resource after startup. Look at the VNC session on first node; you should see the clock.

```
[oracle@vx0306 less09]$ cat sol_09_04_05.sh
#!/bin/ksh
#
# sol_09_04_05.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

/u01/crs11g/bin/crs_start myClock

sleep 30

/u01/crs11g/bin/crs_stat -t
[oracle@vx0306 less09]$
```

```
[oracle@vx0306 less09]$ ./sol_09_04_05.sh
Attempting to start 'myClock' on member 'vx0313'
Start of 'myClock' on member 'vx0313' succeeded.
Name           Type           Target    State     Host
------------------------------------------------------------
myClock        application    ONLINE    ONLINE    vx0313
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora....B2.inst application    ONLINE    ONLINE    vx0313
ora.RDB.db     application    ONLINE    ONLINE    vx0306
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
ora....SM2.asm application    ONLINE    ONLINE    vx0313
ora....13.lsnr application    ONLINE    ONLINE    vx0313
ora.vx0313.gsd application    ONLINE    ONLINE    vx0313
ora.vx0313.ons application    ONLINE    ONLINE    vx0313
ora.vx0313.vip application    ONLINE    ONLINE    vx0313
[oracle@vx0306 less09]$
```

6) Kill the xclock application on the node on which it was started. What do you observe?

   a) You should see the clock restarting on VNC session on first node.

```
[oracle@vx0306 less09]$ cat sol_09_04_06.sh
#!/bin/ksh
#
# sol_09_04_06.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

echo "CLOCKPID=\`ps -ef | grep xclock | grep -v grep | grep -v
xclock_app | awk '{ print \$2 }'\`" >
/home/oracle/solutions/less09/killclockpid.sh
echo "sudo kill -9 \$CLOCKPID" >>
/home/oracle/solutions/less09/killclockpid.sh

chmod 777 /home/oracle/solutions/less09/killclockpid.sh

/home/oracle/solutions/less09/killclockpid.sh

scp /home/oracle/solutions/less09/killclockpid.sh
$z:/home/oracle/solutions/less09/

ssh $z /home/oracle/solutions/less09/killclockpid.sh
[oracle@vx0306 less09]$
```

```
[oracle@vx0306 less09]$ ./sol_09_04_06.sh
usage: kill [ -s signal | -p ] [ -a ] pid ...
       kill -l [ signal ]
killclockpid.sh                              100%  112
0.1KB/s   00:00
[oracle@vx0306 less09]$
```

7) Determine the node on which myClock is currently running.

```
[oracle@vx0306 less09]$ cat sol_09_04_07.sh
#!/bin/ksh
#
# sol_09_04_07.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

/u01/crs11g/bin/crs_stat -t
[oracle@vx0306 less09]$


[oracle@vx0306 less09]$ ./sol_09_04_07.sh
Name           Type           Target    State     Host
------------------------------------------------------------
myClock        application    ONLINE    ONLINE    vx0313
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora....B2.inst application    ONLINE    ONLINE    vx0313
ora.RDB.db     application    ONLINE    ONLINE    vx0306
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
ora....SM2.asm application    ONLINE    ONLINE    vx0313
ora....13.lsnr application    ONLINE    ONLINE    vx0313
ora.vx0313.gsd application    ONLINE    ONLINE    vx0313
ora.vx0313.ons application    ONLINE    ONLINE    vx0313
ora.vx0313.vip application    ONLINE    ONLINE    vx0313
[oracle@vx0306 less09]$
```

8) Repeat the following sequence twice: kill the xclock application and wait for it to be restarted. After the last restart, what do you observe?

   a) You should see that after three kills, myClock is relocated to the other node. This is due to the *ra* attribute that you defined for myClock.

```
[oracle@vx0306 less09]$ cat sol_09_04_08.sh
#!/bin/ksh
#
# sol_09_04_08.sh
#
```

```
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

cd /home/oracle/solutions/less09

echo "CLOCKPID=\`ps -ef | grep xclock | grep -v grep | grep -v
xclock_app | awk '{ print \$2 }'\`" > killclockpid.sh
echo "sudo kill -9 \$CLOCKPID" >> killclockpid.sh

chmod 777 killclockpid.sh

./killclockpid.sh

scp killclockpid.sh $z:/home/oracle/solutions/less09
ssh $z "chmod 777
/home/oracle/solutions/less09/killclockpid.sh"
ssh $z /home/oracle/solutions/less09/killclockpid.sh

sleep 20

./killclockpid.sh

ssh $z /home/oracle/solutions/less09/killclockpid.sh

sleep 20

/u01/crs11g/bin/crs_stat -t
[oracle@vx0306 less09]$


[oracle@vx0306 less09]$ ./sol_09_04_08.sh
usage: kill [ -s signal | -p ] [ -a ] pid ...
       kill -l [ signal ]
killclockpid.sh                              100%  112
0.1KB/s   00:00
usage: kill [ -s signal | -p ] [ -a ] pid ...
       kill -l [ signal ]
Name           Type           Target    State     Host
------------------------------------------------------------
myClock        application    ONLINE    ONLINE    vx0306
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora....B2.inst application    ONLINE    ONLINE    vx0313
ora.RDB.db     application    ONLINE    ONLINE    vx0306
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
ora....SM2.asm application    ONLINE    ONLINE    vx0313
```

```
ora....13.lsnr application    ONLINE    ONLINE    vx0313
ora.vx0313.gsd application    ONLINE    ONLINE    vx0313
ora.vx0313.ons application    ONLINE    ONLINE    vx0313
ora.vx0313.vip application    ONLINE    ONLINE    vx0313
[oracle@vx0306 less09]$
```

9) Stop, unregister myClock resource, and remove the *.scr* and *.cap* files that were created.

```
[oracle@vx0306 less09]$ cat sol_09_04_09.sh
#!/bin/ksh
#
# sol_09_04_09.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

/u01/crs11g/bin/crs_stop myClock

/u01/crs11g/bin/crs_stat -t

sudo /u01/crs11g/bin/crs_unregister myClock

/u01/crs11g/bin/crs_stat -t

sudo rm /u01/crs11g/crs/profile/myClock.cap
sudo rm /u01/crs11g/crs/script/crsclock_action.scr

ssh $z "sudo rm /u01/crs11g/crs/profile/myClock.cap"
ssh $z "sudo rm /u01/crs11g/crs/script/crsclock_action.scr"
[oracle@vx0306 less09]$



[oracle@vx0306 less09]$ ./sol_09_04_09.sh
Attempting to stop `myClock` on member `vx0306`
Stop of `myClock` on member `vx0306` succeeded.
Name           Type           Target    State     Host
------------------------------------------------------------
myClock        application    OFFLINE   OFFLINE
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora....B2.inst application    ONLINE    ONLINE    vx0313
ora.RDB.db     application    ONLINE    ONLINE    vx0306
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
ora....SM2.asm application    ONLINE    ONLINE    vx0313
```

```
ora....13.lsnr application    ONLINE    ONLINE    vx0313
ora.vx0313.gsd application    ONLINE    ONLINE    vx0313
ora.vx0313.ons application    ONLINE    ONLINE    vx0313
ora.vx0313.vip application    ONLINE    ONLINE    vx0313
Name           Type           Target    State     Host
------------------------------------------------------------
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora....B2.inst application    ONLINE    ONLINE    vx0313
ora.RDB.db     application    ONLINE    ONLINE    vx0306
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
ora....SM2.asm application    ONLINE    ONLINE    vx0313
ora....13.lsnr application    ONLINE    ONLINE    vx0313
ora.vx0313.gsd application    ONLINE    ONLINE    vx0313
ora.vx0313.ons application    ONLINE    ONLINE    vx0313
ora.vx0313.vip application    ONLINE    ONLINE    vx0313
rm: cannot remove `/u01/crs11g/crs/profile/myClock.cap': No
such file or directory
[oracle@vx0306 less09]$
```

Practices for Lesson 10

Practice 10-1: Diagnosing Oracle Clusterware Components

The following short exercises are intended to make you familiar with various aspects of inspecting and diagnosing your Oracle Clusterware components including log file locations and assorted utilities provided to aid in data collection activities.

1) In this exercise you stop the crsd process uncleanly and inspect the logs that are generated. Run the script cleanlog.sh as root. Using the ps command find the process ID for the crsd process and kill it with a signal 9. Wait a few moments and change directory to /u01/crs11g/hostname and inspect the various log files that are generated.

   a) Change directory to $HOME/solutions/less10 and run the cleanlog.sh using sudo.

```
[oracle@vx0306 less10]$ cat sol_10_01_01a.sh
# sol_10_01_01a.sh

cd $HOME/solutions/less10

sudo ./cleanlog.sh
[oracle@vx0306 less10]$


[oracle@vx0306 less10]$ cat cleanlog.sh
# cleanlog.sh

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

HOST=`hostname|cut -c 1-11`

cd /u01/crs11g/log/$y
find . -type f -exec rm {} \;
[oracle@vx0306 less10]$


[oracle@vx0306 less10]$ ./sol_10_01_01a.sh
[oracle@vx0306 less10]$
```

   b) Using the ps command find the process ID of the crsd process. Use the kill command with sudo to kill the process since it is owned by root.

```
[oracle@vx0306 less10]$ cat sol_10_01_01b.sh
# sol_10_01_01b.sh

ps -ef|grep -i crsd

ps -ef | grep -i "crsd" | grep -v grep | grep -v killcrs | awk
'{print "sudo kill -9 " $2 }' >
/home/oracle/solutions/less10/z.sh

chmod 777 /home/oracle/solutions/less10/z.sh

echo "Killing crsd..."
```

```
/home/oracle/solutions/less10/z.sh
[oracle@vx0306 less10]$



[oracle@vx0306 less10]$ ./sol_10_01_01b.sh
root      3220     1  0 Nov16 ?        00:00:00 /bin/sh
/etc/init.d/init.crsd run
root      7658  3220  0 Nov16 ?        00:07:17
/u01/crs11g/bin/crsd.bin reboot
oracle   26952 26950  0 02:54 pts/1    00:00:00 grep -i crsd
Killing crsd...
[oracle@vx0306 less10]$
```

c) Change directory to /u01/crs11g/log/<hostname> and find the log files generated by the termination of the crsd process.

```
[oracle@vx0306 less10]$ cat sol_10_01_01c.sh
# sol_10_01_01c.sh

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

HOST=`hostname|cut -c 1-11`
cd /u01/crs11g/log/$y
find . -type f -print
[oracle@vx0306 less10]$


[oracle@vx0306 less10]$ ./sol_10_01_01c.sh
./crsd/crsdOUT.log
./crsd/crsd.log
./racg/ora.RDB.db.log
./alertvx0306.log
[oracle@vx0306 less10]$
```

d) Inspect the various files that were generated, starting with the Clusterware Alert log.

```
[oracle@vx0306 less10]$ cat sol_10_01_01d.sh
# sol_10_01_01d.sh

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

HOST=`hostname|cut -c 1-11`
cd /u01/crs11g/log/$y
cat ./alert$y.log
cat ./crsd/crsd.log
cat ./client/ocr*
cat ./client/css*
```

Practice 10-1: Diagnosing Oracle Clusterware Components (continued)

```
cat ./racg/ora*
[oracle@vx0306 less10]$



[oracle@vx0306 less10]$ ./sol_10_01_01d.sh
Oracle Database 11g CRS Release 11.1.0.6.0 - Production
Copyright 1996, 2007 Oracle. All rights reserved.
2007-11-19 02:55:02.303
[crsd(27280)]CRS-1012:The OCR service started on node vx0306.
2007-11-19 02:55:14.234
[crsd(27280)]CRS-1201:CRSD started on node vx0306.
Oracle Database 11g CRS Release 11.1.0.6.0 - Production
Copyright 1996, 2007 Oracle. All rights reserved.
2007-11-19 02:57:00.698: [ default][3056539872]prlsndmain:
olsnodes successful!!
Oracle Database 11g CRS Release 11.1.0.6.0 - Production
Copyright 1996, 2007 Oracle. All rights reserved.
2007-11-19 02:55:11.535: [ COMMCRS][3054398368]clsc_connect:
(0x816b678) no listener at
(ADDRESS=(PROTOCOL=ipc)(KEY=ora_crsqs))

2007-11-19 02:55:11.535: [    RACG][3055859392]
[27501][3055859392][ora.RDB.db]: clsrcqryapi: crs_qstat error

[oracle@vx0306 less10]$ vi sol_10_01_01d.sh
[oracle@vx0306 less10]$ ./sol_10_01_01d.sh
Oracle Database 11g CRS Release 11.1.0.6.0 - Production
Copyright 1996, 2007 Oracle. All rights reserved.
2007-11-19 02:55:02.303
[crsd(27280)]CRS-1012:The OCR service started on node vx0306.
2007-11-19 02:55:14.234
[crsd(27280)]CRS-1201:CRSD started on node vx0306.
Oracle Database 11g CRS Release 11.1.0.6.0 - Production
Copyright 1996, 2007 Oracle. All rights reserved.
2007-11-19 02:55:01.532: [ default][3055683264] CRS Daemon
Starting
2007-11-19 02:55:01.533: [ CRSMAIN][3055683264] Checking the
OCR device
2007-11-19 02:55:01.566: [ CRSMAIN][3055683264] Connecting to
the CSS Daemon
2007-11-19 02:55:01.636: [ CRSMAIN][3055683264] CRSD running
as the Privileged user

2007-11-19 02:55:02.005: [  CLSVER][3055683264] Active Version
from OCR:11.1.0.6.0
2007-11-19 02:55:02.005: [  CLSVER][3055683264] Active Version
and Software Version are same
2007-11-19 02:55:02.005: [ CRSMAIN][3055683264] Initializing
OCR
```

Practice 10-1: Diagnosing Oracle Clusterware Components (continued)

```
2007-11-19 02:55:02.076: [  OCRRAW][3055683264]proprioo: for
disk 0 (/dev/sdb1), id match (1), my id set
(125597460,2105976786) total id sets (1), 1st set
(125597460,2105976786), 2nd set (0,0) my votes (1), total
votes (2)
2007-11-19 02:55:02.076: [  OCRRAW][3055683264]proprioo: for
disk 1 (/dev/sdb2), id match (1), my id set
(125597460,2105976786) total id sets (1), 1st set
(125597460,2105976786), 2nd set (0,0) my votes (1), total
votes (2)
2007-11-19 02:55:02.319: [    CRSD][3055683264] ENV Logging
level for Module: allcomp  0
2007-11-19 02:55:02.321: [    CRSD][3055683264] ENV Logging
level for Module: default  0
2007-11-19 02:55:02.326: [    CRSD][3055683264] ENV Logging
level for Module: OCRRAW  0
2007-11-19 02:55:02.338: [    CRSD][3055683264] ENV Logging
level for Module: OCROSD  0
2007-11-19 02:55:02.346: [    CRSD][3055683264] ENV Logging
level for Module: OCRCAC  0
2007-11-19 02:55:02.348: [    CRSD][3055683264] ENV Logging
level for Module: COMMCRS  0
2007-11-19 02:55:02.350: [    CRSD][3055683264] ENV Logging
level for Module: COMMNS  0
2007-11-19 02:55:02.352: [    CRSD][3055683264] ENV Logging
level for Module: CRSUI  0
2007-11-19 02:55:02.353: [    CRSD][3055683264] ENV Logging
level for Module: CRSCOMM  0
2007-11-19 02:55:02.356: [    CRSD][3055683264] ENV Logging
level for Module: CRSRTI  0
2007-11-19 02:55:02.357: [    CRSD][3055683264] ENV Logging
level for Module: CRSMAIN  0
2007-11-19 02:55:02.359: [    CRSD][3055683264] ENV Logging
level for Module: CRSPLACE  0
2007-11-19 02:55:02.360: [    CRSD][3055683264] ENV Logging
level for Module: CRSAPP  0
2007-11-19 02:55:02.367: [    CRSD][3055683264] ENV Logging
level for Module: CRSRES  0
2007-11-19 02:55:02.376: [    CRSD][3055683264] ENV Logging
level for Module: CRSOCR  0
2007-11-19 02:55:02.377: [    CRSD][3055683264] ENV Logging
level for Module: CRSTIMER  0
2007-11-19 02:55:02.379: [    CRSD][3055683264] ENV Logging
level for Module: CRSEVT  0
2007-11-19 02:55:02.389: [    CRSD][3055683264] ENV Logging
level for Module: CRSD  0
2007-11-19 02:55:02.396: [    CRSD][3055683264] ENV Logging
level for Module: CLUCLS  0
2007-11-19 02:55:02.398: [    CRSD][3055683264] ENV Logging
level for Module: CLSVER  0
```

```
2007-11-19 02:55:02.400: [    CRSD][3055683264] ENV Logging
level for Module: CSSCLNT  0
2007-11-19 02:55:02.405: [    CRSD][3055683264] ENV Logging
level for Module: OCRAPI  0
2007-11-19 02:55:02.416: [    CRSD][3055683264] ENV Logging
level for Module: OCRUTL  0
2007-11-19 02:55:02.418: [    CRSD][3055683264] ENV Logging
level for Module: OCRMSG  0
2007-11-19 02:55:02.419: [    CRSD][3055683264] ENV Logging
level for Module: OCRCLI  0
2007-11-19 02:55:02.426: [    CRSD][3055683264] ENV Logging
level for Module: OCRSRV  0
2007-11-19 02:55:02.436: [    CRSD][3055683264] ENV Logging
level for Module: OCRMAS  0
2007-11-19 02:55:02.436: [ CRSMAIN][3055683264] Filename is
/u01/crs11g/crs/init/vx0306.pid
[  clsdmt][2800208800]Listening to
(ADDRESS=(PROTOCOL=ipc)(KEY=vx0306DBG_CRSD))
2007-11-19 02:55:02.549: [ CRSMAIN][3055683264] Using
Authorizer location: /u01/crs11g/crs/auth/
2007-11-19 02:55:02.570: [ CRSMAIN][3055683264] Initializing
RTI
2007-11-19 02:55:02.866: [ CRSMAIN][3055683264] Initializing
EVMMgr
2007-11-19 02:55:02.866: [CRSTIMER][2783157152] Timer Thread
Starting.
2007-11-19 02:55:03.761: [ CRSMAIN][3055683264] CRSD locked
during state recovery, please wait.
2007-11-19 02:55:05.318: [  CRSRES][3055683264] ora.vx0306.vip
check shows ONLINE
2007-11-19 02:55:06.517: [  CRSRES][3055683264] ora.vx0306.gsd
check shows ONLINE
2007-11-19 02:55:07.727: [  CRSRES][3055683264] ora.vx0306.ons
check shows ONLINE
2007-11-19 02:55:08.967: [  CRSRES][3055683264]
ora.vx0306.LISTENER_VX0306.lsnr check shows ONLINE
2007-11-19 02:55:10.307: [  CRSRES][3055683264]
ora.vx0306.ASM1.asm check shows ONLINE
2007-11-19 02:55:11.598: [  CRSRES][3055683264] ora.RDB.db
check shows ONLINE
2007-11-19 02:55:14.038: [  CRSRES][3055683264]
ora.RDB.RDB1.inst check shows ONLINE
2007-11-19 02:55:14.137: [ CRSMAIN][3055683264] CRSD
recovered, unlocked.
2007-11-19 02:55:14.138: [ CRSMAIN][3055683264] QS socket on:
(ADDRESS=(PROTOCOL=ipc)(KEY=ora_crsqs))
2007-11-19 02:55:14.207: [ CRSMAIN][3055683264] CRSD UI socket
on: (ADDRESS=(PROTOCOL=ipc)(KEY=CRSD_UI_SOCKET))
2007-11-19 02:55:14.233: [ CRSMAIN][3055683264] E2E socket on:
(ADDRESS=(PROTOCOL=tcp)(HOST=vx0306-priv)(PORT=49896))
```

Practice 10-1: Diagnosing Oracle Clusterware Components (continued)

```
2007-11-19 02:55:14.234: [ CRSMAIN][3055683264] Starting
Threads
2007-11-19 02:55:14.234: [ CRSMAIN][3055683264] CRS Daemon
Started.
2007-11-19 02:55:14.234: [ CRSMAIN][2730208160] Starting
runCommandServer for (UI = 1, E2E = 0). 0
2007-11-19 02:55:14.234: [ CRSMAIN][2728106912] Starting
runCommandServer for (UI = 1, E2E = 0). 1
cat: ./client/ocr*: No such file or directory
Oracle Database 11g CRS Release 11.1.0.6.0 - Production
Copyright 1996, 2007 Oracle. All rights reserved.
2007-11-19 03:02:01.261: [ default][3056130272]prlsndmain:
olsnodes successful!!
Oracle Database 11g CRS Release 11.1.0.6.0 - Production
Copyright 1996, 2007 Oracle. All rights reserved.
2007-11-19 02:57:00.698: [ default][3056539872]prlsndmain:
olsnodes successful!!
Oracle Database 11g CRS Release 11.1.0.6.0 - Production
Copyright 1996, 2007 Oracle. All rights reserved.
2007-11-19 02:55:11.535: [ COMMCRS][3054398368]clsc_connect:
(0x816b678) no listener at
(ADDRESS=(PROTOCOL=ipc)(KEY=ora_crsqs))

2007-11-19 02:55:11.535: [    RACG][3055859392]
[27501][3055859392][ora.RDB.db]: clsrcqryapi: crs_qstat error

[oracle@vx0306 less10]$
```

2) Use the crsctl command to check the health of your clusterware. Use the CLUVFY command to check the viability of the cluster nodeapps.

   a) Run the crsctl command from the /u01/crs11g/bin directory as shown below.

```
[oracle@vx0306 less10]$ cat sol_10_01_02a.sh
# sol_10_01_02a.sh

cd /u01/crs11g/bin
./crsctl check crs
[oracle@vx0306 less10]$


[oracle@vx0306 less10]$ ./sol_10_01_02a.sh
Cluster Synchronization Services appears healthy
Cluster Ready Services appears healthy
Event Manager appears healthy
[oracle@vx0306 less10]$
```

   b) Change directory to /stage/10gR2/rdbms/clusterware/cluvfy and run CLUVFY using the runcluvfy.sh script as shown below to check the nodeapps.

```
[oracle@vx0306 less10]$ cat sol_10_01_02b.sh
# sol_10_01_02b.sh
```

```
/u01/crs11g/bin/cluvfy comp nodeapp -n all -verbose
[oracle@vx0306 less10]$



[oracle@vx0306 less10]$ ./sol_10_01_02b.sh

Verifying node application existence

Checking node application existence...

Checking existence of VIP node application
  Node Name     Required                  Status
Comment
  ------------  ------------------------  --------------------
----  ----------
  vx0306        yes                       exists
passed
  vx0313        yes                       exists
passed
Result: Check passed.

Checking existence of ONS node application
  Node Name     Required                  Status
Comment
  ------------  ------------------------  --------------------
----  ----------
  vx0306        no                        exists
passed
  vx0313        no                        exists
passed
Result: Check passed.

Checking existence of GSD node application
  Node Name     Required                  Status
Comment
  ------------  ------------------------  --------------------
----  ----------
  vx0306        no                        exists
passed
  vx0313        no                        exists
passed
Result: Check passed.


Verification of node application existence was successful.
[oracle@vx0306 less10]$
```

3) Using the crs_stat command find all crs configuration data for the VIP resource located on your second node.

```
[oracle@vx0306 less10]$ cat sol_10_01_03.sh
# sol_10_01_03.sh

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

HOST=`hostname|cut -c 1-10`
cd /u01/crs11g/bin
./crs_stat -p ora.${z}.vip
[oracle@vx0306 less10]$



[oracle@vx0306 less10]$ ./sol_10_01_03.sh
NAME=ora.vx0313.vip
TYPE=application
ACTION_SCRIPT=/u01/crs11g/bin/racgwrap
ACTIVE_PLACEMENT=1
AUTO_START=1
CHECK_INTERVAL=15
DESCRIPTION=CRS application for VIP on a node
FAILOVER_DELAY=0
FAILURE_INTERVAL=0
FAILURE_THRESHOLD=0
HOSTING_MEMBERS=vx0313
OPTIONAL_RESOURCES=
PLACEMENT=favored
REQUIRED_RESOURCES=
RESTART_ATTEMPTS=0
SCRIPT_TIMEOUT=60
START_TIMEOUT=0
STOP_TIMEOUT=0
UPTIME_THRESHOLD=7d
USR_ORA_ALERT_NAME=
USR_ORA_CHECK_TIMEOUT=0
USR_ORA_CONNECT_STR=/ as sysdba
USR_ORA_DEBUG=0
USR_ORA_DISCONNECT=false
USR_ORA_FLAGS=
USR_ORA_IF=eth0
USR_ORA_INST_NOT_SHUTDOWN=
USR_ORA_LANG=
USR_ORA_NETMASK=255.255.252.0
USR_ORA_OPEN_MODE=
USR_ORA_OPI=false
USR_ORA_PFILE=
USR_ORA_PRECONNECT=none
USR_ORA_SRV=
USR_ORA_START_TIMEOUT=0
USR_ORA_STOP_MODE=immediate
```

Practice 10-1: Diagnosing Oracle Clusterware Components (continued)

```
USR_ORA_STOP_TIMEOUT=0
USR_ORA_VIP=10.216.4.75

[oracle@vx0306 less10]$
```

4) Determine the file(s) the OCR is using. Determine the total space available and what is currently being used.

   a) Run the ocrcheck command from the /u01/crs11g/bin directory as shown below.

```
[oracle@vx0306 less10]$ cat sol_10_01_04.sh
# sol_10_01_04.sh

cd /u01/crs11g/bin
./ocrcheck
[oracle@vx0306 less10]$



[oracle@vx0306 less10]$ ./sol_10_01_04.sh
Status of Oracle Cluster Registry is as follows :
         Version                  :          2
         Total space (kbytes)     :     616652
         Used space (kbytes)      :       3848
         Available space (kbytes) :     612804
         ID                       :  992212279
         Device/File Name         :  /dev/sdb1
                                    Device/File integrity
check succeeded
         Device/File Name         :  /dev/sdb2
                                    Device/File integrity
check succeeded

         Cluster registry integrity check succeeded

[oracle@vx0306 less10]$
```

   b) In this example, the OCR is using the file /dev/sdb1 and /dev/sdb2. The size of the file is 616652k and of that, 3848k is currently in use.

Practice 10-2: Fixing Oracle Clusterware issues

This practice introduces an error into the cluster synchronization service on your cluster. From your solutions directory, run the lab10-2-prep.sh script to prepare for the practice. Then run the breakcss.sh script to introduce the error condition. Observe the results and inspect your logs to identify and diagnose the problem. Repair the problem and return your cluster to normal operation.

1) Change your directory to $HOME/solutions/less10 and execute the lab10-2-prep.sh

```
[oracle@vx0306 less10]$ cat lab10-2-prep.sh
# lab10-2-prep.sh

# voting disk backup
dd if=/dev/sdb5 of=/home/oracle/solutions/less10/vdisk.bak
bs=4k 2>/dev/null
dd if=/dev/sdb5 of=/home/oracle/solutions/less10/vdisk.bak2
bs=8k 2>/dev/null

sudo cp /etc/init.d/init.cssd /home/oracle
sudo sed -i '43i/bin/sleep 30' /etc/init.d/init.cssd


[oracle@vx0306 less10]$


[oracle@vx0306 less10]$ ./lab10-2-prep.sh
[oracle@vx0306 less10]$
```

2) From the $HOME/solutions/less10 directory, execute the breakcss.sh script. The script stops the cluster nodeapps and changes the cluster configuration. The error is introduced at the end of the script execution.

   a) Your nodes will now reboot in reaction to the problem that was introduced. There is something seriously wrong!

```
[oracle@vx0306 less10]$ cat breakcss.sh
# breakcss.sh

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/srvctl stop database -d $DBNAME
```

```
/u01/crs11g/bin/srvctl stop asm -n $y

/u01/crs11g/bin/srvctl stop asm -n $z

/u01/crs11g/bin/srvctl stop nodeapps -n $z

/u01/crs11g/bin/srvctl stop nodeapps -n $y

/u01/crs11g/bin/crs_stat -t

ssh $z sudo /u01/crs11g/bin/crsctl stop crs

sleep 60

sudo /u01/crs11g/bin/crsctl stop crs

sleep 60

sudo /u01/crs11g/bin/crsctl delete css votedisk /dev/sdb6 -
force
sudo /u01/crs11g/bin/crsctl delete css votedisk /dev/sdb7 -
force

sudo /u01/crs11g/bin/crsctl start crs

sleep 300

/u01/crs11g/bin/crs_stat -t

ssh $z sudo /u01/crs11g/bin/crsctl start crs

sleep 300

/u01/crs11g/bin/crs_stat -t

while true
do
dd if=/dev/zero of=/dev/sdb5 bs=1M count=20
done

[oracle@vx0306 less10]$



[oracle@vx0306 less10]$ ./breakcss.sh
Name           Type           Target    State     Host
------------------------------------------------------------
ora....B1.inst application    OFFLINE   OFFLINE
ora....B2.inst application    OFFLINE   OFFLINE
ora.RDB.db     application    OFFLINE   OFFLINE
ora....SM1.asm application    OFFLINE   OFFLINE
ora....06.lsnr application    OFFLINE   OFFLINE
```

Practice 10-2: Fixing Oracle Clusterware issues (continued)

```
ora.vx0306.gsd application    OFFLINE   OFFLINE
ora.vx0306.ons application    OFFLINE   OFFLINE
ora.vx0306.vip application    OFFLINE   OFFLINE
ora....SM2.asm application    OFFLINE   OFFLINE
ora....13.lsnr application    OFFLINE   OFFLINE
ora.vx0313.gsd application    OFFLINE   OFFLINE
ora.vx0313.ons application    OFFLINE   OFFLINE
ora.vx0313.vip application    OFFLINE   OFFLINE
Stopping resources.
This could take several minutes.
Successfully stopped Oracle Clusterware resources
Stopping Cluster Synchronization Services.
Shutting down the Cluster Synchronization Services daemon.
Shutdown request successfully issued.
Stopping resources.
This could take several minutes.
Successfully stopped Oracle Clusterware resources
Stopping Cluster Synchronization Services.
Shutting down the Cluster Synchronization Services daemon.
Shutdown request successfully issued.
Successful deletion of voting disk /dev/sdb6.
Successful deletion of voting disk /dev/sdb7.
Attempting to start Oracle Clusterware stack
The CRS stack will be started shortly

Name           Type           Target    State     Host
------------------------------------------------------------
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora....B2.inst application    ONLINE    OFFLINE
ora.RDB.db     application    ONLINE    ONLINE    vx0306
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
ora....SM2.asm application    ONLINE    OFFLINE
ora....13.lsnr application    ONLINE    OFFLINE
ora.vx0313.gsd application    ONLINE    OFFLINE
ora.vx0313.ons application    ONLINE    OFFLINE
ora.vx0313.vip application    ONLINE    ONLINE    vx0306
Attempting to start Oracle Clusterware stack
The CRS stack will be started shortly
Name           Type           Target    State     Host
------------------------------------------------------------
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora....B2.inst application    ONLINE    ONLINE    vx0313
ora.RDB.db     application    ONLINE    ONLINE    vx0306
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
```

```
ora....SM2.asm application    ONLINE    ONLINE    vx0313
ora....13.lsnr application    ONLINE    ONLINE    vx0313
ora.vx0313.gsd application    ONLINE    ONLINE    vx0313
ora.vx0313.ons application    ONLINE    ONLINE    vx0313
ora.vx0313.vip application    ONLINE    ONLINE    vx0313
20+0 records in
20+0 records out
20+0 records in
20+0 records out
20+0 records in
20+0 records out
20+0 records in
20+0 records out
20+0 records in
20+0 records out
20+0 records in
20+0 records out
20+0 records in
20+0 records out
20+0 records in
20+0 records out
-- At this point your session gets disconnected!
```

3) From your classroom PC, attempt to ping the first node in your cluster. When you can ping the node successfully, log in.

```
[oracle@vx0309 ~]$ ping 10.216.4.19
PING 10.216.4.19 (10.216.4.19) 56(84) bytes of data.
64 bytes from 10.216.4.19: icmp_seq=0 ttl=64 time=10.2 ms
64 bytes from 10.216.4.19: icmp_seq=1 ttl=64 time=0.352 ms
64 bytes from 10.216.4.19: icmp_seq=2 ttl=64 time=0.323 ms
64 bytes from 10.216.4.19: icmp_seq=3 ttl=64 time=0.325 ms

--- 10.216.4.19 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.323/2.817/10.271/4.303 ms, pipe 2
[oracle@vx0301 ~]$
```

4) You must now inspect your Oracle Clusterware log files to find the problem.

   a) When the node is stable, change your directory to your log directory, /u01/crs11g/log/<host name>. List the log files. The first file you should look at is the clusterware alert log. In this example, that file is alerted-.log.

```
[oracle@vx0306 less10]$ cat sol_10_02_04.sh
# sol_10_02_04.sh

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

cd /u01/crs11g/log/$y*

ls -al
```

```
[oracle@vx0306 less10]$
[oracle@vx0308 less10]$ ./sol_10_02_04.sh
total 52
drwxr-xr-t  8 root   oinstall  4096 Nov 22 02:33 .
drwxr-xr-x  4 oracle oinstall  4096 Nov 26 02:04 ..
drwxr-x---  2 oracle oinstall  4096 Nov 22 02:33 admin
-rw-rw-r--  1 oracle oinstall 18421 Nov 26 02:48
alertvx0308.log
drwxrwx---  2 oracle oinstall  4096 Nov 26 02:48 client
drwxr-x---  2 root   oinstall  4096 Nov 22 02:34 crsd
drwxr-x---  4 oracle oinstall  4096 Nov 26 02:54 cssd
drwxr-x---  2 oracle oinstall  4096 Nov 22 05:32 evmd
drwxrwx--T  5 oracle oinstall  4096 Nov 22 02:42 racg
[oracle@vx0308 less10]$
```

5) Looking at the alert log, what could be the cause of the problem?
   a) You can see that the voting disk appears to be intermittently offline. This would certainly cause problems with CSS.

```
[oracle@vx0308 less10]$ cat sol_10_02_05.sh
# sol_10_02_05.sh

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

cd /u01/crs11g/log/$y*

view alert*.log
[oracle@vx0308 less10]$

[oracle@vx0308 less10]$ ./sol_10_02_05.sh

…
[crsd(6213)]CRS-1005:The OCR upgrade was completed. Version
has changed from 185599488 to 185599488. Details in
/u01/crs11g/log/vx0308/crsd/crsd.log.
2007-11-26 01:39:29.697
[cssd(7551)]CRS-1612:node vx0309 (2) at 50% heartbeat fatal,
eviction in 14.080 seconds
2007-11-26 01:39:36.768
[cssd(7551)]CRS-1611:node vx0309 (2) at 75% heartbeat fatal,
eviction in 7.010 seconds
2007-11-26 01:39:40.807
[cssd(7551)]CRS-1610:node vx0309 (2) at 90% heartbeat fatal,
eviction in 2.202 seconds
2007-11-26 01:39:41.817
[cssd(7551)]CRS-1610:node vx0309 (2) at 90% heartbeat fatal,
eviction in 1.192 seconds
```

Practice 10-2: Fixing Oracle Clusterware issues (continued)

```
2007-11-26 01:39:42.827
[cssd(7551)]CRS-1610:node vx0309 (2) at 90% heartbeat fatal,
eviction in 0.182 seconds
2007-11-26 01:39:43.788
[cssd(7551)]CRS-1607:CSSD evicting node vx0309. Details in
/u01/crs11g/log/vx0308/cssd/ocssd.log.
[cssd(7551)]CRS-1601:CSSD Reconfiguration complete. Active
nodes are vx0308 .
2007-11-26 01:39:43.911
[crsd(6213)]CRS-1204:Recovering CRS resources for node vx0309.
[cssd(7551)]CRS-1603:CSSD on node vx0308 shutdown by user.
2007-11-26 01:45:20.142
[cssd(6454)]CRS-1605:CSSD voting file is online: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
[cssd(6454)]CRS-1601:CSSD Reconfiguration complete. Active
nodes are vx0308 .
2007-11-26 01:45:27.868
[evmd(5062)]CRS-1401:EVMD started on node vx0308.
2007-11-26 01:45:27.996
[crsd(5835)]CRS-1005:The OCR upgrade was completed. Version
has changed from 185599488 to 185599488. Details in
/u01/crs11g/log/vx0308/crsd/crsd.log.
2007-11-26 01:45:27.997
[crsd(5835)]CRS-1012:The OCR service started on node vx0308.
2007-11-26 01:45:28.987
[crsd(5835)]CRS-1201:CRSD started on node vx0308.
2007-11-26 01:45:29.218
[crsd(5835)]CRS-1205:Auto-start failed for the CRS resource .
Details in vx0308.
2007-11-26 01:45:29.232
[crsd(5835)]CRS-1205:Auto-start failed for the CRS resource .
Details in vx0308.
[cssd(6454)]CRS-1601:CSSD Reconfiguration complete. Active
nodes are vx0308 vx0309 .
2007-11-26 01:51:15.029
[cssd(6454)]CRS-1604:CSSD voting file is offline: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:51:23.029
[cssd(6454)]CRS-1605:CSSD voting file is online: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:51:24.225
[cssd(6454)]CRS-1604:CSSD voting file is offline: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:51:33.413
[cssd(6454)]CRS-1605:CSSD voting file is online: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:51:34.470
[cssd(6454)]CRS-1604:CSSD voting file is offline: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:51:43.894
[cssd(6454)]CRS-1605:CSSD voting file is online: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
```

```
2007-11-26 01:51:46.128
[cssd(6454)]CRS-1604:CSSD voting file is offline: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:51:53.913
[cssd(6454)]CRS-1605:CSSD voting file is online: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:51:55.017
[cssd(6454)]CRS-1604:CSSD voting file is offline: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:52:04.424
[cssd(6454)]CRS-1605:CSSD voting file is online: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:52:06.618
[cssd(6454)]CRS-1604:CSSD voting file is offline: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:52:14.439
[cssd(6454)]CRS-1605:CSSD voting file is online: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:52:15.597
[cssd(6454)]CRS-1604:CSSD voting file is offline: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:52:24.802
[cssd(6454)]CRS-1605:CSSD voting file is online: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:52:25.838
[cssd(6454)]CRS-1604:CSSD voting file is offline: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:52:35.053
[cssd(6454)]CRS-1605:CSSD voting file is online: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:53:39.748
[cssd(6454)]CRS-1604:CSSD voting file is offline: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:53:47.775
[cssd(6454)]CRS-1605:CSSD voting file is online: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:53:48.984
[cssd(6454)]CRS-1604:CSSD voting file is offline: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 01:53:48.984
[cssd(6454)]CRS-1604:CSSD voting file is offline: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 02:02:07.242
[cssd(7569)]CRS-1605:CSSD voting file is online: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
2007-11-26 02:02:07.254
[cssd(7569)]CRS-1604:CSSD voting file is offline: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
[cssd(7569)]CRS-1601:CSSD Reconfiguration complete. Active
nodes are vx0308 vx0309 .
2007-11-26 02:02:08.937
[crsd(6260)]CRS-1012:The OCR service started on node vx0308.
```

```
2007-11-26 02:02:08.941
[evmd(6235)]CRS-1401:EVMD started on node vx0308.
2007-11-26 02:02:09.907
[crsd(6260)]CRS-1201:CRSD started on node vx0308.
2007-11-26 02:02:16.220
[cssd(7569)]CRS-1605:CSSD voting file is online: /dev/sdb5.
Details in /u01/crs11g/log/vx0308/cssd/ocssd.log.
…
```

6) The alert log indicates that further information can be found. Where would you have to look at?

   a) In the /u01/crs11g/log/<host name>/cssd/ocssd.log. The real problem is revealed here. You can see that the voting disk, /dev/sdb5 is corrupted. This was the cause of the reboot.

```
[oracle@vx0308 less10]$ cat sol_10_02_06.sh
# sol_10_02_06.sh

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

view /u01/crs11g/log/$y*/cssd/ocssd.log
[oracle@vx0308 less10]$

[oracle@vx0308 less10]$ ./sol_10_02_06.sh

…
[    CSSD]2007-11-26 01:39:36.767 [2941873056] >WARNING:
clssnmPollingThread: node vx0309 (2) at 75% heartbeat fatal,
eviction in 7.010 seconds
[    CSSD]2007-11-26 01:39:40.807 [2941873056] >WARNING:
clssnmPollingThread: node vx0309 (2) at 90% heartbeat fatal,
eviction in 2.970 seconds
[    CSSD]2007-11-26 01:39:41.817 [2941873056] >WARNING:
clssnmPollingThread: node vx0309 (2) at 90% heartbeat fatal,
eviction in 1.960 seconds
[    CSSD]2007-11-26 01:39:42.827 [2941873056] >WARNING:
clssnmPollingThread: node vx0309 (2) at 90% heartbeat fatal,
eviction in 0.950 seconds
[    CSSD]2007-11-26 01:39:43.787 [2941873056] >TRACE:
clssnmPollingThread: Eviction started for node vx0309 (2),
flags 0x0001, state 3, wt4c 0
[    CSSD]2007-11-26 01:39:43.787 [2925087648] >TRACE:
clssnmDoSyncUpdate: Initiating sync 91161491
[    CSSD]2007-11-26 01:39:43.787 [2925087648] >TRACE:
clssnmDoSyncUpdate: diskTimeout set to (27000)ms
[    CSSD]2007-11-26 01:39:43.787 [2925087648] >TRACE:
clssnmSetupAckWait: Ack message type (11)
[    CSSD]2007-11-26 01:39:43.787 [2925087648] >TRACE:
clssnmSetupAckWait: node(1) is ALIVE
```

```
[    CSSD]2007-11-26 01:39:43.787 [2925087648] >TRACE:
clssnmSendSync: syncSeqNo(91161491)
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmWaitForAcks: Ack message type(11), ackCount(1)
[    CSSD]2007-11-26 01:39:43.788 [3000888224] >TRACE:
clssnmHandleSync: diskTimeout set to (27000)ms
[    CSSD]2007-11-26 01:39:43.788 [3000888224] >TRACE:
clssnmHandleSync: Acknowledging sync: src[1] srcName[vx0308]
seq[1] sync[91161491]
[    CSSD]2007-11-26 01:39:43.788 [3056544128] >USER:
NMEVENT_SUSPEND [00][00][00][06]
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmWaitForAcks: done, msg type(11)
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmNeedConfReq: No configuration to change
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmDoSyncUpdate: Terminating node 2, vx0309,
misstime(30010) state(5)
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmSetupAckWait: Ack message type (13)
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmSetupAckWait: node(1) is ACTIVE
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmSendVote: syncSeqNo(91161491)
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmWaitForAcks: Ack message type(13), ackCount(1)
[    CSSD]2007-11-26 01:39:43.788 [3000888224] >TRACE:
clssnmSendVoteInfo: node(1) syncSeqNo(91161491)
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmWaitForAcks: done, msg type(13)
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmCheckDskInfo: Checking disk info...
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmEvict: Start
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmEvict: Evicting node 2, vx0309, birth 91161446, death
91161491, impendingrcfg 1, stateflags 0x1
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmEvict: Fencing node 2, vx0309, with SAGE
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmWaitOnEvictions: Start
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmWaitOnEvictions: node 2, undead 0, reqid 0
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmSetupAckWait: Ack message type (15)
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmSetupAckWait: node(1) is ACTIVE
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmSendUpdate: syncSeqNo(91161491)
[    CSSD]2007-11-26 01:39:43.788 [2925087648] >TRACE:
clssnmWaitForAcks: Ack message type(15), ackCount(1)
```

Practice 10-2: Fixing Oracle Clusterware issues (continued)

```
[    CSSD]2007-11-26 01:39:43.788 [3000888224] >TRACE:
clssnmUpdateNodeState: node 0, state (0/0) unique (0/0)
prevConuni(0) birth (0/0) (old/new)
[    CSSD]2007-11-26 01:39:43.788 [3000888224] >TRACE:
clssnmUpdateNodeState: node 1, state (3/3) unique
(1195926668/1195926668) prevConuni(0) birth
(91161490/91161490) (old/new)
[    CSSD]2007-11-26 01:39:43.788 [3000888224] >TRACE:
clssnmUpdateNodeState: node 2, state (5/0) unique
(1195720907/1195720907) prevConuni(1195720907) birth
(91161446/91161446) (old/new)
[    CSSD]2007-11-26 01:39:43.788 [3000888224] >TRACE:
clssnmDeactivateNode: node 2 (vx0309) left cluster

[    CSSD]2007-11-26 01:39:43.788 [3000888224] >USER:
clssnmHandleUpdate: SYNC(91161491) from node(1) completed
[    CSSD]2007-11-26 01:39:43.788 [3000888224] >USER:
clssnmHandleUpdate: NODE 1 (vx0308) IS ACTIVE MEMBER OF
CLUSTER
[    CSSD]2007-11-26 01:39:43.788 [3000888224] >TRACE:
clssnmHandleUpdate: diskTimeout set to (200000)ms
[    CSSD]2007-11-26 01:39:43.789 [2925087648] >TRACE:
clssnmWaitForAcks: done, msg type(15)
[    CSSD]2007-11-26 01:39:43.789 [2925087648] >TRACE:
clssnmDoSyncUpdate: Sync 91161491 complete!
[    CSSD]2007-11-26 01:39:43.789 [2899909536] >TRACE:
clssgmReconfigThread:  started for reconfig (91161491)
[    CSSD]2007-11-26 01:39:43.789 [2899909536] >USER:
NMEVENT_RECONFIG [00][00][00][02]
[    CSSD]2007-11-26 01:39:43.789 [2899909536] >TRACE:
clssgmCleanupGrocks: cleaning up grock crs_version type 2
[    CSSD]2007-11-26 01:39:43.789 [2899909536] >TRACE:
clssgmCleanupOrphanMembers: cleaning up remote mbr(2)
grock(crs_version) birth(91161446/0)
[    CSSD]2007-11-26 01:39:43.789 [2899909536] >TRACE:
clssgmCleanupGrocks: cleaning up grock EVMDMAIN type 2
[    CSSD]2007-11-26 01:39:43.789 [2899909536] >TRACE:
clssgmCleanupOrphanMembers: cleaning up remote mbr(2)
grock(EVMDMAIN) birth(91161446/0)
[    CSSD]2007-11-26 01:39:43.789 [2899909536] >TRACE:
clssgmCleanupGrocks: cleaning up grock CRSDMAIN type 2
[    CSSD]2007-11-26 01:39:43.789 [2899909536] >TRACE:
clssgmCleanupGrocks: cleaning up grock ocr_vx_cluster03 type 2
[    CSSD]2007-11-26 01:39:43.789 [2899909536] >TRACE:
clssgmCleanupGrocks: cleaning up grock _ORA_CRS_MEMBER_vx0308
type 3
[    CSSD]2007-11-26 01:39:43.789 [2899909536] >TRACE:
clssgmCleanupGrocks: cleaning up grock #CSS_CLSSOMON type 2
[    CSSD]2007-11-26 01:39:43.789 [2899909536] >TRACE:
clssgmEstablishConnections: 1 nodes in cluster incarn 91161491
[    CSSD]2007-11-26 01:39:43.789 [2950532000] >TRACE:
clssgmPeerListener: connects done (1/1)
```

```
[    CSSD]2007-11-26 01:39:43.789 [2899909536] >TRACE:
clssgmEstablishMasterNode: MASTER for 91161491 is node(1)
birth(91161490)
[    CSSD]2007-11-26 01:39:43.789 [2899909536] >TRACE:
clssgmChangeMasterNode: requeued 0 RPCs
[    CSSD]2007-11-26 01:39:43.789 [2899909536] >TRACE:
clssgmMasterCMSync: Synchronizing group/lock status
[    CSSD]2007-11-26 01:39:43.789 [2899909536] >TRACE:
clssgmMasterSendDBDone: group/lock status synchronization
complete
[    CSSD]CLSS-3000: reconfiguration successful, incarnation
91161491 with 1 nodes

[    CSSD]CLSS-3001: local node number 1, master node number 1

[    CSSD]2007-11-26 01:39:43.848 [2899909536] >TRACE:
clssgmReconfigThread:  completed for reconfig(91161491), with
status(1)
[    CSSD]2007-11-26 01:40:13.684 [2984102816] >WARNING:
clssgmShutDown: Received explicit shutdown request from
client.
[    CSSD]2007-11-26 01:40:13.684 [2984102816] >TRACE:
clssgmClientShutdown: Aborting client (0x835f580) proc
(0x835ecc8)
[    CSSD]2007-11-26 01:40:13.684 [2984102816] >TRACE:
clssgmClientShutdown: Aborting client (0x8364128) proc
(0x835ef20)
[    CSSD]2007-11-26 01:40:13.684 [2984102816] >TRACE:
clssgmClientShutdown: Aborting client (0x8364580) proc
(0x835ef20)
[    CSSD]2007-11-26 01:40:13.684 [2984102816] >TRACE:
clssgmClientShutdown: Aborting client (0x8364af8) proc
(0x835ef20)
[    CSSD]2007-11-26 01:40:13.685 [2984102816] >TRACE:
clssgmClientShutdown: Aborting client (0x835f320) proc
(0x835fdf8)
[    CSSD]2007-11-26 01:40:13.685 [2984102816] >TRACE:
clssgmClientShutdown: waited 0 seconds on 5 IO capable clients
[    CSSD]2007-11-26 01:40:13.685 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835ecc8), pid (7479)
[    CSSD]2007-11-26 01:40:13.685 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835ef20), pid (6213)
[    CSSD]2007-11-26 01:40:13.685 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835ef20), pid (6213)
[    CSSD]2007-11-26 01:40:13.685 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835ef20), pid (6213)
[    CSSD]2007-11-26 01:40:13.685 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835ef20), pid (6213)
[    CSSD]2007-11-26 01:40:13.685 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835ef20), pid (6213)
[    CSSD]2007-11-26 01:40:13.685 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835fdf8), pid (6223)
```

Practice 10-2: Fixing Oracle Clusterware issues (continued)

```
[    CSSD]2007-11-26 01:40:13.685 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835ecc8), pid (7479)
[    CSSD]2007-11-26 01:40:13.685 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835ef20), pid (6213)
[    CSSD]2007-11-26 01:40:13.685 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835ef20), pid (6213)
[    CSSD]2007-11-26 01:40:13.685 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835ef20), pid (6213)
[    CSSD]2007-11-26 01:40:13.685 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835fdf8), pid (6223)
[    CSSD]2007-11-26 01:40:13.688 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835ecc8), pid (7479)
[    CSSD]2007-11-26 01:40:13.688 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835ef20), pid (6213)
[    CSSD]2007-11-26 01:40:13.688 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835ef20), pid (6213)
[    CSSD]2007-11-26 01:40:13.688 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835ef20), pid (6213)
[    CSSD]2007-11-26 01:40:13.688 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835fdf8), pid (6223)
[    CSSD]2007-11-26 01:40:13.709 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835ecc8), pid (7479)
[    CSSD]2007-11-26 01:40:13.709 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835fdf8), pid (6223)
[    CSSD]2007-11-26 01:40:13.742 [2984102816] >TRACE:
clssgmClientShutdown: I/O capable proc (0x835fdf8), pid (6223)
[    CSSD]2007-11-26 01:40:13.764 [2984102816] >WARNING:
clssgmClientShutdown: graceful shutdown completed.
[  clsdmt]Listening to
(ADDRESS=(PROTOCOL=ipc)(KEY=vx0308DBG_CSSD))
[    CSSD]2007-11-26 01:45:19.974 >USER:    Oracle Database
10g CSS Release 11.1.0.6.0 Production Copyright 1996, 2004
Oracle.  All rights reserved.
[    CSSD]2007-11-26 01:45:19.974 >USER:    CSS daemon log for
node vx0308, number 1, in cluster vx_cluster03
[    CSSD]2007-11-26 01:45:19.984 [3055966592] >TRACE:
clssscmain: local-only set to false
[    CSSD]2007-11-26 01:45:20.001 [3055966592] >TRACE:
clssnmReadNodeInfo: added node 1 (vx0308) to cluster
[    CSSD]2007-11-26 01:45:20.016 [3055966592] >TRACE:
clssnmReadNodeInfo: added node 2 (vx0309) to cluster
[    CSSD]2007-11-26 01:45:20.020 [3055966592] >WARNING:
clssnmReadWallet: Open Wallet returned 28759
[    CSSD]2007-11-26 01:45:20.020 [3055966592] >WARNING:
clssnmInitNMInfo: Node not configured for node kill
[    CSSD]2007-11-26 01:45:20.037 [3028274080] >TRACE:
clssnm_skgxninit: Compatible vendor clusterware not in use
[    CSSD]2007-11-26 01:45:20.037 [3028274080] >TRACE:
clssnm_skgxnmon: skgxn init failed
[    CSSD]2007-11-26 01:45:20.044 [3055966592] >TRACE:
clssnmNMInitialize: Network heartbeat thresholds are:
```

```
impending reconfig 15000 ms, reconfig start (misscount) 30000
ms
[    CSSD]2007-11-26 01:45:20.046 [3055966592] >TRACE:
clssnmNMInitialize: Voting file I/O timeouts are: short 27000
ms, long 200000 ms
[    CSSD]2007-11-26 01:45:20.052 [3055966592] >TRACE:
clssnmDiskStateChange: state from 1 to 2 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:45:20.052 [3028274080] >TRACE:
clssnmvDPT: spawned for disk 0 (/dev/sdb5)
[    CSSD]2007-11-26 01:45:20.119 [3028274080] >TRACE:
clssnmDiskStateChange: state from 2 to 4 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:45:20.142 [3055966592] >TRACE:
clssnmFatalInit: fatal mode enabled
[    CSSD]2007-11-26 01:45:20.184 [3019201440] >TRACE:
clssnmvKillBlockThread: spawned for disk 0 (/dev/sdb5) initial
sleep interval (1000)ms
[    CSSD]2007-11-26 01:45:20.188 [3028274080] >TRACE:
clssnmReadDskHeartbeat: node 2, vx0309, has a disk HB, but no
network HB, DHB has rcfg 91161491, wrtcnt, 338751, LATS
136615880, lastSeqNo 338751, timestamp 1196062746/394546570
[    CSSD]2007-11-26 01:45:20.200 [3001019296] >TRACE:
clssnmClusterListener: Listening on
(ADDRESS=(PROTOCOL=tcp)(HOST=vx0308-priv)(PORT=49895))

[    CSSD]2007-11-26 01:45:20.207 [3001019296] >TRACE:
clssnmClusterListener: Probing node vx0309 (2),
probcon(0x8282438)
[    CSSD]2007-11-26 01:45:20.218 [3001019296] >TRACE:
clsc_send_msg: (0x82820a8) NS err (12571, 12560), transport
(530, 111, 0)

[    CSSD]2007-11-26 01:45:20.218 [3001019296] >TRACE:
clssnmDiscHelper: vx0309, node(2) connection failed, con
(0x8282438), probe(0x8282438)
[    CSSD]2007-11-26 01:45:20.218 [2992360352] >TRACE:
clssgmDeathChkThread: Spawned
[    CSSD]2007-11-26 01:45:20.268 [2983967648] >TRACE:
clssgmclientlsnr: listening on
(ADDRESS=(PROTOCOL=ipc)(KEY=Oracle_CSS_LclLstnr_vx_cluster03_1
))
[    CSSD]2007-11-26 01:45:20.268 [2983967648] >TRACE:
clssgmclientlsnr: listening on
(ADDRESS=(PROTOCOL=ipc)(KEY=OCSSD_LL_vx0308_vx_cluster03))
[    CSSD]2007-11-26 01:45:20.268 [2983967648] >TRACE:
clssgmclientlsnr: listening on
(ADDRESS=(PROTOCOL=ipc)(KEY=OCSSD_LL_vx0308_))
[    CSSD]2007-11-26 01:45:20.296 [2949995424] >TRACE:
clssgmPeerListener: Listening on
(ADDRESS=(PROTOCOL=tcp)(DEV=22)(HOST=10.196.30.29)(PORT=18702)
)
[    CSSD]2007-11-26 01:45:27.387 [2924551072] >TRACE:
clssnmRcfgMgrThread: Local Join
```

```
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >WARNING:
clssnmLocalJoinEvent: takeover succ
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmDoSyncUpdate: Initiating sync 0
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmDoSyncUpdate: diskTimeout set to (27000)ms
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmSetupAckWait: Ack message type (11)
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmSetupAckWait: node(1) is ALIVE
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmSetFirstIncarn: Incarnation 0
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmSetFirstIncarn: Node 1 incarnation 91161492
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmSetFirstIncarn: Node 2 incarnation 91161491
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmSetFirstIncarn: Incarnation set to 91161493
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmSendSync: syncSeqNo(91161493)
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmWaitForAcks: Ack message type(11), ackCount(1)
[    CSSD]2007-11-26 01:45:27.388 [3001019296] >TRACE:
clssnmHandleSync: diskTimeout set to (27000)ms
[    CSSD]2007-11-26 01:45:27.388 [3001019296] >TRACE:
clssnmHandleSync: Acknowledging sync: src[1] srcName[vx0308]
seq[1] sync[91161493]
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmWaitForAcks: done, msg type(11)
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmNeedConfReq: No configuration to change
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmDoSyncUpdate: node(1) is transitioning from joining
state to active state
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmSetupAckWait: Ack message type (13)
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmSetupAckWait: node(1) is ACTIVE
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmSendVote: syncSeqNo(91161493)
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmWaitForAcks: Ack message type(13), ackCount(1)
[    CSSD]2007-11-26 01:45:27.388 [3001019296] >TRACE:
clssnmSendVoteInfo: node(1) syncSeqNo(91161493)
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmWaitForAcks: done, msg type(13)
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmCheckDskInfo: Checking disk info...
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmCheckDskInfo: diskTimeout set to (200000)ms
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmEvict: Start
```

```
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmWaitOnEvictions: Start
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmSetupAckWait: Ack message type (15)
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmSetupAckWait: node(1) is ACTIVE
[    CSSD]2007-11-26 01:45:27.388 [2924551072] >TRACE:
clssnmSendUpdate: syncSeqNo(91161493)
[    CSSD]2007-11-26 01:45:27.389 [2924551072] >TRACE:
clssnmWaitForAcks: Ack message type(15), ackCount(1)
[    CSSD]2007-11-26 01:45:27.389 [3001019296] >TRACE:
clssnmUpdateNodeState: node 0, state (0/0) unique (0/0)
prevConuni(0) birth (0/0) (old/new)
[    CSSD]2007-11-26 01:45:27.389 [3001019296] >TRACE:
clssnmUpdateNodeState: node 1, state (2/3) unique
(1196063119/1196063119) prevConuni(0) birth
(91161493/91161493) (old/new)
[    CSSD]2007-11-26 01:45:27.389 [3001019296] >TRACE:
clssnmUpdateNodeState: node 2, state (0/0) unique (0/0)
prevConuni(0) birth (0/0) (old/new)
[    CSSD]2007-11-26 01:45:27.410 [3001019296] >TRACE:
clssscInitSAGEFencing: kgzf fence initialization successfully
started
[    CSSD]2007-11-26 01:45:27.411 [3001019296] >USER:
clssnmHandleUpdate: SYNC(91161493) from node(1) completed
[    CSSD]2007-11-26 01:45:27.411 [3001019296] >USER:
clssnmHandleUpdate: NODE 1 (vx0308) IS ACTIVE MEMBER OF
CLUSTER
[    CSSD]2007-11-26 01:45:27.411 [3001019296] >TRACE:
clssnmHandleUpdate: diskTimeout set to (200000)ms
[    CSSD]2007-11-26 01:45:27.411 [2924551072] >TRACE:
clssnmWaitForAcks: done, msg type(15)
[    CSSD]2007-11-26 01:45:27.411 [2924551072] >TRACE:
clssnmDoSyncUpdate: Sync 0 complete!
[    CSSD]2007-11-26 01:45:27.417 [3055966592] >USER:
NMEVENT_SUSPEND [00][00][00][00]
[    CSSD]2007-11-26 01:45:27.429 [2899372960] >TRACE:
clssgmReconfigThread:  started for reconfig (91161493)
[    CSSD]2007-11-26 01:45:27.429 [2899372960] >TRACE:   KGZF:
context successfully initialized, API version 1.1

[    CSSD]2007-11-26 01:45:27.429 [2899372960] >TRACE:
clssscSAGEInitFenceCompl: kgzf fence initialization
successfully completed
[    CSSD]2007-11-26 01:45:27.429 [2899372960] >USER:
NMEVENT_RECONFIG [00][00][00][02]
[    CSSD]2007-11-26 01:45:27.429 [2899372960] >TRACE:
clssgmEstablishConnections: 1 nodes in cluster incarn 91161493
[    CSSD]2007-11-26 01:45:27.429 [2949995424] >TRACE:
clssgmPeerListener: connects done (1/1)
```

```
[    CSSD]2007-11-26 01:45:27.429 [2899372960] >TRACE:
clssgmEstablishMasterNode: MASTER for 91161493 is node(1)
birth(91161493)
[    CSSD]2007-11-26 01:45:27.429 [2899372960] >TRACE:
clssgmChangeMasterNode: requeued 0 RPCs
[    CSSD]2007-11-26 01:45:27.429 [2899372960] >TRACE:
clssgmMasterCMSync: Synchronizing group/lock status
[    CSSD]2007-11-26 01:45:27.429 [2899372960] >TRACE:
clssgmMasterSendDBDone: group/lock status synchronization
complete
[    CSSD]CLSS-3000: reconfiguration successful, incarnation
91161493 with 1 nodes

[    CSSD]CLSS-3001: local node number 1, master node number 1

[    CSSD]2007-11-26 01:45:27.430 [2899372960] >TRACE:
clssgmReconfigThread:  completed for reconfig(91161493), with
status(1)
[    CSSD]2007-11-26 01:45:27.512 [2983967648] >TRACE:
clssgmCommonAddMember: clsomon joined
(1/0x1000000/#CSS_CLSSOMON)
[    CSSD]2007-11-26 01:46:21.593 [3001019296] >TRACE:
clssnmConnComplete: MSGSRC 2, type 5, node 2, flags 0x0001,
con 0x8370ac0, probe (nil), nodekillsz 0
[    CSSD]2007-11-26 01:46:21.593 [3001019296] >TRACE:
clssnmConnComplete: node 2, vx0309, con(0x8370ac0),
probcon((nil)), ninfcon((nil)), node unique 1196063173, prev
unique 0, msg unique 1196063173 node state 0
[    CSSD]2007-11-26 01:46:21.593 [3001019296] >TRACE:
clssnmsendConnAck: node 2, node state 0
[    CSSD]2007-11-26 01:46:21.593 [3001019296] >TRACE:
clssnmSendConnAck: connected to node 2 , ninfcon (0x8370ac0),
state (0)
[    CSSD]2007-11-26 01:46:21.593 [3001019296] >TRACE:
clssnmConnComplete: connecting to node 2 (con 0x8370ac0),
ninfcon (0x8370ac0), state (0)
[    CSSD]2007-11-26 01:46:21.593 [3001019296] >TRACE:
clssnmConnComplete: ninf->killinfosz 0
[    CSSD]2007-11-26 01:46:21.941 [2924551072] >TRACE:
clssnmDoSyncUpdate: Initiating sync 91161494
[    CSSD]2007-11-26 01:46:21.941 [2924551072] >TRACE:
clssnmDoSyncUpdate: diskTimeout set to (27000)ms
[    CSSD]2007-11-26 01:46:21.941 [2924551072] >TRACE:
clssnmSetupAckWait: Ack message type (11)
[    CSSD]2007-11-26 01:46:21.941 [2924551072] >TRACE:
clssnmSetupAckWait: node(1) is ALIVE
[    CSSD]2007-11-26 01:46:21.941 [2924551072] >TRACE:
clssnmSetupAckWait: node(2) is ALIVE
[    CSSD]2007-11-26 01:46:21.941 [2924551072] >TRACE:
clssnmSendSync: syncSeqNo(91161494)
[    CSSD]2007-11-26 01:46:21.941 [3001019296] >TRACE:
clssnmHandleSync: diskTimeout set to (27000)ms
```

```
[    CSSD]2007-11-26 01:46:21.942 [2924551072] >TRACE:
clssnmWaitForAcks: Ack message type(11), ackCount(2)
[    CSSD]2007-11-26 01:46:21.942 [3001019296] >TRACE:
clssnmHandleSync: Acknowledging sync: src[1] srcName[vx0308]
seq[5] sync[91161494]
[    CSSD]2007-11-26 01:46:21.942 [3055966592] >USER:
NMEVENT_SUSPEND [00][00][00][02]
[    CSSD]2007-11-26 01:46:21.942 [2924551072] >TRACE:
clssnmWaitForAcks: done, msg type(11)
[    CSSD]2007-11-26 01:46:21.942 [2924551072] >TRACE:
clssnmNeedConfReq: No configuration to change
[    CSSD]2007-11-26 01:46:21.942 [2924551072] >TRACE:
clssnmDoSyncUpdate: node(2) is transitioning from joining
state to active state
[    CSSD]2007-11-26 01:46:21.942 [2924551072] >TRACE:
clssnmSetupAckWait: Ack message type (13)
[    CSSD]2007-11-26 01:46:21.942 [2924551072] >TRACE:
clssnmSetupAckWait: node(1) is ACTIVE
[    CSSD]2007-11-26 01:46:21.942 [2924551072] >TRACE:
clssnmSetupAckWait: node(2) is ACTIVE
[    CSSD]2007-11-26 01:46:21.942 [2924551072] >TRACE:
clssnmSendVote: syncSeqNo(91161494)
[    CSSD]2007-11-26 01:46:21.942 [2924551072] >TRACE:
clssnmWaitForAcks: Ack message type(13), ackCount(2)
[    CSSD]2007-11-26 01:46:21.942 [3001019296] >TRACE:
clssnmSendVoteInfo: node(1) syncSeqNo(91161494)
[    CSSD]2007-11-26 01:46:21.943 [2924551072] >TRACE:
clssnmWaitForAcks: done, msg type(13)
[    CSSD]2007-11-26 01:46:21.943 [2924551072] >TRACE:
clssnmCheckDskInfo: Checking disk info...
[    CSSD]2007-11-26 01:46:21.943 [2924551072] >TRACE:
clssnmEvict: Start
[    CSSD]2007-11-26 01:46:21.943 [2924551072] >TRACE:
clssnmWaitOnEvictions: Start
[    CSSD]2007-11-26 01:46:21.943 [2924551072] >TRACE:
clssnmSetupAckWait: Ack message type (15)
[    CSSD]2007-11-26 01:46:21.943 [2924551072] >TRACE:
clssnmSetupAckWait: node(1) is ACTIVE
[    CSSD]2007-11-26 01:46:21.943 [2924551072] >TRACE:
clssnmSetupAckWait: node(2) is ACTIVE
[    CSSD]2007-11-26 01:46:21.943 [2924551072] >TRACE:
clssnmSendUpdate: syncSeqNo(91161494)
[    CSSD]2007-11-26 01:46:21.943 [2924551072] >TRACE:
clssnmWaitForAcks: Ack message type(15), ackCount(2)
[    CSSD]2007-11-26 01:46:21.943 [3001019296] >TRACE:
clssnmUpdateNodeState: node 0, state (0/0) unique (0/0)
prevConuni(0) birth (0/0) (old/new)
[    CSSD]2007-11-26 01:46:21.943 [3001019296] >TRACE:
clssnmUpdateNodeState: node 1, state (3/3) unique
(1196063119/1196063119) prevConuni(0) birth
(91161493/91161493) (old/new)
```

```
[    CSSD]2007-11-26 01:46:21.943 [3001019296] >TRACE:
clssnmUpdateNodeState: node 2, state (2/3) unique
(1196063173/1196063173) prevConuni(0) birth
(91161494/91161494) (old/new)
[    CSSD]2007-11-26 01:46:21.943 [3001019296] >USER:
clssnmHandleUpdate: SYNC(91161494) from node(1) completed
[    CSSD]2007-11-26 01:46:21.943 [3001019296] >USER:
clssnmHandleUpdate: NODE 1 (vx0308) IS ACTIVE MEMBER OF
CLUSTER
[    CSSD]2007-11-26 01:46:21.943 [3001019296] >USER:
clssnmHandleUpdate: NODE 2 (vx0309) IS ACTIVE MEMBER OF
CLUSTER
[    CSSD]2007-11-26 01:46:21.943 [3001019296] >TRACE:
clssnmHandleUpdate: diskTimeout set to (200000)ms
[    CSSD]2007-11-26 01:46:21.943 [2924551072] >TRACE:
clssnmWaitForAcks: done, msg type(15)
[    CSSD]2007-11-26 01:46:21.943 [2924551072] >TRACE:
clssnmDoSyncUpdate: Sync 91161494 complete!
[    CSSD]2007-11-26 01:46:21.945 [2899372960] >TRACE:
clssgmReconfigThread:  started for reconfig (91161494)
[    CSSD]2007-11-26 01:46:21.945 [2899372960] >USER:
NMEVENT_RECONFIG [00][00][00][06]
[    CSSD]2007-11-26 01:46:21.946 [2899372960] >TRACE:
clssgmEstablishConnections: 2 nodes in cluster incarn 91161494
[    CSSD]2007-11-26 01:46:22.023 [2949995424] >TRACE:
clssgmInitialRecv: (0x83831b0) accepted a new connection from
node 2 born at 91161494 active (2, 2), vers (11,1,1,2)
[    CSSD]2007-11-26 01:46:22.023 [2949995424] >TRACE:
clssgmInitialRecv: conns done (2/2)
[    CSSD]2007-11-26 01:46:22.023 [2899372960] >TRACE:
clssgmEstablishMasterNode: MASTER for 91161494 is node(1)
birth(91161493)
[    CSSD]2007-11-26 01:46:22.024 [2899372960] >TRACE:
clssgmMasterCMSync: Synchronizing group/lock status
[    CSSD]2007-11-26 01:46:22.028 [2899372960] >TRACE:
clssgmMasterSendDBDone: group/lock status synchronization
complete
[    CSSD]CLSS-3000: reconfiguration successful, incarnation
91161494 with 2 nodes

[    CSSD]CLSS-3001: local node number 1, master node number 1

[    CSSD]2007-11-26 01:46:22.030 [2899372960] >TRACE:
clssgmReconfigThread:  completed for reconfig(91161494), with
status(1)
[    CSSD]2007-11-26 01:51:14.745 [3019201440] >ERROR:
clssnmvDiskKillCheck: voting disk corrupted
(0x00000000,0x00000000) (0//dev/sdb5)
[    CSSD]2007-11-26 01:51:14.745 [3019201440] >TRACE:
clssnmDiskStateChange: state from 4 to 3 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:51:23.029 [3028274080] >TRACE:
clssnmDiskStateChange: state from 3 to 4 disk (0//dev/sdb5)
```

Practice 10-2: Fixing Oracle Clusterware issues (continued)

```
[    CSSD]2007-11-26 01:51:24.225 [3019201440] >ERROR:
clssnmvDiskKillCheck: voting disk corrupted
(0x00000000,0x00000000) (0//dev/sdb5)
[    CSSD]2007-11-26 01:51:24.225 [3019201440] >TRACE:
clssnmDiskStateChange: state from 4 to 3 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:51:33.413 [3028274080] >TRACE:
clssnmDiskStateChange: state from 3 to 4 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:51:34.470 [3019201440] >ERROR:
clssnmvDiskKillCheck: voting disk corrupted
(0x00000000,0x00000000) (0//dev/sdb5)
[    CSSD]2007-11-26 01:51:34.470 [3019201440] >TRACE:
clssnmDiskStateChange: state from 4 to 3 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:51:43.894 [3028274080] >TRACE:
clssnmDiskStateChange: state from 3 to 4 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:51:34.470 [3019201440] >ERROR:
clssnmvDiskKillCheck: voting disk corrupted
(0x00000000,0x00000000) (0//dev/sdb5)
[    CSSD]2007-11-26 01:51:34.470 [3019201440] >TRACE:
clssnmDiskStateChange: state from 4 to 3 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:51:43.894 [3028274080] >TRACE:
clssnmDiskStateChange: state from 3 to 4 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:51:45.548 [3019201440] >ERROR:
clssnmvDiskKillCheck: voting disk corrupted
(0x00000000,0x00000000) (0//dev/sdb5)
[    CSSD]2007-11-26 01:51:45.548 [3019201440] >TRACE:
clssnmDiskStateChange: state from 4 to 3 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:51:53.913 [3028274080] >TRACE:
clssnmDiskStateChange: state from 3 to 4 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:51:55.016 [3019201440] >ERROR:
clssnmvDiskKillCheck: voting disk corrupted
(0x00000000,0x00000000) (0//dev/sdb5)
[    CSSD]2007-11-26 01:51:55.016 [3019201440] >TRACE:
clssnmDiskStateChange: state from 4 to 3 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:52:04.424 [3028274080] >TRACE:
clssnmDiskStateChange: state from 3 to 4 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:52:06.098 [3019201440] >ERROR:
clssnmvDiskKillCheck: voting disk corrupted
(0x00000000,0x00000000) (0//dev/sdb5)
[    CSSD]2007-11-26 01:52:06.098 [3019201440] >TRACE:
clssnmDiskStateChange: state from 4 to 3 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:52:14.439 [3028274080] >TRACE:
clssnmDiskStateChange: state from 3 to 4 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:52:15.597 [3019201440] >ERROR:
clssnmvDiskKillCheck: voting disk corrupted
(0x00000000,0x00000000) (0//dev/sdb5)
[    CSSD]2007-11-26 01:52:15.597 [3019201440] >TRACE:
clssnmDiskStateChange: state from 4 to 3 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:52:24.802 [3028274080] >TRACE:
clssnmDiskStateChange: state from 3 to 4 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:52:25.836 [3019201440] >ERROR:
clssnmvDiskKillCheck: voting disk corrupted
(0x00000000,0x00000000) (0//dev/sdb5)
```

```
[    CSSD]2007-11-26 01:52:25.836 [3019201440] >TRACE:
clssnmDiskStateChange: state from 4 to 3 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:52:35.053 [3028274080] >TRACE:
clssnmDiskStateChange: state from 3 to 4 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:53:39.731 [3019201440] >ERROR:
clssnmvDiskKillCheck: voting disk corrupted
(0x00000000,0x00000000) (0//dev/sdb5)
[    CSSD]2007-11-26 01:53:39.731 [3019201440] >TRACE:
clssnmDiskStateChange: state from 4 to 3 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:53:47.775 [3028274080] >TRACE:
clssnmDiskStateChange: state from 3 to 4 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:53:48.984 [3028274080] >ERROR:
Internal Error Information:
  Category: 1234
  Location: fwrite_faile
  Other: fwrite unable to write buffer
  Dep: 5

[    CSSD]2007-11-26 01:53:48.984 [3028274080] >ERROR:
clssnmvWriteBlocks: write failed 1 at offset 17 of /dev/sdb5
[    CSSD]2007-11-26 01:53:48.984 [3028274080] >TRACE:
clssnmDiskStateChange: state from 4 to 3 disk (0//dev/sdb5)
[    CSSD]2007-11-26 01:53:48.984 [3019201440] >ERROR:
clssnmvDiskKillCheck: voting disk corrupted
(0x00000000,0x00000000) (0//dev/sdb5)
[  clsdmt]Listening to
(ADDRESS=(PROTOCOL=ipc)(KEY=vx0308DBG_CSSD))
[    CSSD]2007-11-26 02:02:06.830 >USER:    Oracle Database
10g CSS Release 11.1.0.6.0 Production Copyright 1996, 2004
Oracle.  All rights reserved.
[    CSSD]2007-11-26 02:02:06.830 >USER:    CSS daemon log for
node vx0308, number 1, in cluster vx_cluster03
[    CSSD]2007-11-26 02:02:06.855 [3056085376] >TRACE:
clssscmain: local-only set to false
[    CSSD]2007-11-26 02:02:06.918 [3056085376] >TRACE:
clssnmReadNodeInfo: added node 1 (vx0308) to cluster
[    CSSD]2007-11-26 02:02:06.964 [3056085376] >TRACE:
clssnmReadNodeInfo: added node 2 (vx0309) to cluster
[    CSSD]2007-11-26 02:02:06.968 [3056085376] >WARNING:
clssnmReadWallet: Open Wallet returned 28759
[    CSSD]2007-11-26 02:02:06.968 [3056085376] >WARNING:
clssnmInitNMInfo: Node not configured for node kill
[    CSSD]2007-11-26 02:02:06.997 [3028392864] >TRACE:
clssnm_skgxninit: Compatible vendor clusterware not in use
[    CSSD]2007-11-26 02:02:06.997 [3028392864] >TRACE:
clssnm_skgxnmon: skgxn init failed
[    CSSD]2007-11-26 02:02:07.016 [3056085376] >TRACE:
clssnmNMInitialize: Network heartbeat thresholds are:
impending reconfig 15000 ms, reconfig start (misscount) 30000
ms
```

Practice 10-2: Fixing Oracle Clusterware issues (continued)

```
[    CSSD]2007-11-26 02:02:07.020 [3056085376] >TRACE:
clssnmNMInitialize: Voting file I/O timeouts are: short 27000
ms, long 200000 ms
[    CSSD]2007-11-26 02:02:07.039 [3056085376] >TRACE:
clssnmDiskStateChange: state from 1 to 2 disk (0//dev/sdb5)
[    CSSD]2007-11-26 02:02:07.039 [3028392864] >TRACE:
clssnmvDPT: spawned for disk 0 (/dev/sdb5)
[    CSSD]2007-11-26 02:02:07.150 [3028392864] >TRACE:
clssnmDiskStateChange: state from 2 to 4 disk (0//dev/sdb5)
[    CSSD]2007-11-26 02:02:07.168 [3019320224] >TRACE:
clssnmvKillBlockThread: spawned for disk 0 (/dev/sdb5) initial
sleep interval (1000)ms
[    CSSD]2007-11-26 02:02:07.173 [3056085376] >TRACE:
clssnmFatalInit: fatal mode enabled
[    CSSD]2007-11-26 02:02:07.212 [3019320224] >ERROR:
clssnmvDiskKillCheck: voting disk corrupted
(0x00000000,0x00000000) (0//dev/sdb5)
[    CSSD]2007-11-26 02:02:07.212 [3019320224] >TRACE:
clssnmDiskStateChange: state from 4 to 3 disk (0//dev/sdb5)
[    CSSD]2007-11-26 02:02:07.212 [3028392864] >TRACE:
clssnmReadDskHeartbeat: node 2, vx0309, has a disk HB, but no
network HB, DHB has rcfg 91161496, wrtcnt, 339634, LATS
184000, lastSeqNo 339634, timestamp 1196064115/395916300
[    CSSD]2007-11-26 02:02:07.227 [3002399648] >TRACE:
clssnmClusterListener: Listening on
(ADDRESS=(PROTOCOL=tcp)(HOST=vx0308-priv)(PORT=49895))

[    CSSD]2007-11-26 02:02:07.239 [2992634784] >TRACE:
clssgmDeathChkThread: Spawned
[    CSSD]2007-11-26 02:02:07.254 [3002399648] >TRACE:
clssnmClusterListener: Probing node vx0309 (2),
probcon(0xb265f128)
[    CSSD]2007-11-26 02:02:07.255 [3002399648] >TRACE:
clssnmConnComplete: MSGSRC 2, type 6, node 2, flags 0x0001,
con 0xb265f128, probe 0xb265f128, nodekillsz 0
[    CSSD]2007-11-26 02:02:07.255 [3002399648] >TRACE:
clssnmConnComplete: node 2, vx0309, con(0xb265f128),
probcon(0xb265f128), ninfcon((nil)), node unique 1196063173,
prev unique 0, msg unique 1196063173 node state 0
[    CSSD]2007-11-26 02:02:07.255 [3002399648] >TRACE:
clssnmConnComplete: connected to node 2 (con 0xb265f128),
ninfcon (0xb265f128), state (0), flag (1037)
[    CSSD]2007-11-26 02:02:07.255 [3002399648] >TRACE:
clssnmConnComplete: ninf->killinfosz 0
[    CSSD]2007-11-26 02:02:07.293 [2984242080] >TRACE:
clssgmclientlsnr: listening on
(ADDRESS=(PROTOCOL=ipc)(KEY=Oracle_CSS_LclLstnr_vx_cluster03_1
))
[    CSSD]2007-11-26 02:02:07.293 [2984242080] >TRACE:
clssgmclientlsnr: listening on
(ADDRESS=(PROTOCOL=ipc)(KEY=OCSSD_LL_vx0308_vx_cluster03))
```

```
[    CSSD]2007-11-26 02:02:07.293 [2984242080] >TRACE:
clssgmclientlsnr: listening on
(ADDRESS=(PROTOCOL=ipc)(KEY=OCSSD_LL_vx0308_))
[    CSSD]2007-11-26 02:02:07.320 [2950134688] >TRACE:
clssgmPeerListener: Listening on
(ADDRESS=(PROTOCOL=tcp)(DEV=22)(HOST=10.196.30.29)(PORT=46494)
)
[    CSSD]2007-11-26 02:02:08.324 [3002399648] >TRACE:
clssnmHandleSync: diskTimeout set to (27000)ms
[    CSSD]2007-11-26 02:02:08.324 [3002399648] >TRACE:
clssnmHandleSync: Acknowledging sync: src[2] srcName[vx0309]
seq[5] sync[91161496]
[    CSSD]2007-11-26 02:02:08.324 [2924690336] >TRACE:
clssnmRcfgMgrThread: initial lastleader(2) unique(1196063173)
[    CSSD]2007-11-26 02:02:08.324 [3002399648] >TRACE:
clssnmSendVoteInfo: node(2) syncSeqNo(91161496)
[    CSSD]2007-11-26 02:02:08.325 [3002399648] >TRACE:
clssnmHandleUpdate: setting initial cluster incarnation to
91161493
[    CSSD]2007-11-26 02:02:08.325 [3002399648] >TRACE:
clssnmUpdateNodeState: node 0, state (0/0) unique (0/0)
prevConuni(0) birth (0/0) (old/new)
[    CSSD]2007-11-26 02:02:08.325 [3002399648] >TRACE:
clssnmUpdateNodeState: node 1, state (1/3) unique
(1196064126/1196064126) prevConuni(0) birth (0/91161496)
(old/new)
[    CSSD]2007-11-26 02:02:08.325 [3002399648] >TRACE:
clssnmUpdateNodeState: node 2, state (4/3) unique
(1196063173/1196063173) prevConuni(0) birth (0/91161494)
(old/new)
[    CSSD]2007-11-26 02:02:08.347 [3002399648] >TRACE:
clssscInitSAGEFencing: kgzf fence initialization successfully
started
[    CSSD]2007-11-26 02:02:08.347 [3002399648] >USER:
clssnmHandleUpdate: SYNC(91161496) from node(2) completed
[    CSSD]2007-11-26 02:02:08.347 [3002399648] >USER:
clssnmHandleUpdate: NODE 1 (vx0308) IS ACTIVE MEMBER OF
CLUSTER
[    CSSD]2007-11-26 02:02:08.347 [3002399648] >USER:
clssnmHandleUpdate: NODE 2 (vx0309) IS ACTIVE MEMBER OF
CLUSTER
[    CSSD]2007-11-26 02:02:08.347 [3002399648] >TRACE:
clssnmHandleUpdate: diskTimeout set to (200000)ms
[    CSSD]2007-11-26 02:02:08.428 [3056085376] >USER:
NMEVENT_SUSPEND [00][00][00][00]
[    CSSD]2007-11-26 02:02:08.438 [2899512224] >TRACE:
clssgmReconfigThread:  started for reconfig (91161496)
[    CSSD]2007-11-26 02:02:08.438 [2899512224] >TRACE:   KGZF:
context successfully initialized, API version 1.1
```

```
[    CSSD]2007-11-26 02:02:08.438 [2899512224] >TRACE:
clssscSAGEInitFenceCompl: kgzf fence initialization
successfully completed
[    CSSD]2007-11-26 02:02:08.438 [2899512224] >USER:
NMEVENT_RECONFIG [00][00][00][06]
[    CSSD]2007-11-26 02:02:08.438 [2899512224] >TRACE:
clssgmEstablishConnections: 2 nodes in cluster incarn 91161496
[    CSSD]2007-11-26 02:02:08.439 [2950134688] >TRACE:
clssgmInitialRecv: (0x82f7430) accepted a new connection from
node 2 born at 91161494 active (2, 2), vers (11,1,1,2)
[    CSSD]2007-11-26 02:02:08.439 [2950134688] >TRACE:
clssgmInitialRecv: conns done (2/2)
[    CSSD]2007-11-26 02:02:08.440 [2899512224] >TRACE:
clssgmEstablishMasterNode: MASTER for 91161496 is node(2)
birth(91161494)
[    CSSD]2007-11-26 02:02:08.440 [2899512224] >TRACE:
clssgmChangeMasterNode: requeued 0 RPCs
[    CSSD]2007-11-26 02:02:08.446 [2950134688] >TRACE:
clssgmHandleDBDone(): src/dest (2/65535) size(68) incarn
91161496
[    CSSD]CLSS-3000: reconfiguration successful, incarnation
91161496 with 2 nodes

[    CSSD]CLSS-3001: local node number 1, master node number 2

[    CSSD]2007-11-26 02:02:08.447 [2899512224] >TRACE:
clssgmReconfigThread:  completed for reconfig(91161496), with
status(1)
[    CSSD]2007-11-26 02:02:08.557 [2950134688] >TRACE:
clssgmCommonAddMember: clsomon joined
(1/0x1000000/#CSS_CLSSOMON)
[    CSSD]2007-11-26 02:02:16.220 [3028392864] >TRACE:
clssnmDiskStateChange: state from 3 to 4 disk (0//dev/sdb5)
[    CSSD]2007-11-26 02:02:16.222 [3028392864] >ERROR:
clssnmvReadFatal: voting device corrupt
(0x00000000/0x00000000/0//dev/sdb5)


…
```

7) Fix the diagnosed problem.

   a) Although Oracle Clusterware is able to function in this situation, it is necessary to recover the voting disk file, /dev/sbd5. A backup was made at the beginning of this practice by the lab10-2-prep.sh script. It is called vdisk.bak and located in your solutions directory. After you stop Oracle Clusterware on both nodes, use the dd command and specify a 4K block size. Execute the command as shown below.

```
[oracle@vx0308 less10]$ sudo /u01/crs11g/bin/crsctl stop crs
Stopping resources.
This could take several minutes.
Successfully stopped Oracle Clusterware resources
```

```
Stopping Cluster Synchronization Services.
Shutting down the Cluster Synchronization Services daemon.
Shutdown request successfully issued.
[oracle@vx0308 less10]$


[oracle@vx0309 ~]$ sudo /u01/crs11g/bin/crsctl stop crs
Stopping resources.
This could take several minutes.
Successfully stopped Oracle Clusterware resources
Stopping Cluster Synchronization Services.
Shutting down the Cluster Synchronization Services daemon.
Shutdown request successfully issued.
[oracle@vx0309 ~]$


[oracle@vx0308 less10]$ cat sol_10_02_07.sh
# sol_10_02_07.sh

dd if=/home/oracle/solutions/less10/vdisk.bak of=/dev/sdb5
bs=4k
[oracle@vx0308 less10]$


[oracle@vx0308 less10]$ ./sol_10_02_07.sh
154224+1 records in
154224+1 records out
[oracle@vx0308 less10]$
```

8) After you fixed the problem, what should you do?

   a) Re-start Oracle Clusterware on both nodes using the crsctl command as root on both nodes as shown below.

```
[oracle@vx0306 less10]$ cat sol_10_02_08.sh
# sol_10_02_08.sh

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

HOST=`hostname|cut -c 1-10`

sudo /u01/crs11g/bin/crsctl start crs

ssh $z sudo /u01/crs11g/bin/crsctl start crs
[oracle@vx0306 less10]$


[oracle@vx0306 less10]$ ./sol_10_02_08.sh
Attempting to start Oracle Clusterware stack
The CRS stack will be started shortly
Attempting to start Oracle Clusterware stack
The CRS stack will be started shortly
```

```
[oracle@vx0306 less10]$
```

9) Using the crs_stat command, check the status of your CRS stack and nodeapps. Be patient, it takes a few minutes for the components to restart.

```
[oracle@vx0306 less10]$ cat sol_10_02_09.sh
# sol_10_02_09.sh

/u01/crs11g/bin/crs_stat
[oracle@vx0306 less10]$


[oracle@vx0308 less10]$ ./sol_10_02_09.sh
NAME=ora.RDB.RDB1.inst
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0308

NAME=ora.RDB.RDB2.inst
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0309

NAME=ora.RDB.db
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0309

NAME=ora.vx0308.ASM1.asm
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0308

NAME=ora.vx0308.LISTENER_VX0308.lsnr
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0308

NAME=ora.vx0308.gsd
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0308

NAME=ora.vx0308.ons
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0308

NAME=ora.vx0308.vip
TYPE=application
TARGET=ONLINE
```

```
STATE=ONLINE on vx0308

NAME=ora.vx0309.ASM2.asm
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0309

NAME=ora.vx0309.LISTENER_VX0309.lsnr
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0309

NAME=ora.vx0309.gsd
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0309

NAME=ora.vx0309.ons
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0309

NAME=ora.vx0309.vip
TYPE=application
TARGET=ONLINE
STATE=ONLINE on vx0309

[oracle@vx0308 less10]$
```

10) The database instances will be the last things that are started and may take several minutes to do so. What could be the cause of that delay? Because it may take too long to restart both instances, you can manually start them if needed.

   a) Take a look at the database alert log on your first node and see what is happening. The delay, of course is caused by instance recovery. Remember, that the problem we introduced at the beginning of the practice caused the Oracle Clusterware stack to crash and reboot the node and this of course crashed the database also.

```
[oracle@vx0308 less10]$ cat sol_10_02_10.sh
#!/bin/bash
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"
```

Practice 10-2: Fixing Oracle Clusterware issues (continued)

```
I2NAME=$DBNAME"2"

cat /u01/app/diag/rdbms/rdb*/RDB*/trace/alert*
sleep 100
/u01/crs11g/bin/srvctl start instance -d $DBNAME -i $I2NAME
/u01/crs11g/bin/srvctl start instance -d $DBNAME -i $I1NAME
[oracle@vx0308 less10]$

[oracle@vx0308 less10]$ ./sol_10_02_10.sh
```

Practices for Lesson 11

IMPORTANT NOTE: Before you start the following labs, make sure VNC is started on both nodes as user oracle on port 5802.
You execute your commands under VNC terminal sessions alternatively on the first and second node.
Before you start the following labs, execute the sol_11_00_00.sh script located in your /home/oracle/solutions/less11 directory. You should execute this script under the VNC terminal session started on the first node as user `oracle`.

```
[oracle@vx0306 less11]$ ./sol_11_00_00.sh
vx0306 being added to access control list
vx0313 being added to access control list
displayenv.sh                                          100%   80
0.1KB/s   00:00
vx0306 being added to access control list
vx0313 being added to access control list
[oracle@vx0306 less11]$
```

The goal of this practice is to remove the second node of your cluster, and add it back again using DBCA and OUI.

Practice 11-1: Remove the Second Instance

The goal of this practice is to remove the Database instance that exists on the second node of your cluster. You use dbca for that task.

1) Connected as user oracle on the first node, use DBCA to remove the second instance of your cluster.

```
[oracle@vx0306 less11]$ cat sol_11_01_01.sh
#!/bin/bash
#
# sol_11_01_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

cd $ORACLE_HOME/bin

./dbca
[oracle@vx0306 less11]$
[oracle@vx0306 less11]$ ./sol_11_01_01.sh
```

a) On the Welcome screen select Oracle Real Application Clusters Database and click Next.

b) On the Operations screen, select Instance Management, and click Next.

c) On the Instance Management screen, select Delete an Instance, and click Next.

d) On the List of cluster databases screen, select your Database Name. Then enter sys in the Username field and oracle1 in the password field. Click Next.

e) On the list of cluster database instances screen, select the instance from second node. It should be seen as an active instance. Click Next.

f) When you are finished, on the same screen, click Finish.

g) In the Database Configuration Assistant dialog box that appears, confirm that the operation is to delete the second instance on the second node and click OK.

h) In the second Database Configuration Assistant dialog box that appears, click OK to confirm that you want to proceed.

i) Wait until the progress bar goes to 100%. On the Database Configuration Assistant dialog box that appears, click No to confirm that you do not want to perform any other operation.

Practice 11-2: Clean Up ASM

The goal of this practice is to remove the ASM instance from the second node of your cluster. You use srvctl for that task.

1) Connected as user oracle from the first node, stop and remove the ASM instance running on your second node by using srvctl. When you are finished, check that the corresponding resource has been removed from your Oracle Clusterware configuration.

```
[oracle@vx0306 less11]$ cat sol_11_02_01.sh
#!/bin/bash
#
# sol_11_02_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/srvctl stop asm -n $z

/u01/crs11g/bin/srvctl remove asm -n $z

/u01/crs11g/bin/crs_stat -t
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_02_01.sh
Name           Type           Target    State     Host
------------------------------------------------------------
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora.RDB.db     application    ONLINE    ONLINE    vx0306
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
ora....13.lsnr application    ONLINE    ONLINE    vx0313
ora.vx0313.gsd application    ONLINE    ONLINE    vx0313
ora.vx0313.ons application    ONLINE    ONLINE    vx0313
ora.vx0313.vip application    ONLINE    ONLINE    vx0313
[oracle@vx0306 less11]$
```

Practice 11-2: Clean Up ASM (continued)

2) Remove the initialization parameter file of that ASM instance on the second node.

```
[oracle@vx0306 less11]$ cat sol_11_02_02.sh
#!/bin/bash
#
# sol_11_02_02.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

ssh $z rm -f /u01/app/oracle/product/11.1.0/asm_1/dbs/*ASM*
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_02_02.sh
[oracle@vx0306 less11]$
```

3) Remove the ASM directories from the second node.

```
[oracle@vx0306 less11]$ cat sol_11_02_03.sh
#!/bin/bash
#
# sol_11_02_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

ssh $z rm -rf /u01/app/oracle/product/11.1.0/asm_1/admin/+ASM
```

```
ssh $z rm -rf /u01/app/oracle/admin/+ASM
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_02_03.sh
[oracle@vx0306 less11]$
```

4) Last thing you can do is to remove the associated ASM entry from the /etc/oratab file on the second node.

```
[oracle@vx0306 less11]$ cat sol_11_02_04.sh
#!/bin/bash
#
# sol_11_02_04.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#


y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

ssh $z sudo "cat /etc/oratab | grep -v +ASM2 >
/home/oracle/solutions/less11/neworatab"

ssh $z sudo cp /etc/oratab /etc/jfvbakoratab

ssh $z sudo rm /etc/oratab

ssh $z sudo cp /home/oracle/solutions/less11/neworatab
/etc/oratab

ssh $z sudo chmod 664 /etc/oratab

ssh $z sudo chown oracle /etc/oratab

[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_02_04.sh
[oracle@vx0306 less11]$
```

Practice 11-3: Remove the Listener

The goal of this practice is to use netca to remove the listener from the second node of your cluster.

1) You can now remove the listener from the node you want to delete. This listener can be from either the ASM home or the Database home depending when it was created. However, we suppose that you created it from your ASM instance in this practice. To remove the listener, you can use NETCA.

```
[oracle@vx0306 less11]$ cat sol_11_03_01.sh
#!/bin/bash
#
# sol_11_03_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

scp /home/oracle/solutions/less11/runnetca.sh
$z:/home/oracle/solutions/less11/runnetca.sh

ssh $z chmod 777 /home/oracle/solutions/less11/runnetca.sh

ssh $z /home/oracle/solutions/less11/runnetca.sh
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ cat runnetca.sh
#!/bin/bash
#
# runnetca.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`


export ORACLE_HOME=/u01/app/oracle/product/11.1.0/asm_1

cd $ORACLE_HOME/bin
```

Practice 11-3: Remove the Listener (continued)

```
export DISPLAY=$y:2.0

./netca
[oracle@vx0306 less11]$



[oracle@vx0306 less11]$ ./sol_11_03_01.sh
…
```

a) On the Configuration screen, select Cluster configuration and click Next.

b) On the Active Nodes screen select only the second node from which you want to remove the listener and click Next.

c) On the Welcome screen, select Listener configuration and click Next.

d) On the Listener screen, select Delete and click Next.

e) On the Select listener screen, select the corresponding listener, normally called LISTENER, and click Next.

f) On the Oracle Net Configuration Assistant dialog box that appears, confirm that you want to remove LISTENER by clicking Yes.

g) On the Listener Deleted screen, you should see that LISTENER was successfully deleted, and click Next.

h) On the Listener Configuration Done screen, you should see that listener configuration is now complete, and click Next.

i) On the Welcome screen, click Finish.

2) Make sure the listener and the instance on the second node are no longer part of the Oracle Clusterware configuration.

```
[oracle@vx0306 less11]$ cat sol_11_03_02.sh
#!/bin/bash
#
# sol_11_03_02.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"
```

Practice 11-3: Remove the Listener (continued)

```
/u01/crs11g/bin/crs_stat -t
[oracle@vx0306 less11]$




[oracle@vx0306 less11]$ ./sol_11_03_02.sh
Name           Type           Target    State     Host
------------------------------------------------------------
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora.RDB.db     application    ONLINE    ONLINE    vx0306
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
ora.vx0313.gsd application    ONLINE    ONLINE    vx0313
ora.vx0313.ons application    ONLINE    ONLINE    vx0313
ora.vx0313.vip application    ONLINE    ONLINE    vx0313
[oracle@vx0306 less11]$
```

Practice 11-4: Remove the Database Software from the Second Node

The goal of this practice is to remove the Database software installation from the second node of your cluster only.

1) On the second node, and connected as user oracle in a VNC session, make sure you have your ORACLE_HOME set to /u01/app/oracle/product/11.1.0/db_1.

2) On the second node, and connected as user oracle, change your current directory to $ORACLE_HOME/oui/bin and execute the following command:
```
./runInstaller -updateNodeList ORACLE_HOME=$ORACLE_HOME
"CLUSTER_NODES=<second node name>" -local
```

```
[oracle@vx0306 less11]$ cat sol_11_04_02.sh
#!/bin/bash
#
# sol_11_04_02.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

scp /home/oracle/solutions/less11/removedb.sh
$z:/home/oracle/solutions/less11/removedb.sh

ssh $z chmod 777 /home/oracle/solutions/less11/removedb.sh

ssh $z "/home/oracle/solutions/less11/removedb.sh $z"
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ cat removedb.sh
#!/bin/bash
#
# removedb.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`
```

Practice 11-4: Remove the Database Software from the Second Node (continued)

```
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

export DISPLAY=$y:2.0

$ORACLE_HOME/oui/bin/runInstaller -updateNodeList
ORACLE_HOME=$ORACLE_HOME "CLUSTER_NODES=$1" -local
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_04_02.sh
removedb.sh                                100%  345 0.3KB/s
00/00
The inventory pointer is located at /etc/oraInst.loc
The inventory is located at /u01/app/oraInventory
'UpdateNodeList' was successful.
Starting Oracle Universal Installer…

Checking swap space: … Passed
Checking Monitor: …    Passed
[oracle@vx0306 less11]$
```

3) On the second node, use OUI from the database home to remove the database software.

```
[oracle@vx0306 less11]$ cat sol_11_04_03.sh
#!/bin/bash
#
# sol_11_04_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

scp /home/oracle/solutions/less11/removedb2.sh
$z:/home/oracle/solutions/less11/removedb2.sh

ssh $z chmod 777 /home/oracle/solutions/less11/removedb2.sh

ssh $z /home/oracle/solutions/less11/removedb2.sh
[oracle@vx0306 less11]$
```

```
[oracle@vx0306 less11]$ cat removedb2.sh
#!/bin/bash
#
# removedb2.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

export DISPLAY=$y:2.0

$ORACLE_HOME/oui/bin/runInstaller
[oracle@vx0306 less11]$



[oracle@vx0306 less11]$ ./sol_11_04_03.sh
…
```

a) On the Welcome screen, click Deinstall Products.

b) On the Inventory window, select your database home name (should be OraDb11g_home1), and click Remove.

c) On the Confirmation window, where you can see the list of products to be removed, click Yes.

d) On the Warning window that appears, click Yes to remove the ORACLE_HOME at the end.

e) Wait for the Deinstall progress bar to complete (100%).

f) On the Inventory window, you should now see that your database home has been removed. On the Inventory window, click Close.

g) On the Welcome screen, click Cancel.

h) On the Exit dialog box, click Yes to exit OUI.

4) On the first node, make sure you export your ORACLE_HOME environment variable set to /u01/app/oracle/product/11.1.0/db_1, and change your current directory to $ORACLE_HOME/oui/bin. Then execute the following command:
`./runInstaller -updateNodeList ORACLE_HOME=$ORACLE_HOME "CLUSTER_NODES=<first node name>"`

```
[oracle@vx0306 less11]$ cat sol_11_04_04.sh
#!/bin/bash
#
# sol_11_04_04.sh
```

Practice 11-4: Remove the Database Software from the Second Node (continued)

```
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

export DISPLAY=$y:2.0

$ORACLE_HOME/oui/bin/runInstaller -updateNodeList
ORACLE_HOME=$ORACLE_HOME "CLUSTER_NODES=$y"
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_04_04.sh
Starting Oracle Universal Installer…

Checking swap space: … Passed
Checking Monitor: …    Passed
The inventory pointer is located at /etc/oraInst.loc
The inventory is located at /u01/app/oraInventory
'UpdateNodeList' was successful.
[oracle@vx0306 less11]$
```

Practice 11-5: Remove the ASM Software from the Second Node

The goal of this practice is to remove the ASM software installation from the second node of your cluster only.

1) On the second node, and connected as user oracle in a VNC session, make sure you have your ORACLE_HOME set to /u01/app/oracle/product/11.1.0/asm_1.

2) On the second node, and connected as user oracle, change your current directory to $ORACLE_HOME/oui/bin and execute the following command:
`./runInstaller -updateNodeList ORACLE_HOME=$ORACLE_HOME "CLUSTER_NODES=<second node name>" -local`

```
[oracle@vx0306 less11]$ cat sol_11_05_02.sh
#!/bin/bash
#
# sol_11_05_02.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

scp /home/oracle/solutions/less11/removeasm.sh
$z:/home/oracle/solutions/less11/removeasm.sh

ssh $z chmod 777 /home/oracle/solutions/less11/removeasm.sh

ssh $z "/home/oracle/solutions/less11/removeasm.sh $z"
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ cat removeasm.sh
#!/bin/bash
#
# removeasm.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

export DISPLAY=$y:2.0
```

```
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/asm_1

$ORACLE_HOME/oui/bin/runInstaller -updateNodeList
ORACLE_HOME=$ORACLE_HOME "CLUSTER_NODES=$1" -local
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_05_02.sh
removeasm.sh                                100%  347 0.3KB/s
00/00
The inventory pointer is located at /etc/oraInst.loc
The inventory is located at /u01/app/oraInventory
'UpdateNodeList' was successful.
Starting Oracle Universal Installer…

Checking swap space: … Passed
Checking Monitor: …    Passed
[oracle@vx0306 less11]$
```

3) On the second node, use OUI from the ASM home to remove the ASM software.

```
[oracle@vx0306 less11]$ cat sol_11_05_03.sh
#!/bin/bash
#
# sol_11_05_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

scp /home/oracle/solutions/less11/removeasm2.sh
$z:/home/oracle/solutions/less11/removeasm2.sh

ssh $z chmod 777 /home/oracle/solutions/less11/removeasm2.sh

ssh $z /home/oracle/solutions/less11/removeasm2.sh
[oracle@vx0306 less11]$

[oracle@vx0306 less11]$ cat removeasm2.sh
#!/bin/bash
```

```
#
# removeasm2.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

export DISPLAY=$y:2.0

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/asm_1

$ORACLE_HOME/oui/bin/runInstaller
[oracle@vx0306 less11]$



[oracle@vx0306 less11]$ ./sol_11_05_03.sh
…
```

a) On the Welcome screen, click Deinstall Products.

b) On the Inventory window, select your ASM home name (should be OraASM11g_home1), and click Remove.

c) On the Confirmation window, where you can see the list of products to be removed, click Yes.

d) On the Warning window that appears, click Yes to remove the ORACLE_HOME at the end.

e) Wait for the Deinstall progress bar to complete (100%).

f) In the Inventory window, you should now see that your ASM home has been removed. In the Inventory window, click Close.

g) On the Welcome screen, click Cancel.

h) In the Exit dialog box, click Yes to exit OUI.

4) On the first node, make sure you export your ORACLE_HOME environment variable set to /u01/app/oracle/product/11.1.0/asm_1, and change your current directory to $ORACLE_HOME/oui/bin. Then execute the following command:
`./runInstaller -updateNodeList ORACLE_HOME=$ORACLE_HOME "CLUSTER_NODES=<first node name>"`

```
[oracle@vx0306 less11]$ cat sol_11_05_04.sh
#!/bin/bash
#
# sol_11_05_04.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#
```

Practice 11-5: Remove the ASM Software from the Second Node (continued)

```
y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/asm_1

export DISPLAY=$y:2.0

$ORACLE_HOME/oui/bin/runInstaller -updateNodeList
ORACLE_HOME=$ORACLE_HOME "CLUSTER_NODES=$y"
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_05_04.sh
Starting Oracle Universal Installer…

Checking swap space: … Passed
Checking Monitor: …    Passed
The inventory pointer is located at /etc/oraInst.loc
The inventory is located at /u01/app/oraInventory
'UpdateNodeList' was successful.
[oracle@vx0306 less11]$
```

Practice 11-6: Remove the Second Node from the OCR

The goal of this practice is to remove the second node ONS configuration from the OCR.

1) On the first node, use the racgons command tool to remove the configuration of the ONS of the second node from the OCR.

```
[oracle@vx0306 less11]$ cat sol_11_06_01.sh
#!/bin/bash
#
# sol_11_06_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

cat /u01/crs11g/opmn/conf/ons.config

/u01/crs11g/bin/racgons remove_config $z
[oracle@vx0306 less11]$



[oracle@vx0306 less11]$ ./sol_11_06_01.sh
localport=6150
useocr=on
allowgroup=true
usesharedinstall=true
racgons: host vx0313 is deleted from ONS configuration.
[oracle@vx0306 less11]$
```

Practice 11-7: Remove the Oracle Clusterware Software from the Second Node

The goal of this practice is to remove the Oracle Clusterware software installation from the second node of your cluster.

1) From the second node as user root, execute the following command:
`<Oracle Clusterware home>/install/rootdelete.sh`

```
[oracle@vx0306 less11]$ cat sol_11_07_01.sh
#!/bin/bash
#
# sol_11_07_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

ssh $z sudo /u01/crs11g/install/rootdelete.sh
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_07_01.sh
Getting local node name
NODE = vx0313
Getting local node name
NODE = vx0313
Stopping resources.
This could take several minutes.
Successfully stopped Oracle Clusterware resources
Stopping Cluster Synchronization Services.
Shutting down the Cluster Synchronization Services daemon.
Shutdown request successfully issued.
Waiting for Cluster Synchronization Services daemon to stop
Cluster Synchronization Services daemon has stopped
Oracle CRS stack is not running.
Oracle CRS stack is down now.
Removing script for Oracle Cluster Ready services
Updating ocr file for downgrade
Cleaning up SCR settings in '/etc/oracle/scls_scr'
Cleaning up Network socket directories

[oracle@vx0306 less11]$
```

Practice 11-7: Remove the Oracle Clusterware Software from the Second Node (continued)

2) From the first node as the root user, execute the following command:
`<Oracle Clusterware home>/install/rootdeletenode.sh <node name to be deleted>,<node number to be deleted>`

```
[oracle@vx0306 less11]$ cat sol_11_07_02.sh
#!/bin/bash
#
# sol_11_07_02.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/olsnodes -n

sudo /u01/crs11g/install/rootdeletenode.sh $z,2
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_07_02.sh
vx0306  1
vx0313  2
CRS-0210: Could not find resource 'ora.vx0313.ons'.
CRS-0210: Could not find resource 'ora.vx0313.vip'.
CRS-0210: Could not find resource 'ora.vx0313.gsd'.
PRKO-2112 : Some or all node applications are not removed
successfully on node: vx0313
CRS-0210: Could not find resource 'ora.vx0313.vip'.CRS-0210:
Could not find resource 'ora.vx0313.ons'.CRS-0210: Could not
find resource 'ora.vx0313.gsd'.
CRS nodeapps are deleted successfully
clscfg: EXISTING configuration version 4 detected.
clscfg: version 4 is 11 Release 1.
Value SYSTEM.crs.versions.vx0313 marked for deletion is not
there. Ignoring.
Successfully deleted 15 values from OCR.
Key SYSTEM.css.interfaces.nodevx0313 marked for deletion is
not there. Ignoring.
Key SYSTEM.crs.versions.vx0313 marked for deletion is not
there. Ignoring.
Successfully deleted 13 keys from OCR.
```

Practice 11-7: Remove the Oracle Clusterware Software from the Second Node (continued)

```
Node deletion operation successful.
'vx0313,2' deleted successfully
[oracle@vx0306 less11]$
```

3) On the second node, connected as user root, make sure you export the ORACLE_HOME environment variable set to /u01/crs11g. Then execute the following command:
`/u01/crs11g/oui/bin/runInstaller -updateNodeList ORACLE_HOME=/u01/crs11g "CLUSTER_NODES=<second node name>" CRS=TRUE -local`

```
[oracle@vx0306 less11]$ cat sol_11_07_03.sh
#!/bin/bash
#
# sol_11_07_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

scp /home/oracle/solutions/less11/removecrs.sh
$z:/home/oracle/solutions/less11/removecrs.sh

ssh $z chmod 777 /home/oracle/solutions/less11/removecrs.sh

ssh $z "/home/oracle/solutions/less11/removecrs.sh $z"
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ cat removecrs.sh
#!/bin/bash
#
# removecrs.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

export DISPLAY=$y:2.0
```

Practice 11-7: Remove the Oracle Clusterware Software from the Second Node (continued)

```
export ORACLE_HOME=/u01/crs11g

$ORACLE_HOME/oui/bin/runInstaller -updateNodeList
ORACLE_HOME=$ORACLE_HOME "CLUSTER_NODES=$1" CRS=TRUE -local
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_07_03.sh
removecrs.sh                                     100%  331 0.3KB/s
00/00
The inventory pointer is located at /etc/oraInst.loc
The inventory is located at /u01/app/oraInventory
'UpdateNodeList' was successful.
Starting Oracle Universal Installer…

Checking swap space: … Passed
Checking Monitor: …    Passed
[oracle@vx0306 less11]$
```

4) On the second node, start OUI from the Oracle Clusterware home and deinstall Oracle Clusterware software on the second node.

```
[oracle@vx0306 less11]$ cat sol_11_07_04.sh
#!/bin/bash
#
# sol_11_07_04.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

scp /home/oracle/solutions/less11/removecrs2.sh
$z:/home/oracle/solutions/less11/removecrs2.sh

ssh $z chmod 777 /home/oracle/solutions/less11/removecrs2.sh

ssh $z /home/oracle/solutions/less11/removecrs2.sh
[oracle@vx0306 less11]$
```

Practice 11-7: Remove the Oracle Clusterware Software from the Second Node (continued)

```
[oracle@vx0306 less11]$ cat removecrs2.sh
#!/bin/bash
#
# removecrs2.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

export DISPLAY=$y:2.0

export ORACLE_HOME=/u01/crs11g

$ORACLE_HOME/oui/bin/runInstaller
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_07_04.sh
…
```

a) On the Welcome screen, click Deinstall Products .

b) In the Inventory window, select your Oracle Clusterware software home name (OraCrs11g_home) and click Remove.

c) In the confirmation window, you should see the list of products to deinstall. Click Yes.

d) In the Warning dialog box, make sure you click No to avoid CRS directory to be removed at the end of this installation. You do it manually afterward.

e) Wait for the progress bar to finish.

f) In the Inventory window, after deinstallation was successful, you should no longer see the Oracle Clusterware software home. Click Close.

g) On the Welcome screen, click Cancel to finish this deinstallation.

h) In the Exit window, click Yes to exit from OUI.

5) On the first node as user root, execute the following command:
`/u01/crs11g/oui/bin/runInstaller -updateNodeList ORACLE_HOME=/u01/crs11g "CLUSTER_NODES=<first node name>" CRS=TRUE`

```
[oracle@vx0306 less11]$ cat sol_11_07_05.sh
#!/bin/bash
#
# sol_11_07_05.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#
```

Practice 11-7: Remove the Oracle Clusterware Software from the Second Node (continued)

```
y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/crs11g

export DISPLAY=$y:2.0

$ORACLE_HOME/oui/bin/runInstaller -updateNodeList
ORACLE_HOME=$ORACLE_HOME "CLUSTER_NODES=$y" CRS=TRUE
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_07_05.sh
Starting Oracle Universal Installer…

Checking swap space: … Passed
Checking Monitor: …    Passed
The inventory pointer is located at /etc/oraInst.loc
The inventory is located at /u01/app/oraInventory
'UpdateNodeList' was successful.
[oracle@vx0306 less11]$
```

6) Check your Oracle Clusterware configuration. What do you observe?

   a) You should not see any remaining resources on the second node. Also, make sure your second node is no longer part of your cluster from a CRS perspective.

```
[oracle@vx0306 less11]$ cat sol_11_07_06.sh
#!/bin/bash
#
# sol_11_07_06.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"
```

Practice 11-7: Remove the Oracle Clusterware Software from the Second Node (continued)

```
I2NAME=$DBNAME"2"

/u01/crs11g/bin/crs_stat -t

/u01/crs11g/bin/olsnodes -n
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_07_06.sh
Name           Type           Target    State     Host
------------------------------------------------------------
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora.RDB.db     application    ONLINE    ONLINE    vx0306
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
vx0306  1
[oracle@vx0306 less11]$
```

7) Remove your Oracle Clusterware, ASM, and Database home directories from your second node.

```
[oracle@vx0306 less11]$ cat sol_11_07_07.sh
#!/bin/bash
#
# sol_11_07_07.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

ssh $z "sudo rm -rf /u01/crs11g"

ssh $z "sudo rm -rf /u01/app/oracle/product/11.1.0/asm_1"

ssh $z "sudo rm -rf /u01/app/oracle/product/11.1.0/db_1"
[oracle@vx0306 less11]$
```

```
[oracle@vx0306 less11]$ ./sol_11_07_07.sh
[oracle@vx0306 less11]$
```

Practice 11-8: Check Prerequisites Before Oracle Clusterware Installation

The goal of this practice is to make sure your environment meets the expected prerequisites before you install the Oracle Clusterware software on the second node again.

1) Before you can proceed with the Oracle Clusterware installation on the node you want to add to your RAC cluster, you need to make sure that all operating system and hardware prerequisites are met. Use Cluster Verify to check Oracle Clusterware pre-installation.

```
[oracle@vx0306 less11]$ cat sol_11_08_01.sh
#!/bin/bash
#
# sol_11_08_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/cluvfy  stage -pre crsinst -n $y,$z -r 11gR1
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_08_01.sh

Performing pre-checks for cluster services setup

Checking node reachability...
Node reachability check passed from node "vx0306".


Checking user equivalence...
User equivalence check passed for user "oracle".

Checking administrative privileges...
User existence check passed for "oracle".
Group existence check passed for "oinstall".
Membership check for user "oracle" in group "oinstall" [as
Primary] passed.
```

```
Administrative privileges check passed.

Checking node connectivity...

Node connectivity check passed for subnet "10.216.4.0" with
node(s) vx0306,vx0313.

WARNING:
Make sure IP address "10.216.96.120" is up and is a valid IP
address on node "vx0313".
Node connectivity check failed for subnet "10.216.96.0".
Node connectivity check passed for subnet "10.196.28.0" with
node(s) vx0306,vx0313.

Interfaces found on subnet "10.216.4.0" that are likely
candidates for VIP:
vx0306 eth0:10.216.4.17 eth0:10.216.4.68
vx0313 eth0:10.216.4.24

Interfaces found on subnet "10.196.28.0" that are likely
candidates for a private interconnect:
vx0306 eth2:10.196.30.27
vx0313 eth2:10.196.30.34

Node connectivity check failed.


Checking system requirements for 'crs'...
Total memory check passed.
Free disk space check passed.
Swap space check passed.
System architecture check passed.
Kernel version check passed.
Package existence check passed for "make-3.80".
Package existence check passed for "binutils-2.15.92.0.2".
Package existence check passed for "gcc-3.4.5".
Package existence check passed for "libaio-0.3.105".
Package existence check passed for "libaio-devel-0.3.105".
Package existence check passed for "libstdc++-3.4.5".
Package existence check passed for "elfutils-libelf-devel-
0.97".
Package existence check passed for "sysstat-5.0.5".
Package existence check passed for "libgcc-3.4.5".
Package existence check passed for "libstdc++-devel-3.4.5".
Package existence check passed for "unixODBC-2.2.11".
Package existence check passed for "unixODBC-devel-2.2.11".
Package existence check passed for "glibc-2.3.4-2.19".
Group existence check passed for "dba".
Group existence check passed for "oinstall".
User existence check passed for "nobody".
```

Practice 11-8: Check Prerequisites Before Oracle Clusterware Installation (continued)

```
System requirement passed for 'crs'

Pre-check for cluster services setup was unsuccessful on all
the nodes.
[oracle@vx0306 less11]$
```

Practice 11-9: Add Oracle Clusterware to the Second Node

The goal of this practice is to install the Oracle Clusterware software on the second node of your cluster.

1) Login as the oracle user execute the addNode.sh script located in your Oracle Clusterware home directory on the first node (/u01/crs11g/oui/bin). Make sure your ORACLE_HOME is exported to /u01/crs11g. This script runs the Oracle Universal Installer. Then add the Oracle Clusterware software to the new node.

```
[oracle@vx0306 less11]$ cat sol_11_09_01.sh
#!/bin/bash
#
# sol_11_09_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/crs11g

cd $ORACLE_HOME/oui/bin

export DISPLAY=$y:2.0

./addNode.sh
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_09_01.sh
…
```

a) On the Welcome screen, click Next.

b) On the Specify Cluster Nodes to Add to Installation screen, enter the public, private, and virtual host name in the first row of the Specify New Nodes table (use the same scheme as for the first node in the Existing Nodes table). Click Next.

c) On the Cluster Node Addition Summary screen, click Install.

d) The Cluster Node Addition Progress screen appears. Wait until OUI asks you to run some scripts.

e) When the Execute Configuration scripts window appears, do the following:

Practice 11-9: Add Oracle Clusterware to the Second Node (continued)

f) Execute /u01/crs11g/install/rootaddnode.sh on the first node as root user.

```
[oracle@vx0306 less11]$ cat sol_11_09_01f.sh
#!/bin/bash
#
# sol_11_09_01f.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

sudo /u01/crs11g/install/rootaddnode.sh
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_09_01f.sh
clscfg: EXISTING configuration version 4 detected.
clscfg: version 4 is 11 Release 1.
Attempting to add 1 new nodes to the configuration
Using ports: CSS=49895 CRS=49896 EVMC=49898 and EVMR=49897.
node <nodenumber>: <nodename> <private interconnect name>
<hostname>
node 2: vx0313 vx0313-priv vx0313
Creating OCR keys for user 'root', privgrp 'root'..
Operation successful.
/u01/crs11g/bin/srvctl add nodeapps -n vx0313 -A vx0313-
vip/255.255.252.0/eth0
[oracle@vx0306 less11]$
```

g) Execute /u01/crs11g/root.sh on the second node as root user.

```
[oracle@vx0306 less11]$ cat sol_11_09_01g.sh
#!/bin/bash
#
# sol_11_09_01g.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`
```

Practice 11-9: Add Oracle Clusterware to the Second Node (continued)

```
DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

ssh $z sudo /u01/crs11g/root.sh
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_09_01g.sh
Checking to see if Oracle CRS stack is already configured
OCR LOCATIONS =  /dev/sdb1,/dev/sdb2
OCR backup directory '/u01/crs11g/cdata/vx_cluster02' does not
exist. Creating now
Setting the permissions on OCR backup directory
Setting up Network socket directories
Oracle Cluster Registry configuration upgraded successfully
clscfg: EXISTING configuration version 4 detected.
clscfg: version 4 is 11 Release 1.
Successfully accumulated necessary OCR keys.
Using ports: CSS=49895 CRS=49896 EVMC=49898 and EVMR=49897.
node <nodenumber>: <nodename> <private interconnect name>
<hostname>
node 1: vx0306 vx0306-priv vx0306
node 2: vx0313 vx0313-priv vx0313
clscfg: Arguments check out successfully.

NO KEYS WERE WRITTEN. Supply -force parameter to override.
-force is destructive and will destroy any previous cluster
configuration.
Oracle Cluster Registry for cluster has already been
initialized
Startup will be queued to init within 30 seconds.
Adding daemons to inittab
Expecting the CRS daemons to be up within 600 seconds.
Cluster Synchronization Services is active on these nodes.
        vx0306
        vx0313
Cluster Synchronization Services is active on all the nodes.
Waiting for the Oracle CRSD and EVMD to start
Oracle CRS stack installed and running under init(1M)

[oracle@vx0306 less11]$
```

h) When both scripts have been executed in the correct order, go back to the Execute Configuration scripts screen and click OK.

Practice 11-9: Add Oracle Clusterware to the Second Node (continued)

i) On the End of Installation screen, click Exit.

j) In the Exit dialog box, click Yes to exit the installer.

2) Look at the Oracle Clusterware resources, and check that you now have nodeaps running on the second node.

```
[oracle@vx0306 less11]$ cat sol_11_09_02.sh
#!/bin/bash
#
# sol_11_09_02.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/crs_stat -t
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_09_02.sh
Name           Type           Target    State     Host
------------------------------------------------------------
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora.RDB.db     application    ONLINE    ONLINE    vx0306
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
ora.vx0313.gsd application    ONLINE    ONLINE    vx0313
ora.vx0313.ons application    ONLINE    OFFLINE
ora.vx0313.vip application    ONLINE    ONLINE    vx0313
[oracle@vx0306 less11]$
```

Practice 11-10: Configure ONS for the Second Node

The goal of this practice is to configure the OCR to include the second node ONS existence.

1) Use racgons add_config command to add second node ONS configuration information to the OCR.

```
[oracle@vx0306 less11]$ cat sol_11_10_01.sh
#!/bin/bash
#
# sol_11_10_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/racgons add_config $z:6251
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_10_01.sh
[oracle@vx0306 less11]$
```

Practice 11-11: Add ASM Software to the Second Node

The goal of this practice is to install the ASM software on the second node of your cluster.

1) Login as the Oracle user execute the addNode.sh script located in your ASM home directory on the first node. This script runs the Oracle Universal Installer.

```
[oracle@vx0306 less11]$ cat sol_11_11_01.sh
#!/bin/bash
#
# sol_11_11_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/asm_1

cd $ORACLE_HOME/oui/bin

export DISPLAY=$y:2.0

./addNode.sh
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_11_01.sh
…
```

a) On the Welcome screen, click Next.

b) On the Specify Cluster Nodes to Add to Installation screen, check the name of your second node in the Specify New Nodes table. Click Next.

c) On the Cluster Node Addition Summary screen, click Install.

d) The Cluster Node Addition Progress screen appears. Wait until OUI asks you to run some scripts.

e) When the Execute Configuration scripts window appears, do the following:

f) Execute /u01/app/oracle/product/11.1.0/asm_1/root.sh on the second node as root user. Just type "return" for all questions ask (default values).

Practice 11-11: Add ASM Software to the Second Node (continued)

```
[oracle@vx0306 less11]$ cat sol_11_11_01f.sh
#!/bin/bash
#
# sol_11_11_01f.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

ssh $z sudo /u01/app/oracle/product/11.1.0/asm_1/root.sh
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_11_01f.sh
Running Oracle 11g root.sh script...

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /u01/app/oracle/product/11.1.0/asm_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]:
The file "oraenv" already exists in /usr/local/bin.  Overwrite
it? (y/n)
[n]:
The file "coraenv" already exists in /usr/local/bin.
Overwrite it? (y/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.
[oracle@vx0306 less11]$
```

g) When the script has been executed, go back to the Execute Configuration scripts window and click OK.

Practice 11-11: Add ASM Software to the Second Node (continued)

h) On the End of Installation screen, click Exit.

i) In the Exit dialog box, click Yes to exit the installer.

Practice 11-12: Add a Listener to the Second Node

The goal of this practice is to configure and start a new listener on the second node of your cluster.

1) Connected as user oracle on the second node, execute netca from the ASM home you just installed. Then, using netca, add a new standard listener to the second node.

```
[oracle@vx0306 less11]$ cat sol_11_12_01.sh
#!/bin/bash
#
# sol_11_12_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

scp /home/oracle/solutions/less11/runnetcaasm.sh
$z:/home/oracle/solutions/less11/runnetcaasm.sh

ssh $z chmod 777 /home/oracle/solutions/less11/runnetcaasm.sh

ssh $z "/home/oracle/solutions/less11/runnetcaasm.sh"
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ cat runnetcaasm.sh
#!/bin/bash
#
# runnetcaasm.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#
y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

export DISPLAY=$y:2.0

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/asm_1

cd $ORACLE_HOME/bin

./netca
```

```
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_12_01.sh
…
```

a) On the Configuration screen, select Cluster configuration and click Next.

b) On the Active Nodes screen, select ONLY the second node (the one you want to add) and click Next.

c) On the Welcome screen, select Listener configuration and click Next.

d) On the Listener screen, select Add and click Next.

e) On the Listener Name screen, enter LISTENER (should be the default value already typed) Click Next.

f) On the Select Protocols screen, select only TCP (should be the default) and click Next.

g) On the TCP/IP Protocol screen, select Use the standard port number of 1521 and click Next.

h) On the More Listeners screen, select No, and click Next.

i) On the Listener Configuration Done screen, you should see a Listener configuration complete message. Click Next.

j) On the Welcome screen, click Finish.

2) Check that the Oracle Clusterware listener resource is created on the second node.

```
[oracle@vx0306 less11]$ cat sol_11_12_02.sh
#!/bin/bash
#
# sol_11_12_02.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/crs_stat -t
[oracle@vx0306 less11]$
```

Practice 11-12: Add a Listener to the Second Node (continued)

```
[oracle@vx0306 less11]$ ./sol_11_12_02.sh
Name           Type           Target    State     Host
------------------------------------------------------------
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora.RDB.db     application    ONLINE    ONLINE    vx0306
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
ora....13.lsnr application    ONLINE    ONLINE    vx0313
ora.vx0313.gsd application    ONLINE    ONLINE    vx0313
ora.vx0313.ons application    ONLINE    OFFLINE
ora.vx0313.vip application    ONLINE    ONLINE    vx0313
[oracle@vx0306 less11]$
```

Practice 11-13: Add Database Software to the Second Node

The goal of this practice is to install the Database software on the second node of your cluster.

1) Login as the oracle user execute the addNode.sh script located in your database home directory on the first node. This script runs the Oracle Universal Installer. Add the database software to your second node.

```
[oracle@vx0306 less11]$ cat sol_11_13_01.sh
#!/bin/bash
#
# sol_11_13_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

cd $ORACLE_HOME/oui/bin

export DISPLAY=$y:2.0

./addNode.sh
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_13_01.sh
…
```

a) On the Welcome screen, click Next.

b) On the “Specify Cluster Nodes to Add to Installation” screen, check the name of your second node in the Specify New Nodes table. Click Next.

c) On the Cluster Node Addition Summary screen, click Install.

d) The Cluster Node Addition Progress screen appears. Wait until OUI asks you to run some scripts.

e) When the Execute Configuration scripts window appears, do the following:

f) Execute /u01/app/oracle/product/11.1.0/db_1/root.sh on the second node as root user. Just type "return" for all questions ask (default values).

Practice 11-13: Add Database Software to the Second Node (continued)

```
[oracle@vx0306 less11]$ cat sol_11_13_01f.sh
#!/bin/bash
#
# sol_11_13_01f.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

ssh $z sudo /u01/app/oracle/product/11.1.0/db_1/root.sh
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_13_01f.sh
Running Oracle 11g root.sh script...

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /u01/app/oracle/product/11.1.0/db_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]:
The file "oraenv" already exists in /usr/local/bin.  Overwrite
it? (y/n)
[n]:
The file "coraenv" already exists in /usr/local/bin.
Overwrite it? (y/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.
[oracle@vx0306 less11]$
```

g) When the script has been executed, go back to the Execute Configuration scripts window and click OK.

Practice 11-13: Add Database Software to the Second Node (continued)

h) On the End of Installation screen, click Exit.

i) In the Exit dialog box, click Yes to exit the installer.

Practice 11-14: Add a Database Instance to the Second Node

The goal of this practice is to add a new instance to your RAC database on the second node using both DBCA and Enterprise Manager.

1) Use DBCA to add an ASM instance to your second node.

```
[oracle@vx0306 less11]$ cat sol_11_14_01.sh
#!/bin/bash
#
# sol_11_14_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/asm_1

cd $ORACLE_HOME/bin

./dbca

[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_14_01.sh
…
```

a) On the Welcome screen, select Oracle Real Application Clusters database and click Next.

b) On the Operations screen, select Configure Automatic Storage Management and click Next.

c) On the Nodes Selection screen, select both nodes, and click Next.

d) In the Database Configuration Assistant dialog box, click Yes to extend your ASM cluster to the second node.

e) In the ASM Credentials dialog box, enter oracle1 and click OK.

f) On the ASM Disk Groups screen, click Finish.

g) In the Database Configuration Assistant dialog box, click No to exit from DBCA.

Practice 11-14: Add a Database Instance to the Second Node (continued)

2) Start your Enterprise Manager Database Control console.

```
[oracle@vx0306 less11]$ cat sol_11_14_02.sh
#!/bin/bash
#
# sol_11_14_01.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1

export ORACLE_SID=$I1NAME

cd $ORACLE_HOME/bin

./emctl start dbconsole

[oracle@vx0306 less11]$

[oracle@vx0306 less11]$ ./sol_11_14_02.sh
Oracle Enterprise Manager 11g Database Control Release
11.1.0.6.0
Copyright (c) 1996, 2007 Oracle Corporation.  All rights
reserved.
https://vx0306.us.oracle.com:1158/em/console/aboutApplication
Starting Oracle Enterprise Manager 11g Database Control ......
started.
------------------------------------------------------------------
----
Logs are generated in directory
/u01/app/oracle/product/11.1.0/db_1/vx0306_RDB1/sysman/log
[oracle@vx0306 less11]$
```

3) Use Enterprise Manager Database Control to add an instance to your existing database.

   a) Open a browser window and connect as SYS to your database control console.

   b) When on the Cluster Database Home page, click the Server tab.

c) On the Server page, click the Add instance link in the Change Database section.

d) On the Add Instance: Cluster Credentials page, enter both Cluster credentials (oracle/oracle) and ASM credentials (sys/oracle1). Make sure that you check the Save as Preferred Credentials check box. When you are finished, click Next.

e) On the Add Instance: Host page, select the second node of your cluster and click Next.

f) On the Add Instance: Review page, click the Submit Job button.

g) On the Confirmation page, click the View Job button.

h) When on the Execution page, refresh you browser page until the job is finished.

4) At this point check that all resources are created and active on both nodes.

```
[oracle@vx0306 less11]$ cat sol_11_14_04.sh
#!/bin/bash
#
# sol_11_14_02.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

DBNAME=`ps -ef | grep dbw0_RDB | grep -v grep | grep -v
callout1 | awk '{ print $8 }' | sed 's/1/''/' | sed
's/ora_dbw0_/''/'`

I1NAME=$DBNAME"1"

I2NAME=$DBNAME"2"

/u01/crs11g/bin/crs_stat -t
[oracle@vx0306 less11]$


[oracle@vx0306 less11]$ ./sol_11_14_04.sh
Name           Type           Target    State     Host
------------------------------------------------------------
ora....B1.inst application    ONLINE    ONLINE    vx0306
ora....B2.inst application    ONLINE    ONLINE    vx0313
ora.RDB.db     application    ONLINE    ONLINE    vx0306
ora....SM1.asm application    ONLINE    ONLINE    vx0306
ora....06.lsnr application    ONLINE    ONLINE    vx0306
ora.vx0306.gsd application    ONLINE    ONLINE    vx0306
ora.vx0306.ons application    ONLINE    ONLINE    vx0306
ora.vx0306.vip application    ONLINE    ONLINE    vx0306
ora....SM2.asm application    ONLINE    ONLINE    vx0313
ora....13.lsnr application    ONLINE    ONLINE    vx0313
```

Practice 11-14: Add a Database Instance to the Second Node (continued)

```
ora.vx0313.gsd application    ONLINE    ONLINE    vx0313
ora.vx0313.ons application    ONLINE    ONLINE    vx0313
ora.vx0313.vip application    ONLINE    ONLINE    vx0313
[oracle@vx0306 less11]$
```

Practices for Lesson 12

There are no practices for this lesson.

Optional Practice: Direct NFS

In this practice, you use Direct NFS to create a new RAC database that resides on NFS storage.

1) Before you can create the database, you need to install a new Oracle Home on your system because Direct NFS requires a special Oracle Disk Manager. Use Oracle Universal Installer to install a new Oracle Home in `/u01/app/oracle/product/11.1.0/dnfs`.

   a) Start Oracle Universal Installer:

```
[oracle@vx0308 opt]$ cat sol_opt_01a.sh
/stage/db/runInstaller

[oracle@vx0308 opt]$

[oracle@vx0308 opt]$ ./ sol_opt_01a.sh
…
```

   b) On the Select a Product to Install screen, select Oracle Database 11*g* and click Next.

   c) On the Select Installation Type screen, select Enterprise Edition and click Next.

   d) On the Install Location screen, enter `/u01/app/oracle` in the Oracle Base field and `/u01/app/oracle/product/11.1.0/dnfs` in the Path field. Then click Next.

   e) On the Specify Hardware Cluster Installation Mode screen, select Cluster Installation, and click Select All button. Then click Next.

   f) On the Product-Specific Prerequisite Checks screen, click Next.

   g) On the Select Configuration Option screen, select Install Software Only and click Next.

   h) On the Privileged Operating System Groups screen, click Next.

   i) On the Summary screen, click Install.

   j) Follow the instructions on the Execute Configuration Scripts screen:

```
[oracle@vx0308 opt]$ cat sol_opt_01j.sh
#!/bin/ksh
#
# sol_opt_01j.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

sudo /u01/app/oracle/product/11.1.0/dnfs/root.sh
```

Optional Practice: Direct NFS (continued)

```
ssh $z "sudo /u01/app/oracle/product/11.1.0/dnfs/root.sh"


[oracle@vx0308 opt]$ ./sol_opt_01j.sh
Running Oracle 11g root.sh script...

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /u01/app/oracle/product/11.1.0/dnfs

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]:
The file "oraenv" already exists in /usr/local/bin.  Overwrite
it? (y/n)
[n]:
The file "coraenv" already exists in /usr/local/bin.
Overwrite it? (y/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.
Running Oracle 11g root.sh script...

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /u01/app/oracle/product/11.1.0/dnfs

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]:
The file "oraenv" already exists in /usr/local/bin.  Overwrite
it? (y/n)
[n]:
The file "coraenv" already exists in /usr/local/bin.
Overwrite it? (y/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.
[oracle@vx0308 opt]$
```

Optional Practice: Direct NFS (continued)

k) Click OK on the Execute Configuration Scripts screen.

l) On the End of Installation screen, click Exit.

m) On the Exit screen, click Yes.

2) After installing the new Oracle Home, you need to create and mount your NFS directory on both nodes.

   a) On the first node and second node, as `root`, create a new directory called `/RACDB`. Make sure that this directory is owned by `oracle` and that enough rights are granted. Then mount your NFS volume to `/RACDB`. You need to know the filer name. Ask your instructor.

```
[oracle@vx0308 opt]$ cat sol_opt_02a.sh
#!/bin/ksh
#
# sol_opt_02a.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

# EDIT THIS VARIABLE FIRST!!
echo "Enter the name of your NFS server:"
read f

sudo mkdir /RACDB
sudo chown oracle:dba /RACDB
sudo chmod 777 /RACDB
sudo mount -t nfs -o rw,nfsvers=3 $f:/vol/$y/$y/RACdb /RACDB

ssh $z "sudo mkdir /RACDB"
ssh $z "sudo chown oracle:dba /RACDB"
ssh $z "sudo chmod 777 /RACDB"
ssh $z "sudo mount -t nfs -o rw,nfsvers=3 $f:/vol/$y/$y/RACdb
/RACDB"

[oracle@vx0308 opt]$


[oracle@vx0308 opt]$ ./sol_opt_02a.sh
Enter the name of your NFS server:
f18
[oracle@vx0308 opt]$
```

   b) (Optional) As `root`, add the following line to `/etc/fstab` on both nodes:
   `<nfs server name>:/vol/<first node name>/<first node name>/RACdb /RACdb nfs rw,nfsvers=3 0 0`

3) Change the Oracle Disk Manager for your new Oracle Home on both nodes.

```
[oracle@vx0308 opt]$ cat sol_opt_03.sh
#!/bin/ksh
#
# sol_opt_03.sh
#
# Must be executed on NODE1 !!!!!!!!!!
#

y=`cat /home/oracle/nodeinfo | sed -n '1,1p'`
z=`cat /home/oracle/nodeinfo | sed -n '2,2p'`

cd /u01/app/oracle/product/11.1.0/dnfs/lib
mv libodm11.so libodm11.so_stub
ln -s libnfsodm11.so libodm11.so


ssh $z "mv /u01/app/oracle/product/11.1.0/dnfs/lib/libodm11.so
/u01/app/oracle/product/11.1.0/dnfs/lib/libodm11.so_stub"
ssh $z "ln -s
/u01/app/oracle/product/11.1.0/dnfs/lib/libnfsodm11.so
/u01/app/oracle/product/11.1.0/dnfs/lib/libodm11.so"

[oracle@vx0308 opt]$ ./sol_opt_03.sh
[oracle@vx0308 opt]$
```

4) You can now use DBCA from your new Oracle Home on the first node to create a new RAC database stored on `/RACDB`.

   a) Start DBCAL:

```
[oracle@vx0308 opt]$ cat sol_opt_04.sh
/u01/app/oracle/product/11.1.0/dnfs/bin/dbca

[oracle@vx0308 opt]$


[oracle@vx0308 opt]$ ./sol_opt_04.sh
…
```

   b) On the Welcome screen, select Oracle Real Application Clusters database and click Next.

   c) On the Operations screen, select Create a Database and click Next.

   d) On the Node Selection screen, click Select All and click Next.

   e) On the Database Templates screen, select General Purpose or Transaction Processing, and then click Next.

   f) On the Database Identification screen, enter `DNFS` in the Global Database Name field and click Next.

   g) On the Management Options screen, uncheck Configure Enterprise Manager and click Next.

Optional Practice: Direct NFS (continued)

h) On the Database Credentials screen, select "Use the Same Administrative Password for All Accounts," enter `oracle1` in both credential fields, and then click Next.

i) On the Storage Option screen, select Cluster File System and click Next.

j) On the Database File Locations screen, select "Use Common Location for All Database Files," enter `/RACDB` in the Database Files Location field, and then click Next.

k) On the Recovery Configuration screen, select Specify Flash Recovery Area, enter `/RACDB` in the Flash Recovery Area field, and then click Next.

l) On the Database Content screen, select Sample Schema and click Next.

m) On the Initialization Parameters screen, click Next.

n) On the Security Settings screen, click Next.

o) On the Automatic Maintenance Tasks screen, click Next.

p) On the Database Storage screen, click Next.

q) On the Creation Options, select Create Database and click Finish.

r) On the Summary screen, click OK.

s) Wait for the progress bar on the Database Configuration Assistant screen to reach 100%.

t) On the Database Configuration Assistant screen that appears, click Exit.

u) Wait for the Start Cluster Database screen to appear.

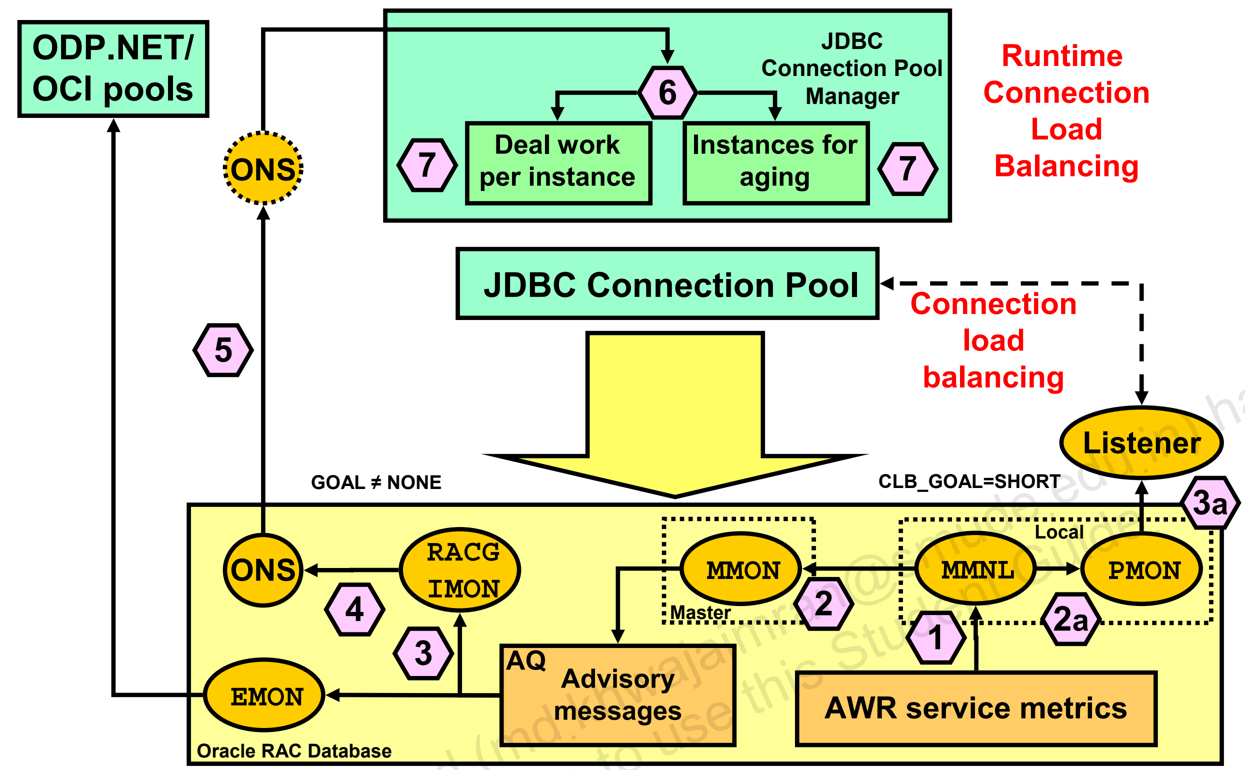
B

Miscellaneous Topics

Contents

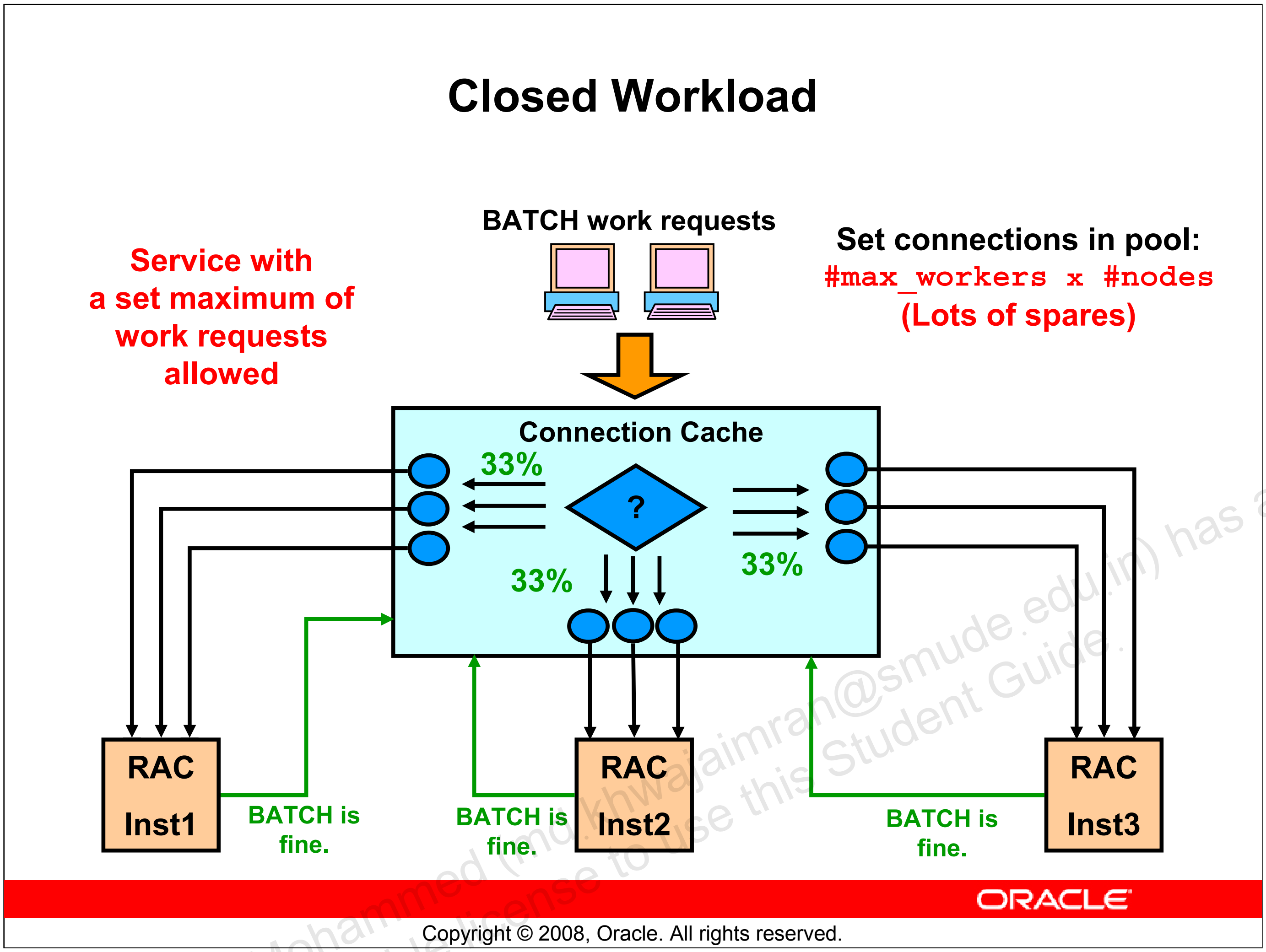
- Additions to Lesson 8
- Additions to Lesson 10

Load Balancing Advisory Workflow in RAC



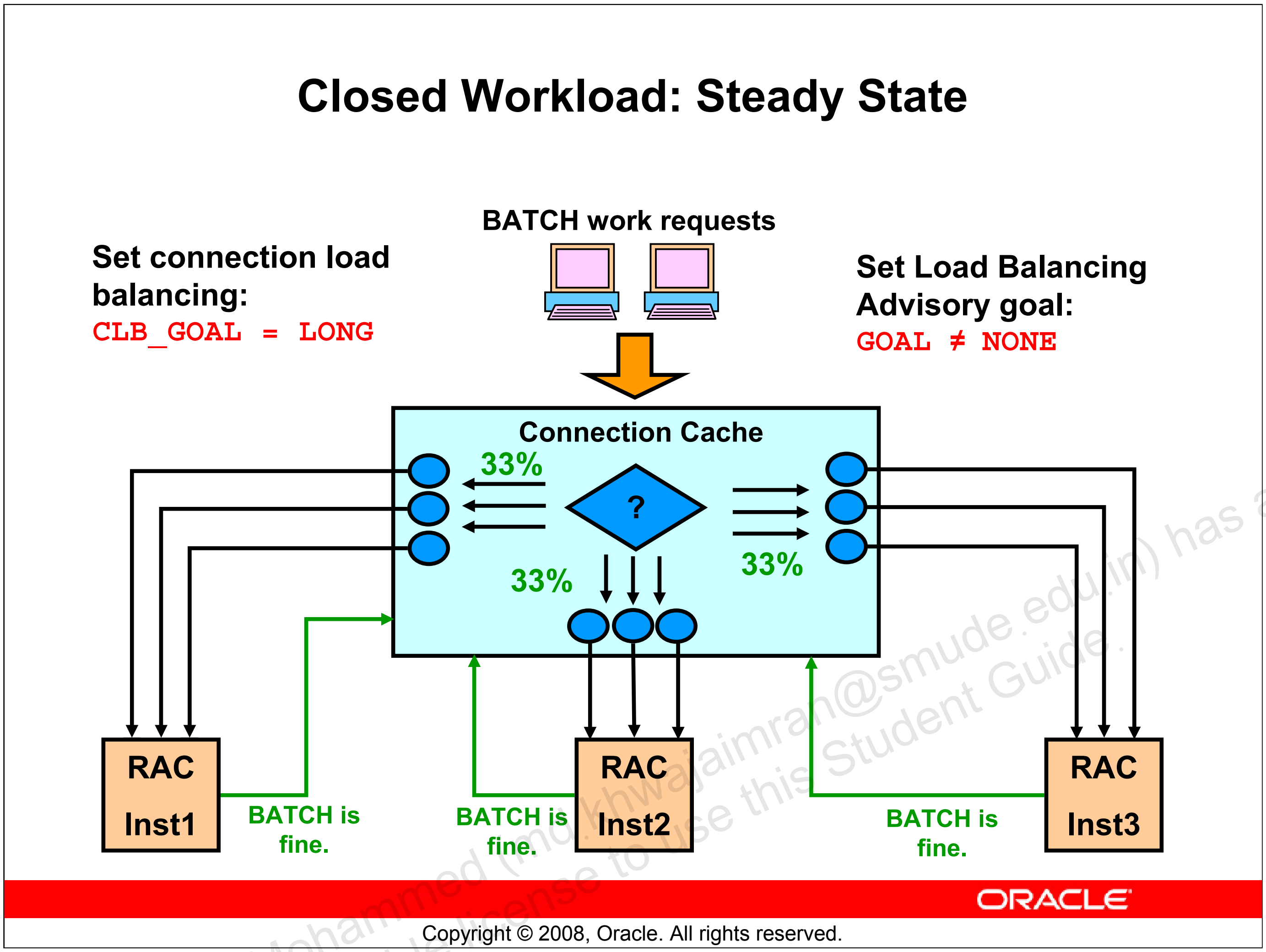
Load Balancing Workflow in RAC

1. MMNL (Manageability MoNitor Light) calculates the service metrics for service goal and resource consumption every five seconds. MMNL derives the service goodness from these data.
2. MMON computes and posts the LBA FAN event to a system queue, and MMNL forwards the service goodness and delta to PMON.
3. IMON (Instance Monitor) and EMON retrieve the event from the queue, and PMON forwards the goodness and delta values to the listeners.
4. IMON posts the LBA FAN event to the local ONS daemon, and EMON posts it to AQ subscribers.
5. The server ONS sends the event to the mid-tier ONS (if used).
6. The mid-tier receives the event and forwards them to subscribers. Each connection pool subscribes to receive events for its own services. On receipt of each event, the Connection Pool Manager refreshes the ratio of work to forward to each RAC instance connection part of the pool. It also ranks the instances to use when aging out connections.
7. Work requests are routed to RAC instances according to the ratios calculated previously. When there are insufficient idle connections to satisfy the ratios, new connections are added to the next best instances determined by the listeners.



Closed Workload

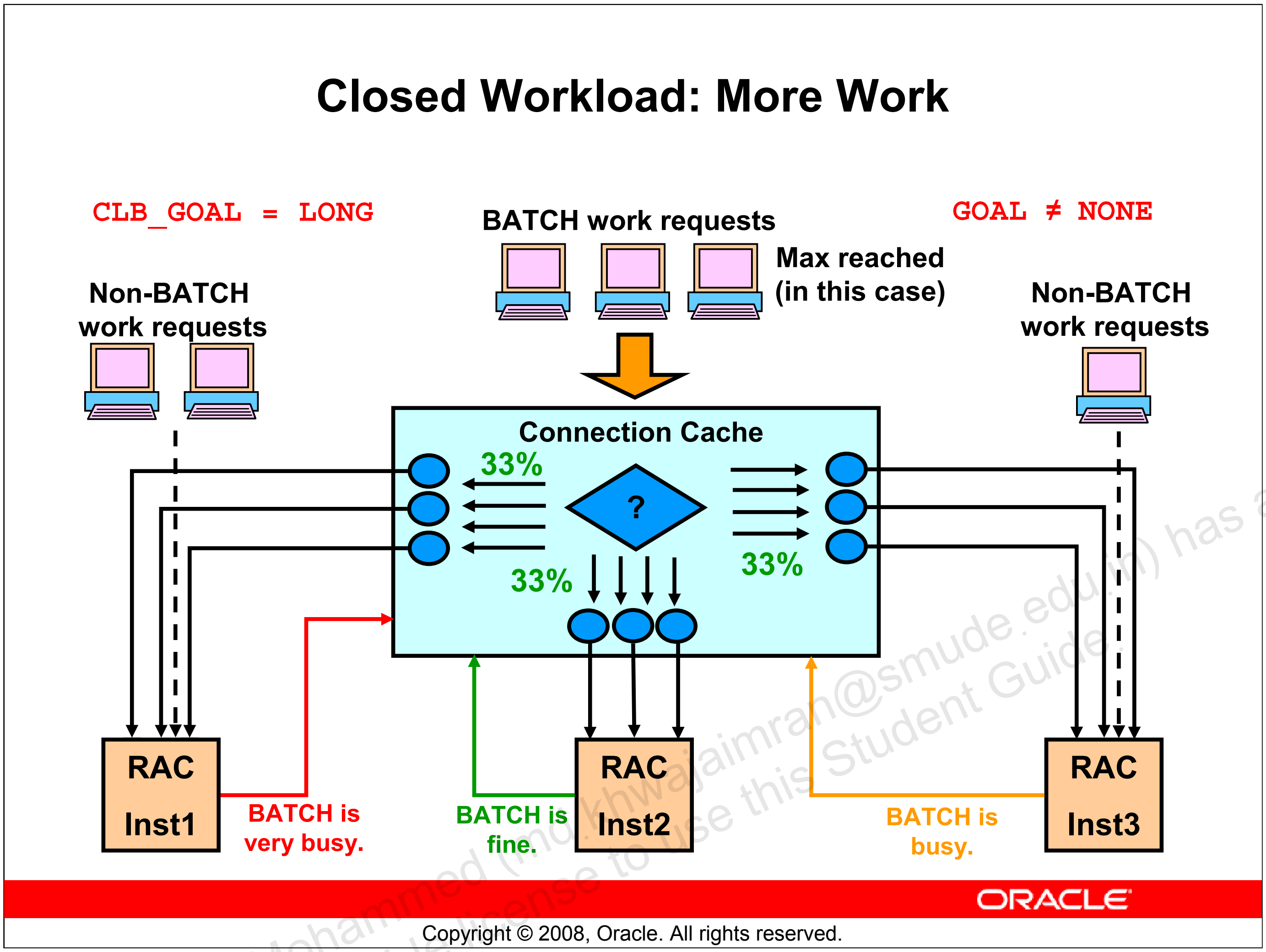
The following slides describe two possible use cases with both connection load balancing, and run-time connection load balancing in a RAC environment. The first case, presented on the slide, shows you what is called a closed workload. In this configuration, you predefine the total number of connections to your RAC instances, and you want them to be evenly distributed on each instance. In the slide example, three connections are established on each instance, and this is the maximum you want to accept. You do so, because you know that you have a predefined maximum number of work requests that can be handled by your connection pool for your service. This is typically the case when you control the application workload that runs on your system. Therefore, the idea is to distribute a maximum number of connections evenly across all your RAC instances for that particular application or service.



Closed Workload: Steady State

You implement this configuration at the RAC level by setting the connection balancing algorithm to LONG to make sure that the listeners distribute the connection requests evenly across all RAC instances. By setting the Load Balancing Advisory to either service time or throughput, you ensure that your connection pool distributes the work requests inside the pool to the best instances according to your Load Balancing Advisory goal.

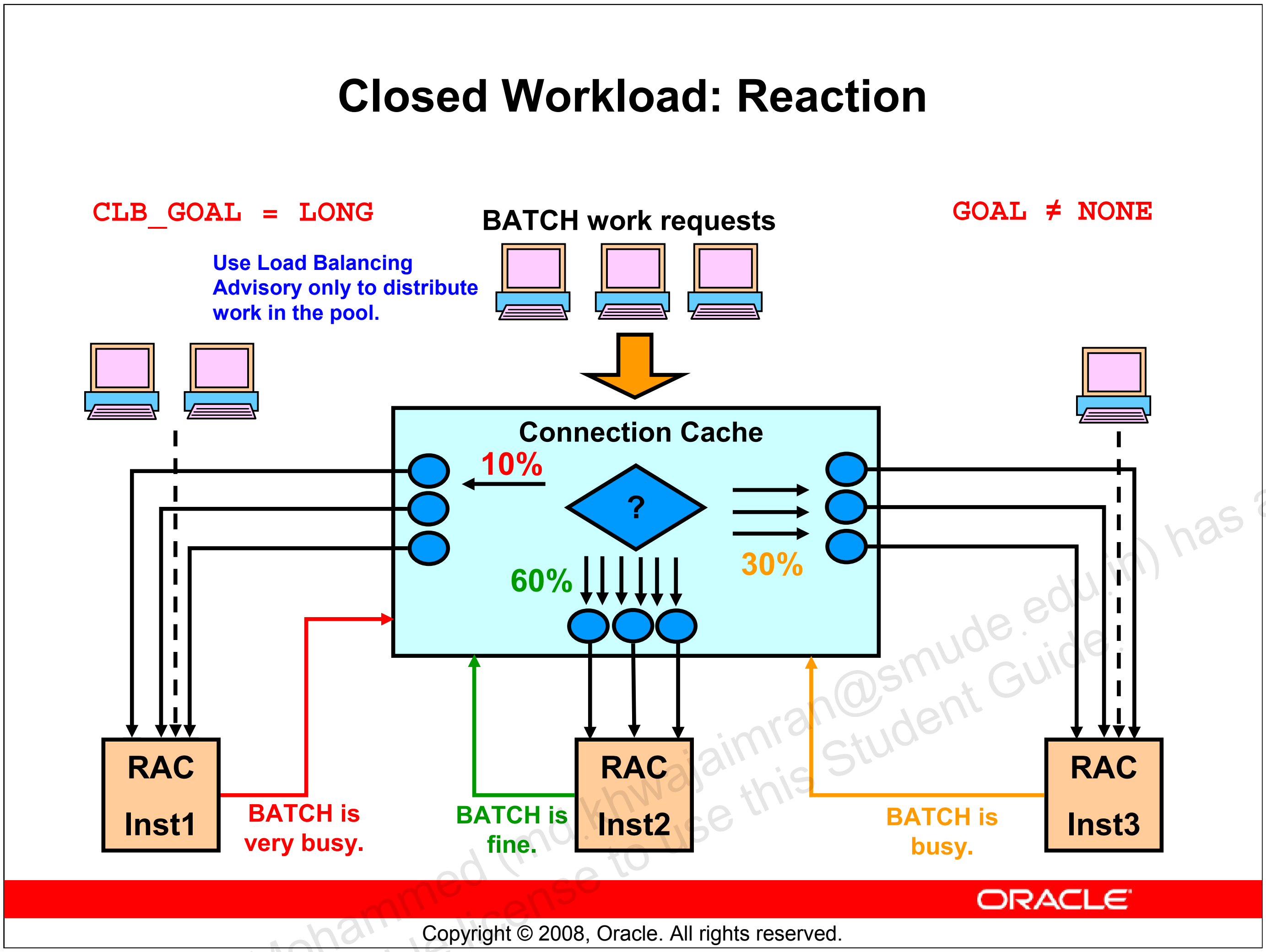
At steady state, all work requests are spread evenly across your RAC instances as shown in the slide. This is because each RAC instance supports the same amount of work using the same amount of resources on each node.



Closed Workload: More Work

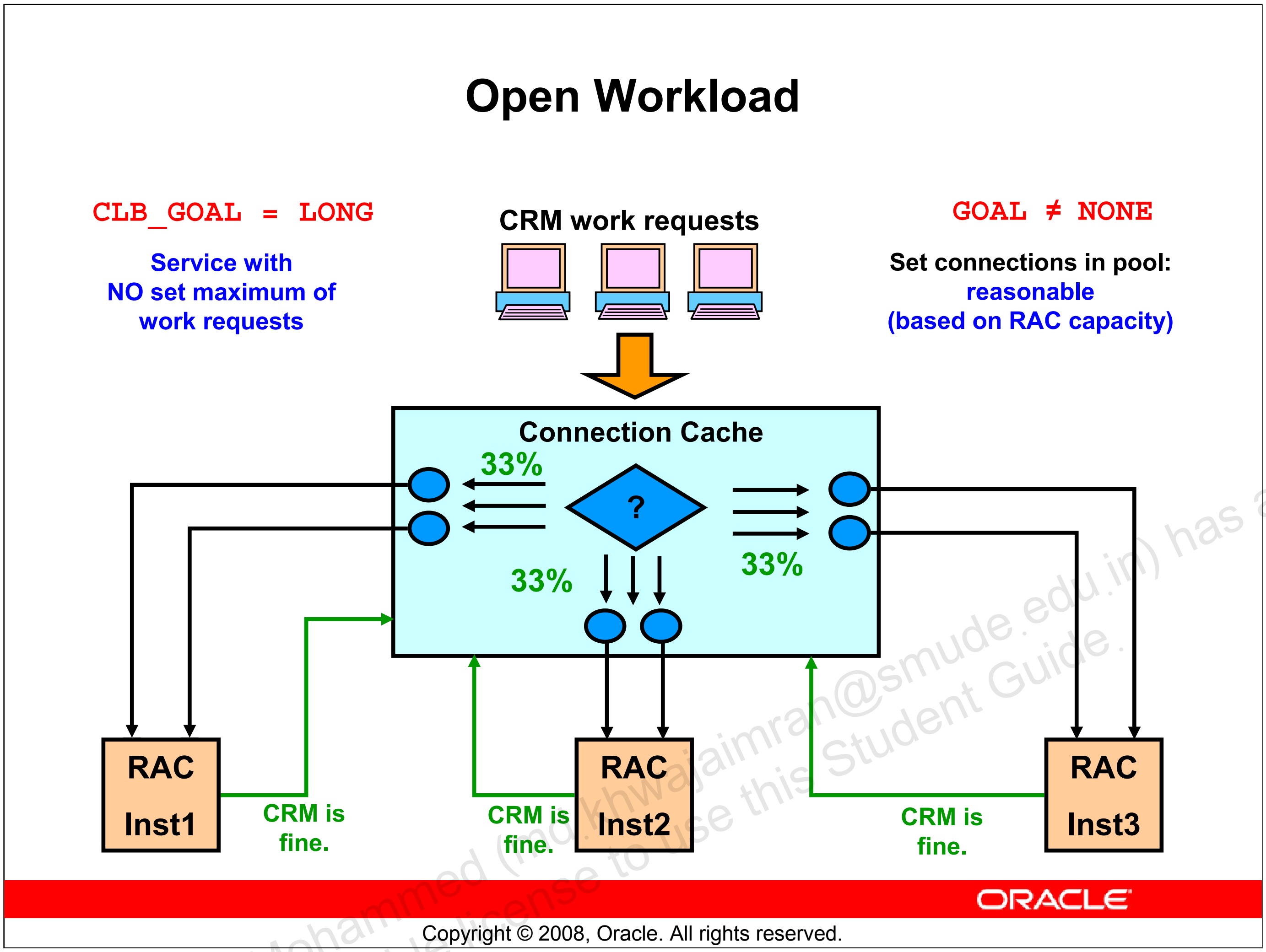
In this slide, more work is added to your RAC system by adding more work requests to your BATCH service as well as by adding new services requests. This ends up overloading the first RAC instance by consuming too many resources on the corresponding node, thereby decreasing its capacity to serve BATCH in an acceptable way. As a result, only the second node, shown in the slide, is still capable to serve BATCH requests normally.

In consequence, Load Balancing Advisory FAN events are sent to your connection pool asking the pool to redistribute most of the BATCH work requests to the second instance, which currently serves BATCH best.



Closed Workload: Reaction

In reaction, your connection pool redistributes 60 percent of the BATCH work requests to the pool of connections already established on the second instance. Here, only the work requests are redistributed. The connections are not redistributed because you define them to make sure that they cannot be saturated by the BATCH work requests.

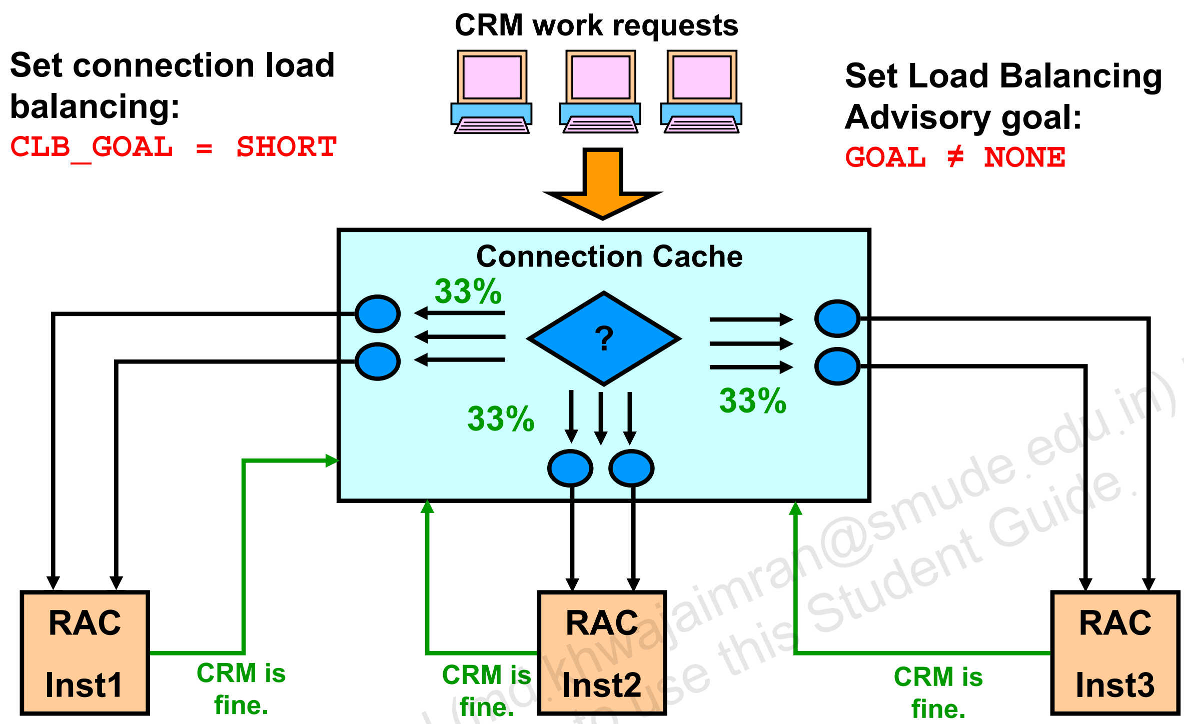


Open Workload

In this case, you do not know in advance how many work requests will have to be handled by your connection pool. So, you cannot predefine them accordingly. This is typically the case with shopping cart type of applications.

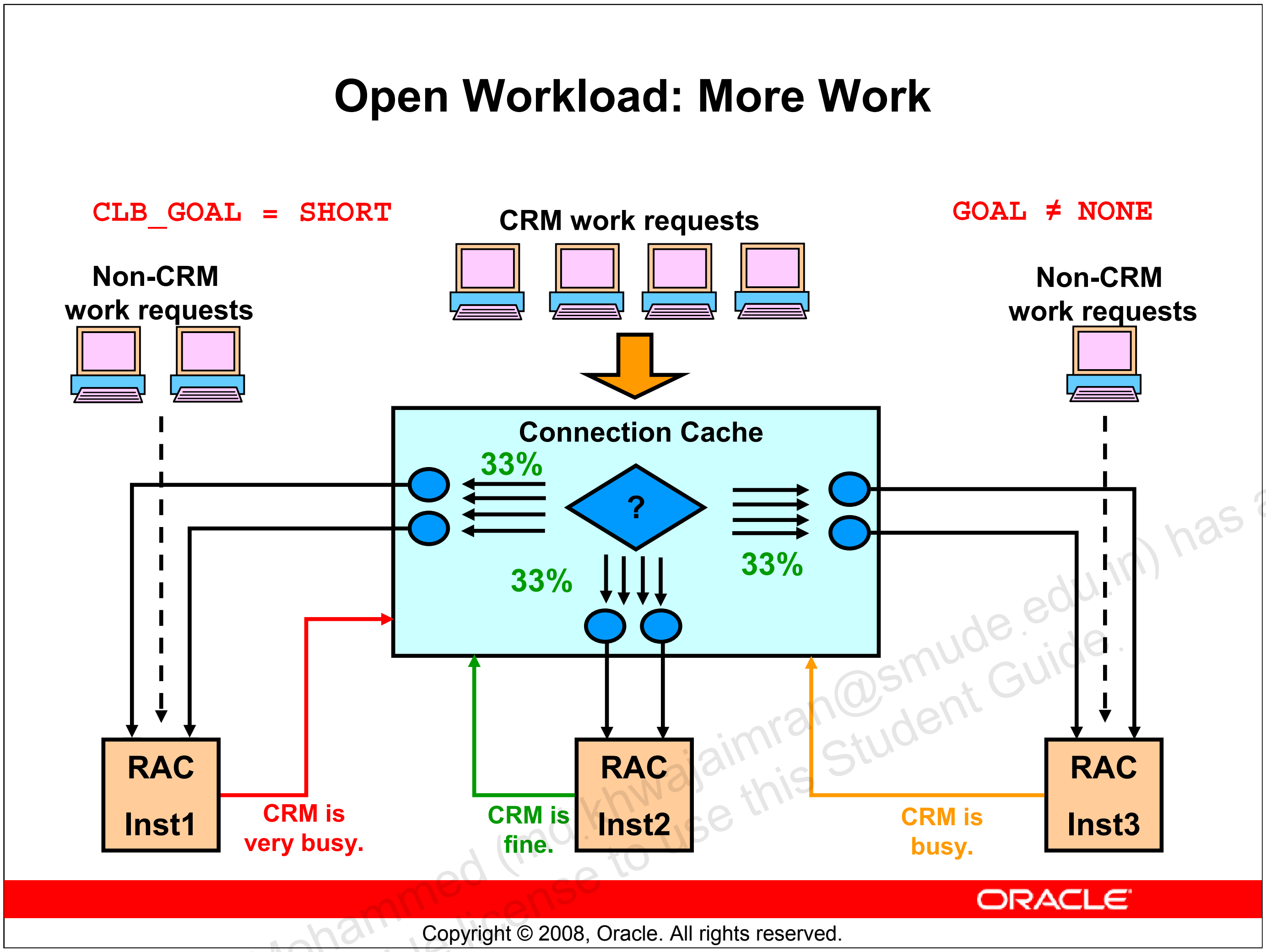
In consequence, you predefine a reasonable amount of connections in your pool, based on your RAC capacity to handle work.

Open Workload: Steady State



Open Workload: Steady State

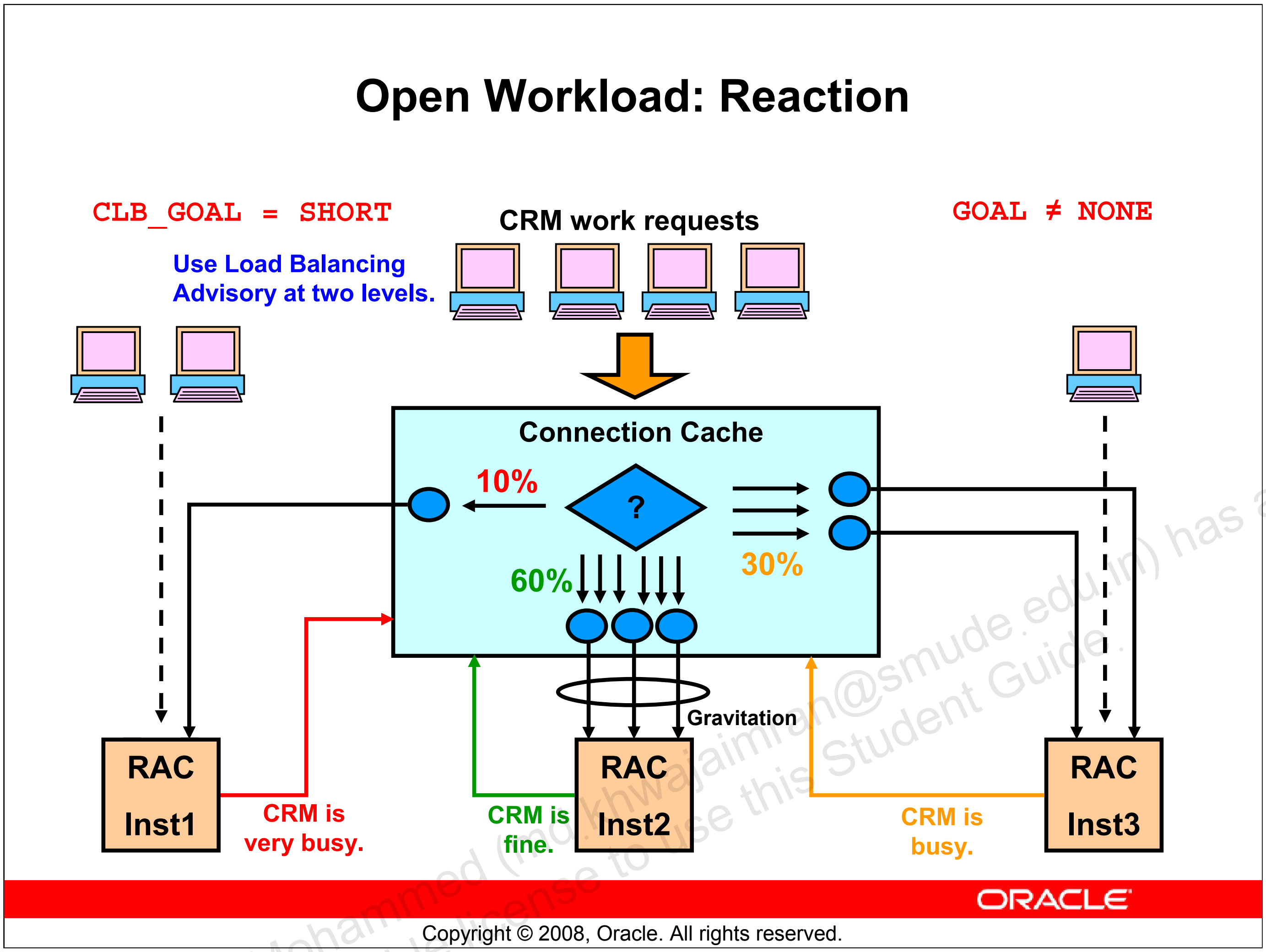
To implement this architecture on your RAC system, you want it to be able not only to redistribute the work requests in your connection pool, but also for your connections to be redistributed to the least loaded nodes because you cannot determine in advance how many connections are needed to avoid their saturation. You do so by setting the connection balancing algorithm to `SHORT` as well as setting your Load Balancing Advisory algorithm to either service time or throughput.



Open Workload: More Work

In this slide, more work is added to your RAC system by adding more work requests to your CRM service as well as by adding new services requests. This ends up overloading the first RAC instance by consuming too many resources on the corresponding node, thereby decreasing its capacity to serve CRM in an acceptable way. As a result, only the second node, shown in the slide, is still capable to serve CRM requests normally.

In consequence, Load Balancing Advisory FAN events are sent to your connection pool asking the pool to redistribute most of the CRM work requests to the second instance, which currently serves CRM best.



Open Workload: Reaction

In reaction to those FAN events, your connection pool has to redistribute the work requests for CRM to the second instance. In this case, because you did not predefine a sufficient number of connections to handle CRM work request, your connection pool must create additional connections to handle the CRM service. Because your connection load balancing algorithm also uses the Load Balancing Advisory information, listeners are going to send those new connection requests to the second instance as illustrated in the slide. This is called connections gravitation.

In this example, not only most of the CRM work requests, but also connection requests, are directed to the second instance.

OCR-Related Tools Debugging

- **OCR tools:**
  - `ocrdump`
  - `ocrconfig`
  - `ocrcheck`
  - `srvctl`
- **Logs generated in** `$ORA_CRS_HOME/log/<hostname>/client/`
- **Debugging control through** `$ORA_CRS_HOME/srvm/admin/ocrlog.ini`

```
mesg_logging_level = 5
comploglvl="OCRAPI:5 ; OCRSRV:5; OCRCAC:5; OCRMAS:5; OCRCONF:5; OCRRAW:5"
comptrclvl="OCRAPI:5 ; OCRSRV:5; OCRCAC:5; OCRMAS:5; OCRCONF:5; OCRRAW:5"
```


OCR-Related Tools Debugging

As you already saw, you can use various tools to manipulate Oracle Cluster Registry (OCR): `ocrdump`, `ocrconfig`, `ocrcheck`, and `srvctl`.

These utilities create log files in `$ORA_CRS_HOME/log/<hostname>/client/`. To change the amount of logging, edit the `$ORA_CRS_HOME/srvm/admin/ocrlog.ini` file. The default logging level is `0`, which basically means minimum logging. When `mesg_logging_level` is set to `0`, which is its default value, only error conditions are logged. You can change this setting to `3` or `5` for detailed logging information.

If that is not enough, you can also change the logging and trace level for each of the components used to manipulate OCR. To do that, edit the entries containing `comploglvl` and `comptrclvl` in `ocrlog.ini`.

The slide shows you the three lines you could add to `ocrlog.ini` to turn on additional debugging information. A typical example where you might have to change the `ocrlog.ini` file is in a situation where you get errors while using either `ocrdump` or `ocrconfig` tools.

Note: You should never execute the commands shown on this slide on your production environment unless explicitly asked by Oracle Support. For additional information about Oracle Clusterware diagnosis, you can also refer to the *Note 353335.1* from Metalink as well as the *Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment* guide.

Enable Resource Debugging

- **Change `USR_ORA_DEBUG` resource attribute to `1` for specific resources:**

```
# crsctl debug log res "ora.atlhp8.vip:1"
```

- **Set the environment variable `_USR_ORA_DEBUG` to `1` in `racgwrap` scripts to debug for all resources:**

```
# Set _USR_ORA_DEBUG to enable more tracing
#_USR_ORA_DEBUG=1 && export _USR_ORA_DEBUG

_USR_ORA_DEBUG=1
export _USR_ORA_DEBUG

$ORACLE_HOME/bin/racgmain "$@"
status=$?
exit $status
```


Enable Resource Debugging

You may be requested to enable tracing to capture additional information for problem resolution with Oracle Clusterware resources when working with Oracle Support. Because the procedures described here may adversely affect performance, perform these activities only with the assistance of Oracle Support.

To generate additional trace information for a particular running resource, as the `root` user, you can use `crsctl` to enable resource debugging using the following syntax:

```
crsctl debug log res "<resource name>:1"
```

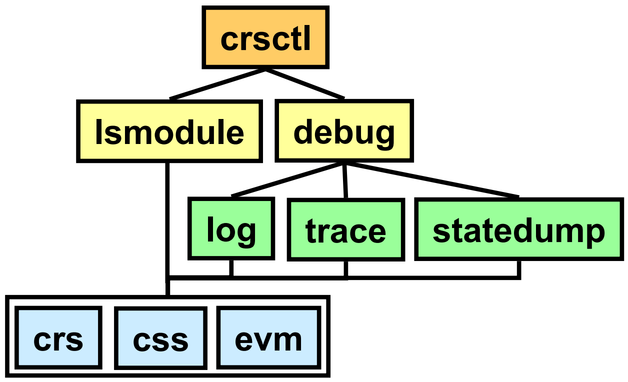
The example in the slide enables debugging for the `ora.atlhp8.vip` resource. This has the effect of setting the environment variable `USER_ORA_DEBUG` to `1` in its resource's profile. This setting is enforced before running the `start`, `stop`, or `check` action scripts for the resource.

If you are asked to enable tracing for all resources, add the following lines to the `racgwrap` script in both `$ORA_CRS_HOME/bin` and `$ORACLE_HOME/bin`:

```
_USR_ORA_DEBUG=1
export _USR_ORA_DEBUG
```

After you capture all trace information, do not forget to either execute corresponding `crsctl debug log res "<resource name>:0"` commands, or to remove the added lines from `racgwrap` scripts to switch off debugging.

Enable Oracle Clusterware Modules Debugging




```
$ crsctl lsmodules crs
The following are the CRS modules :
 CRSUI      CRSCOMM    CRSRTI     CRSMAIN    CRSPLACE   CRSAPP   CRSRES
 CRSCOMM    CRSOCR     CRSTIMER   CRSEVT     CRSD       CLUCLS   CSSCLNT
 COMMCRS    COMMNS
```

```
# crsctl debug log crs CRSEVT:5,CRSAPP:5,CRSTIMER:5,CRSRES:5
```


Enable Oracle Clusterware Modules Debugging

The main Oracle Clusterware daemons (crsd, cssd, evmd) use various internal modules during their execution. You can use crsctl commands as the root user to enable dynamic debugging for the Oracle Clusterware modules.

The crsctl lsmodules crs|css|evm commands are used to list the module's components that can be used for debugging. The example in the slide lists the ones for crs.

When asked by Oracle Support, you can then use commands such as the following to enable addition logging:

- crsctl debug log <module name> <component>:<debugging level>, where <module name> is the name of the module, crs, evm, or css; <component name> is the name of the corresponding component obtained using the crsctl lsmodules command; and <debugging level> is a level from 1 to 5
- crsctl debug statedump crs|css|evm dump state information for crs, css, or evm modules
- crsctl debug trace css|crs|evm dump crs, css, or evm in-memory tracing caches

The example in the slide shows you how to dynamically enable additional logging (level 5) for the following CRS components: CRSEVT, CRSAPP, CRSTIMER, and CRSRES.

Note: ***Never*** execute these commands on a production system without explicit guidance from Oracle Support.

Enable Tracing for Java-Based Tools

- **To enable tracing for:**
  - cluvfy
  - netca
  - vipca
  - srvctl
  - gsdctl
- **Set SRVM_TRACE to TRUE:**

```
$ export SRVM_TRACE=TRUE
$ srvctl config database -d xwkE > /tmp/srvctl.trc
$ cat /tmp/srvctl.trc
/u01/app/oracle/product/Crs/jdk/jre/bin/java -classpath
/u01/app/oracle/product/Crs/jlib/netcfg.jar: …
srvctl.jar -DTRACING.ENABLED=true -DTRACING.LEVEL=2
oracle.ops.opsctl.OPSCTLDriver config database -d xwkE
[main] [9:47:27:454] [OPSCTLDriver.setInternalDebugLevel:165]
tracing is true at level 2 to file null
[main] [9:47:27:496] [OPSCTLDriver.<init>:95]  Security manager
is set …
```

ORACLE

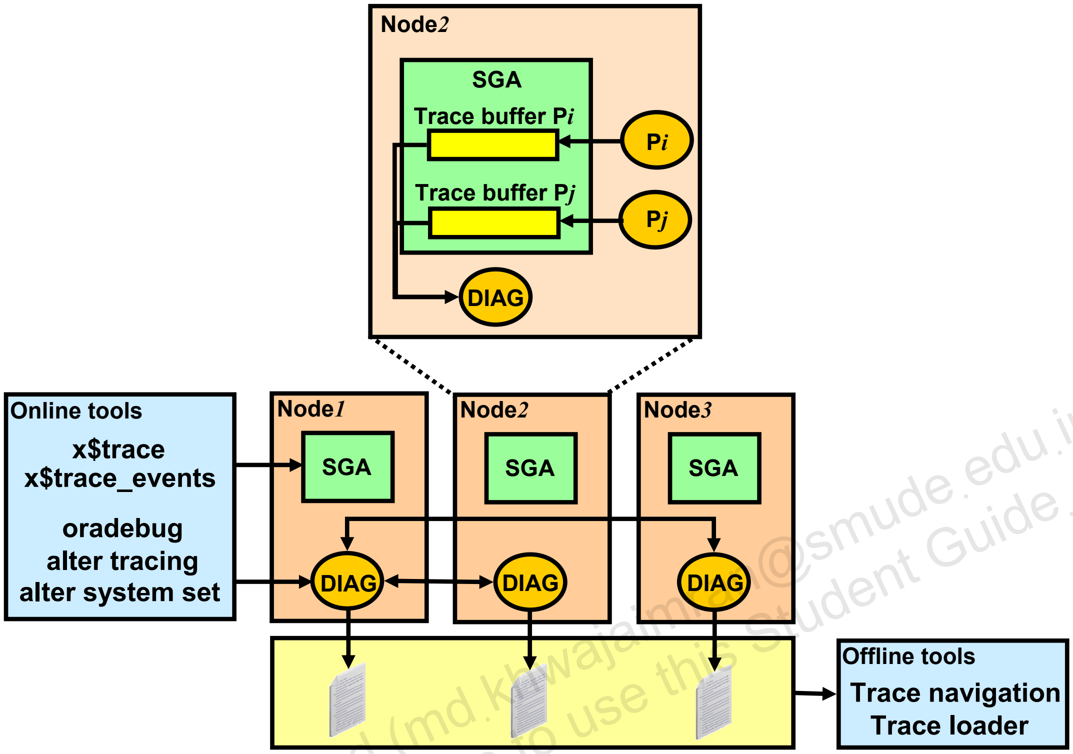
Enable Tracing for Java-Based Tools

All Java-based tools and utilities that are available in RAC are invoked by executing scripts of the same name as the tool or utility. This includes the Cluster Verification Utility (`cluvfy`), the Database Configuration Assistant (`dbca`), the Database Upgrade Assistant (`dbua`), the Net Configuration Assistant (`netca`), the Virtual Internet Protocol Configuration Assistant (`vipca`), Server Control (`srvctl`), and the Global Services Daemon (`gsdctl`). For example, to run the Database Configuration Assistant, enter the command `dbca`.

By default, Oracle enables traces for `dbca` and `dbua`. The resulting log files are written to `$ORACLE_HOME/cfgtoollogs/dbca/` and `$ORACLE_HOME/cfgtoollogs/dbua/`, respectively. For `cluvfy`, `gsdctl`, `srvctl`, and `vipca`, you can set the `SRVM_TRACE` environment variable to `TRUE` to make the system generate traces. Traces are written to either log files or standard output. For example, the system writes traces to log files in `$ORA_CRS_HOME/cv/log/` for `cluvfy`. However, as shown in the slide, it writes traces directly to the standard output for `srvctl`.

Note: *Never* execute these commands on a production system without explicit guidance from Oracle Support.

RAC Diagnostic Infrastructure: Overview



RAC Diagnostic Infrastructure: Overview

The slide shows you the diagnostic architecture of RAC instances in your cluster. This architecture enables trace processing to incur very low overhead. The DIAG process was introduced in Oracle9*i* Database to manage all diagnostics-related activities, acting as a broker between online debugging tools and regular database processes. All debugging commands are issued through the DIAG process on the same node to reach their intended targets. This DIAG process then coordinates with DIAG processes on other nodes of the same cluster to complete the commands. Activities such as setting trace levels, archiving the in-memory trace logs to files, taking memory/crash dumps are done by the DIAG processes resulting in very little overhead to the database server.

By default, minimal tracing is always on for foreground and background processes, and all trace information is written into in-memory buffers within the System Global Area (SGA) instead of being written into files directly. Via the online diagnostic tools, you can instruct DIAG to set trace levels, archive the in-memory trace logs to files, and take memory dumps. This can be done for one or all processes on all instances. Offline tools then transform the archived logs into human-readable formats, load them into database for query, or display them with the GUI interface of the Trace Navigation tool used by Oracle Support. All these trace files have `.trw` as their file extension, so they can be distinguished from regular process trace files. Also, these trace files are circular, similar to the memory buffers to limit the file size.

DIAG Trace Control Interfaces

- Initialization parameter `TRACE_ENABLED`
- SQL:
  - `X$TRACE_EVENTS` and `X$TRACE`
  - `ALTER TRACING ON|OFF|ENABLE|DISABLE|FLUSH`
  - `ALTER SYSTEM SET`
- `oradebug`:
  - RAC aware
  - Clusterwide hang analyzer
  - System and process state dumps

DIAG Trace Control Interfaces

The diagnostic tracing facility provides various controls on its tracing behavior. You can specify the controls through either:

- Initialization parameter during instance startup: `TRACE_ENABLED`, is visible to the users to enable or disable the tracing mechanism. By default, its value is `TRUE`, and minimal tracing takes place. However, you can set it to `FALSE` to meet some high-end benchmark requirements.
- SQL statements: At run time of an instance, tracing behavior can also be modified through SQL statements such as:
  - ALTER TRACING to turn on or turn off the tracing mechanism, to flush trace buffers to disk, and to enable or disable trace events
  - ALTER SYSTEM SET to change value of initialization parameters
- Fixed table views: As shown in the slide, there are two fixed table views related to this tracing mechanism. They are used for online monitoring of tracing characteristics and contents of trace buffers in SGA.
- `oradebug` during run time: It is possible to execute `oradebug` commands on remote instances from a local instance and get the results back to the local node. You can use the `-r` and `-g` options to do that.

Note: ***Never*** execute these commands on a production system without explicit guidance from Oracle Support.

DIAG Crash Dumps

- Crash dump (clusterwide) by DIAG processes when detecting the death of a fatal Oracle process:
  - Local system state dump
  - Remote SGA buffers dump
- If a RAC instance crashes:
  - Second last process to terminate
  - Allowed to dump trace
- Local DIAG coordinates dumping of trace buffers on remote nodes.
- Dumps in the `cdmp_<timestamp>` directory under background dump destination

DIAG Crash Dumps

Crash dump is one of the most important features of the DIAG process. During an instance crash, DIAG sends out a dump message to peer DIAG processes in the cluster and then creates a system state dump. Each remote DIAG process then flushes its trace information to the disks. That way a cluster snapshot can be reused later by Oracle Support for better diagnostics.

During the instance cleanup procedure, DIAG is the second last process to be terminated because it needs to perform trace flushing to the file system. By default, the terminating process, usually PMON, gives a small amount of time to DIAG for dumping.

Instance freezing is not required in order to obtain the snapshot of traces across all instances. It is because all traces with execution history required for diagnosis are already stored in the memory buffers and are dumped to file after a DIAG process receives the crash notification. Traces for the moment of crash are likely to be in this history.

Note: A dump directory, named `cdmp_<timestamp>`, is created at the `BACKGROUND_DUMP_DEST` location and all trace dump files are placed in this directory on the remote nodes.

DIAG Trace Control Interface: Examples

```
SQL> alter tracing enable "10425:10:135";
```
1

```
SQL> SELECT trclevel,status,procs FROM x$trace_events
  2> WHERE  event=10425;
```
2

```
SQL> alter tracing disable "ALL";
```
3

```
SQL> alter tracing flush;
```
4

```
SQL> oradebug setmypid
Statement processed.
SQL> oradebug setinst all
Statement processed.
SQL> oradebug -g def hanganalyze 3
Hang Analysis in
/u01/app/oracle/diag/rdbms/racdb/racdb1/trace/racdb1_diag_11347.trc
SQL>
```
5

DIAG Trace Control Interface: Examples

The slide shows you some possible examples of use as long as you can connect normally to your database:

1. The first example is used from a SQL*Plus session to turn on tracing for event 10425 at level 10 for process ID 135.
2. You can then query `X$TRACE_EVENTS` to determine which events are traced (`STATUS=1`) at which level. This view contains 1,000 event statuses. You can also query `X$TRACE` to retrieve trace records directly from the SGA.
3. This statement is used to disable tracing for all events.
4. This is used to archive trace logs related to your process. Using this command, you can also flush trace buffers of all other processes. Using some undocumented parameters, it is also possible to continuously flush trace buffers to disk instead of having to manually flush them.
5. This example shows you how to use the hang analyzer throughout your RAC cluster. After you are connected to SQL*Plus as `SYSDBA`, you attach `oradebug` to your process. Then you set the instance list to all instances in your cluster, and execute the hang analyzer at level 3. This executes a hang analysis clusterwide. When executed, the command returns the name of the trace file containing the result of this analysis.

Note: ***Never*** execute these commands on a production system without explicit guidance from Oracle Support.

Hang Analysis with 11*g*R1: Example

```
$ sqlplus /nolog

SQL*Plus: Release 11.1.0.6.0 - Production on Fri Nov 2 22:51:24 2007

Copyright (c) 1982, 2007, Oracle.  All rights reserved.

SQL> set _prelim on
SQL> connect / as sysdba
Prelim connection established
SQL> oradebug setorapname reco
Oracle pid: 19, Unix process pid: 11381, image:
oracle@edcdr12p1.us.oracle.com (RECO)
SQL> oradebug dump hanganalyze_global 1
Statement processed.
SQL> oradebug dump systemstate_global 267
Statement processed.
SQL> oradebug setorapname diag
Oracle pid: 4, Unix process pid: 11347, image:
oracle@edcdr12p1.us.oracle.com (DIAG)
SQL> oradebug tracefile_name
/u01/app/oracle/diag/rdbms/racdb1/racdb11/trace/racdb11_diag_11347.trc
SQL>
```


Getting Hang Analysis with 11gR1 Example

Sometimes even connecting to the instance can hang. In this case there is a slightly unsafe (uses remote Oradebug commands) but useful mechanism to collect diagnostic dumps.

This mechanism leverages prelim connections that were introduced in Oracle Database 10.1. Prelim connections create an authenticated foreground that just attaches to the SGA. This is enough to execute Oradebug commands. At the same time, since no process state object, session state object, etc. are created, you are guaranteed to be able to connect to the hung instance.

The above example shows you such a case where you cannot access your instance normally.

The cluster-wide hang analysis dump will be in the local diag trace file. The per-instance system state dump will be in the diag trace file for each instance.

WARNING: REMOTE ORADEBUG COMMANDS ARE UNSAFE. ONLY RUN THESE COMMAND WITH EXPLICIT GUIDANCE FROM ORACLE SUPPORT.

NOTE: The example specifies the RECO process, not DIAG, because the system will use RECO to message DIAG to do the global hang analysis and system state dumps.

In addition, Oracle Database 11*g*R1 introduces the `V$WAIT_CHAINS` view (shown in Enterprise Manager) which displays information about blocked sessions. A wait chain comprises sessions that are blocked by one another. Each row represents a blocked and blocker session pair. If a wait chain is not a cyclical wait chain, then the last row for the chain does not have a blocker.

New 11*g*R1 Background Processes

Background Process Name	Short Description
ACMS	Ensure distributed SGA memory updates
DBRM	DataBase Resource Manager
DIAG	Performs diagnostic dumps
DIA0	Hang detection and deadlock resolution
EMNC	Event management and notifications coordination
GTXn	Provides XA support in RAC
KATE	ASM metafile proxy
MARK	Mark AU as stale when disk goes offline
PING	Measure interconnect latency
VKTM	Provides clock information in RAC

New 11*g*R1 Background Processes

- ACMSAtomic Controlfile to Memory Server per-instance process is an agent that contributes to ensuring a distributed SGA memory update is either globally committed on success or globally aborted in the event of a failure in an Oracle RAC environment.
- DBRM: DataBase Resource Manager process is responsible for setting resource plans and other resource manager related tasks.
- DIAG: DIAGnosability process performs diagnostic dumps and executes global oradebug commands.
- DIA0: DIAgnosability process 0 (only 0 is currently being used) is responsible for hang detection and deadlock resolution.
- EMNC: EMoN Coordinator (event monitor coordinator) is the background server process used for database event management and notifications.
- GTXj: Global Txn process j (0-j) processes provide transparent support for XA global transactions in an Oracle RAC environment. The database autotunes the number of these processes based on the workload of XA global transactions. Global transaction processes are only seen in an Oracle RAC environment.
- KATE: Konductor of ASM Temporary Errands performs proxy I/O to an ASM metafile when a disk goes offline.
- MARK: Mark Au for Resync Koordinator marks ASM allocation units as stale following a missed write to an offline disk.

New 11*g*R1 Background Processes (continued)

- PING: Interconnect latency measurement
- VKTM: Virtual Keeper of TiMe process: Is responsible for providing a wall-clock time (updated every second) and reference-time counter (updated every 20 ms and available only when running at elevated priority)

Appendix C
Practice Environment Scripts

Practice Environment Scripts

Included in the course files on the cluster nodes is a collection of scripts that can be used to set up the complete cluster database environment according to the practice instructions. In addition, the scripts can be used to catch up to any point in the practices if you encounter any unforeseen difficulties. The scripts are located in `/home/oracle/PCPS`. The directory is initially restricted to prevent inadvertent execution of the scripts. To remove the access restrictions from the PCPS directory, execute the `/home/oracle/viewdir.sh` script.

```
$ pwd
/home/oracle
$ ./viewdir.sh
$ cd PCPS
```

The scripts can be used to install and configure one component at a time in parallel with the practice instructions or the entire cluster database can be built with a single command including RMAN and Services configuration and OCR and Voting Disk backups.

Note: These scripts must be run on the first node from the PCPS directory. The first node can be determined by viewing the `/home/oracle/nodeinfo` file.

```
$ cat /home/oracle/nodeinfo
eg6608   ← First node
eg6610
```

The following actions can be performed:

`1. make network`
This command checks whether the RAC network setup is correct by pinging around and checking `/etc/hosts`. Note that pinging to the VIPs returns an error when the clusterware is not installed or if the VIPs are down for any reason such as disabling the nodeapps.

```
$ pwd
/home/oracle/PCPS

$ make network
./check_network
Checking public addresses.
Pinging eg6608: Received reply.
Pinging eg6610: Received reply.
Checking private addresses.
Pinging eg6608-priv: Received reply.
Pinging eg6610-priv: Received reply.
Checking VIP addresses.
Pinging eg6608-vip:
```

```
Received no reply - Expected if CRS is not yet installed!
Pinging eg6610-vip: Received no reply - Expected if CRS is not
yet installed!
```

2. make raw
This checks the raw-devices setup by reading from or writing to them. The purpose of this check is to verify the setup of the shared disks.
Note: Do not perform the "write" test after the clusterware is installed.

```
$ make raw
./check_raw
Type of test? (read/write): write
WARNING: This will overwrite your raw slices!!
Continue? (yes/no): yes
Writing to /dev/sdb1: OK.
Writing to /dev/sdb2: OK.
Writing to /dev/sdb5: OK.
Writing to /dev/sdb6: OK.
Writing to /dev/sdb7: OK.
Writing to /dev/sdc: OK.
Writing to /dev/sdd: OK.
Writing to /dev/sde: OK.
Writing to /dev/sdf: OK.
```

3. make ssh
This configures user equivalency between your RAC nodes.

```
$ make ssh
./configure_ssh
Setting up SSH user equivalency.
oracle@eg6610's password: oracle@eg6610's password:
Checking SSH user equivalency.
eg6608
eg6610
```

4. make crs
This silently installs the clusterware after prompting you for the cluster name and database name.

```
$ make crs
./install_crs
Enter the cluster name: clustorl01
Enter the database name: RDBA

Starting Oracle Universal Installer...

Checking Temp space: must be greater than 80 MB.   Actual 6573 MB
Passed
Checking swap space: must be greater than 150 MB.   Actual 4095
MB    Passed
```

```
Preparing to launch Oracle Universal Installer from
/tmp/OraInstall2008-08-14_12-36-32PM. Please wait ...
Oracle Universal Installer, Version 11.1.0.6.0 Production
Copyright (C) 1999, 2007, Oracle. All rights reserved.
You can find the log of this install session at:
 /u01/app/oraInventory/logs/installActions2008-08-14_12-36-
32PM.log
README.11gR1....................................... 100% Done.

...

Setting the permissions on OCR backup directory
Setting up Network socket directories
Oracle Cluster Registry configuration upgraded successfully
clscfg: EXISTING configuration version 4 detected.
clscfg: version 4 is 11 Release 1.
Successfully accumulated necessary OCR keys.
Using ports: CSS=49895 CRS=49896 EVMC=49898 and EVMR=49897.
node <nodenumber>: <nodename> <private interconnect name>
<hostname>
node 1: eg6608 eg6608-priv eg6608
node 2: eg6610 eg6610-priv eg6610
clscfg: Arguments check out successfully.

...


Done.
HA Resource                         Target     State
-----------                         ------     -----
ora.eg6608.gsd                      ONLINE     ONLINE on eg6608
ora.eg6608.ons                      ONLINE     ONLINE on eg6608
ora.eg6608.vip                      ONLINE     ONLINE on eg6608
ora.eg6610.gsd                      ONLINE     ONLINE on eg6610
ora.eg6610.ons                      ONLINE     ONLINE on eg6610
ora.eg6610.vip                      ONLINE     ONLINE on eg6610
```


5. make asmhome

This silently installs the Oracle database software for the ASM home.

```
$ make asmhome
./install_asmhome
Starting Oracle Universal Installer...

Checking Temp space: must be greater than 80 MB.   Actual 6572 MB
Passed
Checking swap space: must be greater than 150 MB.   Actual 4095
MB    Passed
Preparing to launch Oracle Universal Installer from
/tmp/OraInstall2008-08-14_12-44-38PM. Please wait ...Oracle
Universal Installer, Version 11.1.0.6.0 Production
Copyright (C) 1999, 2007, Oracle. All rights reserved.

You can find the log of this install session at:
```

```
 /u01/app/oraInventory/logs/installActions2008-08-14_12-44-
38PM.log
........................................... 100% Done.

...

The installation of Oracle Database 11g was successful.
Please check '/u01/app/oraInventory/logs/silentInstall2008-08-
14_12-44-38PM.log' for more details.
running root.sh
```

6. make asm

This silently installs the Oracle software for the ASM home, configures ASM instances, and creates the +DATA and +FRA disk groups.

```
$ make asm
./create_asm
creating ASM instances and ASM diskgroups.
Look at the log file
"/u01/app/oracle/cfgtoollogs/dbca/silent.log" for further
details.
HA Resource                         Target     State
-----------                         ------     -----
ora.eg6608.ASM1.asm                 ONLINE     ONLINE on eg6608
ora.eg6610.ASM2.asm                 ONLINE     ONLINE on eg6610
```

7. make dbhome

```
$ make dbhome
./install_dbhome
Starting Oracle Universal Installer...

Checking Temp space: must be greater than 80 MB.   Actual 6571 MB
Passed
Checking swap space: must be greater than 150 MB.   Actual 4095
MB    Passed
Preparing to launch Oracle Universal Installer from
/tmp/OraInstall2008-08-14_01-05-23PM. Please wait ...Oracle
Universal Installer, Version 11.1.0.6.0 Production
Copyright (C) 1999, 2007, Oracle. All rights reserved.

You can find the log of this install session at:
 /u01/app/oraInventory/logs/installActions2008-08-14_01-05-
23PM.log
.............................................. 100% Done.

...

End of install phases.(Thu Aug 14 14:15:29 EDT 2008)
WARNING:
The following configuration scripts need to be executed as the
"root" user in each cluster node.
#!/bin/sh
```

```
#Root script to run
/u01/app/oracle/product/11.1.0/db_1/root.sh #On nodes
eg6608,eg6610
To execute the configuration scripts:
    1. Open a terminal window
    2. Log in as "root"
    3. Run the scripts in each cluster node

The installation of Oracle Database 11g was successful.
Please check '/u01/app/oraInventory/logs/silentInstall2008-08-
14_01-05-23PM.log' for more details.
running root.sh
```

8. `make db`

This silently creates the RAC database and installs the sample schemas. If you did not install the clusterware by using `make crs`, you will be prompted for the database name here.

```
$ make db
./create_db
creating the RDBA database.
Copying database files
...
Creating and starting Oracle instance
...
Creating cluster database views
...
Completing Database Creation
...
Look at the log file "Look at the log file
"/u01/app/oracle/cfgtoollogs/dbca/RDBA/RDBA.log" for further
details.

...
HA Resource                            Target     State
-----------                            ------     -----
ora.RDBA.RDBA1.inst                    ONLINE     ONLINE on eg6608
ora.RDBA.RDBA2.inst                    ONLINE     ONLINE on eg6610
ora.RDBA.db                            ONLINE     ONLINE on eg6610
```

9. `make rman`

This configures the cluster database for archivelog mode and updates the necessary RMAN persistent configuration parameters.

```
$ make rman
./configure_rman
Stopping the RDBA database instances.
ORACLE instance started.
Database mounted.
...
Database altered.

System altered.
```

```
Database closed.
Database dismounted.
ORACLE instance shut down.
Starting the RDBA database instances.

Recovery Manager: Release 11.1.0.6.0 - Production on Thu Aug 14
13:26:46 2008

Copyright (c) 1982, 2007, Oracle.  All rights reserved.

connected to target database: RDBA (DBID=871507933)
RMAN>
using target database control file instead of recovery catalog
new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored
RMAN>

Recovery Manager complete.
HA Resource                            Target     State
-----------                            ------     -----
ora.RDBA.RDBA1.inst                    ONLINE     ONLINE on eg6608
ora.RDBA.RDBA2.inst                    ONLINE     ONLINE on eg6610
ora.RDBA.db                            ONLINE     ONLINE on eg6610
```

10. `make service`

This creates a cluster service called SERV1, activates the service, and configures the `tnsnames.ora` files on both cluster nodes to support the service.

```
$ make service
./configure_service
Creating the SERV1 Service.
Starting the SERV1 Service.
Configuring the SERV1 Service tnsnames.
HA Resource                            Target     State
-----------                            ------     -----
ora.RDBA.SERV1.RDBA1.srv               ONLINE     ONLINE on eg6608
ora.RDBA.SERV1.cs                      ONLINE     ONLINE on eg6608
```

11. `make ocr`

This mirrors the OCR file.

```
$ make ocr
./configure_ocr_mirror
Adding /dev/sdb2 as the OCR mirror.
Status of Oracle Cluster Registry is as follows :
 Version                  :          2
 Total space (kbytes)     :     616652
 Used space (kbytes)      :       4300
 Available space (kbytes) :     612352
 ID                       : 1473866508
```

```
 Device/File Name         :  /dev/sdb1
                            Device/File integrity check succeeded
 Device/File Name         :  /dev/sdb2
                            Device/File integrity check succeeded

 Cluster registry integrity check succeeded
```

12. make vdisk

This mirrors the voting disk file.

```
$ make vdisk
./configure_vdisk_mirror
Now formatting voting disk: /dev/sdb6.
Successful addition of voting disk /dev/sdb6.
Now formatting voting disk: /dev/sdb7.
Successful addition of voting disk /dev/sdb7.
 0.     0    /dev/sdb5
 1.     0    /dev/sdb6
 2.     0    /dev/sdb7
Located 3 voting disk(s).
```

13. make bashrc

This configures the `.bashrc` file on both nodes for the oracle account to set shell variables and to create useful functions to switch homes as required. These functions include:

- `crs`: Set variables for the clusterware home.
- `asm`: Set variables for the ASM home.
- `db`: Set variables for the cluster database home.

```
$ make bashrc
./configure_bashrc
Configuring .bashrc
```

14. make all

This executes steps 3 through 13 and 16.

```
$ make all
./configure_ssh
Setting up SSH user equivalency.
oracle@eg6610's password: oracle@eg6610's password:
Checking SSH user equivalency.
eg6608
eg6610
./install_crs
Enter the cluster name: clustor101
Enter the database name: RDBA
Starting Oracle Universal Installer...
```

```
Checking installer requirements...

...


located 3 votedisk(s).
./configure_rlwrap
Compiling and installing rlwrap.
./configure_bashrc
Configuring .bashrc
```

15. `make progress`

This checks and reports on which components have already been installed or configured on this cluster.

```
$ make progress
./check_progress
SSH user equivalency has been configured.
Oracle Clusterware is installed.
The ASM instances are created.
The database software is installed.
The RDBA database is created.
The SERV1 Service has been created.
The OCR is mirrored.
The voting disk is mirrored.
RMAN has been configured.
rlwrap is installed.
.bashrc is configured.
```

16. `make rlwrap`

This compiles and installs `rlwrap` to give SQL*Plus and RMAN history and command-line editing functionality by typing:

- `rlwrap sqlplus`
- `rlwrap rman`

```
$ make rlwrap
./configure_rlwrap
Compiling and installing rlwrap.
```

17. `make clean`

This disables the clusterware on both nodes and reboots both nodes. Wait for the reboot to complete. The next time this suite is used to install something, it will automatically finish the cleanup. To force the cleanup to finish immediately, run `make clean` again. After this is finished, the environment is refreshed.

```
$ cd /home/oracle/PCPS
$ make clean
./util_cleanup
About to remove EVERYTHING from this cluster!
Continue? (yes/no): yes
Setting up SSH user equivalency.
oracle@eg6610's password:
Checking SSH user equivalency.
eg6608
eg6610
Pseudo-terminal will not be allocated because stdin is not a
terminal.

Rebooting eg6610
Rebooting eg6608


...

$ cd PCPS
$ make clean <After node 1 reboots, run make clean a second time>
./util_cleanup
continuing cleanup procedure.
Done.
```