Oracle 11*g*: RAC and Grid Infrastructure Administration Accelerated

Volume III - Student Guide

D60488GC11

Edition 1.1

September 2010

D69313

ORACLE®

Authors

James Womack

James Spiller

Technical Contributors

David Brower
Mark Fuller
Barb Lundhild
Mike Leatherman
S. Matt Taylor
Jean-Francois Verrier
Rick Wessman

Technical Reviewers

Christopher Andrews
Christian Bauwens
Jonathan Creighton
Michael Cebulla
Al Flournoy
Andy Fortunak
Joel Goodman
Michael Hazel
Pete Jones
Jerry Lee
Markus Michalewicz
Peter Sharman
Ranbir Singh
Richard Strohm
Janet Stern
Linda Smalley
Branislav Valny
Doug Williams

Graphics

Satish Bettegowda

Editors

Nita Pavitran
Arijit Ghosh
Aju Kumar
Daniel Milne

Publishers

Shaik Mahaboob Basha
Nita Brozowski
Jayanthy Keshavamurthy

Contents

8 Administering ASM Disk Groups

9 Administering ASM Files, Directories, and Templates

Appendix C: High Availability of Connections

Appendix D: Oracle RAC One Node

Appendix E: Cloning Oracle Clusterware

Appendix A
Practices and Solutions

Table of Contents

Practices for Lesson 1

The practice for Lesson 1 is a discovery of the classroom environment.

Practice 1-1: Discovering Cluster Environment

In this practice, you will explore the classroom and cluster environment in preparation for cluster installation and management.

1) In an Oracle cluster, there are pieces of information that are important for a successful installation. Several of these are unique to each cluster. In order to make this practice document generic to allow every participant to use the same document and specific enough that each participant can be successful, an environment file has been created called `/home/oracle/labs/st_env.sh`. This file holds the information that is specific to your cluster. This script is not a script supplied with Oracle software. This script was developed specifically for this class.

2) Open a VNC session as the oracle OS user on your first node. (The instructor will give this node name.) Click the VNC icon on the desktop. In the VNC Viewer:Connection Details box, enter the name of your first node with ':1' as a suffix. In the VNC Viewer:Authentication box, enter the password for the `oracle` user. The password is `0racle.` The first character is a zero, not a capital “O”. This is also the root password.

3) In the VNC session, open a terminal window. Right-click in the VNC session window, and select Open Terminal from the shortcut menu.

4) Find and list the `/home/oracle/labs/st_env.sh` script.

```
$ cat /home/oracle/labs/st_env.sh
#Common environment Variables
NODELIST=`cat /home/oracle/nodeinfo`
NODE1=host01
NODE2=host02
NODE3=host03
CLUSTER_NAME=cluster01
GNS_NAME=host01-gns
# Hardware environment definitions
export ST_SOFTWARE_STAGE_GRID=/stage/clusterware/Disk1
export ST_SOFTWARE_STAGE_DB=/stage/database/Disk1
export ST_SOFTWARE_STAGE_DEINST=/stage/deinstall
export ST_NODE1=$NODE1
export ST_NODE2=$NODE2
export ST_NODE3=$NODE3
export ST_CLUSTER_NAME=$CLUSTER_NAME
export ST_NODE3_VIP=${NODE3}-vip
export ST_NODE_DOMAIN=example.com
export ST_NODE_LIST=${NODE1},${NODE2}
export ST_NODE_LIST2=${NODE1},${NODE2},${NODE3}
export ST_GNS_NAME=${GNS_NAME}.${ST_NODE_DOMAIN}
# Software Ownership definitions
export ST_GRID_OWNER=grid
export ST_GRID_OWNER_HOME=/home/grid
export ST_DB1_OWNER=oracle
export ST_DB1_OWNER_HOME=/home/oracle
# Grid Infrastructure definitions
export ST_ASM_INSTANCE1=+ASM1
```

```
export ST_ASM_INSTANCE2=+ASM2
export ST_ASM_INSTANCE3=+ASM3
export ST_ASM_HOME=/u01/app/11.2.0/grid
# Database 1 definitions
export ST_DB1_NAME=orcl
export ST_DB1_INSTANCE1=orcl1
export ST_DB1_INSTANCE2=orcl2
export ST_DB1_INSTANCE3=orcl3
export
ST_DB1_HOME=/u01/app/oracle/acfsmount/11.2.0/sharedhome/dbhome
_1
# Database 2 definitions (standby, rcat for example)
export ST_DB2_NAME=orcl
export ST_DB2_INSTANCE1=orcl1
export ST_DB2_INSTANCE2=orcl2
export ST_DB2_INSTANCE3=orcl3
export ST_DB2_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
# Database 3 definitions
export ST_DB3_NAME=
export ST_DB3_INSTANCE1=
```

5) The environment for this class can change. The node names can be different for every cluster and the cluster names can be different for each setup inside the class. Discover and record values for your cluster.
   Note: The environment variables created for this class begin with `ST_` and are referenced by prefixing the variable with `$`. For example, the `ST_NODE1` variable is referenced with `$ST_NODE1`.

   a) Set the environment variables found in `st_env.sh` for your terminal session. To set the variables in a session from a script, you must "source" the script—that is, cause it to run in the same shell. This is done by using the "`.`" command as shown in the code example.

```
$ . /home/oracle/labs/st_env.sh
```

   b) What are the short names for the nodes of your cluster?
   These nodes are called host01, host02, and host03 in the practices that follow. You will substitute your node names or use the environment variable to refer to the node. For example, `$ST_NODE2` refers to your second node, and in the code output, is shown as host02. Your nodes will be named something else. Record the names of your nodes.

   First node (`$ST_NODE1`) ________________

   Second node (`$ST_NODE2`) _________________

   Third node (`$ST_NODE3`) _________________

```
$ echo $ST_NODE1
host01              << Your node names may be different
$ echo $ST_NODE2
```

Practice 1-1: Discovering Cluster Environment (continued)

```
host02
$ echo $ST_NODE3
host03
```

6) What is the name of your cluster?

   Cluster name ($ST_CLUSTER_NAME) ________________

```
$ echo $ST_CLUSTER_NAME
cluster01          << Your cluster name mayl be different
```

7) What is the node domain?

   Node domain ($ST_NODE_DOMAIN) _____________________

```
$ echo $ST_NODE_DOMAIN
example.com
```

8) What is the name of your GNS ($ST_GNS_NAME)__________________________

```
$ echo $ST_GNS_NAME
cluster01-gns.example.com   << Your gns name may be different
```

9) Verify that the environment variables for your cluster have been set in your environment. Use the env | grep ST_ command.

```
$ env | grep ST_
ST_DB1_NAME=orcl
ST_DB1_OWNER_HOME=/home/oracle
ST_SOFTWARE_STAGE_GRID=/stage/clusterware/Disk1
ST_CLUSTER_NAME=cluster01
ST_NODE1=host01
ST_NODE2=host02
ST_NODE3=host03
ST_GRID_OWNER=grid
ST_ASM_HOME=/u01/app/11.2.0/grid
ST_NODE_LIST=host01,host02
ST_GRID_OWNER_HOME=/home/grid
ST_DB2_INSTANCE2=orcl2
ST_GNS_NAME=cluster01-gns.example.com
ST_DB2_INSTANCE3=orcl3
ST_SOFTWARE_STAGE_DB=/stage/database/Disk1
ST_DB2_NAME=orcl
ST_DB2_INSTANCE1=orcl1
ST_DB1_INSTANCE1=orcl1
ST_DB1_INSTANCE3=orcl3
ST_DB1_INSTANCE2=orcl2
ST_DB1_OWNER=oracle
ST_DB2_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
ST_NODE3_VIP=host03-vip
ST_DB3_NAME=
ST_ASM_INSTANCE3=+ASM3
ST_ASM_INSTANCE2=+ASM2
ST_ASM_INSTANCE1=+ASM1
```

Practice 1-1: Discovering Cluster Environment (continued)

```
ST_DB3_INSTANCE1=
ST_NODE_LIST2=host01,host02,host03
ST_SOFTWARE_STAGE_DEINST=/stage/deinstall
ST_DB1_HOME=/u01/app/oracle/acfsmount/11.2.0/sharedhome/dbhome
_1
ST_NODE_DOMAIN=example.com
```

Practices for Lesson 2

In the practices for this lesson, you will perform the tasks that are prerequisites to successfully installing Oracle Grid Infrastructure. You will configure ASMLib to manage your shared disks and finally, you will install and verify Oracle Grid Infrastructure 11.2. In addition, you will create ASM disk groups and an ACFS file system by using ASMCA.

Practice 2-1: Performing Preinstallation Tasks for Oracle Grid Infrastructure

In this practice, you perform various tasks that are required before installing Oracle Grid Infrastructure. These tasks include:

- Setting up required groups and users
- Creating base directory
- Configuring Network Time Protocol (NTPD)
- Setting shell limits
- Editing profile entries
- Configuring ASMLib and shared disks

1) From a graphical terminal session, make sure that the groups `asmadmin`, `asmdba`, and `asmoper` exist (`cat /etc/group`). Make sure that the user `grid` exists with the primary group of oinstall and the secondary groups of `asmadmin`, `asmdba`, and `asmoper`. Make sure that the `oracle` user's primary group is oinstall with secondary groups of `dba`, `oper`, and `asmdba`. Running the script `/home/oracle/labs/less_02/usrgrp.sh` as the `root` user will complete all these tasks. **Perform this step on all three of your nodes**.

```
$ cat /home/oracle/labs/less_02/usrgrp.sh
#!/bin/bash

groupadd -g 503 oper
groupadd -g 505 asmdba
groupadd -g 506 asmoper
groupadd -g 504 asmadmin

grep -q grid /etc/passwd
UserGridExists=$?
if [[ $UserGridExists == 0 ]]; then
usermod -g oinstall -G asmoper,asmdba,asmadmin grid
else
useradd -u 502 -g oinstall -G asmoper,asmdba,asmadmin grid
fi
echo oracle | passwd --stdin grid
usermod -g oinstall -G dba,oper,asmdba oracle

$ su -
Password: 0racle << password is not displayed

# . /home/oracle/labs/st_env.sh

   <<< On node 1 >>>

# /home/oracle/labs/less_02/usrgrp.sh
Creating mailbox file: File exists
Changing password for user grid.
passwd: all authentication tokens updated successfully.

   <<< On node 2 >>>
```

Practice 2-1: Performing Preinstallation Tasks for Oracle Grid Infrastructure (continued)

```
# ssh $ST_NODE2 /home/oracle/labs/less_02/usrgrp.sh
The authenticity of host 'gr7273 (10.196.180.73)' can't be
established.
RSA key fingerprint is
4a:8c:b8:48:51:04:2e:60:e4:f4:e6:39:13:39:48:8f.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'gr7273,10.196.180.73' (RSA) to the
list of known hosts.
root's password: 0racle << password is not displayed
Creating mailbox file: File exists
Changing password for user grid.
passwd: all authentication tokens updated successfully.
   <<< On node 3 >>>

# ssh $ST_NODE3 /home/oracle/labs/less_02/usrgrp.sh
The authenticity of host 'host03 (10.196.180.14)' can't be
established.
RSA key fingerprint is
4a:8c:b8:48:51:04:2e:60:e4:f4:e6:39:13:39:48:8f.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'host03,10.196.180.14' (RSA) to the
list of known hosts.
root's password: 0racle << password is not displayed
Creating mailbox file: File exists
Changing password for user grid.
passwd: all authentication tokens updated successfully.
```

2) As the `root` user, create the `oracle` and `grid` user base directories. **Perform this step on all three of your nodes**.

```
# mkdir -p /u01/app/grid
# chown -R grid:oinstall /u01/app
# chmod -R 775  /u01/app/grid
# mkdir -p /u01/app/oracle
# chown -R oracle:oinstall /u01/app/oracle
```

3) View the `/etc/sysconfig/ntpd` file and confirm that the –x option is specified to address slewing. If necessary, change the file, and then restart the `ntpd` service with the `service ntpd restart` command. **Perform this step on all three of your nodes**.

```
[root]# cat /etc/sysconfig/ntpd
# Drop root to id 'ntp:ntp' by default.
OPTIONS="-x -u ntp:ntp -p /var/run/ntpd.pid"
```

Practice 2-1: Performing Preinstallation Tasks for Oracle Grid Infrastructure (continued)

```
# Set to 'yes' to sync hw clock after successful ntpdate
SYNC_HWCLOCK=no

# Additional options for ntpdate
NTPDATE_OPTIONS=""
[root]#
```

4) As the `root` user, start the local naming cache daemon on all three nodes with the `service nscd start` command. To make sure `nscd` starts at reboot, execute the `chkconfig nscd` command. **Perform these steps on all three of your nodes.**

```
[root]# service nscd start
Starting nscd: [  OK  ]
[root]# chkconfig nscd

[root]# ssh $ST_NODE2 service nscd start
root's password: 0racle << password is not displayed
Starting nscd: [  OK  ]
[root]# ssh $ST_NODE2 chkconfig nscd

[root]# ssh $ST_NODE3 service nscd start
root's password: 0racle << password is not displayed
Starting nscd: [  OK  ]
[root]# ssh $ST_NODE3 chkconfig nscd
```

5) As the `root` user, run the `/home/oracle/labs/less_02/limits.sh` script. This script replaces the profile for the `oracle` and `grid` users and replaces `/etc/profile`. It replaces the `/etc/security/limits.conf` file with a new one with entries for `oracle` and `grid`. Cat the `/home/oracle/labs/less_02/bash_profile` and `/home/oracle/labs/less_02/profile` to view the new files. **Perform this step on all three of your nodes**.

```
# cat /home/oracle/labs/less_02/bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin
export PATH
```

Practice 2-1: Performing Preinstallation Tasks for Oracle Grid Infrastructure (continued)

```
umask 022

# cat /home/oracle/labs/less_02/profile
# /etc/profile

# System wide environment and startup programs, for login
setup
# Functions and aliases go in /etc/bashrc

pathmunge () {
        if ! echo $PATH | /bin/egrep -q "(^|:)$1($|:)" ; then
           if [ "$2" = "after" ] ; then
              PATH=$PATH:$1
           else
              PATH=$1:$PATH
           fi
        fi
}

# ksh workaround
if [ -z "$EUID" -a -x /usr/bin/id ]; then
        EUID=`id -u`
        UID=`id -ru`
fi

# Path manipulation
if [ "$EUID" = "0" ]; then
        pathmunge /sbin
        pathmunge /usr/sbin
        pathmunge /usr/local/sbin
fi

# No core files by default
ulimit -S -c 0 > /dev/null 2>&1

if [ -x /usr/bin/id ]; then
        USER="`id -un`"
        LOGNAME=$USER
        MAIL="/var/spool/mail/$USER"
fi

HOSTNAME=`/bin/hostname`
HISTSIZE=1000

if [ -z "$INPUTRC" -a ! -f "$HOME/.inputrc" ]; then
    INPUTRC=/etc/inputrc
fi
```

Practice 2-1: Performing Preinstallation Tasks for Oracle Grid Infrastructure (continued)

```
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE INPUTRC

for i in /etc/profile.d/*.sh ; do
    if [ -r "$i" ]; then
        . $i
    fi
done

if [ $USER = "oracle" ] || [ $USER = "grid" ]; then
        umask 022
        if [ $SHELL = "/bin/ksh" ]; then
             ulimit -p 16384
             ulimit -n 65536
        else
             ulimit -u 16384 -n 65536
        fi
fi

unset i
unset pathmunge

# cat /home/oracle/labs/less_02/limits.conf

#        - priority - the priority to run user process with
#        - locks - max number of file locks the user can hold
#        - sigpending - max number of pending signals
#        - msgqueue - max memory used by POSIX message queues
(bytes)
#        - nice - max nice priority allowed to raise to
#        - rtprio - max realtime priority
#<domain>      <type>  <item>         <value>

#*               soft    core            0
#*               hard    rss             10000
#@student        hard    nproc           20
#@faculty        soft    nproc           20
#@faculty        hard    nproc           50
#ftp             hard    nproc           0
#@student        -       maxlogins       4
# End of file
oracle  soft  nofile     131072
oracle  hard  nofile     131072
oracle  soft  nproc 131072
oracle  hard  nproc 131072
oracle  soft  core  unlimited
oracle  hard  core  unlimited
oracle  soft  memlock    3500000
oracle  hard  memlock    3500000
grid    soft    nofile  131072
```

Practice 2-1: Performing Preinstallation Tasks for Oracle Grid Infrastructure (continued)

```
grid    hard    nofile  131072
grid    soft    nproc   131072
grid    hard    nproc   131072
grid    soft    core    unlimited
grid    hard    core    unlimited
grid    soft    memlock 3500000
grid    hard    memlock 3500000
# Recommended stack hard limit 32MB for oracle installations
# oracle   hard   stack    32768

# cat /home/oracle/labs/less_02/limits.sh
cp /home/oracle/labs/less_02/profile /etc/profile
cp /home/oracle/labs/less_02/bash_profile
/home/oracle/.bash_profile
cp /home/oracle/labs/less_02/bash_profile
/home/grid/.bash_profile
cp /home/oracle/labs/less_02/limits.conf
/etc/security/limits.conf

   <<< On Node 1 >>>
# /home/oracle/labs/less_02/limits.sh

   <<< On Node 2 >>>
# ssh $ST_NODE2 /home/oracle/labs/less_02/limits.sh
root@gr7213's password: 0racle << password is not displayed

   <<< On Node 3 >>>>
# ssh $ST_NODE3 /home/oracle/labs/less_02/limits.sh
root@host03's password: 0racle << password is not displayed

#
```

6) As `root`, execute the `oracleasm configure -i` command to configure the Oracle ASM library driver. The owner should be `grid` and the group should be `asmadmin`. Make sure that the driver loads and scans disks on boot. **Perform this step on all three of your nodes**.

```
   <<< On Node 1 >>>
# oracleasm configure -i
Configuring the Oracle ASM library driver.

This will configure the on-boot properties of the Oracle ASM
library
driver.  The following questions will determine whether the
driver is
loaded on boot and what permissions it will have.  The current
values
```

```
will be shown in brackets ('[]').  Hitting <ENTER> without
typing an
answer will keep that current value.  Ctrl-C will abort.

Default user to own the driver interface []: grid
Default group to own the driver interface []: asmadmin
Start Oracle ASM library driver on boot (y/n) [n]: y
Scan for Oracle ASM disks on boot (y/n) [y]: y
Writing Oracle ASM library driver configuration: done

  <<< On Node 2 >>>
# ssh $ST_NODE2 oracleasm configure -i

 … output not shown

  <<< On Node 3 >>>
# ssh $ST_NODE3 oracleasm configure -i

 … output not shown
```

7) **(On the first node only)** Create the ASM disks needed for the practices. The `/home/oracle/labs/less_02/createdisk.sh` script has been provided to do this for you. Look at the script, and then execute it as the `root` user. **Perform this step on the first node only.**

```
# cat /home/oracle/labs/less_02/createdisk.sh
oracleasm init
oracleasm createdisk ASMDISK01 /dev/sda1
oracleasm createdisk ASMDISK02 /dev/sda2
oracleasm createdisk ASMDISK03 /dev/sda3
oracleasm createdisk ASMDISK04 /dev/sda5
oracleasm createdisk ASMDISK05 /dev/sda6
oracleasm createdisk ASMDISK06 /dev/sda7
oracleasm createdisk ASMDISK07 /dev/sda8
oracleasm createdisk ASMDISK08 /dev/sda9
oracleasm createdisk ASMDISK09 /dev/sda10
oracleasm createdisk ASMDISK10 /dev/sda11
oracleasm createdisk ASMDISK11 /dev/sdb1
oracleasm createdisk ASMDISK12 /dev/sdb2
oracleasm createdisk ASMDISK13 /dev/sdb3
oracleasm createdisk ASMDISK14 /dev/sdb5


# /home/oracle/labs/less_02/createdisk.sh
Creating /dev/oracleasm mount point: /dev/oracleasm
Loading module "oracleasm": oracleasm
Mounting ASMlib driver filesystem: /dev/oracleasm
```

Practice 2-1: Performing Preinstallation Tasks for Oracle Grid Infrastructure (continued)

```
Writing disk header: done
Instantiating disk: done
Writing disk header: done
Instantiating disk: done
Writing disk header: done
Instantiating disk: done
Writing disk header: done
Instantiating disk: done
Writing disk header: done
Instantiating disk: done
Writing disk header: done
Instantiating disk: done
Writing disk header: done
Instantiating disk: done
Writing disk header: done
Instantiating disk: done
Writing disk header: done
Instantiating disk: done
Writing disk header: done
Instantiating disk: done
Writing disk header: done
Instantiating disk: done
Writing disk header: done
Instantiating disk: done
Writing disk header: done
Instantiating disk: done
Writing disk header: done
Instantiating disk: done

#
```

8) As the `root` user, scan the disks to make sure that they are available with the `oracleasm scandisks` command. Perform an `oracleasm listdisks` command to make sure all the disks have been configured. **Perform this step on all three of your nodes.**

```
# oracleasm exit

# oracleasm init
Creating /dev/oracleasm mount point: /dev/oracleasm
Loading module "oracleasm": oracleasm
Mounting ASMlib driver filesystem: /dev/oracleasm

# oracleasm scandisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
Instantiating disk "ASMDISK01"
```

Practice 2-1: Performing Preinstallation Tasks for Oracle Grid Infrastructure (continued)

```
Instantiating disk "ASMDISK02"
Instantiating disk "ASMDISK03"
Instantiating disk "ASMDISK04"
Instantiating disk "ASMDISK05"
Instantiating disk "ASMDISK06"
Instantiating disk "ASMDISK07"
Instantiating disk "ASMDISK08"
Instantiating disk "ASMDISK09"
Instantiating disk "ASMDISK10"
Instantiating disk "ASMDISK11"
Instantiating disk "ASMDISK12"
Instantiating disk "ASMDISK13"
Instantiating disk "ASMDISK14"

# oracleasm listdisks
ASMDISK01
ASMDISK02
ASMDISK03
ASMDISK04
ASMDISK05
ASMDISK06
ASMDISK07
ASMDISK08
ASMDISK09
ASMDISK10
ASMDISK12
ASMDISK13
ASMDISK14

#
```

Practice 2-2: Installing Oracle Grid Infrastructure

In this practice, you install Oracle Grid Infrastructure.

1) Use the Oracle Universal Installer (runInstaller) to install Oracle Grid Infrastructure. You need to know your cluster name and GNS name and address. You also need to know your cluster subdomain. Your instructor will provide this information. **YOU MUST USE THE NAMES ASSIGNED TO YOU BY YOUR INSTRUCTOR. Failure to use the proper names will result in a failed or inoperative cluster installation.**

For this example, assume the following:

- Your assigned cluster nodes are **host01**, **host02**, and **host03**.
- Your instructor has assigned the cluster name **cluster01** and GNS name **cluster01-gns.example.com** with an IP address of **192.0.2.155**.
- Your assigned cluster subdomain is **cluster01.example.com**.

These values are for illustrational purposes only. PLEASE USE THE VALUES GIVEN TO YOU BY YOUR INSTRUCTOR.

a) As the `grid` user, start a VNC session on any available port for the `grid` user. Change the Xstartup file for vnc. Before starting the installation, make sure that the DNS server can resolve your GNS name. The GNS name in the classroom environment is available with the `echo $ST_GNS_NAME` command. Record the IP address you get for `$ST_GNS_NAME` ___________________

```
[oracle]$ su - grid
Password: 0racle  << password not displayed
[grid]$ id
uid=502(grid) gid=501(oinstall)
groups=501(oinstall),504(asmadmin),505(asmdba),506(asmoper)

[grid]$ mkdir .vnc

[grid]$ cp /home/oracle/.vnc/xstartup /home/grid/.vnc/xstartup

[grid]$ vncserver :3  <= Starting VNC on port 5803

You will require a password to access your desktops.

Password: oracle  << password not displayed
Verify: oracle  << password not displayed
xauth:  creating new authority file /home/grid/.Xauthority

New 'host03.example.com:3 (grid)' desktop is
host03.example.com:3

Starting applications specified in /home/grid/.vnc/xstartup
Log file is /home/grid/.vnc/host03.example.com:3.log
```

```
[grid]$ . /home/oracle/labs/st_env.sh
[grid]$ echo $ST_GNS_NAME
cluster01-gns.example.com
[grid]$ nslookup $ST_GNS_NAME
Server:         192.0.2.14
Address:        192.0.2.14#53

Name:   cluster01-gns.example.com
Address: 192.0.2.155
```

b) To start a VNC viewer session, click the VNC icon on your desktop, and connect to the `grid` user's session. In keeping with the example above, you would enter *`host01`*`:3`. Open a terminal window and change directory to the staged software location provided by your instructor and start the OUI by executing the `runInstaller` command from the `$ST_SOFTWARE_STAGE_GRID` directory.

```
[grid]$ id
uid=502(grid) gid=501(oinstall) groups=501(oinstall) ...
[grid]$ . /home/oracle/labs/st_env.sh
[grid]$ cd $ST_SOFTWARE_STAGE_GRID

[grid]$ ./runInstaller
```

c) On the Select Installation Option page, select the "Install and Configure Grid Infrastructure for a Cluster" option and click Next.

d) On the Select Installation Type page, select Advanced Installation and click Next.

e) On the Product Languages page, select all languages and click Next.

f) The "Grid Plug and Play Information" page appears next. Make sure that the Configure GNS check box is selected. Input the proper data carefully. **DO NOT GUESS** here. If you are unsure of a value, **PLEASE ASK YOUR INSTRUCTOR**. You must input:

- Cluster Name
- SCAN Name
- SCAN Port
- GNS Subdomain
- GNS Address

The values given here are based on the example data presented earlier. To continue with the example, the following values would be assigned:

- Cluster Name: **cluster01** *(Yours may be different Use the value recorded in practice 1 step 6.)*

- SCAN Name: **cluster01-scan.cluster01.example.com** *(Yours may be different.)*
- SCAN Port: **1521** (DEFAULT)
- GNS Sub Domain: **cluster01.example.com** *(Yours may be different. The subdomain is $ST_CLUSTER_NAME.$ST_NODE_DOMAIN.)*
- GNS Address: **192.0.2.155** *(Yours may be different. Use the value you found in practice 2-2 step 1a.)*

Hint: If you enter the cluster name (for example, cluster01), and then enter GNS Sub Domain (for example, cluster01.example.com), the SCAN name will autofill correctly. Leave SCAN Port to default to 1521 and enter the IP address for your GNS. Verify all data entered on this page, and then click Next.

g) On the Cluster Node Information page, you add **your second node only**. **DO NOT for any reason install to all three nodes.** Click the Add button and enter the fully qualified name of your second node into the box and click OK. Your second node should appear in the window under your first node. The Virtual IP name values for both nodes will be AUTO. Click the SSH Connectivity button. Enter the `grid` password, which is `0racle`. Click the Setup button. A dialog box stating that you have successfully established passwordless SSH connectivity appears. Click OK to close the dialog box. Click Next to continue.

h) On the Specify Network Usage page, you must configure the correct interface types for the listed network interface. Your instructor will indicate the proper usage for each of your interfaces. Again, **DO NOT GUESS**. The systems that were used to develop the course had four interfaces: eth0 (storage network), eth1 (storage network), eth2 (private network), and eth3 (public network). Using that example, eth0 and eth1 would be marked “Do Not Use,” eth2 would be marked Private, and eth3 would be marked Public. Again, this is only an example. Check with your instructor for proper network interface usage. When you have correctly assigned the interface types, click Next to continue.

i) On the Storage Option Information page, select Automatic Storage Management (ASM) and click Next.

j) On the Create ASM Disk Group page, make sure that Disk Group Name is DATA and Redundancy is Normal. In the Add Disks region, select ORCL:ASMDISK01 ORCL:ASMDISK02, ORCL:ASMDISK03, and ORCL:ASMDISK04. Click Next.

k) On the ASM Passwords page, click the Use Same Password for these accounts button. In the Specify Password field, enter **`oracle_4U`** and confirm it in the Confirm Password field. Click Next to continue.

l) Select the “Do not use Intelligent Platform Management Interface (IPMI)” option on the Failure Isolation page and click Next to continue.

m) On the Privileged Operating System Groups page, select “asmdba” for the ASM Database Administrator (OSDBA) group, “asmoper” for the ASM Instance Administration Operator (OSOPER) group, and “asmadmin” for the ASM Instance Administrator (OSASM) group. Click Next to continue.

n) On the Specify Installation Location page, make sure that Oracle Base is `/u01/app/grid` and Software Location is `/u01/app/11.2.0/grid`. Click Next.

o) On the Create Inventory page, Inventory Directory should be `/u01/app/oraInventory` and the oraInventory Group Name should be oinstall. Click Next.

p) On the Perform System Prerequisites page, the Installer checks whether all the systems involved in the installation meet the minimum system requirements for that platform. If the check is successful, click Next. If any deficiencies are found, click the “Fix & Check Again” button. The Execute Fixup Script dialog box appears. You are instructed to execute a script as `root` on each node involved in the installation— two nodes in this case. Open a terminal window on the first node, become the `root` user, and set up the classroom environment variables with the `st_env.sh` script. Execute the fixup script on your first node, and then ssh to your second node and repeat the process. Exit the `root` user session on the first node, and then exit the terminal session. When the script has been run on each node, click OK to close the dialog box. Click Next to continue.

```
   <<< On first node >>>
[grid]$ su - root
Password: 0racle    << password not displayed
[root]# . /home/oracle/labs/st_env.sh
[root]# /tmp/CVU_11.2.0.1.0_grid/runfixup.sh
Response file being used is
:/tmp/CVU_11.2.0.1.0_grid/fixup.response
Enable file being used is
:/tmp/CVU_11.2.0.1.0_grid/fixup.enable
Log file location: /tmp/CVU_11.2.0.1.0_grid/orarun.log
Setting Kernel Parameters...
fs.file-max = 327679
fs.file-max = 6815744
net.ipv4.ip_local_port_range = 9000 65500
net.core.wmem_max = 262144
net.core.wmem_max = 1048576
#
   <<< On Second Node >>>
[root]# ssh $ST_NODE2 /tmp/CVU_11.2.0.1.0_grid/runfixup.sh
root@host02's password: 0racle      <= password not echoed
Response file being used is
:/tmp/CVU_11.2.0.1.0_grid/fixup.response
Enable file being used is
:/tmp/CVU_11.2.0.1.0_grid/fixup.enable
```

```
Log file location: /tmp/CVU_11.2.0.1.0_grid/orarun.log
Setting Kernel Parameters...
fs.file-max = 327679
fs.file-max = 6815744
net.ipv4.ip_local_port_range = 9000 65500
net.core.wmem_max = 262144
net.core.wmem_max = 1048576
[root]# exit
logout
[grid]$ exit
```

q) Click Finish on the Summary screen. From this screen, you can monitor the progress of the installation.

r) When the remote operations have finished, the Execute Configuration Scripts window appears. You are instructed to run the `orainstRoot.sh` and `root.sh` scripts as the `root` user on both nodes. Open a terminal window and as the `root` user set the classroom environment variables. **Note:** You must wait until the `root.sh` script finishes running on the first node before executing it on the second node.

```
# su -
Password: Oracle    << password not displayed

# . /home/oracle/labs/st_env.sh

(On the first node)

# /u01/app/oraInventory/orainstRoot.sh
Changing permissions of /u01/app/oraInventory.
Adding read,write permissions for group.
Removing read,write,execute permissions for world.

Changing groupname of /u01/app/oraInventory to oinstall.
The execution of the script is complete.

# /u01/app/11.2.0/grid/root.sh
Running Oracle 11g root.sh script...

The following environment variables are set as:
    ORACLE_OWNER= grid
    ORACLE_HOME=  /u01/app/11.2.0/grid

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.
2009-08-25 14:46:20: Parsing the host name
2009-08-25 14:46:20: Checking for super user privileges
2009-08-25 14:46:20: User has super user privileges
Using configuration parameter file:
/u01/app/11.2.0/grid/crs/install/crsconfig_params
Creating trace directory
LOCAL ADD MODE
Creating OCR keys for user 'root', privgrp 'root'..
Operation successful.
  root wallet
  root wallet cert
  root cert export
  peer wallet
  profile reader wallet
  pa wallet
  peer wallet keys
  pa wallet keys
  peer cert request
  pa cert request
  peer cert
  pa cert
  peer root cert TP
  profile reader root cert TP
  pa root cert TP
  peer pa cert TP
  pa peer cert TP
  profile reader pa cert TP
  profile reader peer cert TP
  peer user cert
  pa user cert
Adding daemon to inittab
CRS-4123: Oracle High Availability Services has been started.
ohasd is starting
CRS-2672: Attempting to start 'ora.gipcd' on 'host01'
CRS-2672: Attempting to start 'ora.mdnsd' on 'host01'
CRS-2676: Start of 'ora.gipcd' on 'host01' succeeded
CRS-2676: Start of 'ora.mdnsd' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.gpnpd' on 'host01'
CRS-2676: Start of 'ora.gpnpd' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'host01'
CRS-2676: Start of 'ora.cssdmonitor' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.cssd' on 'host01'
CRS-2672: Attempting to start 'ora.diskmon' on 'host01'
CRS-2676: Start of 'ora.diskmon' on 'host01' succeeded
CRS-2676: Start of 'ora.cssd' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.ctssd' on 'host01'
CRS-2676: Start of 'ora.ctssd' on 'host01' succeeded
```

Practice 2-2: Installing Oracle Grid Infrastructure (continued)

```
ASM created and started successfully.

DiskGroup DATA created successfully.

clscfg: -install mode specified
Successfully accumulated necessary OCR keys.
Creating OCR keys for user 'root', privgrp 'root'..
Operation successful.
CRS-2672: Attempting to start 'ora.crsd' on 'host01'
CRS-2676: Start of 'ora.crsd' on 'host01' succeeded
CRS-4256: Updating the profile
Successful addition of voting disk
0c270623caf14f07bf57a7e9a1eb5f5c.
Successful addition of voting disk
04f565f5c6ed4f1cbf7444c2b24ebf1a.
Successful addition of voting disk
34c5c13ae40a4f21bf950e0de2777cdf.
Successfully replaced voting disk group with +DATA.
CRS-4256: Updating the profile
CRS-4266: Voting file(s) successfully replaced
##  STATE    File Universal Id                File Name Disk
group
--  -----    -----------------                --------- ------
---
 1. ONLINE   0c270623caf14f07bf57a7e9a1eb5f5c (ORCL:ASMDISK01)
[DATA]
 2. ONLINE   04f565f5c6ed4f1cbf7444c2b24ebf1a (ORCL:ASMDISK02)
[DATA]
 3. ONLINE   34c5c13ae40a4f21bf950e0de2777cdf (ORCL:ASMDISK03)
[DATA]
Located 3 voting disk(s).
CRS-2673: Attempting to stop 'ora.crsd' on 'host01'
CRS-2677: Stop of 'ora.crsd' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.asm' on 'host01'
CRS-2677: Stop of 'ora.asm' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.ctssd' on 'host01'
CRS-2677: Stop of 'ora.ctssd' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.cssdmonitor' on 'host01'
CRS-2677: Stop of 'ora.cssdmonitor' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'host01'
CRS-2677: Stop of 'ora.cssd' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.gpnpd' on 'host01'
CRS-2677: Stop of 'ora.gpnpd' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.gipcd' on 'host01'
CRS-2677: Stop of 'ora.gipcd' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.mdnsd' on 'host01'
CRS-2677: Stop of 'ora.mdnsd' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.mdnsd' on 'host01'
CRS-2676: Start of 'ora.mdnsd' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.gipcd' on 'host01'
CRS-2676: Start of 'ora.gipcd' on 'host01' succeeded
```

```
CRS-2672: Attempting to start 'ora.gpnpd' on 'host01'
CRS-2676: Start of 'ora.gpnpd' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'host01'
CRS-2676: Start of 'ora.cssdmonitor' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.cssd' on 'host01'
CRS-2672: Attempting to start 'ora.diskmon' on 'host01'
CRS-2676: Start of 'ora.diskmon' on 'host01' succeeded
CRS-2676: Start of 'ora.cssd' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.ctssd' on 'host01'
CRS-2676: Start of 'ora.ctssd' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.asm' on 'host01'
CRS-2676: Start of 'ora.asm' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'host01'
CRS-2676: Start of 'ora.crsd' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.evmd' on 'host01'
CRS-2676: Start of 'ora.evmd' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.asm' on 'host01'
CRS-2676: Start of 'ora.asm' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.DATA.dg' on 'host01'
CRS-2676: Start of 'ora.DATA.dg' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.registry.acfs' on 'host01'
CRS-2676: Start of 'ora.registry.acfs' on 'host01' succeeded

host01     2009/08/25 14:53:37
/u01/app/11.2.0/grid/cdata/host01/backup_20090825_145337.olr
Preparing packages for installation...
cvuqdisk-1.0.7-1
Configure Oracle Grid Infrastructure for a Cluster ...
succeeded
Updating inventory properties for clusterware
Starting Oracle Universal Installer...

Checking swap space: must be greater than 500 MB.   Actual
3007 MB    Passed
The inventory pointer is located at /etc/oraInst.loc
The inventory is located at /u01/app/oraInventory
'UpdateNodeList' was successful.
#
```

(On the second node AFTER the root.sh script finishes on the first node)

```
# ssh $ST_NODE2 /u01/app/oraInventory/orainstRoot.sh
root's password: 0racle    << password not displayed
Changing permissions of /u01/app/oraInventory.
Adding read,write permissions for group.
Removing read,write,execute permissions for world.

Changing groupname of /u01/app/oraInventory to oinstall.
The execution of the script is complete.


# ssh $ST_NODE2 /u01/app/11.2.0/grid/root.sh
```

```
Running Oracle 11g root.sh script...

The following environment variables are set as:
    ORACLE_OWNER= grid
    ORACLE_HOME=  /u01/app/11.2.0/grid

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.
2009-08-25 14:56:42: Parsing the host name
2009-08-25 14:56:42: Checking for super user privileges
2009-08-25 14:56:42: User has super user privileges
Using configuration parameter file:
/u01/app/11.2.0/grid/crs/install/crsconfig_params
Creating trace directory
LOCAL ADD MODE
Creating OCR keys for user 'root', privgrp 'root'..
Operation successful.
Adding daemon to inittab
CRS-4123: Oracle High Availability Services has been started.
ohasd is starting
CRS-4402: The CSS daemon was started in exclusive mode but
found an active CSS daemon on node host01, number 1, and is
terminating
An active cluster was found during exclusive startup,
restarting to join the cluster
CRS-2672: Attempting to start 'ora.mdnsd' on 'host02'
CRS-2676: Start of 'ora.mdnsd' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.gipcd' on 'host02'
CRS-2676: Start of 'ora.gipcd' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.gpnpd' on 'host02'
CRS-2676: Start of 'ora.gpnpd' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'host02'
CRS-2676: Start of 'ora.cssdmonitor' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.cssd' on 'host02'
CRS-2672: Attempting to start 'ora.diskmon' on 'host02'
CRS-2676: Start of 'ora.diskmon' on 'host02' succeeded
CRS-2676: Start of 'ora.cssd' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.ctssd' on 'host02'
CRS-2676: Start of 'ora.ctssd' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.drivers.acfs' on 'host02'
CRS-2676: Start of 'ora.drivers.acfs' on 'host02' succeeded
```

```
CRS-2672: Attempting to start 'ora.asm' on 'host02'
CRS-2676: Start of 'ora.asm' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'host02'
CRS-2676: Start of 'ora.crsd' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.evmd' on 'host02'
CRS-2676: Start of 'ora.evmd' on 'host02' succeeded

host02     2009/08/25 15:01:15
/u01/app/11.2.0/grid/cdata/host02/backup_20090825_150115.olr
Preparing packages for installation...
cvuqdisk-1.0.7-1
Configure Oracle Grid Infrastructure for a Cluster ...
succeeded
Updating inventory properties for clusterware
Starting Oracle Universal Installer...

Checking swap space: must be greater than 500 MB.   Actual
3007 MB    Passed
The inventory pointer is located at /etc/oraInst.loc
The inventory is located at /u01/app/oraInventory
'UpdateNodeList' was successful.
#
```

s) After the scripts are executed on both nodes, click the OK button to close the dialog box. The configuration assistants will continue to execute from the Setup page.

t) When the configuration assistants have finished, click the Close button on the Finish page to exit the Installer.

2) When the installation finishes, you should verify the installation. You should check to make sure that the software stack is running, as it should. Execute the `crsctl stat res -t` command:

```
[grid]$ /u01/app/11.2.0/grid/bin/crsctl stat res -t

--------------------------------------------------------------
NAME           TARGET  STATE        SERVER       STATE_DETAILS
--------------------------------------------------------------
Local Resources
--------------------------------------------------------------
ora.DATA1.dg
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.LISTENER.lsnr
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.asm
               ONLINE  ONLINE       host01             Started
               ONLINE  ONLINE       host02             Started
```

```
ora.eons
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.gsd
               OFFLINE OFFLINE      host01
               OFFLINE OFFLINE      host02
ora.net1.network
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.ons
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.registry.acfs
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
--------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE       host02
ora.LISTENER_SCAN2.lsnr
      1        ONLINE  ONLINE       host01
ora.LISTENER_SCAN3.lsnr
      1        ONLINE  ONLINE       host01
ora.gns
      1        ONLINE  ONLINE       host01
ora.gns.vip
      1        ONLINE  ONLINE       host01
ora.host01.vip
      1        ONLINE  ONLINE       host01
ora.host02.vip
      1        ONLINE  ONLINE       host02
ora.oc4j
      1        OFFLINE OFFLINE
ora.scan1.vip
      1        ONLINE  ONLINE       host02
ora.scan2.vip
      1        ONLINE  ONLINE       host01
ora.scan3.vip
      1        ONLINE  ONLINE       host01
The inventory is located at /u01/app/oraInventory
```

3) Use the `dig` command to check your GNS and make sure that it is being updated with your cluster addresses. In this example, the IP address is the one assigned to your GNS.

```
# dig @192.0.2.155 cluster01-scan.cluster01.example.com

# dig @${ST_GNS_NAME} ${ST_CLUSTER_NAME}-
scan.${CLUSTER_NAME}.${ST_NODE_DOMAIN}
```

```
; <<>> DiG 9.3.4-P1 <<>> @cluster01-gns cluster01-
scan.cluster01.example.com
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43003
;; flags: qr aa; QUERY: 1, ANSWER: 3, AUTHORITY: 1,
ADDITIONAL: 1

;; QUESTION SECTION:
;cluster01-scan.cluster01.example.com.        IN      A

;; ANSWER SECTION:
cluster01-scan.cluster01.example.com. 120 IN  A
192.0.2.231
cluster01-scan.cluster01.example.com. 120 IN  A
192.0.2.229
cluster01-scan.cluster01.example.com. 120 IN  A
192.0.2.232

;; AUTHORITY SECTION:
cluster01-gns-vip.cluster01.example.com. 10800 IN NS
cluster01-gns-vip.cluster01.example.com.

;; ADDITIONAL SECTION:
cluster01-gns-vip.cluster01.example.com. 10800 IN A
192.0.2.155

;; Query time: 55 msec
;; SERVER: 192.0.2.155#53(192.0.2.155)
;; WHEN: Tue Aug 25 15:42:56 2009
;; MSG SIZE  rcvd: 174
```

4) Use the `dig` command to make sure that your name server is properly forwarding address requests in your cluster subdomain back to your GNS for resolution.

```
# cat /etc/resolv.conf
domain example.com
nameserver 10.216.104.27
search example.com

# dig @10.216.104.27 ${ST_CLUSTER_NAME}-
scan.${CLUSTER_NAME}.${ST_NODE_DOMAIN}

; <<>> DiG 9.3.4-P1 <<>> @10.216.104.27 cluster01-
scan.cluster01.example.com
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14358
```

```
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 1,
ADDITIONAL: 1

;; QUESTION SECTION:
; cluster01-scan.cluster01.example.com.        IN      A

;; ANSWER SECTION:
cluster01-scan.cluster01.example.com. 120 IN  A
10.196.180.214
cluster01-scan.cluster01.example.com. 120 IN  A
10.196.180.233
cluster01-scan.cluster01.example.com. 120 IN  A
10.196.182.230

;; AUTHORITY SECTION:
cluster01.example.com.     86400   IN      NS      cluster01-
gns.example.com.

;; ADDITIONAL SECTION:
cluster01-gns.example.com. 86400   IN      A
10.196.183.12

;; Query time: 62 msec
;; SERVER: 10.216.104.27#53(10.216.104.27)
;; WHEN: Tue Sep 29 14:53:49 2009
;; MSG SIZE  rcvd: 137
```

Practice 2-3: Creating ASM Disk Groups

In this practice, you create additional ASM disk groups to support the activities in the rest of the course. You create a disk group to hold ACFS file systems and another disk group to hold the Fast Recovery Area (FRA).

1) In the VNC session for the `grid` user, open a terminal window as the `grid` user, and set the oracle environment with the oraenv tool to the +ASM1 instance.

```
[grid]$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid
```

2) Start the ASM Configuration Assistant (ASMCA).

```
$ asmca
```

3) Create a disk group named ACFS with four disks and external redundancy—choose disks ASMDISK05 through ASMDISK08. Set the disk group attribute ADVM Compatibility to 11.2.0.0.0.

Step	Screen/Page Description	Choices or Values
a.	Configure ASM :DiskGroups	Click Create.
b.	Create DiskGroup	Enter: Disk Group Name: ACFS In the Redundancy section, select “External (None).” In the Select Member Disk section, select: ASMDISK05 ASMDISK06 ASMDISK07 ASMDISK08 Click Show Advanced Options. In the Disk Group Attributes section, set ADVM Compatibility to 11.2.0.0.0. Click OK.
c.	Disk Group:Creation	Click OK. Exit ASMCA when finished.

4) Using ASMCMD, create a disk group named FRA over the disks ASMDISK09 through ASMDISK11 with external redundancy. Using the command

```
asmcmd mkdg /home/oracle/labs/less_02/FRA_dg_config.xml
```

Review the `FRA_dg_config.xml` file, and then execute the command.

```
[grid]$ cat /home/oracle/labs/less_02/FRA_dg_config.xml
<dg name="FRA" redundancy="external">
<dsk>   <dsk string="ORCL:ASMDISK09"/>
<dsk>   <dsk string="ORCL:ASMDISK10"/>
<dsk>   <dsk string="ORCL:ASMDISK11"/>
 <a name="compatible.asm" value="11.2"/>
```

Practice 2-3: Creating ASM Disk Groups (continued)

```
 <a name="compatible.rdbms" value="11.2"/>
<a name="compatible.advm" value="11.2"/>
</dg>
[grid]$ asmcmd mkdg /home/oracle/labs/less_02/FRA_dg_config.xml
[grid]$
```

5) Use ASMCMD to confirm the creation of the FRA disk group and to see which disks are included in the FRA disk group.

```
[grid]$ asmcmd
ASMCMD> lsdg
State    Type    Rebal  Sector  Block       AU  Total_MB
Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks
Voting_files  Name
MOUNTED  EXTERN  N         512   4096  1048576     10644
4396                0          4396              0
N  ACFS/
MOUNTED  NORMAL  N         512   4096  1048576     10648
6249            1111           2569              0
N  DATA/
MOUNTED  EXTERN  N         512   4096  1048576      7984
7928                0          7928              0
N  FRA/
ASMCMD> lsdsk -G FRA
Path
ORCL:ASMDISK09
ORCL:ASMDISK10
ORCL:ASMDISK11
ASMCMD> exit
[grid]$
```

6) Mount the FRA disk group on your second node.
Note: The ASMCMD commands operate only on the local node, so the FRA disk group is mounted only on your first node.

```
[grid}$ . /home/oracle/labs/st_env.sh
[grid]$ ssh $ST_NODE2
[grid@host02] $ . oraenv
ORACLE_SID = [grid] ? +ASM2
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid
[grid@ host02]$ asmcmd mount FRA
[grid@ host02]$ asmcmd lsdg
State    Type    Rebal  Sector  Block       AU  Total_MB
Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks
Voting_files  Name
MOUNTED  EXTERN  N         512   4096  1048576      9996
3748                0          3748              0
N  ACFS/
MOUNTED  NORMAL  N         512   4096  1048576      9998
5556            1126           2215              0
N  DATA/
```

Practice 2-3: Creating ASM Disk Groups (continued)

```
MOUNTED  EXTERN  N         512   4096  1048576      7497
7398              0           7398             0
N  FRA/
[grid@ host02]$ exit
```

Practice 2-4: Creating ACFS File System

In this practice, you create an ASM volume to use as the shared ORACLE_HOME for a database that is being used as the Enterprise Manager repository. Create this volume in the disk group called ACFS, the volume name is DBHOME_1, and the mount point is `/u01/app/oracle/acfsmount/11.2.0/sharedhome`.

1) Start the ASM Configuration Assistant (ASMCA).

```
$ asmca
```

2) Create an ASM volume named DBHOME_1 with size 6 GB in the ACFS disk group.

Step	Screen/Page Description	Choices or Values
a.	Configure ASM :DiskGroups	Click the Volumes tab.
b.	Configure ASM :Volumes	Click Create.
c.	Create Volume	Enter: Volume Name: DBHOME_1 Size: 6 G Bytes Click OK.
d.	Volume: Creation	Click OK.

3) Open a terminal window and become the `root` user, password `0racle`. Create the mount point directory at `/u01/app/oracle/acfsmount/11.2.0/sharedhome`.

Do this on all three nodes.

```
$ su - root
Password: 0racle   << password is not displayed
[root]# mkdir -p /u01/app/oracle/acfsmount/11.2.0/sharedhome

[root]# . /home/oracle/labs/st_env.sh

[root]# ssh $ST_NODE2 mkdir -p
/u01/app/oracle/acfsmount/11.2.0/sharedhome
root@host02's password: 0racle   << password is not displayed

[root]# ssh $ST_NODE3 mkdir -p
/u01/app/oracle/acfsmount/11.2.0/sharedhome
root@host03's password: 0racle   << password is not displayed
```

4) Create the ACFS file system, and mount and register the ACFS file system.

Step	Screen/Page Description	Choices or Values
a.	Configure ASM: Volumes	Click the ASM Cluster File Systems tab.
b.	Configure ASM: ASM Cluster File	Click Create.

Practice 2-4: Creating ACFS File System (continued)

Step	Screen/Page Description	Choices or Values
	Systems	
c.	Create ASM Cluster File System	Verify that Database Home File System is selected. Enter Database Home Mountpoint: `/u01/app/oracle/acfsmount/11.2.0/sharedhome` Database Home Owner Name: `oracle` Database Home Owner Group: `oinstall` Click OK.
d.	Database Home: Run ACFS Script	Copy the script shown to the root terminal window.
e.	Root terminal window	Execute the script: `#` **`/u01/app/grid/cfgtoollogs/asmca/scripts/acfs_script.sh`** `#`
f.	Database Home: Run ACFS Script	Click Close.
g.	Configure ASM: ASM Cluster File Systems	Click Exit.
h.	ASM Configuration Assistant	Click Yes.

5) Exit the root terminal window.

```
# exit
$
```

Practice 2-5: Installing a Silent Database

In this practice, you start a silent installation of an Oracle RAC database that will be used in this course to enable Enterprise Manager Database Control.

1) In a VNC session for the `oracle` user (*host01*:1). Establish the `ssh` user equivalency for the `oracle` user between your two cluster nodes. Open a terminal window on one as the `oracle` user and name the other in place of remote_host in the example.

```
[oracle]$ cd /home/oracle/labs/silent_inst
[oracle]$ ./ssh_setup.sh
oracle@host02's password: 0racle  << password not displayed
Warning: Permanently added 'gr7213,10.196.180.13' (RSA) to the
list of known hosts.
oracle@host02's password: 0racle  << password not displayed
oracle@host02's password: 0racle  << password not displayed
id_rsa.pub                                    100%  395
0.4KB/s   00:00
oracle@host02's password: 0racle  << password not displayed
[oracle]$
```

2) Confirm that the `oracle` user in this session is a member of the required groups. The required groups are: dba, oinstall, oper, and asmdba. If all the groups do not appear, use the `su - oracle` command to reset the user groups.
Note: The groups are only reset for this terminal window.

```
$ id
uid=501(oracle) gid=502(oinstall)
groups=501(dba),502(oinstall)
[oracle]$ su - oracle
Password: 0racle  << password not displayed
[oracle@gr7212 ~]$ id
uid=501(oracle) gid=502(oinstall)
groups=501(dba),502(oinstall),503(oper),505(asmdba)
```

3) In the same terminal window, use the classroom environment variables run `/home/oracle/labs/st_env.sh`.

```
$ . /home/oracle/labs/st_env.sh
```

4) In a terminal window, as the `oracle` user, change directory to the following or to a directory specified by your instructor:

```
$ cd $ST_SOFTWARE_STAGE_DB
```

5) Start the silent installation by using the following command: (Be sure to provide the parameters. The installer shows a warning that can be ignored [WARNING] [INS-35421].This option installs a single instance database only.)

```
$ ./runInstaller -silent -responseFile \
/home/oracle/labs/silent_inst/db.rsp -waitforcompletion \
ORACLE_HOST=`hostname` CLUSTER_NODES=$ST_NODE_LIST
Starting Oracle Universal Installer...
```

Practice 2-5: Installing a Silent Database (continued)

```
Checking Temp space: must be greater than 80 MB.   Actual
29270 MB    Passed
Checking swap space: must be greater than 150 MB.   Actual
3971 MB    Passed
Preparing to launch Oracle Universal Installer from
/tmp/OraInstall2009-10-02_11-51-20AM. Please wait ...[WARNING]
[INS-35421] This option installs a single instance database
only.
   CAUSE: You have chosen to perform a Desktop class install
on a cluster. This option will not install Oracle RAC.
   ACTION: If you wish to install Oracle RAC, choose to
perform a Server class install.
[WARNING] [INS-35421] This option installs a single instance
database only.
   CAUSE: You have chosen to perform a Desktop class install
on a cluster. This option will not install Oracle RAC.
   ACTION: If you wish to install Oracle RAC, choose to
perform a Server class install.
You can find the log of this install session at:
 /u01/app/oraInventory/logs/installActions2009-10-02_11-51-
20AM.log
[WARNING] [INS-35421] This option installs a single instance
database only.
   CAUSE: You have chosen to perform a Desktop class install
on a cluster. This option will not install Oracle RAC.
   ACTION: If you wish to install Oracle RAC, choose to
perform a Server class install.
```

6) Execute the `root.sh` script **on both nodes** to complete the installation.

```
-- On the first node

# /u01/app/oracle/acfsmount/11.2.0/sharedhome/dbhome_1/root.sh
Check
/u01/app/oracle/acfsmount/11.2.0/sharedhome/dbhome_1/install/r
oot_host01_2009-09-11_10-35-54.log for the output of root
script
#

-- On the second node

# /u01/app/oracle/acfsmount/11.2.0/sharedhome/dbhome_1/root.sh
Check
/u01/app/oracle/acfsmount/11.2.0/sharedhome/dbhome_1/install/r
oot_host02_2009-09-11_10-43-07.log for the output of root
script
#
```

Practices for Lesson 3

In these practices, you will verify, stop, and start Oracle Clusterware. You will add and remove Oracle Clusterware configuration files and backup the Oracle Cluster Registry and the Oracle Local Registry.

Practice 3-1: Verifying, Starting, and Stopping Oracle Clusterware

In this practice, you check the status of Oracle Clusterware using both the operating system commands and the `crsctl` utility. You will also start and stop Oracle Clusterware.

1) Connect to the first node of your cluster as the `grid` user. You can use the `oraenv` script to define `ORACLE_SID`, `ORACLE_HOME`, `PATH`, `ORACLE_BASE`, and `LD_LIBRARY_PATH` for your environment.

```
$ id
uid=502(grid) gid=501(oinstall)
groups=501(oinstall),504(asmadmin),505(asmdba),506(asmoper)
$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid
```

2) Using the operating system commands, verify that the Oracle Clusterware daemon processes are running on the current node. (**Hint:** Most of the Oracle Clusterware daemon processes have names that end with `d.bin`.)

```
$ pgrep -l d.bin
12895 ohasd.bin
14838 mdnsd.bin
14850 gipcd.bin
14862 gpnpd.bin
14916 ocssd.bin
15062 octssd.bin
15166 oclskd.bin
15181 crsd.bin
15198 evmd.bin
15222 oclskd.bin
15844 gnsd.bin
24709 oclskd.bin
```

3) Using the `crsctl` utility, verify that Oracle Clusterware is running on the current node.

```
$ crsctl check crs
CRS-4638: Oracle High Availability Services is online
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
```

4) Verify the status of all cluster resources that are being managed by Oracle Clusterware for all nodes.

```
$ crsctl stat res -t
-------------------------------------------------------------
NAME           TARGET  STATE       SERVER    STATE_DETAILS
```

Practice 3-1: Verifying, Starting, and Stopping Oracle Clusterware (continued)

```
--------------------------------------------------------------
Local Resources
--------------------------------------------------------------
ora.ACFS.dg
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.DATA.dg
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.FRA.dg
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.LISTENER.lsnr
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.acfs.dbhome1.acfs
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.asm
               ONLINE  ONLINE       host01       Started
               ONLINE  ONLINE       host02       Started
ora.eons
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.gsd
               OFFLINE OFFLINE      host01
               OFFLINE OFFLINE      host02
ora.net1.network
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.ons
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.registry.acfs
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE       host02
ora.LISTENER_SCAN2.lsnr
      1        ONLINE  ONLINE       host01
ora.LISTENER_SCAN3.lsnr
      1        ONLINE  ONLINE       host01
ora.gns
      1        ONLINE  ONLINE       host01
ora.gns.vip
      1        ONLINE  ONLINE       host01
ora.host01.vip
      1        ONLINE  ONLINE       host01
ora.host02.vip
```

Practice 3-1: Verifying, Starting, and Stopping Oracle Clusterware (continued)

```
      1        ONLINE  ONLINE       host02
ora.oc4j
      1        OFFLINE OFFLINE
ora.orcl.db
      1        ONLINE  ONLINE       host01      Open
      2        ONLINE  ONLINE       host02      Open
ora.scan1.vip
      1        ONLINE  ONLINE       host02
ora.scan2.vip
      1        ONLINE  ONLINE       host01
ora.scan3.vip
      1        ONLINE  ONLINE       host01
```

5) Attempt to stop Oracle Clusterware on the current node while logged in as the `grid` user. What happens and why?

```
$ crsctl stop crs
CRS-4563: Insufficient user privileges.
CRS-4000: Command Stop failed, or completed with errors.
```

6) Switch to the `root` account and stop Oracle Clusterware only on the current node. Exit the switch user command when the stop succeeds.

```
$ su -
Password: 0racle   << Password is not displayed

# /u01/app/11.2.0/grid/bin/crsctl stop crs
CRS-2791: Starting shutdown of Oracle High Availability
Services-managed resources on 'host01'
CRS-2673: Attempting to stop 'ora.crsd' on 'host01'
CRS-2790: Starting shutdown of Cluster Ready Services-managed
resources on 'host01'
CRS-2673: Attempting to stop 'ora.LISTENER.lsnr' on 'host01'
CRS-2673: Attempting to stop 'ora.FRA.dg' on 'host01'
CRS-2673: Attempting to stop 'ora.orcl.db' on 'host01'
CRS-2673: Attempting to stop 'ora.registry.acfs' on 'host01'
CRS-2673: Attempting to stop 'ora.gns' on 'host01'
CRS-2673: Attempting to stop 'ora.LISTENER_SCAN3.lsnr' on
'host01'
CRS-2673: Attempting to stop 'ora.LISTENER_SCAN2.lsnr' on
'host01'
CRS-2677: Stop of 'ora.LISTENER.lsnr' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.host01.vip' on 'host01'
CRS-2677: Stop of 'ora.LISTENER_SCAN3.lsnr' on 'host01'
succeeded
CRS-2673: Attempting to stop 'ora.scan3.vip' on 'host01'
CRS-2677: Stop of 'ora.LISTENER_SCAN2.lsnr' on 'host01'
succeeded
CRS-2673: Attempting to stop 'ora.scan2.vip' on 'host01'
CRS-2677: Stop of 'ora.host01.vip' on 'host01' succeeded
```

Practice 3-1: Verifying, Starting, and Stopping Oracle Clusterware (continued)

```
CRS-2672: Attempting to start 'ora.host01.vip' on 'host02'
CRS-2677: Stop of 'ora.scan3.vip' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.scan3.vip' on 'host02'
CRS-2677: Stop of 'ora.scan2.vip' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.scan2.vip' on 'host02'
CRS-2677: Stop of 'ora.gns' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.gns.vip' on 'host01'
CRS-2677: Stop of 'ora.gns.vip' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.gns.vip' on 'host02'
CRS-2677: Stop of 'ora.FRA.dg' on 'host01' succeeded
CRS-2676: Start of 'ora.scan2.vip' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.LISTENER_SCAN2.lsnr' on
'host02'
CRS-2676: Start of 'ora.gns.vip' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.gns' on 'host02'
CRS-2676: Start of 'ora.scan3.vip' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.LISTENER_SCAN3.lsnr' on
'host02'
CRS-2674: Start of 'ora.host01.vip' on 'host02' failed
CRS-2679: Attempting to clean 'ora.host01.vip' on 'host02'
CRS-2674: Start of 'ora.gns' on 'host02' failed
CRS-2679: Attempting to clean 'ora.gns' on 'host02'
CRS-2681: Clean of 'ora.gns' on 'host02' succeeded
CRS-2681: Clean of 'ora.host01.vip' on 'host02' succeeded
CRS-2676: Start of 'ora.LISTENER_SCAN2.lsnr' on 'host02'
succeeded
CRS-2676: Start of 'ora.LISTENER_SCAN3.lsnr' on 'host02'
succeeded
CRS-2677: Stop of 'ora.orcl.db' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.acfs.dbhome1.acfs' on
'host01'
CRS-2673: Attempting to stop 'ora.DATA.dg' on 'host01'
CRS-2677: Stop of 'ora.DATA.dg' on 'host01' succeeded
CRS-5014: Agent "/u01/app/11.2.0/grid/bin/orarootagent.bin"
timed out starting process
"/u01/app/11.2.0/grid/bin/acfssinglefsmount" for action
"stop": details at "(:CLSN00009:)" in
"/u01/app/11.2.0/grid/log/host01/agent/crsd/orarootagent_root/
orarootagent_root.log"
(:CLSN00009:)Utils:execCmd aborted
CRS-2675: Stop of 'ora.acfs.dbhome1.acfs' on 'host01' failed
CRS-2679: Attempting to clean 'ora.acfs.dbhome1.acfs' on
'host01'
CRS-2675: Stop of 'ora.registry.acfs' on 'host01' failed
CRS-2679: Attempting to clean 'ora.registry.acfs' on 'host01'
CRS-2678: 'ora.acfs.dbhome1.acfs' on 'host01' has experienced
an unrecoverable failure
CRS-0267: Human intervention required to resume its
availability.
```

```
CRS-2673: Attempting to stop 'ora.asm' on 'host01'
CRS-2677: Stop of 'ora.asm' on 'host01' succeeded
CRS-2681: Clean of 'ora.registry.acfs' on 'host01' succeeded
CRS-2794: Shutdown of Cluster Ready Services-managed resources
on 'host01' has failed
CRS-2675: Stop of 'ora.crsd' on 'host01' failed
CRS-2795: Shutdown of Oracle High Availability Services-
managed resources on 'host01' has failed
CRS-4687: Shutdown command has completed with error(s).
CRS-4000: Command Stop failed, or completed with errors.
```

7) If the `crsctl` utility fails to stop Oracle Clusterware, reissue the command on the same node.

```
# /u01/app/11.2.0/grid/bin/crsctl stop crs
CRS-2791: Starting shutdown of Oracle High Availability
Services-managed resources on 'host01'
CRS-2673: Attempting to stop 'ora.crsd' on 'host01'
CRS-2790: Starting shutdown of Cluster Ready Services-managed
resources on 'host01'
CRS-2679: Attempting to clean 'ora.acfs.dbhome1.acfs' on
'host01'
CRS-2673: Attempting to stop 'ora.ACFS.dg' on 'host01'
CRS-2681: Clean of 'ora.acfs.dbhome1.acfs' on 'host01'
succeeded
CRS-2677: Stop of 'ora.ACFS.dg' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.ons' on 'host01'
CRS-2673: Attempting to stop 'ora.eons' on 'host01'
CRS-2677: Stop of 'ora.ons' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.net1.network' on 'host01'
CRS-2677: Stop of 'ora.net1.network' on 'host01' succeeded
CRS-2677: Stop of 'ora.eons' on 'host01' succeeded
CRS-2792: Shutdown of Cluster Ready Services-managed resources
on 'host01' has completed
CRS-2677: Stop of 'ora.crsd' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.cssdmonitor' on 'host01'
CRS-2673: Attempting to stop 'ora.ctssd' on 'host01'
CRS-2673: Attempting to stop 'ora.evmd' on 'host01'
CRS-2673: Attempting to stop 'ora.asm' on 'host01'
CRS-2673: Attempting to stop 'ora.mdnsd' on 'host01'
CRS-2673: Attempting to stop 'ora.drivers.acfs' on 'host01'
CRS-2677: Stop of 'ora.cssdmonitor' on 'host01' succeeded
CRS-2677: Stop of 'ora.evmd' on 'host01' succeeded
CRS-2677: Stop of 'ora.mdnsd' on 'host01' succeeded
CRS-5016: Process "/u01/app/11.2.0/grid/bin/acfsload" spawned
by agent "/u01/app/11.2.0/grid/bin/orarootagent.bin" for
action "stop" failed: details at "(:CLSN00010:)" in
"/u01/app/11.2.0/grid/log/host01/agent/ohasd/orarootagent_root
/orarootagent_root.log"
Waiting for ASM to shutdown.
```

Practice 3-1: Verifying, Starting, and Stopping Oracle Clusterware (continued)

```
acfsload: ACFS-9118: oracleadvm.ko driver in use - can not
unload.
acfsload: ACFS-9118: oracleoks.ko driver in use - can not
unload.
CRS-2677: Stop of 'ora.drivers.acfs' on 'host01' succeeded
CRS-2677: Stop of 'ora.ctssd' on 'host01' succeeded
CRS-2677: Stop of 'ora.asm' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'host01'
CRS-2677: Stop of 'ora.cssd' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.gpnpd' on 'host01'
CRS-2673: Attempting to stop 'ora.diskmon' on 'host01'
CRS-2677: Stop of 'ora.gpnpd' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.gipcd' on 'host01'
CRS-2677: Stop of 'ora.gipcd' on 'host01' succeeded
CRS-2677: Stop of 'ora.diskmon' on 'host01' succeeded
CRS-2793: Shutdown of Oracle High Availability Services-
managed resources on 'host01' has completed
CRS-4133: Oracle High Availability Services has been stopped.
# exit
```

8) Attempt to check the status of Oracle Clusterware now that it has been successfully stopped.

```
$ crsctl check crs
CRS-4639: Could not contact Oracle High Availability Services

$ crsctl check cluster
CRS-4639: Could not contact Oracle High Availability Services
CRS-4000: Command Check failed, or completed with errors.
```

9) Connect to the second node of your cluster and verify that Oracle Clusterware is still running on that node. You may need to set your environment for the second node by using the oraenv utility.

```
$ . /home/oracle/labs/st_env.sh
$ ssh $ST_NODE2
Last login: Thu Aug 27 17:28:29 2009 from host01.example.com

$ . oraenv
ORACLE_SID = [grid] ? +ASM2
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid

$ crsctl check crs
CRS-4638: Oracle High Availability Services is online
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
```

10) Verify that all cluster resources are running on the second node, stopped on the first node, and that the VIP resources from the first node have migrated or failed over to the second node. The `ora.oc4j` and the `ora.gsd` resources are expected to be offline. Exit the connection to the second node when done.

```
$ crsctl stat res -t
----------------------------------------------------------
NAME           TARGET  STATE        SERVER   STATE_DETAILS
----------------------------------------------------------
Local Resources
----------------------------------------------------------
ora.ACFS.dg
               ONLINE  ONLINE       host02
ora.DATA.dg
               ONLINE  ONLINE       host02
ora.FRA.dg
               ONLINE  ONLINE       host02
ora.LISTENER.lsnr
               ONLINE  ONLINE       host02
ora.acfs.dbhome1.acfs
               ONLINE  ONLINE       host02
ora.asm
               ONLINE  ONLINE       host02
ora.eons
               ONLINE  ONLINE       host02
ora.gsd
               OFFLINE OFFLINE      host02
ora.net1.network
               ONLINE  ONLINE       host02
ora.ons
               ONLINE  ONLINE       host02
ora.registry.acfs
               ONLINE  ONLINE       host02
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE       host02
ora.LISTENER_SCAN2.lsnr
      1        ONLINE  ONLINE       host02
ora.LISTENER_SCAN3.lsnr
      1        ONLINE  ONLINE       host02
ora.gns
      1        ONLINE  ONLINE       host02
ora.gns.vip
      1        ONLINE  ONLINE       host02
ora.host01.vip
      1        ONLINE  INTERMEDIATE host02    FAILED OVER
ora.host02.vip
      1        ONLINE  ONLINE       host02
ora.oc4j
      1        OFFLINE OFFLINE
```

Practice 3-1: Verifying, Starting, and Stopping Oracle Clusterware (continued)

```
ora.orcl.db
      1        ONLINE  OFFLINE
      2        ONLINE  ONLINE       host02
ora.scan1.vip
      1        ONLINE  ONLINE       host02
ora.scan2.vip
      1        ONLINE  ONLINE       host02
ora.scan3.vip
      1        ONLINE  ONLINE       host02

$ exit
Connection to host02 closed.
$
```

11) Restart Oracle Clusterware on the first node as the `root` user. Return to the `grid` account and verify the results.
Note: You may need to check the status of all the resources several times until they all have been restarted. You can tell that they are all complete when the `ora.orcl.db` resource has a State Details of Open. It may take several minutes to completely restart all resources.

```
$ su -
Password: 0racle  << Password is not displayed

# /u01/app/11.2.0/grid/bin/crsctl start crs
CRS-4123: Oracle High Availability Services has been started.
# exit
$

$ crsctl stat res -t
-----------------------------------------------------------
NAME           TARGET  STATE        SERVER    STATE_DETAILS
-----------------------------------------------------------
Local Resources
-----------------------------------------------------------
ora.ACFS.dg
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.DATA.dg
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.FRA.dg
               OFFLINE OFFLINE      host01
               ONLINE  ONLINE       host02
ora.LISTENER.lsnr
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.acfs.dbhome1.acfs
               ONLINE  ONLINE       host01
```

Practice 3-1: Verifying, Starting, and Stopping Oracle Clusterware (continued)

```
               ONLINE  ONLINE       host02
ora.asm
               ONLINE  ONLINE       host01      Started
               ONLINE  ONLINE       host02
ora.eons
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.gsd
               OFFLINE OFFLINE      host01
               OFFLINE OFFLINE      host02
ora.net1.network
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.ons
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.registry.acfs
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE       host01
ora.LISTENER_SCAN2.lsnr
      1        ONLINE  ONLINE       host02
ora.LISTENER_SCAN3.lsnr
      1        ONLINE  ONLINE       host02
ora.gns
      1        ONLINE  ONLINE       host02
ora.gns.vip
      1        ONLINE  ONLINE       host02
ora.host01.vip
      1        ONLINE  ONLINE       host01
ora.host02.vip
      1        ONLINE  ONLINE       host02
ora.oc4j
      1        OFFLINE OFFLINE
ora.orcl.db
      1        ONLINE  ONLINE       host01       Open
      2        ONLINE  ONLINE       host02
ora.scan1.vip
      1        ONLINE  ONLINE       host01
ora.scan2.vip
      1        ONLINE  ONLINE       host02
ora.scan3.vip
      1        ONLINE  ONLINE       host02
```

Practice 3-2: Adding and Removing Oracle Clusterware Configuration Files

In this practice, you determine the current location of your voting disks and Oracle Cluster Registry (OCR) files. You will then add another OCR location and remove it.

1) Use the `crsctl` utility to determine the location of the voting disks that are currently used by your Oracle Clusterware installation.

```
$ crsctl query css votedisk
##  STATE    File Universal Id                File Name Disk
group
--  -----    -----------------                --------- -----
-----
 1. ONLINE   0f7a12c15ece4fcebf4dfe8ab209406e (ORCL:ASMDISK01)
[DATA]
 2. ONLINE   718cf2c254b44f21bfb8d41fbf21fdd5 (ORCL:ASMDISK02)
[DATA]
 3. ONLINE   449422aa69534f81bfc54e41087d677c (ORCL:ASMDISK03)
[DATA]
Located 3 voting disk(s).
```

2) Use the `ocrcheck` utility to determine the location of the Oracle Clusterware Registry (OCR) files.

```
$ ocrcheck
Status of Oracle Cluster Registry is as follows :
         Version                  :          3
         Total space (kbytes)     :     262120
         Used space (kbytes)      :       3228
         Available space (kbytes) :     258892
         ID                       : 1581544792
         Device/File Name         :      +DATA
                                    Device/File integrity
check succeeded

                                    Device/File not configured

                                    Device/File not configured

                                    Device/File not configured

                                    Device/File not configured

         Cluster registry integrity check succeeded

         Logical corruption check bypassed due to non-
privileged user
```

3) Verify that the `FRA` ASM disk group is currently online for all nodes using the `crsctl` utility.

```
$ crsctl stat res ora.FRA.dg -t
-----------------------------------------------------------
```

Practice 3-2: Adding and Removing Oracle Clusterware Configuration Files (continued)

```
NAME           TARGET  STATE        SERVER   STATE_DETAILS
-----------------------------------------------------------
Local Resources
----------------------------------------------------------
ora.FRA.dg
               OFFLINE OFFLINE      host01
               ONLINE  ONLINE       host02
```

4) If the FRA ASM disk group is not online, use the asmcmd utility to mount the FRA disk group as the grid user.
   Note: This step may not be necessary if it is already in an online state on each node. Verify the results. You may have to run the commands on each node.

```
$ asmcmd mount FRA
$

$ crsctl stat res ora.FRA.dg -t
-----------------------------------------------------------
NAME           TARGET  STATE        SERVER   STATE_DETAILS
-----------------------------------------------------------
Local Resources
----------------------------------------------------------
ora.FRA.dg
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
```

5) Switch to the root account and add a second OCR location that is to be stored in the FRA ASM disk group. Use the ocrcheck command to verify the results.

```
$ su -
Password: 0racle << Password is not displayed

# /u01/app/11.2.0/grid/bin/ocrconfig -add +FRA

# /u01/app/11.2.0/grid/bin/ocrcheck
Status of Oracle Cluster Registry is as follows :
         Version                  :          3
         Total space (kbytes)     :     262120
         Used space (kbytes)      :       3228
         Available space (kbytes) :     258892
         ID                       : 1581544792
         Device/File Name         :      +DATA
                                    Device/File integrity
check succeeded
         Device/File Name         :       +FRA
                                    Device/File integrity
check succeeded

                                    Device/File not configured
```

Practice 3-2: Adding and Removing Oracle Clusterware Configuration Files (continued)

```
                                    Device/File not configured

                                    Device/File not configured

         Cluster registry integrity check succeeded

         Logical corruption check succeeded
```

6) Examine the contents of the `ocr.loc` configuration file to see the changes made to the file referencing the new OCR location.

```
# cat /etc/oracle/ocr.loc
#Device/file  getting replaced by device +FRA
ocrconfig_loc=+DATA
ocrmirrorconfig_loc=+FRA
```

7) Open a connection to your second node as the `root` user, and remove the second OCR file that was added from the first node. Exit the remote connection and verify the results when completed.

```
# . /home/oracle/labs/st_env.sh
# ssh $ST_NODE2
root@host02's password: 0racle << Password is not displayed
Last login: Tue Aug 25 13:04:32 2009 from host01.example.com

# /u01/app/11.2.0/grid/bin/ocrconfig -delete +FRA

# exit
Connection to host02 closed.

# /u01/app/11.2.0/grid/bin/ocrcheck
Status of Oracle Cluster Registry is as follows :
         Version                  :          3
         Total space (kbytes)     :     262120
         Used space (kbytes)      :       3228
         Available space (kbytes) :     258892
         ID                       : 1581544792
         Device/File Name         :      +DATA
                                    Device/File integrity
check succeeded

                                    Device/File not configured

                                    Device/File not configured

                                    Device/File not configured
```

Practice 3-2: Adding and Removing Oracle Clusterware Configuration Files (continued)

```
                                  Device/File not configured

         Cluster registry integrity check succeeded

         Logical corruption check succeeded
```

Practice 3-3: Performing a Backup of the OCR and OLR

In this practice, you determine the location of the Oracle Local Registry (OLR) and perform backups of the OCR and OLR files.

1) Use the `ocrconfig` utility to list the automatic backups of the Oracle Cluster Registry (OCR) and the node or nodes on which they have been performed.

```
# /u01/app/11.2.0/grid/bin/ocrconfig -showbackup

host02     2009/08/27 20:17:20
/u01/app/11.2.0/grid/cdata/cluster03/backup00.ocr

host01     2009/08/27 14:53:56
/u01/app/11.2.0/grid/cdata/cluster03/backup01.ocr

host01     2009/08/27 10:53:54
/u01/app/11.2.0/grid/cdata/cluster03/backup02.ocr

host01     2009/08/26 02:53:44
/u01/app/11.2.0/grid/cdata/cluster03/day.ocr

host01     2009/08/24 18:53:34
/u01/app/11.2.0/grid/cdata/cluster03/week.ocr
PROT-25: Manual backups for the Oracle Cluster Registry are
not available
```

2) Perform a manual backup of the OCR.

```
# /u01/app/11.2.0/grid/bin/ocrconfig -manualbackup

host02     2009/08/27 21:37:49
/u01/app/11.2.0/grid/cdata/cluster03/backup_20090827_213749.oc
r
```

3) Perform a logical backup of the OCR, storing the file in the `/home/oracle` directory.

```
# /u01/app/11.2.0/grid/bin/ocrconfig -export
/home/oracle/ocr.backup
```

4) Display only the manual backups that have been performed and identify the node for which the backup was stored. Do logical backup appear in the display?

```
# /u01/app/11.2.0/grid/bin/ocrconfig -showbackup manual

host02     2009/08/27 21:37:49
/u01/app/11.2.0/grid/cdata/cluster03/backup_20090827_213749.oc
r
```

5) Determine the location of the Oracle Local Registry (OLR) using the `ocrcheck` utility.

```
# /u01/app/11.2.0/grid/bin/ocrcheck -local
Status of Oracle Local Registry is as follows :
```

Practice 3-3: Performing a Backup of the OCR and OLR (continued)

```
         Version                  :          3
         Total space (kbytes)     :     262120
         Used space (kbytes)      :       2204
         Available space (kbytes) :     259916
         ID                       : 1965295944
         Device/File Name         :
/u01/app/11.2.0/grid/cdata/host01.olr
                                    Device/File integrity
check succeeded

         Local registry integrity check succeeded

         Logical corruption check succeeded
```

6) Perform a logical backup of the OLR, storing the file in the `/home/oracle` directory.

```
# /u01/app/11.2.0/grid/bin/ocrconfig -local -export
/home/oracle/olr.backup
```

Practices for Lesson 4

In this practice, you will add a third node to your cluster.

Practice 4-1: Adding a Third Node to Your Cluster

The goal of this lab is to extend your cluster to a third node.

Before you start this lab, make sure that you went through the following steps on your third node: lab 2-1: 1, 2, 3, 4, 5, 6, 8.

In this practice, you extend your cluster to the third node given to you.

Note

- **All the scripts you need to execute this lab are located on your first node in the `/home/oracle/labs/node_addition` directory**.
- **Unless specified otherwise, you are connected on your first node as the `grid` user using a terminal session**.

1) Set up the `ssh` user equivalence for the `grid` user between your first node and your third node.

```
$  /home/oracle/solutions/configure_ssh
Setting up SSH user equivalency.
grid@gr7213's password:
grid@gr7214's password:
Checking SSH user equivalency.
gr7212
gr7213
gr7214

$
```

2) Make sure that you can connect from your first node to the third one without being prompted for passwords.

```
$ . /home/oracle/labs/st_env.sh
$ ssh $ST_NODE3 date
Fri Sep  4 10:38:10 EDT 2009
$
```

3) Make sure that you set up your environment variables correctly for the `grid` user to point to your grid installation.

```
$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid
$
```

4) Check your pre-grid installation for your third node using the Cluster Verification Utility. This fails because the fixup scripts have not been run on the third node.

```
$ . /home/oracle/labs/st_env.sh
$
$ cluvfy stage -pre crsinst -n $ST_NODE3

Performing pre-checks for cluster services setup
```

```
Checking node reachability...
Node reachability check passed from node "gr7212"


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

Checking node connectivity...

Checking hosts config file...

Verification of the hosts config file successful

Node connectivity passed for subnet "10.216.52.0" with node(s)
gr7214
TCP connectivity check passed for subnet "10.216.52.0"

Node connectivity passed for subnet "10.216.96.0" with node(s)
gr7214
TCP connectivity check passed for subnet "10.216.96.0"

Node connectivity passed for subnet "10.196.28.0" with node(s)
gr7214
TCP connectivity check passed for subnet "10.196.28.0"

Node connectivity passed for subnet "10.196.180.0" with
node(s) gr7214
TCP connectivity check passed for subnet "10.196.180.0"


Interfaces found on subnet "10.216.52.0" that are likely
candidates for VIP are:
gr7214 eth0:10.216.54.232

Interfaces found on subnet "10.216.96.0" that are likely
candidates for a private interconnect are:
gr7214 eth1:10.216.98.155

Interfaces found on subnet "10.196.28.0" that are likely
candidates for a private interconnect are:
gr7214 eth2:10.196.31.14

Interfaces found on subnet "10.196.180.0" that are likely
candidates for a private interconnect are:
gr7214 eth3:10.196.180.14

Node connectivity check passed

Total memory check passed
Available memory check passed
```

```
Swap space check passed
Free disk space check passed for "gr7214:/tmp"
User existence check passed for "grid"
Group existence check passed for "oinstall"
Group existence check passed for "dba"
Membership check for user "grid" in group "oinstall" [as
Primary] passed
Membership check for user "grid" in group "dba" failed
Check failed on nodes:
        gr7214
Run level check passed
Hard limits check passed for "maximum open file descriptors"
Soft limits check passed for "maximum open file descriptors"
Hard limits check passed for "maximum user processes"
Soft limits check passed for "maximum user processes"
System architecture check passed
Kernel version check passed
Kernel parameter check passed for "semmsl"
Kernel parameter check passed for "semmns"
Kernel parameter check passed for "semopm"
Kernel parameter check passed for "semmni"
Kernel parameter check passed for "shmmax"
Kernel parameter check passed for "shmmni"
Kernel parameter check passed for "shmall"
Kernel parameter check failed for "file-max"
Check failed on nodes:
        gr7214
Kernel parameter check passed for "ip_local_port_range"
Kernel parameter check passed for "rmem_default"
Kernel parameter check passed for "rmem_max"
Kernel parameter check passed for "wmem_default"
Kernel parameter check failed for "wmem_max"
Check failed on nodes:
        gr7214
Kernel parameter check passed for "aio-max-nr"
Package existence check passed for "make-3.81"
Package existence check passed for "binutils-2.17.50.0.6"
Package existence check passed for "gcc-4.1.2"
Package existence check passed for "gcc-c++-4.1.2"
Package existence check passed for "libgomp-4.1.2"
Package existence check passed for "libaio-0.3.106"
Package existence check passed for "glibc-2.5-24"
Package existence check passed for "compat-libstdc++-33-3.2.3"
Package existence check passed for "elfutils-libelf-0.125"
Package existence check passed for "elfutils-libelf-devel-
0.125"
Package existence check passed for "glibc-common-2.5"
Package existence check passed for "glibc-devel-2.5"
Package existence check passed for "glibc-headers-2.5"
Package existence check passed for "libaio-devel-0.3.106"
Package existence check passed for "libgcc-4.1.2"
```

```
Package existence check passed for "libstdc++-4.1.2"
Package existence check passed for "libstdc++-devel-4.1.2"
Package existence check passed for "sysstat-7.0.2"
Package existence check passed for "unixODBC-2.2.11"
Package existence check passed for "unixODBC-devel-2.2.11"
Package existence check passed for "ksh-20060214"
Check for multiple users with UID value 0 passed
Current group ID check passed
Core file name pattern consistency check passed.

User "grid" is not part of "root" group. Check passed
Default user file creation mask check passed

Starting Clock synchronization checks using Network Time
Protocol(NTP)...

NTP Configuration file check started...
NTP Configuration file check passed

Checking daemon liveness...
Liveness check passed for "ntpd"

NTP daemon slewing option check passed

NTP daemon's boot time configuration check for slewing option
passed

NTP common Time Server Check started...
Check of common NTP Time Server passed

Clock time offset check from NTP Time Server started...
Clock time offset check passed

Clock synchronization check using Network Time Protocol(NTP)
passed


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

5) Generate the fixup script for your third node with `cluvfy` with the `–fixup` option.

```
$ cluvfy stage -pre crsinst -n $ST_NODE3 –fixup
Performing pre-checks for cluster services setup

Checking node reachability...
Node reachability check passed from node "host01"


Checking user equivalence...
```

```
User equivalence check passed for user "grid"

Checking node connectivity...

Checking hosts config file...

Verification of the hosts config file successful

Node connectivity passed for subnet "10.216.52.0" with node(s)
host03
TCP connectivity check passed for subnet "10.216.52.0"

Node connectivity passed for subnet "10.216.96.0" with node(s)
host03
TCP connectivity check passed for subnet "10.216.96.0"

Node connectivity passed for subnet "10.196.28.0" with node(s)
host03
TCP connectivity check passed for subnet "10.196.28.0"

Node connectivity passed for subnet "10.196.180.0" with
node(s) host03
TCP connectivity check passed for subnet "10.196.180.0"


Interfaces found on subnet "10.216.52.0" that are likely
candidates for VIP are:
host03 eth0:10.216.54.232

Interfaces found on subnet "10.216.96.0" that are likely
candidates for a private interconnect are:
host03 eth1:10.216.98.155

Interfaces found on subnet "10.196.28.0" that are likely
candidates for a private interconnect are:
host03 eth2:10.196.31.14

Interfaces found on subnet "10.196.180.0" that are likely
candidates for a private interconnect are:
host03 eth3:10.196.180.14

Node connectivity check passed

Total memory check passed
Available memory check passed
Swap space check passed
Free disk space check passed for " host03:/tmp"
User existence check passed for "grid"
Group existence check passed for "oinstall"
Group existence check passed for "dba"
```

```
Membership check for user "grid" in group "oinstall" [as
Primary] passed
Membership check for user "grid" in group "dba" failed
Check failed on nodes:
        host03
Run level check passed
Hard limits check passed for "maximum open file descriptors"
Soft limits check passed for "maximum open file descriptors"
Hard limits check passed for "maximum user processes"
Soft limits check passed for "maximum user processes"
System architecture check passed
Kernel version check passed
Kernel parameter check passed for "semmsl"
Kernel parameter check passed for "semmns"
Kernel parameter check passed for "semopm"
Kernel parameter check passed for "semmni"
Kernel parameter check passed for "shmmax"
Kernel parameter check passed for "shmmni"
Kernel parameter check passed for "shmall"
Kernel parameter check failed for "file-max"
Check failed on nodes:
        host03
Kernel parameter check passed for "ip_local_port_range"
Kernel parameter check passed for "rmem_default"
Kernel parameter check passed for "rmem_max"
Kernel parameter check passed for "wmem_default"
Kernel parameter check failed for "wmem_max"
Check failed on nodes:
        host03
Kernel parameter check passed for "aio-max-nr"
Package existence check passed for "make-3.81"
Package existence check passed for "binutils-2.17.50.0.6"
Package existence check passed for "gcc-4.1.2"
Package existence check passed for "gcc-c++-4.1.2"
Package existence check passed for "libgomp-4.1.2"
Package existence check passed for "libaio-0.3.106"
Package existence check passed for "glibc-2.5-24"
Package existence check passed for "compat-libstdc++-33-3.2.3"
Package existence check passed for "elfutils-libelf-0.125"
Package existence check passed for "elfutils-libelf-devel-
0.125"
Package existence check passed for "glibc-common-2.5"
Package existence check passed for "glibc-devel-2.5"
Package existence check passed for "glibc-headers-2.5"
Package existence check passed for "libaio-devel-0.3.106"
Package existence check passed for "libgcc-4.1.2"
Package existence check passed for "libstdc++-4.1.2"
Package existence check passed for "libstdc++-devel-4.1.2"
Package existence check passed for "sysstat-7.0.2"
Package existence check passed for "unixODBC-2.2.11"
Package existence check passed for "unixODBC-devel-2.2.11"
```

Practice 4-1: Adding a Third Node to Your Cluster (continued)

```
Package existence check passed for "ksh-20060214"
Check for multiple users with UID value 0 passed
Current group ID check passed
Core file name pattern consistency check passed.

User "grid" is not part of "root" group. Check passed
Default user file creation mask check passed

Starting Clock synchronization checks using Network Time
Protocol(NTP)...

NTP Configuration file check started...
NTP Configuration file check passed

Checking daemon liveness...
Liveness check passed for "ntpd"

NTP daemon slewing option check passed

NTP daemon's boot time configuration check for slewing option
passed

NTP common Time Server Check started...
Check of common NTP Time Server passed

Clock time offset check from NTP Time Server started...
Clock time offset check passed

Clock synchronization check using Network Time Protocol(NTP)
passed

Fixup information has been generated for following node(s):
host03
Please run the following script on each node as "root" user to
execute the fixups:
'/tmp/CVU_11.2.0.1.0_grid/runfixup.sh'

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

6) Run the fixup script as directed.

```
$ ssh root@${ST_NODE3} /tmp/CVU_11.2.0.1.0_grid/runfixup.sh
root@host03's password:
Response file being used is
:/tmp/CVU_11.2.0.1.0_grid/fixup.response
Enable file being used is
:/tmp/CVU_11.2.0.1.0_grid/fixup.enable
Log file location: /tmp/CVU_11.2.0.1.0_grid/orarun.log
Setting Kernel Parameters...
fs.file-max = 327679
```

```
fs.file-max = 6815744
net.ipv4.ip_local_port_range = 9000 65500
net.core.wmem_max = 262144
net.core.wmem_max = 1048576
uid=502(grid) gid=502(oinstall)
groups=505(asmdba),506(asmoper),504(asmadmin),502(oinstall)
$
```

7) Using the Cluster Verification Utility, make sure that you can add your third node to the cluster.

```
$ . /home/oracle/labs/st_env.sh
$
$ cluvfy stage -pre nodeadd -n $ST_NODE3

Performing pre-checks for node addition

Checking node reachability...
Node reachability check passed from node "host01"


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

Checking node connectivity...

Checking hosts config file...

Verification of the hosts config file successful

Check: Node connectivity for interface "eth3"
Node connectivity passed for interface "eth3"

Node connectivity check passed


Checking CRS integrity...

CRS integrity check passed

Checking shared resources...
Shared resources check for node addition failed

Check failed on nodes:  <-------- You can safely ignore
        host03

Checking node connectivity...

Checking hosts config file...

Verification of the hosts config file successful
```

```
Node connectivity passed for subnet "10.216.52.0" with node(s)
host01,host02,host03
TCP connectivity check passed for subnet "10.216.52.0"

Node connectivity passed for subnet "10.216.96.0" with node(s)
host01,host02,host03
TCP connectivity check passed for subnet "10.216.96.0"

Node connectivity passed for subnet "10.196.28.0" with node(s)
host01,host02,host03
TCP connectivity check passed for subnet "10.196.28.0"

Node connectivity passed for subnet "10.196.180.0" with
node(s) host01,host02,host03
TCP connectivity check failed for subnet "10.196.180.0"


Interfaces found on subnet "10.216.52.0" that are likely
candidates for VIP are:
host01 eth0:10.216.54.233
host02 eth0:10.216.54.234
host03 eth0:10.216.54.235

Interfaces found on subnet "10.216.96.0" that are likely
candidates for a private interconnect are:
host01 eth1:10.216.101.101
host02 eth1:10.216.96.144
host03 eth1:10.216.100.226

Interfaces found on subnet "10.196.28.0" that are likely
candidates for a private interconnect are:
host01 eth2:10.196.31.15
host02 eth2:10.196.31.16
host03 eth2:10.196.31.17

Interfaces found on subnet "10.196.180.0" that are likely
candidates for a private interconnect are:
host01 eth3:10.196.180.15 eth3:10.196.181.239
eth3:10.196.183.15 eth3:10.196.182.229 eth3:10.196.180.232
host02 eth3:10.196.180.16 eth3:10.196.181.231
eth3:10.196.181.244
host03 eth3:10.196.180.17

Node connectivity check passed

Total memory check passed
Available memory check passed
Swap space check passed
Free disk space check passed for "host03:/tmp"
User existence check passed for "grid"
```

```
Run level check passed
Hard limits check passed for "maximum open file descriptors"
Soft limits check passed for "maximum open file descriptors"
Hard limits check passed for "maximum user processes"
Soft limits check passed for "maximum user processes"
System architecture check passed
Kernel version check passed
Kernel parameter check passed for "semmsl"
Kernel parameter check passed for "semmns"
Kernel parameter check passed for "semopm"
Kernel parameter check passed for "semmni"
Kernel parameter check passed for "shmmax"
Kernel parameter check passed for "shmmni"
Kernel parameter check passed for "shmall"
Kernel parameter check passed for "file-max"
Kernel parameter check passed for "ip_local_port_range"
Kernel parameter check passed for "rmem_default"
Kernel parameter check passed for "rmem_max"
Kernel parameter check passed for "wmem_default"
Kernel parameter check passed for "wmem_max"
Kernel parameter check passed for "aio-max-nr"
Package existence check passed for "make-3.81"
Package existence check passed for "binutils-2.17.50.0.6"
Package existence check passed for "gcc-4.1.2"
Package existence check passed for "gcc-c++-4.1.2"
Package existence check passed for "libgomp-4.1.2"
Package existence check passed for "libaio-0.3.106"
Package existence check passed for "glibc-2.5-24"
Package existence check passed for "compat-libstdc++-33-3.2.3"
Package existence check passed for "elfutils-libelf-0.125"
Package existence check passed for "elfutils-libelf-devel-
0.125"
Package existence check passed for "glibc-common-2.5"
Package existence check passed for "glibc-devel-2.5"
Package existence check passed for "glibc-headers-2.5"
Package existence check passed for "libaio-devel-0.3.106"
Package existence check passed for "libgcc-4.1.2"
Package existence check passed for "libstdc++-4.1.2"
Package existence check passed for "libstdc++-devel-4.1.2"
Package existence check passed for "sysstat-7.0.2"
Package existence check passed for "unixODBC-2.2.11"
Package existence check passed for "unixODBC-devel-2.2.11"
Package existence check passed for "ksh-20060214"
Check for multiple users with UID value 0 passed

User "grid" is not part of "root" group. Check passed

Starting Clock synchronization checks using Network Time
Protocol(NTP)...

NTP Configuration file check started...
```

```
NTP Configuration file check passed

Checking daemon liveness...
Liveness check passed for "ntpd"

NTP daemon slewing option check passed

NTP daemon's boot time configuration check for slewing option
passed

NTP common Time Server Check started...
Check of common NTP Time Server passed

Clock time offset check from NTP Time Server started...
Clock time offset check passed

Clock synchronization check using Network Time Protocol(NTP)
passed


Pre-check for node addition was unsuccessful on all the nodes.
$
```

8) Add your third node to the cluster from your first node:

```
$ . /home/oracle/labs/st_env.sh
$
$ cd $ORACLE_HOME/oui/bin
$
$ ./addNode.sh -silent "CLUSTER_NEW_NODES={$ST_NODE3}"
"CLUSTER_NEW_VIRTUAL_HOSTNAMES={$ST_NODE3_VIP}"
Starting Oracle Universal Installer...

Checking swap space: must be greater than 500 MB.   Actual
3755 MB    Passed
Oracle Universal Installer, Version 11.2.0.1.0 Production
Copyright (C) 1999, 2009, Oracle. All rights reserved.


Performing tests to see whether nodes host02,host03 are
available
...............................................................
. 100% Done.

.
-----------------------------------------------------------------
---------------
Cluster Node Addition Summary
Global Settings
   Source: /u01/app/11.2.0/grid
   New Nodes
```

Practice 4-1: Adding a Third Node to Your Cluster (continued)

```
Space Requirements
   New Nodes
      host03
         /: Required 4.76GB : Available 32.38GB
Installed Products
   Product Names
      Oracle Grid Infrastructure 11.2.0.1.0
      Sun JDK 1.5.0.17.0
      Installer SDK Component 11.2.0.1.0
      Oracle One-Off Patch Installer 11.2.0.0.2
      Oracle Universal Installer 11.2.0.1.0
      Oracle Configuration Manager Deconfiguration 10.3.1.0.0
      Enterprise Manager Common Core Files 10.2.0.4.2
      Oracle DBCA Deconfiguration 11.2.0.1.0
      Oracle RAC Deconfiguration 11.2.0.1.0
      Oracle Quality of Service Management (Server) 11.2.0.1.0
      Installation Plugin Files 11.2.0.1.0
      Universal Storage Manager Files 11.2.0.1.0
      Oracle Text Required Support Files 11.2.0.1.0
      Automatic Storage Management Assistant 11.2.0.1.0
      Oracle Database 11g Multimedia Files 11.2.0.1.0
      Oracle Multimedia Java Advanced Imaging 11.2.0.1.0
      Oracle Globalization Support 11.2.0.1.0
      Oracle Multimedia Locator RDBMS Files 11.2.0.1.0
      Oracle Core Required Support Files 11.2.0.1.0
      Bali Share 1.1.18.0.0
      Oracle Database Deconfiguration 11.2.0.1.0
      Oracle Quality of Service Management (Client) 11.2.0.1.0
      Expat libraries 2.0.1.0.1
      Oracle Containers for Java 11.2.0.1.0
      Perl Modules 5.10.0.0.1
      Secure Socket Layer 11.2.0.1.0
      Oracle JDBC/OCI Instant Client 11.2.0.1.0
      Oracle Multimedia Client Option 11.2.0.1.0
      LDAP Required Support Files 11.2.0.1.0
      Character Set Migration Utility 11.2.0.1.0
      Perl Interpreter 5.10.0.0.1
      PL/SQL Embedded Gateway 11.2.0.1.0
      OLAP SQL Scripts 11.2.0.1.0
      Database SQL Scripts 11.2.0.1.0
      Oracle Extended Windowing Toolkit 3.4.47.0.0
      SSL Required Support Files for InstantClient 11.2.0.1.0
      SQL*Plus Files for Instant Client 11.2.0.1.0
      Oracle Net Required Support Files 11.2.0.1.0
      Oracle Database User Interface 2.2.13.0.0
      RDBMS Required Support Files for Instant Client
11.2.0.1.0
      Enterprise Manager Minimal Integration 11.2.0.1.0
      XML Parser for Java 11.2.0.1.0
      Oracle Security Developer Tools 11.2.0.1.0
      Oracle Wallet Manager 11.2.0.1.0
```

```
      Enterprise Manager plugin Common Files 11.2.0.1.0
      Platform Required Support Files 11.2.0.1.0
      Oracle JFC Extended Windowing Toolkit 4.2.36.0.0
      RDBMS Required Support Files 11.2.0.1.0
      Oracle Ice Browser 5.2.3.6.0
      Oracle Help For Java 4.2.9.0.0
      Enterprise Manager Common Files 10.2.0.4.2
      Deinstallation Tool 11.2.0.1.0
      Oracle Java Client 11.2.0.1.0
      Cluster Verification Utility Files 11.2.0.1.0
      Oracle Notification Service (eONS) 11.2.0.1.0
      Oracle LDAP administration 11.2.0.1.0
      Cluster Verification Utility Common Files 11.2.0.1.0
      Oracle Clusterware RDBMS Files 11.2.0.1.0
      Oracle Locale Builder 11.2.0.1.0
      Oracle Globalization Support 11.2.0.1.0
      Buildtools Common Files 11.2.0.1.0
      Oracle RAC Required Support Files-HAS 11.2.0.1.0
      SQL*Plus Required Support Files 11.2.0.1.0
      XDK Required Support Files 11.2.0.1.0
      Agent Required Support Files 10.2.0.4.2
      Parser Generator Required Support Files 11.2.0.1.0
      Precompiler Required Support Files 11.2.0.1.0
      Installation Common Files 11.2.0.1.0
      Required Support Files 11.2.0.1.0
      Oracle JDBC/THIN Interfaces 11.2.0.1.0
      Oracle Multimedia Locator 11.2.0.1.0
      Oracle Multimedia 11.2.0.1.0
      HAS Common Files 11.2.0.1.0
      Assistant Common Files 11.2.0.1.0
      PL/SQL 11.2.0.1.0
      HAS Files for DB 11.2.0.1.0
      Oracle Recovery Manager 11.2.0.1.0
      Oracle Database Utilities 11.2.0.1.0
      Oracle Notification Service 11.2.0.0.0
      SQL*Plus 11.2.0.1.0
      Oracle Netca Client 11.2.0.1.0
      Oracle Net 11.2.0.1.0
      Oracle JVM 11.2.0.1.0
      Oracle Internet Directory Client 11.2.0.1.0
      Oracle Net Listener 11.2.0.1.0
      Cluster Ready Services Files 11.2.0.1.0
      Oracle Database 11g 11.2.0.1.0
--------------------------------------------------------------
---------------


Instantiating scripts for add node (Tuesday, September 1, 2009
2:46:41 PM EDT)
.
1% Done.
```

Practice 4-1: Adding a Third Node to Your Cluster (continued)

```
Instantiation of add node scripts complete

Copying to remote nodes (Tuesday, September 1, 2009 2:46:45 PM
EDT)
...............................................................
.................................
96% Done.
Home copied to new nodes

Saving inventory on nodes (Tuesday, September 1, 2009 2:51:33
PM EDT)
.
100% Done.
Save inventory complete
WARNING:A new inventory has been created on one or more nodes
in this session. However, it has not yet been registered as
the central inventory of this system.
To register the new inventory please run the script at
'/u01/app/oraInventory/orainstRoot.sh' with root privileges on
nodes 'host03'.
If you do not register the inventory, you may not be able to
update or patch the products you installed.
The following configuration scripts need to be executed as the
"root" user in each cluster node.
/u01/app/oraInventory/orainstRoot.sh #On nodes host03
/u01/app/11.2.0/grid/root.sh #On nodes host03
To execute the configuration scripts:
    1. Open a terminal window
    2. Log in as "root"
    3. Run the scripts in each cluster node

The Cluster Node Addition of /u01/app/11.2.0/grid was
successful.
Please check '/tmp/silentInstall.log' for more details.
$
```

9) Connected as the `root` user on your third node using a terminal session, execute the following scripts: `/u01/app/oraInventory/orainstRoot.sh` and `/u01/app/11.2.0/grid/root.sh`.

```
[grid]$ . /home/oracle/labs/st_env.sh
[grid]$ ssh root@${ST_NODE3}
root@host03's password:
Last login: Tue Sep 29 09:59:03 2009 from host01.example.com
# /u01/app/oraInventory/orainstRoot.sh
Creating the Oracle inventory pointer file (/etc/oraInst.loc)
Changing permissions of /u01/app/oraInventory.
Adding read,write permissions for group.
Removing read,write,execute permissions for world.

Changing groupname of /u01/app/oraInventory to oinstall.
```

```
The execution of the script is complete.
#
# /u01/app/11.2.0/grid/root.sh
Running Oracle 11g root.sh script...

The following environment variables are set as:
    ORACLE_OWNER= grid
    ORACLE_HOME=  /u01/app/11.2.0/grid

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.
2009-09-01 15:05:27: Parsing the host name
2009-09-01 15:05:27: Checking for super user privileges
2009-09-01 15:05:27: User has super user privileges
Using configuration parameter file:
/u01/app/11.2.0/grid/crs/install/crsconfig_params
Creating trace directory
LOCAL ADD MODE
Creating OCR keys for user 'root', privgrp 'root'..
Operation successful.
Adding daemon to inittab
CRS-4123: Oracle High Availability Services has been started.
ohasd is starting
CRS-4402: The CSS daemon was started in exclusive mode but
found an active CSS daemon on node host01, number 1, and is
terminating
An active cluster was found during exclusive startup,
restarting to join the cluster
CRS-2672: Attempting to start 'ora.mdnsd' on 'host03'
CRS-2676: Start of 'ora.mdnsd' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.gipcd' on 'host03'
CRS-2676: Start of 'ora.gipcd' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.gpnpd' on 'host03'
CRS-2676: Start of 'ora.gpnpd' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'host03'
CRS-2676: Start of 'ora.cssdmonitor' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.cssd' on 'host03'
CRS-2672: Attempting to start 'ora.diskmon' on 'host03'
CRS-2676: Start of 'ora.diskmon' on 'host03' succeeded
CRS-2676: Start of 'ora.cssd' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.ctssd' on 'host03'
```

```
CRS-2676: Start of 'ora.ctssd' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.drivers.acfs' on 'host03'
CRS-2676: Start of 'ora.drivers.acfs' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.asm' on 'host03'
CRS-2676: Start of 'ora.asm' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'host03'
CRS-2676: Start of 'ora.crsd' on 'host03' succeeded
CRS-2672: Attempting to start 'ora.evmd' on 'host03'
CRS-2676: Start of 'ora.evmd' on 'host03' succeeded
clscfg: EXISTING configuration version 5 detected.
clscfg: version 5 is 11g Release 2.
Successfully accumulated necessary OCR keys.
Creating OCR keys for user 'root', privgrp 'root'..
Operation successful.

host03     2009/09/01 15:09:15
/u01/app/11.2.0/grid/cdata/host03/backup_20090901_150915.olr
Preparing packages for installation...
cvuqdisk-1.0.7-1
Configure Oracle Grid Infrastructure for a Cluster ...
succeeded
Updating inventory properties for clusterware
Starting Oracle Universal Installer...

Checking swap space: must be greater than 500 MB.   Actual
4095 MB    Passed
The inventory pointer is located at /etc/oraInst.loc
The inventory is located at /u01/app/oraInventory
'UpdateNodeList' was successful.
#
```

10) From your first node, check that your cluster is integrated and that the cluster is not divided into separate parts.

```
$ . /home/oracle/labs/st_env.sh
$
$ cluvfy stage -post nodeadd -n $ST_NODE3

Performing post-checks for node addition

Checking node reachability...
Node reachability check passed from node "host01"


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

Checking node connectivity...

Checking hosts config file...
```

Practice 4-1: Adding a Third Node to Your Cluster (continued)

```
Verification of the hosts config file successful

Check: Node connectivity for interface "eth3"
Node connectivity passed for interface "eth3"

Node connectivity check passed


Checking cluster integrity...


Cluster integrity check passed


Checking CRS integrity...

CRS integrity check passed

Checking shared resources...
Shared resources check for node addition passed


Checking node connectivity...

Checking hosts config file...

Verification of the hosts config file successful

Node connectivity passed for subnet "10.216.52.0" with node(s)
host03,host02,host01
TCP connectivity check passed for subnet "10.216.52.0"

Node connectivity passed for subnet "10.216.96.0" with node(s)
host03,host02,host01
TCP connectivity check passed for subnet "10.216.96.0"

Node connectivity passed for subnet "10.196.28.0" with node(s)
host03,host02,host01
TCP connectivity check passed for subnet "10.196.28.0"

Node connectivity passed for subnet "10.196.180.0" with
node(s) host03,host02,host01
TCP connectivity check passed for subnet "10.196.180.0"


Interfaces found on subnet "10.216.52.0" that are likely
candidates for VIP are:
host03 eth0:10.216.54.235
host02 eth0:10.216.54.234
host01 eth0:10.216.54.233
```

Practice 4-1: Adding a Third Node to Your Cluster (continued)

```
Interfaces found on subnet "10.216.96.0" that are likely
candidates for a private interconnect are:
host03 eth1:10.216.100.226
host02 eth1:10.216.96.144
host01 eth1:10.216.101.101

Interfaces found on subnet "10.196.28.0" that are likely
candidates for a private interconnect are:
host03 eth2:10.196.31.17
host02 eth2:10.196.31.16
host01 eth2:10.196.31.15

Interfaces found on subnet "10.196.180.0" that are likely
candidates for a private interconnect are:
host03 eth3:10.196.180.17 eth3:10.196.182.229
eth3:10.196.180.224
host02 eth3:10.196.180.16 eth3:10.196.181.231
eth3:10.196.181.244
host01 eth3:10.196.180.15 eth3:10.196.181.239
eth3:10.196.183.15 eth3:10.196.180.232

Node connectivity check passed


Checking node application existence...

Checking existence of VIP node application (required)
Check passed.

Checking existence of ONS node application (optional)
Check passed.

Checking existence of GSD node application (optional)
Check ignored.

Checking existence of EONS node application (optional)
Check passed.

Checking existence of NETWORK node application (optional)
Check passed.


Checking Single Client Access Name (SCAN)...

Checking name resolution setup for "cl7215-
scan.cl7215.example.com"...

Verification of SCAN VIP and Listener setup passed

User "grid" is not part of "root" group. Check passed
```

```
Checking if Clusterware is installed on all nodes...
Check of Clusterware install passed

Checking if CTSS Resource is running on all nodes...
CTSS resource check passed


Querying CTSS for time offset on all nodes...
Query of CTSS for time offset passed

Check CTSS state started...
CTSS is in Observer state. Switching over to clock
synchronization checks using NTP


Starting Clock synchronization checks using Network Time
Protocol(NTP)...

NTP Configuration file check started...
NTP Configuration file check passed

Checking daemon liveness...
Liveness check passed for "ntpd"

NTP daemon slewing option check passed

NTP daemon's boot time configuration check for slewing option
passed

NTP common Time Server Check started...
Check of common NTP Time Server passed

Clock time offset check from NTP Time Server started...
Clock time offset check passed

Clock synchronization check using Network Time Protocol(NTP)
passed


Oracle Cluster Time Synchronization Services check passed

Post-check for node addition was successful.
$
```

11) Make sure that the FRA and ASCF ASM disk groups are mounted on all three nodes.

```


[grid@host01]$ crsctl stat res ora.FRA.dg -t
--------------------------------------------------------------------
------------------
```

```
NAME           TARGET  STATE        SERVER
STATE_DETAILS
--------------------------------------------------------------
------------------
Local Resources
--------------------------------------------------------------
------------------
ora.FRA.dg
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
               OFFLINE OFFLINE      host03

[grid@host01]$ crsctl stat res ora.ACFS.dg -t
--------------------------------------------------------------
------------------
NAME           TARGET  STATE        SERVER
STATE_DETAILS
--------------------------------------------------------------
------------------
Local Resources
--------------------------------------------------------------
------------------
ora.ACFS.dg
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
               OFFLINE OFFLINE      host03

[grid@host01]$ ssh $ST_NODE3
Last login: Sat Sep 12 13:16:53 2009 from host01.example.com
[grid@host03]$ . oraenv
ORACLE_SID = [grid] ? +ASM3
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid
[grid@host03]$ asmcmd mount ACFS
[grid@host03]$ asmcmd mount FRA
[grid@host03]$ exit
logout
Connection to host03 closed.
[grid@host01]$ crsctl stat res ora.FRA.dg -t
--------------------------------------------------------------
------------------
NAME           TARGET  STATE        SERVER
STATE_DETAILS
--------------------------------------------------------------
------------------
Local Resources
--------------------------------------------------------------
------------------
ora.FRA.dg
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
```

Practice 4-1: Adding a Third Node to Your Cluster (continued)

```
               ONLINE  ONLINE       host03

[grid@host011]$ crsctl stat res ora.ACFS.dg -t
--------------------------------------------------------------
------------------
NAME           TARGET  STATE        SERVER
STATE_DETAILS
--------------------------------------------------------------
------------------
Local Resources
--------------------------------------------------------------
------------------
ora.ACFS.dg
               ONLINE  ONLINE       host01
               ONLINE  ONLINE       host02
               ONLINE  ONLINE       host03
[grid@host01 ~]$
$
```

Practices for Lesson 5

In this practice, you will use Oracle Clusterware to protect the Apache application.

Practice 5-1: Protecting the Apache Application

In this practice, you use Oracle Clusterware to protect the Apache application. To do this, you create an application VIP for Apache (HTTPD), an action script, and a resource.

1) As root, source the environment script `/home/oracle/labs/st_env.sh`. Then verify that the Apache RPMs; `httpd`, `httpd-devel`, and `httpd-manual` are installed on your first two nodes.

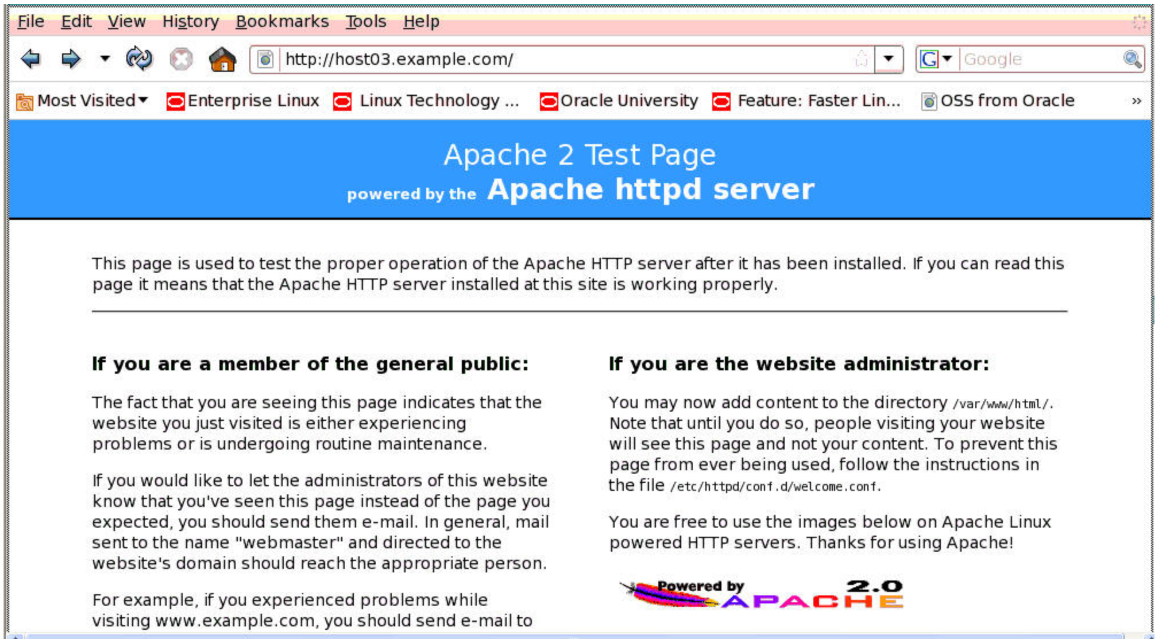
```
# su -
Password: 0racle  <<password not displayed

# . /home/oracle/labs/st_env.sh

# rpm -qa|grep httpd

httpd-2.2.3-22.0.1.el5
httpd-devel-2.2.3-22.0.1.el5
httpd-manual-2.2.3-22.0.1.el5

 [root]# . /home/oracle/labs/st_env.sh

Repeat on second node
 [root] ssh $ST_NODE2 rpm -qa|grep httpd
root@host02's password: 0racle  <<password not displayed
httpd-2.2.3-22.0.1.el5
httpd-devel-2.2.3-22.0.1.el5
httpd-manual-2.2.3-22.0.1.el5
 [root]#
```

2) As the `root` user, start the Apache application on your first node with the `apachectl start` command.

```
# apachectl start
```

*From a VNC session on one of your three node*s, access the Apache test page on your first node. For example, if your first node was named host01, the HTTP address would look something like this:

http://host01.example.com



After you have determined that Apache is working properly, repeat this step on your second host. After you have determined that Apache is working correctly on both nodes, you can stop Apache with the `apachectl stop` command.

3) Create an action script to control the application. This script must be accessible by all nodes on which the application resource can be located.

   a) As the `root` user, create a script on the first node called `apache.scr` in `/usr/local/bin` that will start, stop, check status, and clean up if the application does not exit cleanly. Make sure that the host specified in the `WEBPAGECHECK` variable is your **first** node. Use the `/home/oracle/labs/less_05/apache.scr.tpl` file as a template for creating the script. Make the script executable and test the script.

```
# cp /home/oracle/labs/less_05/apache.tpl
/usr/local/bin/apache.scr
# vi /usr/local/bin/apache.scr

#!/bin/bash

HTTPDCONFLOCATION=/etc/httpd/conf/httpd.conf
WEBPAGECHECK=http://host01.example.com:80/icons/apache_pb.gif
case $1 in
'start')
/usr/sbin/apachectl -k start -f $HTTPDCONFLOCATION
RET=$?
;;
'stop')
/usr/sbin/apachectl -k stop
```

Practice 5-1: Protecting the Apache Application (continued)

```
RET=$?
;;
'clean')
/usr/sbin/apachectl -k stop
RET=$?
;;
'check')
/usr/bin/wget -q --delete-after $WEBPAGECHECK
RET=$?
;;
*)
RET=0
;;
esac
# 0: success; 1 : error
if [ $RET -eq 0 ]; then
exit 0
else
exit 1
fi

Save the file

# chmod 755 /usr/local/bin/apache.scr
# apache.scr start
Verify web page
# apache.scr stop
Web page should no longer display
```

b) As `root`, create a script on the second node called `apache.scr` in `/usr/bin/local` that will start, stop, check status, and clean up if the application does not exit cleanly. Make sure that the host specified in the `WEBPAGECHECK` variable is your **second** node. Use the `/home/oracle/labs/less_05/apache.scr.tpl` file as a template for creating the script. Make the script executable and test the script.

```
# ssh $ST_NODE2
root@host02's password: 0racle << password not displayed
# cp /home/oracle/labs/less_05/apache.tpl
/usr/local/bin/apache.scr
# vi /usr/local/bin/apache.scr

#!/bin/bash

HTTPDCONFLOCATION=/etc/httpd/conf/httpd.conf
WEBPAGECHECK=http://host02.example.com:80/icons/apache_pb.gif
case $1 in
'start')
/usr/sbin/apachectl -k start -f $HTTPDCONFLOCATION
RET=$?
```

```
;;
'stop')
/usr/sbin/apachectl -k stop
RET=$?
;;
'clean')
/usr/sbin/apachectl -k stop
RET=$?
;;
'check')
/usr/bin/wget -q --delete-after $WEBPAGECHECK
RET=$?
;;
*)
RET=0
;;
esac
# 0: success; 1 : error
if [ $RET -eq 0 ]; then
exit 0
else
exit 1
fi
```

Save the file

```
# chmod 755 /usr/local/bin/apache.scr
# apache.scr start
```

Verify web page

```
# apache.scr stop
```

Web page should no longer display

4) Next, you must validate the return code of a check failure using the new script. The Apache server should NOT be running on either node. Run `apache.scr check` and immediately test the return code by issuing an `echo $?` command. This must be run immediately after the `apache.scr check` command because the shell variable `$?` holds the exit code of the previous command run from the shell. An unsuccessful check should return an exit code of 1. You should do this on both nodes.

```
# apache.scr check
# echo $?
1
```

Repeat on second node

Practice 5-1: Protecting the Apache Application (continued)

5) As the `grid` user, create a server pool for the resource called `myApache_sp`. This pool contains your first two hosts and is a child of the Generic pool.

```
$ id
uid=502(grid) gid=501(oinstall)
groups=501(oinstall),504(asmadmin),505(asmdba),506(asmoper)

$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid
$ /u01/app/11.2.0/grid/bin/crsctl add serverpool myApache_sp
-attr "PARENT_POOLS=Generic, SERVER_NAMES=$ST_NODE1 $ST_NODE2"
```

6) Check the status of the new pool on your cluster.

```
$ /u01/app/11.2.0/grid/bin/crsctl status server -f
NAME=host01
STATE=ONLINE
ACTIVE_POOLS=Generic ora.orcl  myApache_sp
STATE_DETAILS=

NAME=host02
STATE=ONLINE
ACTIVE_POOLS=Generic ora.orcl  myApache_sp
STATE_DETAILS=
...
```

7) Add the Apache Resource, which can be called myApache, to the `myApache_sp` subpool that has Generic as a parent. It must be performed as `root` because the resource requires root authority because of listening on the default privileged port 80. Set CHECK_INTERVAL to `30`, RESTART_ATTEMPTS to `2`, and PLACEMENT to `restricted`.

```
# su -
Password: 0racle << Password not displayed
# id
uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel
)
# /u01/app/11.2.0/grid/bin/crsctl add resource myApache -type
cluster_resource -attr
"ACTION_SCRIPT=/usr/local/bin/apache.scr,
PLACEMENT='restricted', SERVER_POOLS=myApache_sp,
CHECK_INTERVAL='30', RESTART_ATTEMPTS='2'"

#
```

8) View the static attributes of the myApache resource with the `crsctl status resource myApache -p -f` command.

```
# /u01/app/11.2.0/grid/bin/crsctl status resource myApache -f
NAME=myApache
TYPE=cluster_resource
STATE=OFFLINE
TARGET=OFFLINE
ACL=owner:root:rwx,pgrp:root:r-x,other::r--
ACTION_FAILURE_TEMPLATE=
ACTION_SCRIPT=/usr/local/bin/apache.scr
ACTIVE_PLACEMENT=0
AGENT_FILENAME=%CRS_HOME%/bin/scriptagent
AUTO_START=restore
CARDINALITY=1
CARDINALITY_ID=0
CHECK_INTERVAL=30
CREATION_SEED=123
CURRENT_RCOUNT=0
DEFAULT_TEMPLATE=
DEGREE=1
DESCRIPTION=
ENABLED=1
FAILOVER_DELAY=0
FAILURE_COUNT=0
FAILURE_HISTORY=
FAILURE_INTERVAL=0
FAILURE_THRESHOLD=0
HOSTING_MEMBERS=
ID=myApache
INCARNATION=0
LAST_FAULT=0
LAST_RESTART=0
LAST_SERVER=
LOAD=1
LOGGING_LEVEL=1
NOT_RESTARTING_TEMPLATE=
OFFLINE_CHECK_INTERVAL=0
PLACEMENT=restricted
PROFILE_CHANGE_TEMPLATE=
RESTART_ATTEMPTS=2
SCRIPT_TIMEOUT=60
SERVER_POOLS=myApache_sp
START_DEPENDENCIES=
START_TIMEOUT=0
STATE_CHANGE_TEMPLATE=
STATE_CHANGE_VERS=0
STATE_DETAILS=
STOP_DEPENDENCIES=
```

Practice 5-1: Protecting the Apache Application (continued)

```
STOP_TIMEOUT=0
UPTIME_THRESHOLD=1h
```

9) Use the `crsctl start resource myApache` command to start the new resource. Use the `crsctl status resource myApache` command to confirm that the resource is online on the first node. If you like, open a browser and point it to your first node as shown in step 2.

```
# /u01/app/11.2.0/grid/bin/crsctl start resource myApache
CRS-2672: Attempting to start 'myApache' on 'host01'
CRS-2676: Start of 'myApache' on 'host01' succeeded

# /u01/app/11.2.0/grid/bin/crsctl status resource myApache
NAME=myApache
TYPE=cluster_resource
TARGET=ONLINE
STATE=ONLINE on host01
```

10) Confirm that Apache is NOT running on your second node. The easiest way to do this is to check for the running `/usr/sbin/httpd -k start -f /etc/httpd/conf/httpd.confd` processes with the `ps` command.

```
# . /home/oracle/labs/st_env.sh
# ssh $ST_NODE2 ps -ef|grep -i "httpd –k"
root@host02's password: 0racle << password is not displayed
#
```

11) Next, simulate a node failure on your first node using the `init` command as `root`. Before issuing the reboot on the first node, open a VNC session on the second node and as the `root` user, execute the `/home/oracle/labs/less_05/monitor.sh` script so that you can monitor the failover.

ON THE FIRST NODE AS THE root USER

```
# init 6        To initiate a reboot, simulating a node failure
```

ON THE second NODE AS THE root USER

```
# cat /home/oracle/labs/less_05/monitor.sh
while true
   do
   ps -ef | grep -i "httpd -k"
 sleep 1
done
```

Practice 5-1: Protecting the Apache Application (continued)

```
# /home/oracle/labs/less_05/monitor.sh      Execute this on the
                                            second node

root   21940 18530  0 11:01 pts/4    00:00:00 grep -i httpd -k
root   21948 18530  0 11:01 pts/4    00:00:00 grep -i httpd -k
root   21951 18530  0 11:01 pts/4    00:00:00 grep -i httpd -k
...
apache   22123 22117  0 11:01 ?        00:00:00
/usr/sbin/httpd -k start -f /etc/httpd/conf/httpd.conf

apache   22124 22117  0 11:01 ?        00:00:00
/usr/sbin/httpd -k start -f /etc/httpd/conf/httpd.conf

apache   22125 22117  0 11:01 ?        00:00:00
/usr/sbin/httpd -k start -f /etc/httpd/conf/httpd.conf
...

Issue a ctl-c to stop the monitoring

#
```

12) Verify the failover from the first node to the second with the `crsctl stat resource myApache –t` command.

```
# /u01/app/11.2.0/grid/bin/crsctl stat resource myApache –t

--------------------------------------------------------------
NAME      TARGET  STATE        SERVER           STATE_DETAILS
--------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------
myApache
      1        ONLINE  ONLINE       host02
```

Practices for Lesson 6

In this lesson you will work with Oracle Clusterware log files and learn to use the `ocrdump` and `cluvfy` utilities.

Practice 6-1: Working with Log Files

In this practice, you will examine the Oracle Clusterware alert log and then package various log files into an archive format suitable to send to My Oracle Support.

1) While connected as the `grid` user to your first node, locate and view the contents of the Oracle Clusterware alert log.

```
$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid

$ . /home/oracle/labs/st_env.sh

$ cd $ORACLE_HOME/log/$ST_NODE1
$ view alert$ST_NODE1.log
Oracle Database 11g Clusterware Release 11.2.0.1.0 -
Production Copyright 1996, 2009 Oracle. All rights reserved.
2009-08-24 14:32:24.580
[client(12578)]CRS-2106:The OLR location
/u01/app/11.2.0/grid/cdata/host01.olr is inaccessible. Details
in /u01/app/11.2.0/grid/log/host01/client/ocrconfig_12578.log.
2009-08-24 14:32:24.933
[client(12578)]CRS-2101:The OLR was formatted using version 3.
2009-08-24 14:32:58.972
[ohasd(12895)]CRS-2112:The OLR service started on node host01.
2009-08-24 14:33:00.090
[ohasd(12895)]CRS-2772:Server 'host01' has been assigned to
pool 'Free'.
2009-08-24 14:33:58.187
[ohasd(12895)]CRS-2302:Cannot get GPnP profile. Error
CLSGPNP_NO_DAEMON (GPNPD daemon is not running).
2009-08-24 14:34:02.853
[cssd(14068)]CRS-1713:CSSD daemon is started in exclusive mode
2009-08-24 14:34:05.230
[cssd(14068)]CRS-1709:Lease acquisition failed for node host01
because no voting file has been configured; Details at
(:CSSNM00031:) in
/u01/app/11.2.0/grid/log/host01/cssd/ocssd.log
2009-08-24 14:34:22.893
[cssd(14068)]CRS-1601:CSSD Reconfiguration complete. Active
nodes are host01
…

:q!
```

2) Navigate to the Oracle Cluster Synchronization Services daemon log directory and determine whether any log archives exist.

```
$ cd ./cssd

$ pwd
```

Practice 6-1: Working with Log Files (continued)

```
/u01/app/11.2.0/grid/log/host01/cssd

$ ls -alt ocssd*
-rw-r--r-- 1 grid oinstall  7564217 Sep  1 14:06 ocssd.log
-rw-r--r-- 1 grid oinstall 52606470 Sep  1 06:06 ocssd.l01
-rw-r--r-- 1 grid oinstall 52527387 Aug 29 22:26 ocssd.l02
-rw-r--r-- 1 root root     52655425 Aug 27 14:18 ocssd.l03
-rw-r--r-- 1 grid oinstall      114 Aug 24 14:34 ocssd.trc
```

3) Switch to the `root` user and set up the environment variables for the Grid Infrastructure. Change to the `/home/oracle/labs` directory and run the `diagcollection.pl` script to gather all log files that can be send to My Oracle Support for problem analysis.

```
$ su -
Password: 0racle << Password is not displayed

# . oraenv
ORACLE_SID = [root] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid

# cd /home/oracle/labs

# diagcollection.pl --collect --crshome /u01/app/11.2.0/grid
Production Copyright 2004, 2008, Oracle.  All rights reserved
Cluster Ready Services (CRS) diagnostic collection tool
The following CRS diagnostic archives will be created in the
local directory.
crsData_host01_20090901_1413.tar.gz -> logs,traces and cores
from CRS home. Note: core files will be packaged only with the
--core option.
ocrData_host01_20090901_1413.tar.gz -> ocrdump, ocrcheck etc
coreData_host01_20090901_1413.tar.gz -> contents of CRS core
files in text format

osData_host01_20090901_1413.tar.gz -> logs from Operating
System
Collecting crs data
/bin/tar:
log/host01/agent/crsd/orarootagent_root/orarootagent_root.log:
file changed as we read it
Collecting OCR data
Collecting information from core files
No corefiles found
Collecting OS logs
```

4) List the resulting log file archives that were generated with the `diagcollection.pl` script.

```
# ls -la *tar.gz
```

Practice 6-1: Working with Log Files (continued)

```
-rw-r--r-- 1 root root 89083873 Sep  1 14:17
crsData_host01_20090901_1413.tar.gz
-rw-r--r-- 1 root root    11821 Sep  1 14:18
ocrData_host01_20090901_1413.tar.gz
-rw-r--r-- 1 root root    98372 Sep  1 14:18
osData_host01_20090901_1413.tar.gz
```

5) Exit the switch user command to return to the `grid` account.

```
# exit
$
```

Practice 6-2: Working with OCRDUMP

In this practice, you will work with the OCRDUMP utility and dump the binary file into both text and XML representations.

1) While connected to the `grid` account, dump the contents of the OCR to the standard output and count the number of lines of output.

```
$ ocrdump -stdout | wc -l
469
```

2) Switch to the `root` user, dump the contents of the OCR to the standard output and count the number of lines of output. Compare your results with the previous step. How do the results differ?

```
$ su -
Password: 0racle << Password is not displayed

# . oraenv
ORACLE_SID = [root] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid

# ocrdump -stdout | wc -l
3243
```

3) Dump the first 25 lines of the OCR to standard output using XML format.

```
# ocrdump -stdout -xml | head -25
<OCRDUMP>

<TIMESTAMP>09/01/2009 16:55:11</TIMESTAMP>
<COMMAND>/u01/app/11.2.0/grid/bin/ocrdump.bin -stdout -xml
</COMMAND>

<KEY>
<NAME>SYSTEM</NAME>
<VALUE_TYPE>UNDEF</VALUE_TYPE>
<VALUE><![CDATA[]]></VALUE>
<USER_PERMISSION>PROCR_ALL_ACCESS</USER_PERMISSION>
<GROUP_PERMISSION>PROCR_READ</GROUP_PERMISSION>
<OTHER_PERMISSION>PROCR_READ</OTHER_PERMISSION>
<USER_NAME>root</USER_NAME>
<GROUP_NAME>root</GROUP_NAME>

<KEY>
<NAME>SYSTEM.version</NAME>
<VALUE_TYPE>UB4 (10)</VALUE_TYPE>
<VALUE><![CDATA[5]]></VALUE>
<USER_PERMISSION>PROCR_ALL_ACCESS</USER_PERMISSION>
<GROUP_PERMISSION>PROCR_READ</GROUP_PERMISSION>
<OTHER_PERMISSION>PROCR_READ</OTHER_PERMISSION>
<USER_NAME>root</USER_NAME>
<GROUP_NAME>root</GROUP_NAME>
```

Practice 6-2: Working with OCRDUMP (continued)

4) Create an XML file dump of the OCR in the `/home/oracle/labs` directory. Name the dump file `ocr_current_dump.xml`.

```
# ocrdump -xml /home/oracle/labs/ocr_current_dump.xml
```

5) Find the node and directory that contains the automatic backup of the OCR from 24 hours ago.

```
# ocrconfig -showbackup

host02     2009/09/01 16:17:55
/u01/app/11.2.0/grid/cdata/cluster01/backup00.ocr

host02     2009/09/01 12:17:54
/u01/app/11.2.0/grid/cdata/cluster01/backup01.ocr

host02     2009/09/01 08:17:53
/u01/app/11.2.0/grid/cdata/cluster01/backup02.ocr

host02     2009/08/31 04:17:44
/u01/app/11.2.0/grid/cdata/cluster01/day.ocr

host01     2009/08/24 18:53:34
/u01/app/11.2.0/grid/cdata/cluster01/week.ocr

host02     2009/08/27 21:37:49
/u01/app/11.2.0/grid/cdata/cluster01/backup_20090827_213749.oc
r
```

6) Copy the 24 hour old automatic backup of the OCR into the `/home/oracle/labs` directory. This is not a dump, but rather an actual backup of the OCR. (If the daily backup of the OCR is not there use the oldest backup on the list.)
Note: It may be necessary to use `scp` if the file is located on a different node. Be sure to use your cluster name in place of cluster01 in the path.

```
# . /home/oracle/labs/st_env.sh
# cp /u01/app/11.2.0/grid/cdata/cluster01/day.ocr
/home/oracle/labs/day.ocr

>>>> Or <<<<<

# scp $ST_NODE2:/u01/app/11.2.0/grid/cdata/cluster01/day.ocr
/home/oracle/labs/day.ocr
root@host02's password: 0racle << Password is not displayed
day.ocr                    100% 7436KB   7.3MB/s   00:00
```

7) Dump the contents of the `day.ocr` backup OCR file in XML format saving the file in the `/home/oracle/labs` directory. Name the file `day_ocr.xml`.

```
# ocrdump -xml -backupfile /home/oracle/labs/day.ocr
/home/oracle/labs/day_ocr.xml
```

Practice 6-2: Working with OCRDUMP (continued)

8) Compare the differences between the `day_ocr.xml` file and the `ocr_current_dump.xml` file to determine all changes made to the OCR in the last 24 hours. Exit the switch user command when done.

```
# diff /home/oracle/labs/day_ocr.xml
/home/oracle/labs/ocr_current_dump.xml
3,5c3,4
< <TIMESTAMP>09/01/2009 17:25:28</TIMESTAMP>
< <DEVICE>/home/oracle/labs/day.ocr</DEVICE>
< <COMMAND>/u01/app/11.2.0/grid/bin/ocrdump.bin -xml -
backupfile /home/oracle/labs/day.ocr
/home/oracle/labs/day_ocr.xml </COMMAND>
---
> <TIMESTAMP>09/01/2009 17:17:56</TIMESTAMP>
> <COMMAND>/u01/app/11.2.0/grid/bin/ocrdump.bin -xml
/home/oracle/labs/ocr_current_dump2 </COMMAND>
8438c8437
< <VALUE><![CDATA[2009/08/31 00:17:43]]></VALUE>
---
> <VALUE><![CDATA[2009/09/01 16:17:55]]></VALUE>
8486c8485
< <VALUE><![CDATA[2009/08/30 20:17:42]]></VALUE>
---
> <VALUE><![CDATA[2009/09/01 12:17:54]]></VALUE>
8534c8533
< <VALUE><![CDATA[2009/08/30 16:17:40]]></VALUE>
---
> <VALUE><![CDATA[2009/09/01 08:17:53]]></VALUE>
8582c8581
< <VALUE><![CDATA[2009/08/29 04:17:29]]></VALUE>
---
> <VALUE><![CDATA[2009/08/31 04:17:44]]></VALUE>
8630c8629
< <VALUE><![CDATA[2009/08/30 04:17:36]]></VALUE>
---
> <VALUE><![CDATA[2009/09/01 04:17:51]]></VALUE>

# exit
$
```

Practice 6-3: Working with CLUVFY

In this practice, you will work with CLUVFY to verify the state of various cluster components.

1) Determine the location of the `cluvfy` utility and its configuration file.

```
$ . /home/oracle/labs/st_env.sh
$ which cluvfy
/u01/app/11.2.0/grid/bin/cluvfy

$ cd $ORACLE_HOME/cv/admin
$ pwd
/u01/app/11.2.0/grid/cv/admin

$ cat cvu_config
# Configuration file for Cluster Verification Utility(CVU)
# Version: 011405
#
# NOTE:
# 1._ Any line without a '=' will be ignored
# 2._ Since the fallback option will look into the environment
variables,
#     please have a component prefix(CV_) for each property to
define a
#     namespace.
#


#Nodes for the cluster. If CRS home is not installed, this
list will be
#picked up when -n all is mentioned in the commandline
argument.
#CV_NODE_ALL=

#if enabled, cvuqdisk rpm is required on all nodes
CV_RAW_CHECK_ENABLED=TRUE

# Fallback to this distribution id
CV_ASSUME_DISTID=OEL4

# Whether X-Windows check should be performed for user
equivalence with SSH
#CV_XCHK_FOR_SSH_ENABLED=TRUE

# To override SSH location
#ORACLE_SRVM_REMOTESHELL=/usr/bin/ssh

# To override SCP location
#ORACLE_SRVM_REMOTECOPY=/usr/bin/scp
```

Practice 6-3: Working with CLUVFY (continued)

2) Display the stage options and stage names that can be used with the `cluvfy` utility.

```
$ cluvfy stage -list


USAGE:
cluvfy stage {-pre|-post} <stage-name> <stage-specific
options>  [-verbose]

Valid stage options and stage names are:
        -post hwos    :  post-check for hardware and operating
system
        -pre  cfs     :  pre-check for CFS setup
        -post cfs     :  post-check for CFS setup
        -pre  crsinst :  pre-check for CRS installation
        -post crsinst :  post-check for CRS installation
        -pre  hacfg   :  pre-check for HA configuration
        -post hacfg   :  post-check for HA configuration
        -pre  dbinst  :  pre-check for database installation
        -pre  acfscfg  :  pre-check for ACFS Configuration.
        -post acfscfg  :  post-check for ACFS Configuration.
        -pre  dbcfg   :  pre-check for database configuration
        -pre  nodeadd :  pre-check for node addition.
        -post nodeadd :  post-check for node addition.
        -post nodedel :  post-check for node deletion.
```

3) Perform a postcheck for the ACFS configuration on all nodes.

```
$ cluvfy stage -post acfscfg -n $ST_NODE_LIST

Performing post-checks for ACFS Configuration

Checking node reachability...
Node reachability check passed from node "host01"


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

Task ACFS Integrity check started...

Starting check to see if ASM is running on all cluster
nodes...

ASM Running check passed. ASM is running on all cluster nodes

Starting Disk Groups check to see if at least one Disk Group
configured...
Disk Group Check passed. At least one Disk Group configured

Task ACFS Integrity check passed
```

```
UDev attributes check for ACFS started...
UDev attributes check passed for ACFS


Post-check for ACFS Configuration was successful.
```

4) Display a list of the component names that can be checked with the `cluvfy` utility.

```
$ cluvfy comp -list


USAGE:
cluvfy comp  <component-name> <component-specific options>  [-
verbose]

Valid components are:
        nodereach : checks reachability between nodes
        nodecon   : checks node connectivity
        cfs       : checks CFS integrity
        ssa       : checks shared storage accessibility
        space     : checks space availability
        sys       : checks minimum system requirements
        clu       : checks cluster integrity
        clumgr    : checks cluster manager integrity
        ocr       : checks OCR integrity
        olr       : checks OLR integrity
        ha        : checks HA integrity
        crs       : checks CRS integrity
        nodeapp   : checks node applications existence
        admprv    : checks administrative privileges
        peer      : compares properties with peers
        software  : checks software distribution
        asm       : checks ASM integrity
        acfs       : checks ACFS integrity
        gpnp      : checks GPnP integrity
        gns       : checks GNS integrity
        scan      : checks SCAN configuration
        ohasd     : checks OHASD integrity
        clocksync      : checks Clock Synchronization
        vdisk      : check Voting Disk Udev settings
```

5) Display the syntax usage help for the space component check of the `cluvfy` utility.

```
$ cluvfy comp space -help


USAGE:
cluvfy comp space  [-n <node_list>] -l <storage_location>
                   -z <disk_space>{B|K|M|G}  [-verbose]
 <node_list> is the comma separated list of non-domain
qualified nodenames, on which the test should be conducted. If
```

```
"all" is specified, then all the nodes in the cluster will be
used for verification.
 <storage_location> is the storage path.
 <disk_space> is the required disk space, in units of
bytes(B),kilobytes(K),megabytes(M) or gigabytes(G).

DESCRIPTION:
Checks for free disk space at the location provided by '-l'
option on all the nodes in the nodelist. If no '-n' option is
given, local node is used for this check.
```

6) Verify that on each node of the cluster that the `/tmp` directory has at least 200 MB of free space in it using the `cluvfy` utility. Use verbose output.

```
$ cluvfy comp space -n $ST_NODE_LIST2 -l /tmp -z 200M -verbose

Verifying space availability

Checking space availability...

Check: Space available on "/tmp"
 Node Name  Available               Required          Comment
 --------  ----------------------  -----------------  --------
 host02    29.14GB (3.055228E7KB)  200MB (204800.0KB)   passed
 host01    29.74GB (3.118448E7KB)  200MB (204800.0KB)   passed
 host03    29.69GB (3.1132244E7KB) 200MB (204800.0KB)   passed
Result: Space availability check passed for "/tmp"
Verification of space availability was successful.
```

Practices for Lesson 7

In these practices, you will adjust ASM initialization parameters, stop and start instances, and monitor the status of instances.

Practice 7-1: Administering ASM Instances

In this practice, you adjust initialization parameters in the SPFILE, and stop and start the ASM instances on local and remote nodes.

1) Disk groups are reconfigured occasionally to move older data to slower disks. Even though these operations occur at scheduled maintenance times in off-peak hours, the rebalance operations do not complete before regular operations resume. There is some performance impact to the regular operations. The setting for the `ASM_POWER_LIMIT` initialization parameter determines the speed of the rebalance operation. Determine the current setting and increase the speed by 2.

   a) Open a terminal window on the first node, become the `grid` user, and set the environment to use the +ASM1 instance. Connect to the +ASM1 instance as `SYS` with the `SYSASM` privilege. What is the setting for ASM_POWER_LIMIT?

```
[oracle]$ su - grid
Password: 0racle  << password is not displayed
[grid]$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid
[grid]$ sqlplus / as sysasm

SQL*Plus: Release 11.2.0.1.0 Production on Thu Aug 27 09:57:16
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Real Application Clusters and Automatic Storage
Management options

SQL> show parameter ASM_POWER_LIMIT

NAME             TYPE        VALUE
---------------- ----------- ------------------------------
asm_power_limit  integer     1
SQL>
```

   b) This installation uses an SPFILE. Use the `ALTER SYSTEM` command to change the ASM_POWER_LIMIT for all nodes.

```
SQL> show parameter SPFILE

NAME             TYPE        VALUE
---------------- ----------- ------------------------------
spfile           string      +DATA/cluster01/asmparameterfi
                             le/registry.253.695669633
```

```
SQL> ALTER SYSTEM set ASM_POWER_LIMIT=3 SCOPE=BOTH SID='*';

System altered.

SQL> show parameter ASM_POWER_LIMIT

NAME             TYPE        VALUE
---------------- ----------- ------------------------------
asm_power_limit  integer     3
SQL>
```

2) You have decided that due to other maintenance operations you want one instance +ASM1 to handle the bulk of the rebalance operation, so you will set the ASM POWER_LIMIT to 1 on instance +ASM2 and 5 on instance +ASM1.

```
SQL> ALTER SYSTEM set ASM_POWER_LIMIT=1 SCOPE=BOTH
  2>         SID='+ASM2';

System altered.

SQL> ALTER SYSTEM set ASM_POWER_LIMIT=5 SCOPE=BOTH
  2>         SID='+ASM1';

System altered

SQL> show parameter ASM_POWER_LIMIT

NAME             TYPE        VALUE
---------------- ----------- ------------------------------
asm_power_limit  integer     5

SQL> column NAME format A16
SQL> column VALUE format 999999
SQL> select inst_id, name, value from GV$PARAMETER
  2> where name like 'asm_power_limit';

   INST_ID NAME             VALUE
---------- ---------------- ------
         1 asm_power_limit  5

         2 asm_power_limit  1

         3 asm_power_limit  3
SQL>
```

3) Exit the SQL*Plus application.

```
SQL> exit
```

4) The ASM instance and all associated applications, ACFS, database, and listener on one node must be stopped for a maintenance operation to the physical cabling. Stop all the applications, ASM, and listener associated with +ASM2 using srvctl.

   a) In a new terminal window, as the `oracle` user, stop Enterprise Manager on your first node.

```
[oracle]$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmount/11.2.0/sharedhome/dbhome
_1 is /u01/app/oracle
[oracle]$ export ORACLE_UNQNAME=orcl
[oracle]$ . /home/oracle/labs/st_env.sh
[oracle]$ emctl stop dbconsole
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009 Oracle Corporation.  All rights
reserved.
https://host01.example.com:1158/em/console/aboutApplication
Stopping Oracle Enterprise Manager 11g Database Control ...
 ...  Stopped.
```

   b) Stop the `orcl` database.

```
[oracle]$ srvctl stop instance -d orcl -n $ST_NODE1
```

   c) Verify that the database is stopped on ST_NODE1. The `pgrep` command shows that no `orcl` background processes are running.

```
[oracle]$ pgrep -lf orcl
```

   d) In a terminal window, become the `grid` OS user, set the oracle environment, set the class environment, and then stop the ASM instance +ASM using the `srvctl stop asm -n $STNODE` command.

```
[oracle]$ su - grid
Password: 0racle  << password is not displayed
[grid@host01 ~]$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid
[grid]$ . /home/oracle/labs/st_env.sh
[grid]$ srvctl stop asm -n $ST_NODE1
PRCR-1014 : Failed to stop resource ora.asm
PRCR-1065 : Failed to stop resource ora.asm
CRS-2529: Unable to act on 'ora.asm' because that would
require stopping or relocating 'ora.DATA.dg', but the force
option was not specified
```

   e) Attempt to stop the ASM instance on ST_NODE1 using the force option, `-f`.

```
[grid]$ srvctl stop asm -n $ST_NODE1 -f
```

```
PRCR-1014 : Failed to stop resource ora.asm
PRCR-1065 : Failed to stop resource ora.asm
CRS-2670: Unable to start/relocate 'ora.asm' because
'ora.ACFS.dg' has a stop-time 'hard' dependency on it
CRS-0222: Resource 'ora.ACFS.dg' has dependency error.
CRS-2670: Unable to start/relocate 'ora.ACFS.dg' because
'ora.acfs.dbhome1.acfs' has a stop-time 'hard' dependency on
it
CRS-0245:  User doesn't have enough privilege to perform the
operation
```

f) The ACFS file system is dependent on the disk group, so `root` must dismount the file system. Become `root` and dismount the `sharedhome` file system. Then attempt to stop ASM. Why does it still fail? Check the mounted volumes with the `df -h` command. (Your devices may be different than those shown here.)

```
[grid]$ su - root
Password: 0racle  << password is not displayed
[root]# . oraenv
ORACLE_SID = [root] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid
[root@host01 ~]# . /home/oracle/labs/st_env.sh
[root@host01 ~]# srvctl stop asm -n $ST_NODE1 -f
PRCR-1014 : Failed to stop resource ora.asm
PRCR-1065 : Failed to stop resource ora.asm
CRS-2673: Attempting to stop 'ora.asm' on 'host01'
ORA-15097: cannot SHUTDOWN ASM instance with connected client
CRS-2675: Stop of 'ora.asm' on 'host01' failed

CRS-2675: Stop of 'ora.asm' on 'host01' failed

[root]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
                      9.7G  4.1G  5.1G  45% /
/dev/xvda1             99M   20M   74M  22% /boot
tmpfs                 1.1G  154M  871M  16% /dev/shm
/dev/mapper/VolGroup01-LogVol00
                       30G  4.5G   24G  16% /u01
nnn.nnn.nnn.nnn:/mnt/shareddisk01/software/software
                       60G   40G   17G  71% /mnt/software
[root]#
```

g) The `df -h` command shows that the `/u01/app/oracle/acfsmount/sharedhome` file system has been dismounted, but the ASM instance still has clients dependent on it. In this case, the Oracle clusterware is dependent on the ASM instance because the OCR is in the ASM disk group.

h) Stop the ASM instance with `crsctl stop cluster -n $ST_NODE1` command. This command will stop all the Cluster services on the node.

```
[root]# crsctl stop cluster -n $ST_NODE1
CRS-2673: Attempting to stop 'ora.crsd' on 'host01'
CRS-2790: Starting shutdown of Cluster Ready Services-managed
resources on 'host01'
CRS-2673: Attempting to stop 'ora.gns' on 'host01'
CRS-2673: Attempting to stop 'ora.LISTENER_SCAN3.lsnr' on
'host01'
CRS-2673: Attempting to stop 'ora.LISTENER.lsnr' on 'host01'
CRS-2673: Attempting to stop 'ora.asm' on 'host01'
CRS-2673: Attempting to stop 'ora.LISTENER_SCAN2.lsnr' on
'host01'
CRS-2677: Stop of 'ora.asm' on 'host01' succeeded
CRS-2677: Stop of 'ora.LISTENER_SCAN3.lsnr' on 'host01'
succeeded
CRS-2673: Attempting to stop 'ora.scan3.vip' on 'host01'
CRS-2677: Stop of 'ora.scan3.vip' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.scan3.vip' on 'host02'
CRS-2677: Stop of 'ora.LISTENER.lsnr' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.host01.vip' on 'host01'
CRS-2677: Stop of 'ora.host01.vip' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.host01.vip' on 'host02'
CRS-2677: Stop of 'ora.LISTENER_SCAN2.lsnr' on 'host01'
succeeded
CRS-2673: Attempting to stop 'ora.scan2.vip' on 'host01'
CRS-2677: Stop of 'ora.scan2.vip' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.scan2.vip' on 'host02'
CRS-2677: Stop of 'ora.gns' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.gns.vip' on 'host01'
CRS-2677: Stop of 'ora.gns.vip' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.gns.vip' on 'host02'
CRS-2676: Start of 'ora.scan2.vip' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.LISTENER_SCAN2.lsnr' on
'host02'
CRS-2676: Start of 'ora.gns.vip' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.gns' on 'host02'
CRS-2676: Start of 'ora.scan3.vip' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.LISTENER_SCAN3.lsnr' on
'host02'
CRS-2674: Start of 'ora.host01.vip' on 'host02' failed
CRS-2679: Attempting to clean 'ora.host01.vip' on 'host02'
CRS-2674: Start of 'ora.gns' on 'host02' failed
CRS-2679: Attempting to clean 'ora.gns' on 'host02'
CRS-2676: Start of 'ora.LISTENER_SCAN2.lsnr' on 'host02'
succeeded
CRS-2676: Start of 'ora.LISTENER_SCAN3.lsnr' on 'host02'
succeeded
CRS-2681: Clean of 'ora.gns' on 'host02' succeeded
CRS-2681: Clean of 'ora.host01.vip' on 'host02' succeeded
```

```
CRS-2673: Attempting to stop 'ora.ons' on 'host01'
CRS-2673: Attempting to stop 'ora.eons' on 'host01'
CRS-2677: Stop of 'ora.ons' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.net1.network' on 'host01'
CRS-2677: Stop of 'ora.net1.network' on 'host01' succeeded
CRS-2677: Stop of 'ora.eons' on 'host01' succeeded
CRS-2792: Shutdown of Cluster Ready Services-managed resources
on 'host01' has completed
CRS-2677: Stop of 'ora.crsd' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.cssdmonitor' on 'host01'
CRS-2673: Attempting to stop 'ora.ctssd' on 'host01'
CRS-2673: Attempting to stop 'ora.evmd' on 'host01'
CRS-2673: Attempting to stop 'ora.asm' on 'host01'
CRS-2677: Stop of 'ora.cssdmonitor' on 'host01' succeeded
CRS-2677: Stop of 'ora.evmd' on 'host01' succeeded
CRS-2677: Stop of 'ora.ctssd' on 'host01' succeeded
CRS-2677: Stop of 'ora.asm' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'host01'
CRS-2677: Stop of 'ora.cssd' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.diskmon' on 'host01'
CRS-2677: Stop of 'ora.diskmon' on 'host01' succeeded
[root]#
```

i) Confirm that the listener has been stopped.

```
[root]# lsnrctl status listener

LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 28-AUG-
2009 08:46:51

Copyright (c) 1991, 2009, Oracle.  All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER)))
TNS-12541: TNS:no listener
 TNS-12560: TNS:protocol adapter error
  TNS-00511: No listener
   Linux Error: 2: No such file or directory
```

5) Restart all the cluster resources on node ST_NODE1.

```
[root]# crsctl start cluster -n $ST_NODE1
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'host01'
CRS-2676: Start of 'ora.cssdmonitor' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.cssd' on 'host01'
CRS-2672: Attempting to start 'ora.diskmon' on 'host01'
CRS-2676: Start of 'ora.diskmon' on 'host01' succeeded
CRS-2676: Start of 'ora.cssd' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.ctssd' on 'host01'
CRS-2676: Start of 'ora.ctssd' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.asm' on 'host01'
CRS-2672: Attempting to start 'ora.evmd' on 'host01'
```

Practice 7-1: Administering ASM Instances (continued)

```
CRS-2676: Start of 'ora.evmd' on 'host01' succeeded
CRS-2676: Start of 'ora.asm' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'host01'
CRS-2676: Start of 'ora.crsd' on 'host01' succeeded
```

6) Verify that the resources, database, and Enterprise manager are restarted on ST_NODE1. The `crsctl status resource -n $ST_NODE1` command shows that ASM is online and the ACFS volume dbhome1 is mounted.

```
[root]# crsctl status resource -n $ST_NODE1
NAME=ora.ACFS.dg
TYPE=ora.diskgroup.type
TARGET=ONLINE
STATE=ONLINE

NAME=ora.DATA.dg
TYPE=ora.diskgroup.type
TARGET=ONLINE
STATE=ONLINE

NAME=ora.FRA.dg
TYPE=ora.diskgroup.type
TARGET=ONLINE
STATE=ONLINE

NAME=ora.LISTENER.lsnr
TYPE=ora.listener.type
TARGET=ONLINE
STATE=ONLINE

NAME=ora.LISTENER_SCAN1.lsnr
TYPE=ora.scan_listener.type
CARDINALITY_ID=1
TARGET=ONLINE
STATE=ONLINE

NAME=ora.acfs.dbhome1.acfs
TYPE=ora.acfs.type
TARGET=ONLINE
STATE=ONLINE

NAME=ora.asm
TYPE=ora.asm.type
TARGET=ONLINE
STATE=ONLINE

NAME=ora.eons
TYPE=ora.eons.type
TARGET=ONLINE
STATE=ONLINE
```

Practice 7-1: Administering ASM Instances (continued)

```
NAME=ora.gsd
TYPE=ora.gsd.type
TARGET=OFFLINE
STATE=OFFLINE

NAME=ora.host01.vip
TYPE=ora.cluster_vip_net1.type
CARDINALITY_ID=1
TARGET=ONLINE
STATE=ONLINE

NAME=ora.net1.network
TYPE=ora.network.type
TARGET=ONLINE
STATE=ONLINE

NAME=ora.ons
TYPE=ora.ons.type
TARGET=ONLINE
STATE=ONLINE

NAME=ora.registry.acfs
TYPE=ora.registry.acfs.type
TARGET=ONLINE
STATE=ONLINE

NAME=ora.scan1.vip
TYPE=ora.scan_vip.type
CARDINALITY_ID=1
TARGET=ONLINE
STATE=ONLINE

[root]#
```

7) In a terminal window, on the first node as the `oracle` user, start the `orcl` instance and Enterprise Manager on ST_NODE1.

```
[oracle]$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmount/11.2.0/sharedhome/dbhome
_1 is /u01/app/oracle
[oracle]$ . /home/oracle/labs/st_env.sh
[oracle]$ srvctl start instance -d orcl -n $ST_NODE1
[oracle]$ export ORACLE_UNQNAME=orcl
[oracle]$ emctl start dbconsole
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009 Oracle Corporation.  All rights
reserved.
https://host01.example.com:1158/em/console/aboutApplication
```

```
Starting Oracle Enterprise Manager 11g Database Control
............ started.
------------------------------------------------------------------
----
Logs are generated in directory
/u01/app/oracle/acfsmount/11.2.0/sharedhome/host01_orcl/sysman
/log
```

8) Determine the Enterprise Manager DB control configuration on the cluster. Notice that dbconsole is running on your first node, an agent is running on your second node, and no EM components have been started on the third node.

```
[oracle]$ emca -displayConfig dbcontrol -cluster

STARTED EMCA at Aug 28, 2009 9:27:22 AM
EM Configuration Assistant, Version 11.2.0.0.2 Production
Copyright (c) 2003, 2005, Oracle.  All rights reserved.

Enter the following information:
Database unique name: orcl
Service name: orcl
Do you wish to continue? [yes(Y)/no(N)]: Y
Aug 28, 2009 9:27:31 AM oracle.sysman.emcp.EMConfig perform
INFO: This operation is being logged at
/u01/app/oracle/cfgtoollogs/emca/orcl/emca_2009_08_28_09_27_21
.log.
Aug 28, 2009 9:27:36 AM oracle.sysman.emcp.EMDBPostConfig
showClusterDBCAgentMessage
INFO:
****************  Current Configuration  ****************
 INSTANCE            NODE           DBCONTROL_UPLOAD_HOST
----------        ----------        ---------------------

orcl              host01            host01.example.com
orcl              host02            host01.example.com


Enterprise Manager configuration completed successfully
FINISHED EMCA at Aug 28, 2009 9:27:36 AM
```

9) In a terminal window, become the `root` user. Set the grid home environment and set the practice environment variables. Stop all the cluster resources on ST_NODE2.

```
[oracle]$ su - root
Password: 0racle  << password is not displayed
[root]# . oraenv
ORACLE_SID = [root] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid
[root]# . ~oracle/labs/st_env.sh
[root]# crsctl stop cluster -n $ST_NODE2
CRS-2673: Attempting to stop 'ora.crsd' on 'host02'
```

```
CRS-2790: Starting shutdown of Cluster Ready Services-managed
resources on 'host02'
CRS-2673: Attempting to stop 'ora.LISTENER_SCAN1.lsnr' on
'host02'
CRS-2673: Attempting to stop 'ora.orcl.db' on 'host02'
CRS-2673: Attempting to stop 'ora.registry.acfs' on 'host02'
CRS-2673: Attempting to stop 'ora.LISTENER.lsnr' on 'host02'
CRS-2677: Stop of 'ora.LISTENER_SCAN1.lsnr' on 'host02'
succeeded
CRS-2673: Attempting to stop 'ora.scan1.vip' on 'host02'
CRS-2677: Stop of 'ora.LISTENER.lsnr' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.host02.vip' on 'host02'
CRS-2677: Stop of 'ora.scan1.vip' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.scan1.vip' on 'host01'
CRS-2677: Stop of 'ora.host02.vip' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.host02.vip' on 'host01'
CRS-2677: Stop of 'ora.registry.acfs' on 'host02' succeeded
CRS-2676: Start of 'ora.host02.vip' on 'host01' succeeded
CRS-2676: Start of 'ora.scan1.vip' on 'host01' succeeded
CRS-2672: Attempting to start 'ora.LISTENER_SCAN1.lsnr' on
'host01'
CRS-2676: Start of 'ora.LISTENER_SCAN1.lsnr' on 'host01'
succeeded
CRS-2677: Stop of 'ora.orcl.db' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.acfs.dbhome1.acfs' on
'host02'
CRS-2673: Attempting to stop 'ora.DATA.dg' on 'host02'
CRS-2677: Stop of 'ora.DATA.dg' on 'host02' succeeded
CRS-2677: Stop of 'ora.acfs.dbhome1.acfs' on 'host02'
succeeded
CRS-2673: Attempting to stop 'ora.ACFS.dg' on 'host02'
CRS-2677: Stop of 'ora.ACFS.dg' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.asm' on 'host02'
CRS-2677: Stop of 'ora.asm' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.ons' on 'host02'
CRS-2673: Attempting to stop 'ora.eons' on 'host02'
CRS-2677: Stop of 'ora.ons' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.net1.network' on 'host02'
CRS-2677: Stop of 'ora.net1.network' on 'host02' succeeded
CRS-2677: Stop of 'ora.eons' on 'host02' succeeded
CRS-2792: Shutdown of Cluster Ready Services-managed resources
on 'host02' has completed
CRS-2677: Stop of 'ora.crsd' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.cssdmonitor' on 'host02'
CRS-2673: Attempting to stop 'ora.ctssd' on 'host02'
CRS-2673: Attempting to stop 'ora.evmd' on 'host02'
CRS-2673: Attempting to stop 'ora.asm' on 'host02'
CRS-2677: Stop of 'ora.cssdmonitor' on 'host02' succeeded
CRS-2677: Stop of 'ora.evmd' on 'host02' succeeded
CRS-2677: Stop of 'ora.ctssd' on 'host02' succeeded
CRS-2677: Stop of 'ora.asm' on 'host02' succeeded
```

```
CRS-2673: Attempting to stop 'ora.cssd' on 'host02'
CRS-2677: Stop of 'ora.cssd' on 'host02' succeeded
CRS-2673: Attempting to stop 'ora.diskmon' on 'host02'
CRS-2677: Stop of 'ora.diskmon' on 'host02' succeeded
[root]#
```

10) What is the status of the orcl database on your cluster?

```
[root]# srvctl status database -d orcl
Instance orcl1 is running on node gr7212
Instance orcl2 is not running on node gr7213
```

11) As the `root` user on your first node, start the cluster on your second node (`ST_NODE2`).

```
[root@host01 ~]# crsctl start cluster -n $ST_NODE2
CRS-2672: Attempting to start 'ora.cssdmonitor' on 'host02'
CRS-2676: Start of 'ora.cssdmonitor' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.cssd' on 'host02'
CRS-2672: Attempting to start 'ora.diskmon' on 'host02'
CRS-2676: Start of 'ora.diskmon' on 'host02' succeeded
CRS-2676: Start of 'ora.cssd' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.ctssd' on 'host02'
CRS-2676: Start of 'ora.ctssd' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.evmd' on 'host02'
CRS-2672: Attempting to start 'ora.asm' on 'host02'
CRS-2676: Start of 'ora.evmd' on 'host02' succeeded
CRS-2676: Start of 'ora.asm' on 'host02' succeeded
CRS-2672: Attempting to start 'ora.crsd' on 'host02'
CRS-2676: Start of 'ora.crsd' on 'host02' succeeded
[root]#
```

12) Did the orcl instance on ST_NODE2 start? Use the `svrctl status database –orcl` command as any of the users (`oracle`, `grid`, `root`) as long as the oracle environment is set for that user.
Note: The database may take a couple of minutes to restart. If the `orcl2` instance is not running, try the status command again, until instance `orcl2` is running.

```
[root]# . oraenv
ORACLE_SID = [root] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid
[root]# srvctl status database -d orcl
Instance orcl1 is running on node host01
Instance orcl2 is running on node host02
```

13) Configure database control to monitor the ASM instance on your first node. Open a terminal window on your first node. Prepare the oracle environment then use lsnrctl status command to determine the listener IP address. Use the second IP address listed.

Practice 7-1: Administering ASM Instances (continued)

```
[oracle]$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmount/11.2.0/sharedhome/dbhome
_1 is /u01/app/oracle
[oracle]$ lsnrctl status

LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 30-SEP-
2009 15:37:15

Copyright (c) 1991, 2009, Oracle.  All rights reserved.

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version
11.2.0.1.0 - Production
Start Date                30-SEP-2009 14:42:04
Uptime                    0 days 0 hr. 55 min. 11 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/11.2.0/grid/network/admin/listener.ora
Listener Log File
/u01/app/grid/diag/tnslsnr/gr7212/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=10.196.180.12)(PORT=
1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=10.196.180.213)(PORT
=1521)))
Services Summary...
Service "+ASM" has 1 instance(s).
  Instance "+ASM1", status READY, has 1 handler(s) for this
service...
Service "orcl.example.com" has 1 instance(s).
  Instance "orcl1", status READY, has 1 handler(s) for this
service...
Service "orclXDB.example.com" has 1 instance(s).
  Instance "orcl1", status READY, has 1 handler(s) for this
service...
The command completed successfully
```

Practice 7-1: Administering ASM Instances (continued)

Step	Screen/Page Description	Choices or Values
a.	Cluster Database Home	Log into EM as SYS/oracle_4U as SYSDBA. In the Instances section, click the link for +ASM1.example.com.
b.	Automatic Storage Management: *Instance name* Home	If the instance appears to be down, click Configure.
c.	Monitoring Configuration	In the Machine Name field enter the IP address that you discovered in Step 3 (preceding). Click Test Connection.
d.	Monitoring Configuration	An information message displays "Test Successful." Click OK.
e.	Automatic Storage Management: *Instance name* Home	Click the browser refresh button. You may have to refresh several times before EM reports that ASM instance is up.

Verify the configuration by starting Enterprise Manager DBcontrol and logging in as sys/oracle_4u as SYSDBA. Use the URL:
`https://first_ node_ name.example.com:1158/em`

Practices for Lesson 8

In these practices, you will add, configure, and remove disk groups, manage rebalance operations, and monitor disk and disk group IO statistics.

Practice 8-1: Administering ASM Disk Groups

In this practice, you will change the configuration of a disk group, and control the resulting rebalance operations. You will determine the connected clients to the existing disk groups, and perform disk group checks.

Because the asmadmin group has only one member, `grid`, open a terminal window and become the `grid` OS user for this practice.

You will use several tools, such as EM, ASMCMD, and ASMCA, to perform the same operations.

1) The FRA disk group has more disks allocated than is needed. So that one disk will be dropped from the FRA disk group, remove ASMDISK11 from the disk group. Use ASMCMD.

   a) As the `grid` OS user on your first node, confirm that the FRA disk group is mounted. If it is not mounted, mount it on your first and second nodes.

```
[grid]$ . oraenv
ORACLE_SID = [+ASM1] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid
[grid]$ . /home/oracle/labs/st_env.sh
[grid]$ asmcmd
ASMCMD> lsdg
State    Type    Rebal  Sector  Block       AU  Total_MB
Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks
Voting_files  Name
MOUNTED  EXTERN  N         512   4096  1048576      9996
3706              0           3706              0
N  ACFS/
MOUNTED  NORMAL  N         512   4096  1048576      9998
5280             1198         2041              0
N  DATA/
ASMCMD> mount FRA
ASMCMD> lsdg
State    Type    Rebal  Sector  Block       AU  Total_MB
Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks
Voting_files  Name
MOUNTED  EXTERN  N         512   4096  1048576      9996
3706              0           3706              0
N  ACFS/
MOUNTED  NORMAL  N         512   4096  1048576      9998
5280             1198         2041              0
N  DATA/
MOUNTED  EXTERN  N         512   4096  1048576      7497
7088              0           7088              0
N  FRA/
ASMCMD> exit

[grid]$ crsctl status resource ora.FRA.dg -t
--------------------------------------------------------------
-----------------
```

```
NAME           TARGET  STATE        SERVER
STATE_DETAILS
--------------------------------------------------------------
------------------
Local Resources
--------------------------------------------------------------
------------------
ora.FRA.dg
               ONLINE  ONLINE       gr7212
               OFFLINE OFFLINE      gr7213
               ONLINE  ONLINE       gr7214

[grid]$ ssh $ST_NODE2
 Last login: Tue Sep 29 16:56:00 2009 from gr7212.example.com
[grid@host02]$ . oraenv
ORACLE_SID = [grid] ? +ASM2
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid
grid@host02]$ asmcmd mount FRA
grid@host02]$ exit
grid]$ crsctl status resource ora.FRA.dg -t
--------------------------------------------------------------
------------------
NAME           TARGET  STATE        SERVER
STATE_DETAILS
--------------------------------------------------------------
------------------
Local Resources
--------------------------------------------------------------
------------------
ora.FRA.dg
               ONLINE  ONLINE       gr7212
               ONLINE  ONLINE       gr7213
               ONLINE  ONLINE       gr7214
```

b) Use the `chdg` command with inline XML. Note that the command is typed without a return, all on one line.

```
chdg <chdg name="FRA" power="5"> <drop>
<dsk name="ASMDISK11"/> </drop> </chdg>
```

```
[grid]$ asmcmd
ASMCMD> chdg <chdg name="FRA" power="5"> <drop>
<dsk name="ASMDISK11"/> </drop> </chdg>
ASMCMD>
```

2) In preparation for adding another disk to the DATA disk group, perform a disk check to verify the disk group metadata. Use the check disk group command `chkdg`.

```
ASMCMD> chkdg DATA
```

Practice 8-1: Administering ASM Disk Groups (continued)

3) Add another disk (ASMDISK12) to the DATA disk group and remove a disk (ASMDISK04), but the rebalance operation must wait until a quiet time and then proceed as quickly as possible. Use Enterprise Manager Database Control.

Step	Screen/Page Description	Choices or Values
a.	Automatic Storage Management: *Instance name* Home	Click Disk Groups tab.
b.	Automatic Storage Management Login	Enter: Username: `SYS` Password: `oracle_4U` Click Login.
c.	Automatic Storage Management: *Instance name* Disk Groups	Click the DATA link.
d.	Disk Group: DATA	Click ADD.
e.	Add Disks	Change Rebalance Power to 0 (this prevents any rebalance). Select Disks: ORCL:ASMDISK12 Click Show SQL.
f.	Show SQL	The SQL statement that will be executed is shown: `ALTER DISKGROUP DATA ADD DISK 'ORCL:ASMDISK12' SIZE 2500 M REBALANCE POWER 0` Click Return.
g.	Add Disks	Click OK.
h.	Add Disks in Progress message displayed	
i.	Disk Group: DATA	Select ASMDISK04. Click Remove.
j.	Confirmation	Click Show Advanced Options. Set Rebalance Power to 0. Click Yes.

4) Perform the pending rebalance operation on disk group DATA.

Step	Screen/Page Description	Choices or Values
a.	Disk Group: DATA	Click the Automatic Storage Management: +ASM1_host01.example.com locator link at the top of page.
b.	Automatic Storage Management: +ASM1_host01.example.com	Select the DATA disk group. Click Rebalance.
c.	Select ASM Instances	Select both ASM instances. Click OK.
d.	Confirmation	Click Show Advanced Options. Set Rebalance Power to 11. Click Yes.

Practice 8-1: Administering ASM Disk Groups (continued)

Step	Screen/Page Description	Choices or Values
e.	Automatic Storage Management: +ASM1_host01.example.com	Update Message: Disk Group DATA rebalance request has been submitted. Click the DATA disk group.
f.	Disk Group: DATA	Observe the Used(%) column for the various disk in the DATA disk group. Notice the status column (ASMDISK04 is marked as DROPPING). Click the browser's Refresh button.
g.	Disk Group: DATA	Notice the change in the Used(%) column. Refresh again. After a few minutes the ASMDISK04 will not appear in the list on Member disks.
h.		Exit EM.

5) Determine which clients are using ASM and ACFS. This is helpful to determine which clients are preventing the shutdown of ASM. Use the ASMCMD `lsct` command. Notice that the expect database client is shown and the ACFS client called asmvol, but the cluster +ASM1 instance is also listed because the OCR, and voting disk files are in the ASM disk group DATA. Open a terminal window on your first node as the `grid` user. For ASMCMD to work, the grid infrastructure environment must be set.

```
[grid]$ . oraenv
ORACLE_SID = [+ASM1] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid [grid]$ asmcmd lsct
DB_Name  Status     Software_Version  Compatible_version
Instance_Name  Disk_Group
+ASM     CONNECTED        11.2.0.1.0          11.2.0.1.0
+ASM1          DATA
asmvol   CONNECTED        11.2.0.1.0          11.2.0.1.0
+ASM1          ACFS
orcl     CONNECTED        11.2.0.1.0          11.2.0.0.0
orcl1          DATA
```

6) Examine the disk activity involving the DATA disk group.

   a) Examine the statistics for all the disk groups and then check the individual disks in the DATA disk group using EM.

Step	Screen/Page Description	Choices or Values
a.	Cluster Database: orcl.example.com: Home	In the Instances section, click the +ASM1.host01.example.com link.
b.	Automatic Storage Management: +ASM1_host01.example.com:	Click the Performance tab.

Practice 8-1: Administering ASM Disk Groups (continued)

Step	Screen/Page Description	Choices or Values
	Home	
c.	Automatic Storage Management: +ASM1_host01.example.com: Performance	Notice the graphs of the various metrics: Response Time, Throughput, Operations per Second, and Operation Size. Almost all the data operations involved the DATA disk group. In the Additional Monitoring Links section, click Disk Group I/O Cumulative Statistics.
d.	Automatic Storage Management Login	Enter: Username: `SYS` Password: `oracle_4U` Click Login.
e.	Disk Group I/O Cumulative Statistics	Notice that almost all of the I/O calls are against the DATA disk group. Expand the DATA disk group to show the individual disk statistics.
f.	Disk Group I/O Cumulative Statistics	Examine the number of reads and writes to each disk in the DATA disk group. The I/O to the disks will not be balanced because ASMDISK12 was just added.
g.		Exit EM.

b) Examine the disk I/O statistics using `lsdsk --statistics` command.

```
[grid]$ asmcmd
ASMCMD> lsdsk --statistics
Reads  Write  Read_Errs  Write_Errs  Read_time  Write_Time
Bytes_Read  Bytes_Written  Voting_File  Path
 6893  10103          0           0    324.968        74.8
664005632      300485632            Y  ORCL:ASMDISK01
10273  11096          0           0    338.432      84.596
743536128      346663936            Y  ORCL:ASMDISK02
 6345  10971          0           0     283.62      59.944
599138304      246011392            Y  ORCL:ASMDISK03
  153      0          0           0          1           0
1568768             0           N  ORCL:ASMDISK05
  124      4          0           0      1.056        .016
1490944         16384           N  ORCL:ASMDISK06
   97      0          0           0        1.1           0
397312             0           N  ORCL:ASMDISK07
   99      0          0           0      1.152           0
405504             0           N  ORCL:ASMDISK08
  897   1027          0           0      3.792       5.224
48025600      118779904           N  ORCL:ASMDISK09
  447    631          0           0      4.644         3.8
23568384       92385280           N  ORCL:ASMDISK10
```

```
 2050   7008          0           0       6.436     199.012
29081600     1234462720           N   ORCL:ASMDISK12
```

c) Examine the disk statistics bytes and time for the DATA disk group with the `iostat -t -G DATA` command.

```
ASMCMD> iostat -t -G DATA
Group_Name Dsk_Name   Reads      Writes    Read_Time Write_Time
DATA       ASMDISK01  664112128  301286912   324.972   74.908
DATA       ASMDISK02  744060416  346926080   338.444   84.624
DATA       ASMDISK03  600055808  246699520   283.656   60.032
DATA       ASMDISK12  30916608   1235182080  6.452     199.116
ASMCMD> exit
```

7) Run the following ASMCMD commands to return the DATA and FRA disk groups to the configuration at the beginning of the practice.

```
[grid]$ asmcmd chdg /home/oracle/labs/less_08/reset_DATA.xml
[grid]$ asmcmd chdg /home/oracle/labs/less_08/reset_FRA.xml
```

Practices for Lesson 9

In this practice, you will administer ASM files, directories, and templates.

Practice 9-1: Administering ASM Files, Directories, and Templates

In this practice, you use several tools to navigate the ASM file hierarchy, manage aliases, manage templates, and move files to different disk regions.

1) ASM is designed to hold database files in a hierarchical structure. Navigate the orcl database files with ASMCMD.

```
[grid]$ asmcmd
ASMCMD> ls
ACFS/
DATA/
FRA/
ASMCMD> ls DATA
ORCL/
cluster02/
ASMCMD> ls DATA/ORCL
CONTROLFILE/
DATAFILE/
ONLINELOG/
PARAMETERFILE/
TEMPFILE/
spfileorcl.ora
ASMCMD> ls -l DATA/ORCL/*
Type           Redund  Striped  Time             Sys  Name

+DATA/ORCL/CONTROLFILE/:
CONTROLFILE    HIGH    FINE     AUG 31 13:00:00  Y
Current.263.695918587

+DATA/ORCL/DATAFILE/:
DATAFILE       MIRROR  COARSE   AUG 31 13:00:00  Y
EXAMPLE.264.695918623
DATAFILE       MIRROR  COARSE   SEP 01 04:00:00  Y
SYSAUX.260.695918435
DATAFILE       MIRROR  COARSE   AUG 31 13:00:00  Y
SYSTEM.258.695918433
DATAFILE       MIRROR  COARSE   AUG 31 13:00:00  Y
UNDOTBS1.259.695918437
DATAFILE       MIRROR  COARSE   AUG 31 13:00:00  Y
UNDOTBS2.257.695918879
DATAFILE       MIRROR  COARSE   AUG 31 13:00:00  Y
USERS.268.695918437

+DATA/ORCL/ONLINELOG/:
ONLINELOG      MIRROR  COARSE   AUG 31 13:00:00  Y
group_1.262.695918591
ONLINELOG      MIRROR  COARSE   AUG 31 13:00:00  Y
group_2.261.695918595
ONLINELOG      MIRROR  COARSE   AUG 31 13:00:00  Y
group_3.265.695918947
```

Practice 9-1: Administering ASM Files, Directories, and Templates (continued)

```
ONLINELOG       MIRROR  COARSE   AUG 31 13:00:00  Y
group_4.256.695918951

+DATA/ORCL/PARAMETERFILE/:
PARAMETERFILE  MIRROR  COARSE   AUG 31 22:00:00  Y
spfile.266.695918955

+DATA/ORCL/TEMPFILE/:
TEMPFILE       MIRROR  COARSE   AUG 31 13:00:00  Y
TEMP.267.695918609
                                                 N
spfileorcl.ora =>
+DATA/ORCL/PARAMETERFILE/spfile.266.695918955
ASMCMD>
```

2) The default structure may not be the most useful for some sites. Create a set of aliases for directories and files to match a file system. Use EM.

Step	Screen/Page Description	Choices or Values
a.	Cluster Database: orcl.example.com	In the Instances section, click the +ASM1_*node_name*.example.com link.
b.	Automatic Storage Management: +ASM1_*node_name*.example.com Home	Click the Disk Groups tab.
c.	Automatic Storage Management Login	Enter: Username: `SYS` Password: `oracle_4U` Click Login.
d.	Automatic Storage Management: +ASM1_host01.example.com DiskGroups	Click the DATA disk group link.
e.	Disk Group: DATA General	Click the Files tab.
f.	Disk Group: DATA Files	Select ORCL. Click Create Directory.
g.	Create Directory	Enter: New Directory: `oradata` Click Show SQL.
h.	Show SQL	The SQL that will be executed is shown `ALTER DISKGROUP DATA ADD DIRECTORY '+DATA/ORCL/oradata'` Click Return.
i.	Create Directory	Click OK.
j.	Disk Group: DATA Files	Expand the ORCL folder. Expand the DATAFILE folder. Select EXAMPLE.nnn.NNNNNN. Click Create Alias.

Step	Screen/Page Description	Choices or Values
k.	Create Alias	Enter: User Alias: **+DATA/ORCL/oradata/example_01.dbf** Click Show SQL.
l.	Show SQL	The SQL that will be executed is shown `ALTER DISKGROUP DATA ADD ALIAS '+DATA/ORCL/oradata/example_01.dbf' FOR '+DATA/ORCL/DATAFILE/EXAMPLE.264.698859675'` Click Return.
m.	Create Alias	Click OK.
n.	Disk Group: DATA: Files	Click the `EXAMPLE.nnn.NNNNN` link.
o.	`EXAMPLE.nnn.NNNNNN`: Properties	Notice the properties that are displayed in the General section. Click OK.
p.	Disk Group: DATA Files	Click the `example_01.dbf` link.
q.	example_01.dbf: Properties	Note that the properties include System Name. Click OK.
r.		Exit EM.

3) Using ASMCMD, navigate to view `example_01.dbf` and display the properties. Using the system name, find the alias. Use the `ls -a` command.

```
[grid]$ asmcmd
ASMCMD> ls +DATA/ORCL/oradata/*
example_01.dbf
ASMCMD> ls -l +DATA/ORCL/oradata/*
Type      Redund  Striped  Time              Sys  Name
                                             N
example_01.dbf => +DATA/ORCL/DATAFILE/EXAMPLE.264.695918623
ASMCMD> ls -a +DATA/ORCL/DATAFILE/example*
+DATA/ORCL/oradata/example_01.dbf => EXAMPLE.264.695918623
```

4) Create a new tablespace. Name the file using a full name. Use EM.

Step	Screen/Page Description	Choices or Values
a.	Cluster Database: orcl.example.com Home	Click the Server tab.
b.	Cluster Database: orcl.example.com Server	In the Storage section, click the Tablespaces link.
c.	Tablespaces	Click Create.
d.	Create Tablespace: General	Enter: Name: `XYZ` In the Datafiles section, click Add.
e.	Add Datafile	Enter: Alias directory: +DATA/ORCL/oradata Alias name: XYZ_01.dbf

Practice 9-1: Administering ASM Files, Directories, and Templates (continued)

Step	Screen/Page Description	Choices or Values
		Click Continue.
f.	Create Tablespace: General	Click OK.
g.	Tablespaces	

5) Create another data file for the XYZ tablespace. Allow the file to receive a default name. Did both the files get system-assigned names?

Step	Screen/Page Description	Choices or Values
a.	Tablespaces	Select XYZ tablespace. Click Edit.
b.	Edit Tablespace: XYZ	In the Datafiles section, click Add.
c.	Add Datafile	Click Continue.
d.	Edit Tablespace: XYZ	Click Show SQL.
e.	Show SQL	Note: The SQL provides only the disk group name. Click Return.
f.	Edit Tablespace: XYZ	Click Apply.
g.	Edit Tablespace: XYZ	In the Datafiles section, note the names of the two files. One name was specified in the previous practice step, xyz_01.dbf, and the other is a system-assigned name. Click the Database tab.
h.	Cluster Database: orcl.example.com	In the Instances section, click the +ASM1_*node_name*.example.com link.
i.	Automatic Storage Management: +ASM1_*node_name*.example.com Home	Click the Disk Groups tab.
j.	Automatic Storage Management Login	Enter: Username: `SYS` Password: `oracle_4U` Click Login.
k.	Automatic Storage Management: +ASM1_host01.example.com DiskGroups	Click the DATA disk group link.
l.	Disk Group: DATA General	Click the Files tab.
m.	Disk Group: DATA Files	Expand the ORCL folder. Expand the DATAFILE folder. Note that there are two system-named files associated with the XYZ tablespace. Expand the oradata folder. Click the xyz_01.dbf link.
n.	XYZ_01.dbf: Properties	Observe that the `xyz_01.dbf` file is an alias to a file with a system name.

Practice 9-1: Administering ASM Files, Directories, and Templates (continued)

Step	Screen/Page Description	Choices or Values
		Click OK.
o.	Disk Group: DATA Files	

6) Move the files for the XYZ tablespace to the hot region of the DATA disk group.

Step	Screen/Page Description	Choices or Values
a.	Disk Group: DATA: Files	Click the General tab.
b.	Disk Group: DATA: General	In the Advanced Attributes section, click Edit.
c.	Edit Advanced Attributes for Disk Group: DATA	Change Database Compatibility to 11.2.0.0. Show SQL.
d.	Show SQL	Notice the SET ATTRIBUTE clause. Click Return.
e.	Edit Advanced Attributes for Disk Group: DATA	Click OK.
f.	Disk Group: DATA General	Click the Files tab.
g.	Disk Group: DATA: Files	Expand the ORCL folder. Expand the oradata folder. Select the `xyz_01.dbf` file. Click Edit File.
h.	Edit File: XYZ_01.dbf	In the Regions section, select the Primary Hot and Mirror Hot options. Click Show SQL.
i.	Show SQL	Note that the SQL statement uses the alias name and attributes clause. Click Return.
j.	Edit File: XYZ_01.dbf	Click Apply.
k.	Disk Group: DATA: Files	Expand the DATAFILES folder. Select the XYZ file that is not in the HOT region. Click Edit File.
l.	Edit File:XYZ.nnn.NNNNN	In the Regions section, select Primary Hot and Mirror Hot options. Click Show SQL.
m.	Show SQL	Note that the SQL statement uses the system name and attributes clause. Click Return.
n.	Edit File: XYZ.nnn.NNNNNN	Click Apply.
o.	Disk Group: DATA: Files	

7) Create a template that changes the default placement of files to the hot region.

Practice 9-1: Administering ASM Files, Directories, and Templates (continued)

Step	Screen/Page Description	Choices or Values
a.	Disk Group: DATA: Files	Click the Templates tab.
b.	Disk Group: DATA: Templates	Click Create.
c.	Create Template	Enter: Template Name: `HOT_FILES` In the Regions section, select the Primary Hot and Mirror Hot options. Click Show SQL.
d.	Show SQL	Note the SQL statement attributes clause. Click Return.
e.	Create Template	Click OK.
f.	Disk Group: DATA: Templates	Note the attributes of the HOT_FILES template compared with the DATAFILE template.

8) Add another data file to the XYZ tablespace using the template. Was the file placed in the HOT region?

Step	Screen/Page Description	Choices or Values
a.	Disk Group: DATA: Templates	Click the Database tab.
b.	Cluster Database: orcl.example.com Home	Click the Server tab.
c.	Cluster Database: orcl.example.com Server	In the Storage section, click the Tablespaces link.
d.	Tablespaces	Select the XYZ tablespace. Click Edit.
e.	Edit Tablespace: XYZ	In the Datafiles section, click Add.
f.	Add Datafile	Change Template to HOT_FILES. Click Continue.
g.	Edit Tablespace: XYZ	Click Show SQL.
h.	Show SQL	Note the data file specification "+DATA(HOT_FILES)." Click Return.
i.	Edit Tablespace: XYZ	Click Apply. Click the Database tab.
j.	Cluster Database: orcl.example.com	In the Instances section, click the +ASM1_*node_name*.example.com link.
k.	Automatic Storage Management: +ASM1_*node_name*.example.com Home	Click the Disk Groups tab.
l.	Automatic Storage Management Login	Enter: Username: `SYS` Password: `oracle_4U` Click Login.

Step	Screen/Page Description	Choices or Values
m.	Automatic Storage Management: +ASM1_host01.example.com DiskGroups	Click the DATA disk group link.
n.	Disk Group: DATA General	Click the Files tab.
o.	Disk Group: DATA Files	Expand the ORCL folder. Expand the DATAFILE folder. Notice that there are three system-named files associated with the XYZ tablespace. All have the HOT and HOT MIRROR attributes set.

9) Create a table in the XYZ tablespace.

   a) In a terminal window, as the `oracle` OS user, use the following command connect to the ORCL database (the password for `SYS` is `oracle_4U`):

   `sqlplus sys@orcl AS SYSDBA`

```
[oracle]$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmount/11.2.0/sharedhome is
/u01/app/oracle
[oracle]$ sqlplus sys@orcl as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 1 12:23:43
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Enter password:oracle_4U  << password is not displayed

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL>
```

   b) Create a large table in the XYZ tablespace called CUST_COPY by executing the `cr_cust_copy.sql` script. This script makes a copy of the `SH.CUSTOMERS` table into the XYZ tablespace.

```
SQL> @/home/oracle/labs/less_09/cr_cust_copy.sql
SQL>
```

Practice 9-1: Administering ASM Files, Directories, and Templates (continued)

```
SQL> CREATE TABLE Cust_copy TABLESPACE XYZ AS
  2  SELECT * FROM SH.CUSTOMERS;

Table created.

SQL>
```

10) Query the new table. Select all the rows to force some read activity with the command: `SELECT * FROM CUST_COPY`. Us the `SET PAGESIZE 300` command to speed up the display processing

```
SQL> SET PAGESIZE 300
SQL> SELECT * FROM CUST_COPY;
… /* rows removed */
    100055 Andrew               Clark
F
              1978 Married              77 Cumberland Avenue
74673      Duncan                                51402
SC
52722      52790
260-755-4130              J: 190,000 - 249,999
11000
Clark@company.com              Customer total         52772
01-JAN-98          A


55500 rows selected.

SQL>
```

11) View the I/O statistics by region. Use EM to view the statistics, and then repeat using ASMCMD.

a) View the I/O statistics by region with Enterprise Manager.

Step	Screen/Page Description	Choices or Values
a.	Disk Group: DATA Files	Click the Performance tab.
b.	Disk Group: DATA: Performance	In the Disk Group I/O Cumulative Statistics section, observe the values for Hot Reads and Hot Writes.

b) In a terminal window, on your first node as the `grid` user, set the oracle environment for the +ASM1 instance. View the I/O statistics by region using ASMCMD.

```
[grid@host01]$ . oraenv
ORACLE_SID = [grid] ? +ASM1
```

```
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid
[grid]$ asmcmd
ASMCMD> iostat --io --region -G DATA
Group_Name  Dsk_Name   Reads   Writes  Cold_Reads  Cold_Writes
Hot_Reads  Hot_Writes
DATA        ASMDISK01  11071   83680   5345        70653
20         263
DATA        ASMDISK02  13056   73855   5411        58344
24         226
DATA        ASMDISK03  46914   48823   36155       29176
31         367
DATA        ASMDISK04  204358  71893   196787      56579
32         362
ASMCMD> exit
```

12) Drop the tablespaces and templates created in this practice.

a) As the `oracle` OS user, connect to the orcl database, and then use the `drop_XYZ.sh` script to drop the XYZ tablespace.

```
[oracle]$ sqlplus sys@orcl as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 1 13:29:05
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Enter password: oracle_4U  << password is not displayed

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> @/home/oracle/labs/less_09/drop_XYZ.sql

SQL>
SQL> DROP TABLESPACE XYZ INCLUDING CONTENTS AND DATAFILES;

Tablespace dropped.

SQL>
SQL> EXIT;
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
```

Practice 9-1: Administering ASM Files, Directories, and Templates (continued)

```
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options
[oracle]$
```

b) As the `grid` OS user, use `asmcmd` to remove the `HOT_FILES` template.

```
[grid]$ asmcmd
ASMCMD> rmtmpl -G DATA HOT_FILES
ASMCMD> exit
[grid@host01 ~]$
```

Practices for Lesson 10

In this practice you will create, register, and mount an ACFS file system. In addition, you will manage ACFS snapshots.

Practice 10-1: Managing ACFS

In this practice, you will create, register, and mount an ACFS file system for general use. You will see the acfs modules that are loaded for ACFS. You create, use, and manage ACFS snapshots.

1) Open a terminal window on your first node and become the `root` user. Use the `lsmod` command to list the currently loaded modules. Use the `grep` command to display only the modes that have the `ora` string in them. Note the first three modules in the list below. These modules are required to enable ADVM and ACFS. The `oracleasm` module is loaded to enable ASMlib management of the ASM disks. Check all three nodes. If the ACFS drivers are not loaded, load them with the *`Grid_home`*`/bin/acfsload` script. The drivers are loaded and registered automatically when an ACFS volume is created with ASMCA. In this case, the configuration must be manually extended to ST_NODE3.

```
[oracle]$ su - root
Password: 0racle  << password is not displayed

/* on ST_NODE1 */

[root@host01]# lsmod | grep ora
oracleacfs            787588  2
oracleadvm            177792  6
oracleoks             226784  2 oracleacfs,oracleadvm
oracleasm              46356  1


[root@host01]# . /home/oracle/labs/st_env.sh

  /* on second node */
[root@host01]# ssh $ST_NODE2 lsmod| grep ora
root@host02's password: 0racle  << password is not displayed
oracleacfs            787588  3
oracleadvm            177792  7
oracleoks             226784  2 oracleacfs,oracleadvm
oracleasm              46356  1

/* on ST_NODE3 */

[root@host01 ~]# ssh $ST_NODE3 lsmod | grep ora
root@host03's password: 0racle  << password is not displayed
oracleacfs            787588  0
oracleadvm            177792  0
oracleoks             226784  2 oracleacfs,oracleadvm
oracleasm              46356  1
[root@host01 ~]# exit
```

Practice 10-1: Managing ACFS (continued)

2) Scenario: Your database application creates a number of image files stored as BFILES and external tables. These must be stored on a shared resource. An ACFS file system meets that requirement. Create an ASM volume and the ACFS file system. The ACFS volume should be 3 GB on the ACFS disk group. The mount point should be `/u01/app/oracle/asfcmounts/images`. These operations can be done with ASMCA as in Practice 2, ASMCMD, Enterprise Manager, or SQL*Plus. The ASMCMD solution is shown here.

   a) Open another terminal window as the `grid` OS user on your first node.

```
[grid@host01]$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid
```

   b) Start ASMCMD as the `grid` OS user.

```
[grid@host01]$ asmcmd
ASMCMD>
```

   c) Create a volume using the `volcreate` command.

```
ASMCMD> volcreate -G ACFS -s 3G IMAGES
ASMCMD>
```

   d) Find the volume device name. Use the `volinfo` command.

```
ASMCMD> volinfo -G ACFS -a
Diskgroup Name: ACFS

         Volume Name: DBHOME_1
         Volume Device: /dev/asm/dbhome_1-407
         State: ENABLED
         Size (MB): 6144
         Resize Unit (MB): 256
         Redundancy: UNPROT
         Stripe Columns: 4
         Stripe Width (K): 128
         Usage: ACFS
         Mountpath:
/u01/app/oracle/acfsmount/11.2.0/sharedhome

         Volume Name: IMAGES
         Volume Device: /dev/asm/images-407
         State: ENABLED
         Size (MB): 3072
         Resize Unit (MB): 256
         Redundancy: UNPROT
         Stripe Columns: 4
         Stripe Width (K): 128
         Usage:
         Mountpath:
```

Practice 10-1: Managing ACFS (continued)

```
ASMCMD>
```

e) Create an ACFS file system in the IMAGES volume—using the terminal window that you used in step 1—as the `root` user. When using the `mkfs -t acfs` command, the volume device must be supplied. Use the volume device name that you found in the previous step.

```
[root@host01 ~]# mkfs -t acfs /dev/asm/images-407
mkfs.acfs: version                   = 11.2.0.1.0.0
mkfs.acfs: on-disk version           = 39.0
mkfs.acfs: volume                    = /dev/asm/images-407
mkfs.acfs: volume size               = 3221225472
mkfs.acfs: Format complete.
[root@host01 ~]#
```

f) Mount the volume—using the terminal window that you used in step 1—as the root user, create the mount point directory, and mount the volume. The volume device is used again in the mount command. Enter the mount command all on one line. Repeat these commands on the second and third nodes of the cluster as the `root` user.

```
/* On first node */

[root@host01 ~]# mkdir -p /u01/app/oracle/acfsmount/images
[root@host01 ~]# mount -t acfs /dev/asm/images-407
/u01/app/oracle/acfsmount/images

/* On second node */

[root@host01 ~]# ssh $ST_NODE2
root@host02's password: 0racle  << password not displayed
[root@host02 ~]# mkdir /u01/app/oracle/acfsmount/images
[root@host02 ~]# mount -t acfs /dev/asm/images-407
/u01/app/oracle/acfsmount/images
[root@host02 ~]# exit
logout


Connection to host02 closed.
[root@host01 ~]#

/* On third node*/

[root@host01]# ssh $ST_NODE3
root@host03's password: 0racle  << password not displayed
[root@host03]# mkdir -p /u01/app/oracle/acfsmount/images
[root@host03]# mount -t acfs /dev/asm/images-407
/u01/app/oracle/acfsmount/images
[root@host03]# exit
```

g) Verify that the volume is mounted.

```
/* On first node */

[root@host01 ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
                      9.7G  2.6G  6.6G  28% /
/dev/xvda1             99M   20M   74M  22% /boot
tmpfs                 1.1G  482M  543M  48% /dev/shm
/dev/mapper/VolGroup01-LogVol00
                       30G  4.7G   24G  17% /u01
nnn.nnn.nnn.nnn:/mnt/shareddisk01/software/software
                       60G   34G   23G  60% /mnt/software
/dev/asm/dbhome_1-407
                      6.0G  4.8G  1.3G  79%
/u01/app/oracle/acfsmount/11.2.0/sharedhome
/dev/asm/images-407   3.0G   43M  3.0G   2%
/u01/app/oracle/acfsmount/images

/* On second node */

[root@host01 ~]# ssh $ST_NODE2 df -h
/u01/app/oracle/acfsmount/images
root@host02's password: 0racle  << password not displayed
Filesystem            Size  Used Avail Use% Mounted on
/dev/asm/images-407   3.0G   73M  3.0G   3%
/u01/app/oracle/acfsmount/images

[root@host01 ~]#


/* On third node */

[root@host01 ~]# ssh $ST_NODE3 df -h
/u01/app/oracle/acfsmount/images
root@host03's password: 0racle  << password not displayed

Filesystem            Size  Used Avail Use% Mounted on
/dev/asm/images-407   3.0G  109M  2.9G   4%
/u01/app/oracle/acfsmount/images

[root@host01 ~]#
```

h) Register the volume and mount point. As the `root` user, run the command: (Use your actual device name.)

`acfsutil registry –a /dev/asm/images-nnn /u01/app/oracle/acfsmount/images`

```
[root@host01]# acfsutil registry -a /dev/asm/images-407
/u01/app/oracle/acfsmount/images
```

```
acfsutil registry: mount point
/u01/app/oracle/acfsmount/images successfully added to Oracle
Registry
```

i) As the `root` user, view the registry status of the volume with the `acfsutil registry -l` command.

```
[root@host01]# acfsutil registry -l
Device : /dev/asm/images-407 : Mount Point :
/u01/app/oracle/acfsmount/images : Options : none : Nodes :
all : Disk Group : ACFS : Volume : IMAGES
[root@host01]#
```

3) An ACFS file system can be resized, and it will automatically resize the volume, if there is sufficient space in the disk group. The images file system is near capacity. Increase the file system by 256 MB. As the `root` user, use the command `acfsutil size +256M /u01/app/oracle/acfsmount/images`.

```
[root@host01]# acfsutil size +256M
/u01/app/oracle/acfsmount/images
acfsutil size: new file system size: 3489660928 (3328MB)
[root@host01 ~]#
```

4) As the `grid` user, check the size of the volume after the resize operation with `asmcmd volinfo`.

```
[grid@host03 ~]$ asmcmd volinfo -G ACFS IMAGES
Diskgroup Name: ACFS

         Volume Name: IMAGES
         Volume Device: /dev/asm/images-407
         State: ENABLED
         Size (MB): 3328
         Resize Unit (MB): 256
         Redundancy: UNPROT
         Stripe Columns: 4
         Stripe Width (K): 128
         Usage: ACFS
         Mountpath: /u01/app/oracle/acfsmount/images

[grid@host03 ~]$
```

5) As root, source the environment script `/home/oracle/labs/st_env.sh`. The IMAGES file system holds the image files for the orcl database owned by the `oracle` user. As the `root` user, change the permissions on the mount point so that the `oracle` user will own the file system on all three nodes.

```
/* on ST_NODE1  */
[root@host01]# . /home/oracle/labs/st_env.sh

[root@host01]# chown oracle:dba
/u01/app/oracle/acfsmount/images
```

Practice 10-1: Managing ACFS (continued)

```
/* On second node */

[root@host01 ~]# ssh $ST_NODE2 chown oracle:dba
/u01/app/oracle/acfsmount/images
root@host02's password: 0racle  << password is not displayed

/* on third node  */

[root@host01]# ssh $ST_NODE3 chown oracle:dba
/u01/app/oracle/acfsmount/images
root@host03's password: 0racle  << password is not displayed

[root@host01 ~]#
```

6) As the `oracle` user, transfer a set of images to `/u01/app/oracle/acfsmount/images`. Unzip the images in `/home/oracle/labs/less_10/images.zip` to the IMAGES file system.

```
[oracle@host01]$ cd /home/oracle/labs/less_10
[oracle@host01]$ unzip images.zip -d
/u01/app/oracle/acfsmount/images
Archive:  images.zip
   creating: /u01/app/oracle/acfsmount/images/gridInstall/
  inflating:
/u01/app/oracle/acfsmount/images/gridInstall/asm.gif
  inflating:
/u01/app/oracle/acfsmount/images/gridInstall/bullet2.gif

… Lines removed …

  inflating:
/u01/app/oracle/acfsmount/images/gridInstall/view_image.gif
 extracting:
/u01/app/oracle/acfsmount/images/gridInstall/white_spacer.gif
[oracle@host01 less_10]$
```

7) Verify that the files have been extracted.

```
 [oracle@host01]$ ls -R  /u01/app/oracle/acfsmount/images
/u01/app/oracle/acfsmount/images:
gridInstall  lost+found

/u01/app/oracle/acfsmount/images/gridInstall:
asm.gif             t20108.gif        t30104.gif   t30119d.gif
bullet2.gif         t20109a.gif       t30105.gif   t30119.gif
bullet.gif          t20109b.gif       t30106.gif   t30120a.gif
divider.gif         t20110.gif        t30107.gif   t30120b.gif
gradient.gif        t20111a.gif       t30108a.gif  t30121d.gif
MoveAllButton.gif   t20111b.gif       t30108.gif   t30123a.gif
MoveButton.gif      t20111c.gif       t30109.gif   t30123b.gif
rpm-oracleasm.gif   t20111.gif        t30110.gif   t30123c.gif
```

```
show_me.gif        t20112.gif        t30111.gif   t30201.gif
t10101.gif         t20113.gif        t30112a.gif  t30202.gif
t10102.gif         t20113h.gif       t30112.gif   t30203.gif
t10103.gif         t20114c.gif       t30113a.gif  t30204a.gif
t10201.gif         t20114login.gif   t30113b.gif  t30204.gif
t10202.gif         t20114server.gif  t30114a.gif  t30205.gif
t10203.gif         t20117add.gif     t30114b.gif  t30206.gif
t10204.gif         t20117crtbs.gif   t30114.gif   t30207.gif
t10205.gif         t20117emctl.gif   t30115a.gif  t30208.gif
t20101.gif         t20117tbs.gif     t30115.gif   t40101.gif
t20102.gif         t20119asm.gif     t30116a.gif  t40102.gif
t20103.gif         t2017emctl.gif    t30116b.gif  t40104.gif
t20104.gif         t30101a.gif       t30116c.gif  t40105a.gif
t20105.gif         t30101b.gif       t30116d.gif  t40105b.gif
t20106.gif         t30101c.gif       t30118b.gif  Thumbs.db
t20107a.gif        t30102.gif        t30119b.gif
view_image.gif
t20107.gif         t30103.gif        t30119c.gif
white_spacer.gif
ls: /u01/app/oracle/acfsmount/images/lost+found: Permission
denied
[oracle@host01]$
```

8) Create a snapshot of the IMAGES file system. Use the ASMCMD utility as the `root` user to execute the command:

```
/sbin/acfsutil snap create snap_001 \
/u01/app/oracle/acfsmount/images
```

```
[root@host01]# /sbin/acfsutil snap create snap_001
/u01/app/oracle/acfsmount/images
acfsutil snap create: Snapshot operation is complete.
```

9) Find the `.SNAP` directory and explore the entries. How much space does the `gridInstall` directory tree use? How much space does the `.ACFS/snaps/snap_001/rgidInstall` directory tree use?

```
[root@host01]# cd /u01/app/oracle/acfsmount/images
[root@host01 images]# ls –la
total 92
drwxrwx--- 5 oracle dba       4096 Sep  7 23:31 .
drwxr-xr-x 4 root   root      4096 Sep  7 11:53 ..
drwxr-xr-x 5 root   root      4096 Sep  7 15:04 .ACFS
drwxr-xr-x 2 oracle oinstall 12288 May  6 16:30 gridInstall
drwx------ 2 root   root     65536 Sep  7 15:04 lost+found
[root@host01]# du -h gridInstall
2.0M    gridInstall
[root@host01]# ls .ACFS
repl  snaps
[root@host01 images]# ls .ACFS/snaps
snap_001
```

Practice 10-1: Managing ACFS (continued)

```
[root@host01 images]# ls .ACFS/snaps/snap_001
gridInstall  lost+found
[root@host01]# du -h .ACFS/snaps/snap_001/gridInstall
2.0M    .ACFS/snaps/snap_001/gridInstall
```

10) Delete the `asm.gif` file from the IMAGES file system.

```
[root@host01 images]# rm gridInstall/asm.gif
rm: remove regular file `gridInstall/asm.gif'? y
```

11) Create another snapshot of the IMAGES file system.

```
[root@host01 images]# /sbin/acfsutil snap create snap_002
/u01/app/oracle/acfsmount/images
acfsutil snap create: Snapshot operation is complete.
```

12) How much space is being used by the snapshots and the files that are stored in the IMAGES file system? Use the `acfsutil info` command to find this information.

```
[root@host01 images]# /sbin/acfsutil info fs
/u01/app/oracle/acfsmount/images
/u01/app/oracle/acfsmount/images
    ACFS Version: 11.2.0.1.0.0
    flags:        MountPoint,Available
    mount time:   Mon Sep 13 16:14:10 2010
    volumes:      1
    total size:   3489660928
    total free:   3308507136
    primary volume: /dev/asm/images-7
        label:
        flags:                 Primary,Available,ADVM
        on-disk version:       39.0
        allocation unit:       4096
        major, minor:          252, 3586
        size:                  3489660928
        free:                  3308507136
        ADVM diskgroup         ACFS
        ADVM resize increment: 268435456
        ADVM redundancy:       unprotected
        ADVM stripe columns:   4
        ADVM stripe width:     131072
    number of snapshots:  2
    snapshot space usage: 2351104
[root@host01 images]#
```

13) Restore the `asm.gif` file to the file system from the snapshot. This can be done with OS commands or from Enterprise Manager. This solution uses the OS commands.

Practice 10-1: Managing ACFS (continued)

a) The snapshot is a sparse file representation of the file system, so you can browse the snapshot as if it were a full file system. All the OS file commands are functional. Find the `asm.gif` file in the snapshot. The `find` command ID is shown in this solution. Perform this operation as the `root` user.

```
[root]$ cd /u01/app/oracle/acfsmount/images
[root]$ find . -name asm.gif
find: ./.ACFS/.fileid: Permission denied
find: ./.ACFS/repl: Permission denied
find: ./.ACFS/snaps/snap_001/.ACFS/.fileid: Permission denied
find: ./.ACFS/snaps/snap_001/.ACFS/repl: Permission denied
find: ./.ACFS/snaps/snap_001/.ACFS/snaps: Permission denied
find: ./.ACFS/snaps/snap_001/lost+found: Permission denied
./.ACFS/snaps/snap_001/gridInstall/asm.gif
find: ./.ACFS/snaps/snap_002/.ACFS/.fileid: Permission denied
find: ./.ACFS/snaps/snap_002/.ACFS/repl: Permission denied
find: ./.ACFS/snaps/snap_002/.ACFS/snaps: Permission denied
find: ./.ACFS/snaps/snap_002/lost+found: Permission denied
find: ./lost+found: Permission denied
```

b) Restore the `asm.gif` file by copying from the snapshot to the original location.

```
[root]$ cp ./.ACFS/snaps/snap_001/gridInstall/asm.gif
./gridInstall/asm.gif
```

14) Dismount the Images file system from all three nodes. This command must be executed by the `root` user. If the directory is busy, use `lsof` to find the user that is holding the directory open and stop that session.

```
[root@host01]# umount /u01/app/oracle/acfsmount/images
umount: /u01/app/oracle/acfsmount/images: device is busy
umount: /u01/app/oracle/acfsmount/images: device is busy

[root@host01]# lsof +d /u01/app/oracle/acfsmount/images
COMMAND   PID   USER   FD   TYPE     DEVICE SIZE NODE NAME
COMMAND   PID USER   FD   TYPE   DEVICE SIZE NODE NAME
lsof     5770 root  cwd    DIR 252,5634 4096    2
/u01/app/oracle/acfsmount/images
lsof     5771 root  cwd    DIR 252,5634 4096    2
/u01/app/oracle/acfsmount/images
bash    23971 root  cwd    DIR 252,5634 4096    2
/u01/app/oracle/acfsmount/images
   /* change directories the root session then continue */

[root@host01]# cd

[root@host01]# umount /u01/app/oracle/acfsmount/images

[root@host01 ~]# ssh $ST_NODE2 umount
/u01/app/oracle/acfsmount/images
root@host02's password: 0racle  << password is not displayed
```

Practice 10-1: Managing ACFS (continued)

```
[root@host01 ~]# ssh $ST_NODE3 umount
/u01/app/oracle/acfsmount/images
root@host03's password: 0racle  << password is not displayed

[root@host01 ~]#
```

Practice 10-1: Managing ACFS (continued)

15) Remove the IMAGES ACFS file system and volume using Enterprise Manager.

Step	Screen/Page Description	Choices or Values
a.	Enter the URL in the browser	https//host01:1158/em
b.	Login	Name: `SYS` Password: `oracle_4U` Connect as `SYSDBA` Click Login.
c.	Cluster Database: orcl.example.com Home	In the Instances section, click the link for +ASM1_host01.example.com.
d.	Automatic Storage Management: +ASM1_host01.example.com Home	Click the ASM Cluster File System tab.
e.	Automatic Storage Management Login	Username: `SYS` Password: `oracle_4U` Click Login.
f.	Automatic Storage Management: +ASM1_host01.example.com ASM Cluster File System	Select /u01/app/oracle/acfsmount/images. Click Deregister.
g.	ASM Cluster File System Host Credentials: host01.example.com	Username: `grid` Password: `0racle` Click Continue.
h.	Deregister ASM Cluster File System: /dev/asm/images-407	Click OK.
i.	Automatic Storage Management: +ASM1_host01.example.com ASM Cluster File System	Click the Disk Groups tab.
j.	Automatic Storage Management: +ASM1_host01.example.com Disk Groups	Click the link for the ACFS disk group.
k.	Disk Group: ACFS General	Click the Volumes tab.
l.	Disk Group: ACFS Volumes	Select the IMAGES volume. Click Delete.
m.	Confirmation	Click Yes.
n.	Disk Group: ACFS General	
o.	Disk Group: ACFS	Click Logout. Close the browser.

Practice 10-2: Uninstalling RAC Database

In this practice, you will uninstall the RAC database, leaving the clusterware installed on three nodes.

1) Open a terminal window as the `oracle` OS user on your first node and set the environment with the `oraenv` and `st_env.sh` commands. Enter the name of the database when prompted.

```
[oracle@host01]$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmount/11.2.0/sharedhome is
/u01/app/oracle
[oracle] $ . /home/oracle/labs/st_env.sh
```

2) Stop Enterprise Manager on your first and second nodes. It was not started on your third node.

```
[oracle@host01]$ export ORACLE_UNQNAME=orcl
[oracle@host01]$ emctl stop dbconsole
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009 Oracle Corporation.  All rights
reserved.
https://host01.example.com:1158/em/console/aboutApplication
Stopping Oracle Enterprise Manager 11g Database Control ...

all attemps to stop oc4j failed... now trying to kill 9
 ...  Stopped.
[oracle@host01]$ ssh $ST_NODE2
Last login: Thu Sep 24 20:56:37 2009 from host01.example.com
[oracle@host02]$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmount/11.2.0/sharedhome is
/u01/app/oracle
[oracle@host02]$ export ORACLE_UNQNAME=orcl
[oracle@host02]$ emctl stop dbconsole
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009 Oracle Corporation.  All rights
reserved.
https://host02.example.com:1158/em/console/aboutApplication
Stopping Oracle Enterprise Manager 11g Database Control ...

 Cannot determine Oracle Enterprise Manager 11g Database
Control process.
/u01/app/oracle/acfsmount/11.2.0/sharedhome/host02_orcl/emctl.
pid does not exist.
[oracle@host02 ~]$ exit
logout
```

```
Connection to host02 closed.
[oracle@host01 ~]$
```

3) Change directories to the `oracle` user's home directory: `/home/oracle`.

```
[oracle@host01]$ cd /home/oracle
```

4) Start the deinstall tool. The tool does an auto discover of the database parameters. The next prompt allows you to change the discovered parameters, answer `n`.

```
[oracle@host01]$ $ORACLE_HOME/deinstall/deinstall -silent
Checking for required files and bootstrapping ...
Please wait ...
Location of logs /u01/app/oraInventory/logs/

############ ORACLE DEINSTALL & DECONFIG TOOL START
############


######################## CHECK OPERATION START
########################
Install check configuration START


Checking for existence of the Oracle home location
/u01/app/oracle/acfsmount/11.2.0/sharedhome/dbhome_1
Oracle Home type selected for de-install is: RACDB
Oracle Base selected for de-install is: /u01/app/oracle
Checking for existence of central inventory location
/u01/app/oraInventory
Checking for existence of the Oracle Grid Infrastructure home
/u01/app/11.2.0/grid
The following nodes are part of this cluster: host01,host02

Install check configuration END


Network Configuration check config START

Network de-configuration trace file location:
/u01/app/oraInventory/logs/netdc_check5746757283308326616.log

Network Configuration check config END

Database Check Configuration START

Database de-configuration trace file location:
/u01/app/oraInventory/logs/databasedc_check7510048152035885516
.log


###### For Database 'orcl' ######
```

Practice 10-2: Uninstalling RAC Database (continued)

```
RAC Database
The nodes on which this database has instances: [host01,
host02]
The instance names: [orcl1, orcl2]
The local instance name on node: orcl1
The diagnostic destination location of the database:
/u01/app/oracle/diag/rdbms/orcl
Storage type used by the Database: ASM
Database Check Configuration END

Enterprise Manager Configuration Assistant START

EMCA de-configuration trace file location:
/u01/app/oraInventory/logs/emcadc_check.log

Checking configuration for database orcl
Enterprise Manager Configuration Assistant END
Oracle Configuration Manager check START
OCM check log file location :
/u01/app/oraInventory/logs//ocm_check7494.log
Oracle Configuration Manager check END

######################### CHECK OPERATION END
#########################


####################### CHECK OPERATION SUMMARY
#######################
Oracle Grid Infrastructure Home is: /u01/app/11.2.0/grid
The cluster node(s) on which the Oracle home exists are:
(Please input nodes seperated by ",", eg:
node1,node2,...)host01,host02
Oracle Home selected for de-install is:
/u01/app/oracle/acfsmount/11.2.0/sharedhome/dbhome_1
Inventory Location where the Oracle home registered is:
/u01/app/oraInventory
The following databases were selected for de-configuration :
orcl
Database unique name : orcl
Storage used : ASM
Will update the Enterprise Manager configuration for the
following database(s): orcl
No Enterprise Manager ASM targets to update
No Enterprise Manager listener targets to migrate
Checking the config status for CCR
host01 : Oracle Home exists with CCR directory, but CCR is not
configured
host02 : Oracle Home exists with CCR directory, but CCR is not
configured
CCR check is finished
```

```
A log of this session will be written to:
'/u01/app/oraInventory/logs/deinstall_deconfig2009-09-28_08-
24-07-AM.out'
Any error messages from this session will be written to:
'/u01/app/oraInventory/logs/deinstall_deconfig2009-09-28_08-
24-07-AM.err'

######################## CLEAN OPERATION START
########################

Enterprise Manager Configuration Assistant START

EMCA de-configuration trace file location:
/u01/app/oraInventory/logs/emcadc_clean.log

Updating Enterprise Manager Database Control configuration for
database orcl
Updating Enterprise Manager ASM targets (if any)
Updating Enterprise Manager listener targets (if any)
Enterprise Manager Configuration Assistant END
Database de-configuration trace file location:
/u01/app/oraInventory/logs/databasedc_clean8536757532603850049
.log
Database Clean Configuration START orcl
This operation may take few minutes.
Database Clean Configuration END orcl

Network Configuration clean config START

Network de-configuration trace file location:
/u01/app/oraInventory/logs/netdc_clean4830516361423529044.log

De-configuring backup files...
Backup files de-configured successfully.

The network configuration has been cleaned up successfully.

Network Configuration clean config END

Oracle Configuration Manager clean START
OCM clean log file location :
/u01/app/oraInventory/logs//ocm_clean7494.log
Oracle Configuration Manager clean END
Oracle Universal Installer clean START

Detach Oracle home
'/u01/app/oracle/acfsmount/11.2.0/sharedhome/dbhome_1' from
the central inventory on the local node : Done
```

```
Delete directory
'/u01/app/oracle/acfsmount/11.2.0/sharedhome/dbhome_1' on the
local node : Done
  <<< following Failed messages are expected >>>
Failed to delete the directory
'/u01/app/oracle/acfsmount/images'. The directory is in use.
Failed to delete the directory
'/u01/app/oracle/acfsmount/11.2.0/sharedhome/.ACFS/.fileid'.
The directory is in use.
Failed to delete the directory
'/u01/app/oracle/acfsmount/11.2.0/sharedhome/.ACFS/repl'. The
directory is in use.
Failed to delete the directory
'/u01/app/oracle/acfsmount/11.2.0/sharedhome/.ACFS/snaps'. The
directory is in use.
Failed to delete the directory
'/u01/app/oracle/acfsmount/11.2.0/sharedhome/.ACFS'. The
directory is not empty.
Failed to delete the directory
'/u01/app/oracle/acfsmount/11.2.0/sharedhome/lost+found'. The
directory is in use.
Failed to delete the directory
'/u01/app/oracle/acfsmount/11.2.0/sharedhome'. The directory
is not empty.
Failed to delete the directory
'/u01/app/oracle/acfsmount/11.2.0'. The directory is not
empty.
Failed to delete the directory '/u01/app/oracle/acfsmount'.
The directory is not empty.
Failed to delete the directory
'/u01/app/oracle/acfsmount/images'. The directory is in use.
Failed to delete the directory
'/u01/app/oracle/acfsmount/11.2.0/sharedhome/.ACFS/.fileid'.
The directory is in use.
Failed to delete the directory
'/u01/app/oracle/acfsmount/11.2.0/sharedhome/.ACFS/repl'. The
directory is in use.
Failed to delete the directory
'/u01/app/oracle/acfsmount/11.2.0/sharedhome/.ACFS/snaps'. The
directory is in use.
Failed to delete the directory
'/u01/app/oracle/acfsmount/11.2.0/sharedhome/.ACFS'. The
directory is not empty.
Failed to delete the directory
'/u01/app/oracle/acfsmount/11.2.0/sharedhome/lost+found'. The
directory is in use.
Failed to delete the directory
'/u01/app/oracle/acfsmount/11.2.0/sharedhome'. The directory
is not empty.
```

```
Failed to delete the directory
'/u01/app/oracle/acfsmount/11.2.0'. The directory is not
empty.
Failed to delete the directory '/u01/app/oracle/acfsmount'.
The directory is not empty.
Failed to delete the directory '/u01/app/oracle'. The
directory is not empty.
Delete directory '/u01/app/oracle' on the local node : Failed
<<<<

Detach Oracle home
'/u01/app/oracle/acfsmount/11.2.0/sharedhome/dbhome_1' from
the central inventory on the remote nodes 'host02' : Done

Delete directory '/u01/app/oracle' on the remote nodes
'host02' : Failed <<<<

The directory '/u01/app/oracle' could not be deleted on the
nodes 'host02'.
Oracle Universal Installer cleanup completed with errors.

Oracle Universal Installer clean END


Oracle install clean START

Clean install operation removing temporary directory
'/tmp/install' on node 'host01'
Clean install operation removing temporary directory
'/tmp/install' on node 'host02'

Oracle install clean END


######################### CLEAN OPERATION END
#########################


####################### CLEAN OPERATION SUMMARY
#######################
Updated Enterprise Manager configuration for database orcl
Successfully de-configured the following database instances :
orcl
Cleaning the config for CCR
As CCR is not configured, so skipping the cleaning of CCR
configuration
CCR clean is finished
Successfully detached Oracle home
'/u01/app/oracle/acfsmount/11.2.0/sharedhome/dbhome_1' from
the central inventory on the local node.
```

```
Successfully deleted directory
'/u01/app/oracle/acfsmount/11.2.0/sharedhome/dbhome_1' on the
local node.
Failed to delete directory '/u01/app/oracle' on the local
node.
Successfully detached Oracle home
'/u01/app/oracle/acfsmount/11.2.0/sharedhome/dbhome_1' from
the central inventory on the remote nodes 'host02'.
Failed to delete directory '/u01/app/oracle' on the remote
nodes 'host02'.
Oracle Universal Installer cleanup completed with errors.

Oracle install successfully cleaned up the temporary
directories.
##############################################################


############# ORACLE DEINSTALL & DECONFIG TOOL END ###########

 [oracle@host01 ~]$
```

5) The `deinstall` tool does not completely remove the ORACLE_HOME directory when ORACLE_HOME is placed in the ACFS file system. Using ASMCA, remove the ACFS file system.

   a) Open a window as the `grid` user on your first node and set the Grid Infrastructure home with `oraenv`.

```
$ su - grid
Password: 0racle    <<< paswword not displayed
[grid]$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base for ORACLE_HOME=/u01/app/11.2.0/grid is
/u01/app/grid
[grid]$
```

   b) Start ASMCA with the `asmca` command as the `grid` user.

```
[grid]$ asmca
```

   c) Dismount the ACFS file system.

Step	Screen/Page Description	Choices or Values
a.	ASM Configuration Assistant	Click the ASM Cluster File Systems tab.
b.	ASM Cluster File System	Right-click the row with Volume DBHOME_1.
c.	In the popup-menu	Select Show Dismount Command.
d.	Dismount ACFS Command	Follow instructions Start a window as the `root` user. Copy the command in the ASMCA window to the terminal window and execute.

Step	Screen/Page Description	Choices or Values
e.	Dismount ACFS Command	Click Close.

d) Remove the ORACLE_HOME directory

Step	Screen/Page Description	Choices or Values
a.	ASM Cluster File System	Right-click the row with Volume DBHOME_1.
b.	In the popup-menu	Select Delete.
c.	Confirm Delete	Click Yes.
d.	ASM Cluster File System: Deletion	Follow instructions Start a window as the `root` user Copy the command in the ASMCA window to the terminal window and execute. Exit the terminal window.
e.	ASM Cluster File System: Deletion	Click close.
f.	ASM Cluster File System	Click Exit.
g.	ASM Configuration Assistant	Click Yes.

6) Close all terminal windows.

Practices for Lesson 11

In this practice, you will install the Oracle Database 11*g* Release 2 software and create a three-node cluster database.

Practice 11-1: Installing the Oracle Database Software

In this practice, you will install the Oracle Database 11*g* Release 2 software on three nodes.

1) Use the VNC session on ST_NODE1 on display :1. This is the oracle user's VNC session. Click the VNC icon on your desktop, enter *`ST_NODE1`*`:1`, substituting the name of your first node for *ST_NODE1*. Enter the password `0racle`.

2) In the VNC session, confirm that you are connected as the `oracle` user with the proper group memberships. Change directory to the staged software location provided by your instructor and start the OUI by executing the `runInstaller` command from the `/`*`staged_software_location`*`/database/Disk1` directory.

```
$ id
uid=501(oracle) gid=502(oinstall)
groups=501(dba),502(oinstall),503(oper),505(asmdba)
$ cd /stage/database/Disk1

$ ./runInstaller
```

   a) On the Configure Security Updates page, deselect the "I wish to receive security updates" check box and click Next. **Note:** If this were a production machine or part of an important test environment, you might consider this option. A dialog box appears making sure that you want to remain uninformed about the updates. Click Yes to close the dialog box and continue.

   b) On the Select Installation Option page, select the "Install database software only" option and click Next.

   c) On the Node Selection page, select Real Application Clusters database installation. Select all three of your assigned hosts and click Next. If the ssh connectivity test fails, click the SSH Connectivity button. Enter the password `0racle` for the `oracle` user. Select the Reuse public and private keys check box and click the Setup button. After the setup completes, click Next to continue.

   d) On the Select Product Languages, promote all languages from the Available Languages window to the Selected Languages window on the right-hand side. Click Next to continue.

   e) On the Select Database Edition, select Enterprise Edition and click Next to continue.

   f) On the Specify Installation edition, the Oracle Base should be `/u01/app/oracle` and the Software Location should be `/u01/app/oracle/product/11.2.0/dbhome_1`. Do not install the database to a shared location. Click Next to continue.

   g) On the Privileged Operating System Groups, select dba as the Database Administrator Group and oper as the Database Operator Group. Click Next to continue.

Practice 11-1: Installing the Oracle Database Software (continued)

h) When the prerequisites have successfully been checked on the Perform Prerequisites Check page, click Next to continue. If any checks fail, click the Fix and Check Again button. Run the scripts on the cluster nodes as `root` as directed, and then click Next.

i) Check the information on the Summary page and click Finish.

j) The Install Progress screen allows you to monitor the progression of the installation.

k) When the files have been copied to all nodes, the Execute Configuration Scripts window is presented. Execute the `/u01/app/oracle/product/11.2.0/dbhome_1/root.sh` script on **all three nodes**.

```
# /u01/app/oracle/product/11.2.0/dbhome_1/root.sh

Running Oracle 11g root.sh script...

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /u01/app/oracle/product/11.2.0/dbhome_1

Enter the full pathname of the local bin directory:
[/usr/local/bin]:
The file "dbhome" already exists in /usr/local/bin.  Overwrite
it? (y/n)
[n]: n
The file "oraenv" already exists in /usr/local/bin.  Overwrite
it? (y/n)
[n]: n
The file "coraenv" already exists in /usr/local/bin.
Overwrite it? (y/n)
[n]: n

Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root.sh script.
Now product-specific root actions will be performed.
Finished product-specific root actions.
#
```

***Run this script on the remaining two nodes before continuing*.**

l) When you have run the `root.sh` scripts on all three nodes, click the OK button to close the Execute Configuration Scripts window.

m) Click the Close button on the Finish page to complete the installation and exit the Installer.

Practice 11-2: Creating a RAC Database

In this practice, you will create a three-node RAC database.

1) From the oracle VNC session that you used to install the database software in the preceding practice, change directory to `/u01/app/oracle/product/11.2.0/dbhome_1/bin` and launch the Database Configuration Assistant by executing the `dbca` command.

```
$ id
uid=500(oracle) gid=501(oinstall)
groups=500(dba),501(oinstall),502(oper),505(asmdba)

$ cd /u01/app/oracle/product/11.2.0/dbhome_1/bin

$ ./dbca
```

   a) On the Welcome page, select Oracle Real Application Clusters database and click Next.

   b) On the Operations page, select "Create a Database" and click Next.

   c) On the Database Templates page, select the General Purpose or Transaction Processing template. Click Next to continue.

   d) On the Database Identification page, select Admin-Managed and enter **`orcl`** in the Global Database Name field. Click the Select All button to install the database to all three of your nodes and click Next.

   e) On the Management Options page, make sure that the Configure Enterprise Manager check box is selected and the "Configure Database Control for local management" option is selected. Click Next to continue.

   f) Select "Use the Same Administrative Password for All Accounts" on the Database Credentials page. Enter `oracle_4U` in the Password and Confirm Password fields and click Next.

   g) On Database File locations, you can specify your storage type and location for your database files. Select Automatic Storage Management (ASM) from the drop-down list for Storage Type. Select Use Oracle-Managed Files and make sure that the Database Area value is +DATA. Click Next to continue.

   h) When the ASM Credentials box appears, enter the ASM password that you used during the clusterware installation practice 2.2, step k. The password should be `oracle_4U`. Enter the correct password and click OK to close the box.

   i) On the Recovery Configuration page, enter `+FRA` in the Recovery Area field and accept the default for the Recovery Area Size. Click Next to continue.

   j) On the Database Content page, select the Sample Schemas check box and click Next.

Practice 11-2: Creating a RAC Database (continued)

k) On the Memory tabbed page of the Initialization Parameters page, make sure that you enter `550` in the Memory Size (SGA and PGA) field. Then click the Character Sets tab, and select Use Unicode on the Database Character Set tabbed page. Click Next.

l) Accept the default values on the Database Storage page and click Next to continue.

m) Select Create Database on the Creation Options page and click Finish.

n) On the Database Configuration Assistant: Summary page, click OK.

o) You can monitor the database creation progress from the Database Configuration Assistant window.

p) At the end of the installation, a dialog box with your database information including the Database Control URL is displayed. Click Exit. This will close the dialog box and end the Database Configuration Assistant.

q) Open a browser and enter the Database Control URL displayed in the previous step. Verify that all instances are up. Verify that all cluster resources are up across all three nodes.

Practices for Lesson 12

In these practices, you will contrast operating system, password file authenticated connections, and Oracle database authenticated connections. You will also learn to stop a complete ORACLE_HOME component stack

Practice 12-1: Operating System and Password File Authenticated Connections

In this practice, you will make both operating system authenticated connections and password file authenticated connections to the database instance. You will also examine problems with the `oraenv` script.

1) Connect to your first node as the `oracle` user and set up your environment variables using the `oraenv` script.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
```

2) Identify all the database instance names that are currently executing on your machine using the Linux `ps` command. **Note:** All database instances have a mandatory background process named `pmon`, and the instance name will be part of the complete process name.

```
$ ps -ef | grep -i pmon
grid      4111     1  0 10:53 ?        00:00:00 asm_pmon_+ASM1
oracle    4987     1  0 10:56 ?        00:00:00 ora_pmon_orcl1
oracle    5299  5257  0 10:58 pts/0    00:00:00 grep -i pmon
```

3) Attempt to make a local connection to the `orcl1` instance using SQL*Plus with the `sysdba` privilege. This is known as operating system authentication because a password is not needed. What happens when trying to connect to the instance?

```
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 10:59:04
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> exit
Disconnected.
$
```

4) Attempt to connect to the instance using a network connection string `@orcl` with the `sysdba` privilege. This is known as password file authentication. Is the connection successful this time?

```
$ sqlplus sys@orcl as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 11:03:35
2009
```

```
Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Enter password: oracle_4U << Password is not displayed

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> exit

$
```

5) Display the values of the environment variables (`ORACLE_BASE`, `ORACLE_HOME`, `ORACLE_SID`, `PATH`, and `LD_LIBRARY_PATH`) that were defined with the `oraenv` script in step 1.

```
$ env | grep ORA
ORACLE_SID=orcl
ORACLE_BASE=/u01/app/oracle
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1

$ env | grep PATH
LD_LIBRARY_PATH=/u01/app/oracle/product/11.2.0/dbhome_1/lib
PATH=/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/orac
le/bin:/u01/app/oracle/product/11.2.0/dbhome_1/bin
```

6) Modify the `ORACE_SID` environment variable to match the actual database instance name for the `orcl` database.

```
$ export ORACLE_SID=orcl1
```

7) Attempt the local connection with system authentication to the `orcl1` instance using SQL*Plus with the `sysdba` privilege. This is the same command as in step 3.

```
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 11:01:48
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
```

```
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL>
```

8) Query the `instance_name` column of the `v$instance` dynamic performance view to validate the instance that you connected with. Exit SQL*Plus when finished.

```
SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
orcl1

SQL> exit
```

Practice 12-2: Oracle Database Authenticated Connections

In this practice, you will make multiple Oracle database authenticated connections to a database instance and notice the effects of load balanced connections.

1) From your first node, connected as the `oracle` user, validate the instance names on each host.

```
$ . /home/oracle/labs/st_env.sh
$ ssh $ST_NODE1 ps -ef | grep pmon
grid      4111     1  0 10:53 ?        00:00:00 asm_pmon_+ASM1
oracle    4987     1  0 10:56 ?        00:00:00 ora_pmon_orcl1
oracle    7932  7779  0 11:56 pts/0    00:00:00 grep pmon

$ ssh $ST_NODE2 ps -ef | grep pmon
grid      4104     1  0 10:53 ?        00:00:00 asm_pmon_+ASM2
oracle    4959     1  0 10:56 ?        00:00:00 ora_pmon_orcl2

$ ssh $ST_NODE3 ps -ef | grep pmon
grid      4096     1  0 10:53 ?        00:00:00 asm_pmon_+ASM3
oracle    4841     1  0 10:55 ?        00:00:00 ora_pmon_orcl3
```

2) Verify the current host name, and then set the environment variables using the `oraenv` script.

```
$ hostname
host01.example.com

$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
```

3) Connect to a database instance by using SQL*Plus with the `system` account. This is known as Oracle database authentication. After it is connected, query the `instance_name` column from the `v$instance` dynamic performance view.
Note: Your instance names may vary from the ones displayed below.

```
$ sqlplus system@orcl

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 12:05:06
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Enter password: oracle_4U << Password is not displayed

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
```

Practice 12-2: Oracle Database Authenticated Connections (continued)

```
Data Mining and Real Application Testing options

SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
orcl2

SQL>
```

4) Use the SQL*Plus `host` command to temporarily exit SQL*Plus and return to the operating system prompt. **Note:** SQL*Plus is still running when this is performed. Validate that you are still on your first node. Repeat step 3 from the operating system prompt to establish a second SQL*Plus session and database instance connection. What instance name did you connect to?

```
SQL> host
$ hostname
host01.example.com

$ sqlplus system@orcl

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 12:13:02
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Enter password: oracle_4U << Password is not displayed

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
orcl1

SQL>
```

5) Use the SQL*Plus `host` command to temporarily exit SQL*Plus and return to the operating system prompt. **Note:** SQL*Plus is still running when this is performed. Validate that you are still on your first node. Repeat step 3 from the operating system prompt to establish a third SQL*Plus session and database instance connection. What instance name did you connect to?

Practice 12-2: Oracle Database Authenticated Connections (continued)

```
SQL> HOST
$ hostname
host01.example.com
[oracle@host01 ~]$ sqlplus system@orcl

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 12:15:52
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Enter password: oracle_4U << Password is not displayed

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
orcl3

SQL>
```

6) Exit the three SQL*Plus sessions that are currently executing on the first node.

```
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

$ exit
exit

SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

$ exit
exit
```

Practice 12-2: Oracle Database Authenticated Connections (continued)

```
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

$ exit << Optional. This will exit your terminal session.
```

Practice 12-3: Stopping a Complete ORACLE_HOME Component Stack

In this practice, you will use the `srvctl` utility to stop all resource components executing from a single home location.

1) From your first node, connect to the `oracle` account and set up the environment variables for the database instance.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$ . /home/oracle/labs/st_env.sh
```

2) Validate that the instances are running on each node of the cluster.
Note: There are several ways this step can be performed.

```
$ ssh $ST_NODE1 ps -ef | grep pmon
grid      4111     1  0 10:53 ?        00:00:00 asm_pmon_+ASM1
oracle    4987     1  0 10:56 ?        00:00:00 ora_pmon_orcl1
oracle    5321  5254  0 12:45 pts/0    00:00:00 grep pmon

$ ssh $ST_NODE2 ps -ef | grep pmon
grid      4104     1  0 10:53 ?        00:00:00 asm_pmon_+ASM2
oracle    4959     1  0 10:56 ?        00:00:00 ora_pmon_orcl2

$ ssh $ST_NODE3 ps -ef | grep pmon
grid      4096     1  0 10:53 ?        00:00:00 asm_pmon_+ASM3
oracle    4841     1  0 10:55 ?        00:00:00 ora_pmon_orcl3

>>>>>>>>>>>>> or <<<<<<<<<<<<<<<<<

$ srvctl status database -d orcl -v
Instance orcl1 is running on node host01
Instance orcl2 is running on node host02
Instance orcl3 is running on node host03

$ srvctl status asm -a
ASM is running on host01,host02,host03
ASM is enabled.

>>>>>>>>>>>>> or <<<<<<<<<<<<<<<<<

$ /u01/app/11.2.0/grid/bin/crsctl status resource ora.orcl.db
NAME=ora.orcl.db
TYPE=ora.database.type
TARGET=ONLINE          , ONLINE          , ONLINE
STATE=ONLINE on host01, ONLINE on host02, ONLINE on host03

$ /u01/app/11.2.0/grid/bin/crsctl status resource ora.asm
NAME=ora.asm
```

Practice 12-3: Stopping a Complete ORACLE_HOME Component Stack (continued)

```
TYPE=ora.asm.type
TARGET=ONLINE          , ONLINE          , ONLINE
STATE=ONLINE on host01, ONLINE on host02, ONLINE on host03
```

3) Display the syntax usage help for the `srvctl status home` command.

```
$ srvctl status home -help

Displays the current state of of all resources for the Oracle
home.

Usage: srvctl status home -o <oracle_home> -s <state_file> -n
<node_name>
    -o <oracle_home>         ORACLE_HOME path
    -s <state_file>   Specify a file path for the 'srvctl stop
home' command to store the state of the resources
    -n <node_name>           Node name
    -h                       Print usage
```

4) Use the `srvctl status home` command to check the state of all resources running from the `/u01/app/oracle/product/11.2.0/dbhome_1` home location. Create the required state file in the `/tmp` directory with the file name `host01_dbhome_state.dmp` for the first node only.

```
$ srvctl status home -o
/u01/app/oracle/product/11.2.0/dbhome_1 -s
/tmp/host01_dbhome_state.dmp -n $ST_NODE1

Database orcl is running on node host01
```

5) Display the syntax usage help for the `srvctl stop home` command.

```
$ srvctl stop home -help

Stops all Oracle clusterware resources that run from the
Oracle home.

Usage: srvctl stop home -o <oracle_home> -s <state_file> -n
<node_name> [-t <stop_options>] [-f]
    -o <oracle_home>         ORACLE_HOME path
    -s <state_file>   Specify a file path for the 'srvctl stop
home' command to store the state of the resources
    -n <node_name>           Node name
    -t <stop_options>        Stop options for the database.
Examples of shutdown options are normal, transactional,
immediate, or abort.
    -f                       Force stop
    -h                       Print usage
```

Practice 12-3: Stopping a Complete ORACLE_HOME Component Stack (continued)

6) Stop all resources executing in the `/u01/app/oracle/product/11.2.0/dbhome_1` home using the state file created in step 4. Do not use the optional parameters identified by square brackets "[]" displayed in the syntax usage help.

```
$ srvctl stop home -o /u01/app/oracle/product/11.2.0/dbhome_1
-s /tmp/host01_dbhome_state.dmp -n $ST_NODE1
$
```

7) Check the status of the database instances on each node.
Note: There are several ways this step can be performed. Do not use the `srvctl status home` command with the same state file created above.

```
$ srvctl status database -d orcl -v
Instance orcl1 is not running on node host01
Instance orcl2 is running on node host02
Instance orcl3 is running on node host03
```

8) Start all resources for the `/u01/app/oracle/product/11.2.0/dbhome_1` home using the state file created by the `stop` command.

```
$ srvctl start home -o /u01/app/oracle/product/11.2.0/dbhome_1
-s /tmp/host01_dbhome_state.dmp -n $ST_NODE1
$
```

9) Check the status of the database instances on each node.
Note: There are several ways this step can be performed.

```
$ srvctl status database -d orcl -v
Instance orcl1 is running on node host01
Instance orcl2 is running on node host02
Instance orcl3 is running on node host03
```

Practices for Lesson 13

In this practice, you will configure ARCHIVELOG mode for your RAC database, configure instance-specific connect strings for RMAN, and configure persistent RMAN settings.

Practice 13-1: Configuring the Archive Log Mode

In this practice, you will configure the archive log mode of a Real Applications Cluster database.

1) From the first node of your cluster, open a terminal session as the `oracle` user and set up the environment variables using the `oraenv` script for the database instance. Change the value of the `ORACLE_SID` variable to allow local system authenticated connections.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle

$ export ORACLE_SID=orcl1
```

2) Make a local connection using operating system authentication to the database instance and then use the `archive log list` SQL command to determine if the database is in archive log mode. Exit SQL*Plus when done.

```
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 15:48:58
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> archive log list
Database log mode              No Archive Mode
Automatic archival             Disabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     11
Current log sequence           12
SQL> exit
$
```

3) Stop the `orcl` database on each node of the cluster using the `srvctl stop database` command.

```
$ srvctl stop database -d orcl
```

4) Verify that the `orcl` database is not running on any node of the cluster using the `srvctl status database` command.

```
$ srvctl status database -d orcl -v
Instance orcl1 is not running on node host01
Instance orcl2 is not running on node host02
Instance orcl3 is not running on node host03
```

5) Make a local connection using operating system authentication to the database instance and then start up the database on the first node only with the `mount` option.

```
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 16:27:17
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup mount
ORACLE instance started.

Total System Global Area  577511424 bytes
Fixed Size                  1338000 bytes
Variable Size             444597616 bytes
Database Buffers          125829120 bytes
Redo Buffers                5746688 bytes
Database mounted.
SQL>
```

6) Issue the `alter database archivelog` SQL command to change the archive mode of the database and then verify the results with the `archive log list` SQL command.

```
SQL> alter database archivelog;

Database altered.

SQL> archive log list
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     11
Next log sequence to archive   12
Current log sequence           12
```

7) Shut down the database instance with the immediate option and exit SQL*Plus. Use the `srvctl` utility to restart the database instances on all nodes of the cluster.

```
SQL> shutdown immediate
ORA-01109: database not open


Database dismounted.
```

```
ORACLE instance shut down.

SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

$ srvctl start database -d orcl
$
```

8) Verify that the `orcl` database is running on all the three nodes of your cluster by using the `srvctl status database` command.

```
$ srvctl status database -d orcl -v
Instance orcl1 is running on node host01
Instance orcl2 is running on node host02
Instance orcl3 is running on node host03
$
```

Practice 13-2: Configuring Specific Instance Connection Strings

In this practice, you will modify the `tnsnames.ora` file to disable connection load balancing and allow specific named instances to be used for connectivity.

1) Examine the `$ORACLE_HOME/network/admin/tnsnames.ora` file. There should be only one entry. This entry allows load balancing of connections as you observed in Practice 12-2.

```
$ cat
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/tnsnames
.ora

# tnsnames.ora Network Configuration File:
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/tnsnames
.ora
# Generated by Oracle configuration tools.

ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = cluster03-
scan.cluster03.example.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl.example.com)
    )
  )
```

2) Execute the `/home/oracle/labs/less_13/fix_tns.sh` script. This script will add three additional entries to the `tnsnames.ora` file that disable load balancing of connections by requiring a specific `INSTANCE_NAME` when used. Examine the changes made to the `tnsnames.ora` file.

```
$ /home/oracle/labs/less_13/fix_tns.sh

$ cat
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/tnsnames
.ora
# tnsnames.ora Network Configuration File:
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/tnsnames
.ora
# Generated by Oracle configuration tools.

ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = cluster03-
scan.cluster03.example.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )
```

```
# Added by lab 13
orcl1 =
   (DESCRIPTION =
     (ADDRESS_LIST =
       (ADDRESS = (PROTOCOL = TCP)(HOST = host01)(PORT=1521))
     )
     (CONNECT_DATA =
       (SERVICE_NAME = orcl)
       (INSTANCE_NAME = orcl1)
     )
   )
orcl2 =
   (DESCRIPTION =
     (ADDRESS_LIST =
       (ADDRESS = (PROTOCOL = TCP)(HOST = host02)(PORT=1521))
     )
     (CONNECT_DATA =
       (SERVICE_NAME = orcl)
       (INSTANCE_NAME = orcl2)
     )
   )
orcl3 =
   (DESCRIPTION =
     (ADDRESS_LIST =
       (ADDRESS = (PROTOCOL = TCP)(HOST = host03)(PORT=1521))
     )
     (CONNECT_DATA =
       (SERVICE_NAME = orcl)
       (INSTANCE_NAME = orcl3)
     )
   )
```

3) Using one of the three new entries in the `tnsnames.ora` file, connect to the system database account using SQL*Plus and verify the instance name to see that it matches the specific entry.

```
$ sqlplus system@orcl2

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 17:21:26
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Enter password: oracle_4U << Password is not displayed

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
```

Practice 13-2: Configuring Specific Instance Connection Strings (continued)

```
Data Mining and Real Application Testing options

SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
orcl2

SQL>
```

4) Use the SQL*Plus `host` command to temporarily exit SQL*Plus and return to the operating system prompt. **Note:** SQL*Plus is still running when this is performed. Repeat step 3 from the operating system prompt to establish a second SQL*Plus session and database instance connection using the same connection string. Verify that the `INSTANCE_NAME` stays the same.

```
SQL> host

$ sqlplus system@orcl2

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 17:26:30
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Enter password: oracle_4U << Password is not displayed

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
orcl2
```

5) Exit both SQL*Plus sessions.

```
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

$ exit
```

Practice 13-2: Configuring Specific Instance Connection Strings (continued)

```
exit

SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

Practice 13-3: Configuring RMAN and Performing Parallel Backups

In this practice, you will designate your first and second nodes of the cluster as nodes responsible for performing parallel backups of the database. The database will be backed up to the +FRA ASM disk group by default.

1) Using the recovery manager utility (RMAN), connect to the `orcl` database as the target database.

```
$ rman target /

Recovery Manager: Release 11.2.0.1.0 - Production on Tue Sep 8
17:31:34 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates.  All
rights reserved.

connected to target database: ORCL (DBID=1224399398)

RMAN>
```

2) Display all of the current RMAN settings.

```
RMAN> show all;

using target database control file instead of recovery catalog
RMAN configuration parameters for database with db_unique_name
ORCL are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP OFF; # default
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK
TO '%F'; # default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO
BACKUPSET; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; #
default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1;
# default
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE
'DEFAULT' OPTIMIZE FOR LOAD TRUE ; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'/u01/app/oracle/product/11.2.0/dbhome_1/dbs/snapcf_orcl1.f';
# default
```

3) Configure RMAN to automatically back up the control file and server parameter file each time any backup operation is performed.

```
RMAN> configure controlfile autobackup on;

new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored
```

4) Configure all backups done to disk to be done in parallel 2 degrees of parallelism.

```
RMAN> configure device type disk parallelism 2;

new RMAN configuration parameters:
CONFIGURE DEVICE TYPE DISK PARALLELISM 2 BACKUP TYPE TO
BACKUPSET;
new RMAN configuration parameters are successfully stored
```

5) Configure channel 1 and channel 2 to use the connect string '`sys/oracle_4U@orcl#`' when performing a parallel backup to disk. Replace the pound sign (#) with 1 for channel 1 and 2 for channel 2, respectively. This will designate your first and second nodes as dedicated backup nodes for the cluster using the node specific connection strings created earlier. Without node specific connection strings, there would be no control over which nodes are being connected to in order to perform the backups.

```
RMAN> configure channel 1 device type disk
connect='sys/oracle_4U@orcl1';

new RMAN configuration parameters:
CONFIGURE CHANNEL 1 DEVICE TYPE DISK CONNECT '*';
new RMAN configuration parameters are successfully stored

RMAN> configure channel 2 device type disk
connect='sys/oracle_4U@orcl2';

new RMAN configuration parameters:
CONFIGURE CHANNEL 2 DEVICE TYPE DISK CONNECT '*';
new RMAN configuration parameters are successfully stored
RMAN>
```

6) Open a second terminal session as the `oracle` account and set up the environment variables for the `orcl` database. Navigate to the `~/labs/less_13` directory, invoke SQL*plus as the `system` user, and run the `monitor_rman.sql` script. Do not exit the first session with the RMAN prompt or this second session with the SQL prompt.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
```

Practice 13-3: Configuring RMAN and Performing Parallel Backups (continued)

```
$ cd /home/oracle/labs/less_13

$ sqlplus system@orcl

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 8 20:37:36
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Enter password: oracle_4U << Password is not displayed

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> @monitor_rman.sql

no rows selected
```

7) In the first session with the RMAN prompt, perform a full database backup with archive logs. The backup should happen only on the designated nodes (your first and second nodes) as the backup nodes. **Do not wait for this step to finish before proceeding to the next step!**

```
RMAN> backup database plus archivelog;


Starting backup at 08-SEP-09
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=31 instance=orcl1 device type=DISK
allocated channel: ORA_DISK_2
channel ORA_DISK_2: SID=55 instance=orcl2 device type=DISK
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=2 sequence=4 RECID=1 STAMP=697048583
input archived log thread=3 sequence=10 RECID=4
STAMP=697062223
channel ORA_DISK_1: starting piece 1 at 08-SEP-09
channel ORA_DISK_2: starting archived log backup set
channel ORA_DISK_2: specifying archived log(s) in backup set
input archived log thread=1 sequence=12 RECID=2
STAMP=697062220
input archived log thread=2 sequence=5 RECID=3 STAMP=697062220
channel ORA_DISK_2: starting piece 1 at 08-SEP-09
channel ORA_DISK_2: finished piece 1 at 08-SEP-09
```

Practice 13-3: Configuring RMAN and Performing Parallel Backups (continued)

```
piece
handle=+FRA/orcl/backupset/2009_09_08/annnf0_tag20090908t20234
7_0.268.697062231 tag=TAG20090908T202347 comment=NONE
channel ORA_DISK_2: backup set complete, elapsed time:
00:00:04
channel ORA_DISK_1: finished piece 1 at 08-SEP-09
piece
handle=+FRA/orcl/backupset/2009_09_08/annnf0_tag20090908t20234
7_0.267.697062229 tag=TAG20090908T202347 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:00:05
Finished backup at 08-SEP-09

Starting backup at 08-SEP-09
using channel ORA_DISK_1
using channel ORA_DISK_2
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001
name=+DATA/orcl/datafile/system.260.696618497
input datafile file number=00005
name=+DATA/orcl/datafile/example.264.696618709
input datafile file number=00004
name=+DATA/orcl/datafile/users.267.696618501
input datafile file number=00006
name=+DATA/orcl/datafile/undotbs2.259.696619015
channel ORA_DISK_1: starting piece 1 at 08-SEP-09
channel ORA_DISK_2: starting full datafile backup set
channel ORA_DISK_2: specifying datafile(s) in backup set
input datafile file number=00002
name=+DATA/orcl/datafile/sysaux.268.696618499
input datafile file number=00003
name=+DATA/orcl/datafile/undotbs1.263.696618499
input datafile file number=00007
name=+DATA/orcl/datafile/undotbs3.258.696619021
channel ORA_DISK_2: starting piece 1 at 08-SEP-09
channel ORA_DISK_2: finished piece 1 at 08-SEP-09
piece
handle=+FRA/orcl/backupset/2009_09_08/nnndf0_tag20090908t20235
4_0.270.697062239 tag=TAG20090908T202354 comment=NONE
channel ORA_DISK_2: backup set complete, elapsed time:
00:01:35
channel ORA_DISK_1: finished piece 1 at 08-SEP-09
piece
handle=+FRA/orcl/backupset/2009_09_08/nnndf0_tag20090908t20235
4_0.269.697062235 tag=TAG20090908T202354 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:01:47
Finished backup at 08-SEP-09
```

```
Starting backup at 08-SEP-09
current log archived
using channel ORA_DISK_1
using channel ORA_DISK_2
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=13 RECID=5
STAMP=697062345
input archived log thread=2 sequence=6 RECID=7 STAMP=697062345
channel ORA_DISK_1: starting piece 1 at 08-SEP-09
channel ORA_DISK_2: starting archived log backup set
channel ORA_DISK_2: specifying archived log(s) in backup set
input archived log thread=3 sequence=11 RECID=6
STAMP=697062345
channel ORA_DISK_2: starting piece 1 at 08-SEP-09
channel ORA_DISK_1: finished piece 1 at 08-SEP-09
piece
handle=+FRA/orcl/backupset/2009_09_08/annnf0_tag20090908t20254
6_0.274.697062347 tag=TAG20090908T202546 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:00:02
channel ORA_DISK_2: finished piece 1 at 08-SEP-09
piece
handle=+FRA/orcl/backupset/2009_09_08/annnf0_tag20090908t20254
6_0.275.697062347 tag=TAG20090908T202546 comment=NONE
channel ORA_DISK_2: backup set complete, elapsed time:
00:00:01
Finished backup at 08-SEP-09

Starting Control File and SPFILE Autobackup at 08-SEP-09
piece
handle=+FRA/orcl/autobackup/2009_09_08/s_697062349.276.6970623
53 comment=NONE
Finished Control File and SPFILE Autobackup at 08-SEP-09

RMAN>
```

8) While the backup is in progress, rerun the query on the second terminal window to monitor the RMAN backup session progress within the cluster. The backup should be done in parallel, with work distributed to both the backup nodes of the cluster. Enter the slash (/) symbol and press the Enter key to rerun the query. It may be necessary to do this multiple times until the output appears.

```
SQL> /

no rows selected

SQL> /
```

Practice 13-3: Configuring RMAN and Performing Parallel Backups (continued)

```
no rows selected
SQL> /

INST_ID  SID  SERIAL# CONTEXT     SOFAR  TOTALWORK %_COMPLETE
------- ---- -------- ------- -------- ---------- ----------
      1   31     1913       1    13308     104960      12.68
      2   55      284       1    23934     101760      23.52

SQL> /

INST_ID SID  SERIAL#  CONTEXT     SOFAR  TOTALWORK %_COMPLETE
------- --- -------- -------- -------- ---------- ----------
      1  31     1913        1    44798     104960      42.68
      2  55      284        1    49278     101760      48.43

SQL> exit
```

9) Run the `/home/oracle/labs/less_13/cleanup_13.sh` script.

```
$ /home/oracle/labs/less_13/cleanup_13.sh
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
```

10) Exit all windows when finished.

Practices for Lesson 14

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

Practice 14-1: ADDM and RAC Part I

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

Note that all the necessary scripts for this lab are located in the `/home/oracle/labs/seq` directory on your first cluster node.

1) Before you start this lab, make sure that you have the necessary TNS entries in the `tnsnames.ora` file located in your ORACLE_HOME for the orcl database. You can execute the following script to create those entries: `add_tnsinstances.sh`.

```
$ cd /home/oracle/labs/seq
$ ./add_tnsinstances.sh
# tnsnames.ora… Network Configuration File:
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/tnsnames
.ora…
# Generated by Oracle configuration tools.


ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = ….example.com)(PORT =
1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )


orcl1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = <host01>)(PORT =
1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = orcl)
      (INSTANCE_NAME = orcl1)
    )
  )

orcl2 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = <host02>)(PORT =
1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = orcl)
      (INSTANCE_NAME = orcl2)
```

Practice 14-1: ADDM and RAC Part I (continued)

```
    )
  )

orcl3 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = <host03>)(PORT =
1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = orcl)
      (INSTANCE_NAME = orcl3)
    )
  )

$
```

2) Execute the `setupseq1.sh` script to set up the necessary configuration for this lab.

```
$ ./setupseq1.sh


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.

 drop user jfv cascade
           *
ERROR at line 1:
ORA-01918: user 'JFV' does not exist


 drop tablespace seq including contents and datafiles
*
ERROR at line 1:
ORA-00959: tablespace 'SEQ' does not exist



Tablespace created.


User created.


Grant succeeded.

 drop sequence s
               *
ERROR at line 1:
```

Practice 14-1: ADDM and RAC Part I (continued)

```
ORA-02289: sequence does not exist


 drop table s purge
            *
ERROR at line 1:
ORA-00942: table or view does not exist


 drop table t purge
            *
ERROR at line 1:
ORA-00942: table or view does not exist



Table created.


Table created.


Index created.


1 row created.


Commit complete.


PL/SQL procedure successfully completed.

$
```

3) Using Database Control, and connected as the `SYS` user, navigate to the Performance page of your Cluster Database.
   a) Click the Performance tab from the Cluster Database Home page.
   b) On the Cluster Database Performance page, make sure that Real Time: 15 Seconds Refresh is selected from the View Data drop-down list.
4) Use PL/SQL to create a new AWR snapshot.

```
$ ./create_snapshot.sh

PL/SQL procedure successfully completed.

$
```

Practice 14-1: ADDM and RAC Part I (continued)

5) Execute the `startseq1.sh` script to generate a workload on all instances of your cluster. Do not wait; proceed with the next step.

```
$ ./startseq1.sh
$ old   7:   insert into t values(v,'&1');
new   7:   insert into t values(v,'orcl2');
old   7:   insert into t values(v,'&1');
new   7:   insert into t values(v,'orcl1');
old   7:   insert into t values(v,'&1');
new   7:   insert into t values(v,'orcl3');

… Do not wait after this point and go to the next step.

PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


$
```

6) Using Database Control, determine the list of blocking locks in your database.
   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 drop-down 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 must see that all sessions are alternatively waiting for the other to release the exclusive lock held on table S.
7) While the scripts are still executing, look at the Average Active Sessions graphic. Then, drill down to the Cluster wait class for the first node. What are your conclusions?
   a) By using the drill-down method of Enterprise Manager, you can quickly identify the top waiting SQL statements and the top waiting sessions on both instances. Here it appears that an `UPDATE` statement on table S is causing most of the waits for the Cluster wait class.
   b) Click Cluster Database in the locator link at the top of the page to return to the Cluster Database Performance page.

Practice 14-1: ADDM and RAC Part I (continued)

c) From there you can now see the Average Active Sessions graph. Make sure that the View Data field is set to Real Time:15 Seconds Refresh. After a few seconds, the graphic must clearly show that the Cluster and Application wait classes are causing most waits. Using the Throughput tabbed page graph underneath the Average Active Sessions graph, you should also notice that the transaction rate is about 250 per second.

d) In the Average Active Sessions graph, click the Cluster link on the right. This takes you to the Active Sessions By Instance: Cluster page.

e) On the Active Sessions By Instance: Cluster page, you will see that the number of active sessions is almost the same on all nodes. Click the first instance's link (instance number 1). This takes you to the Active Sessions Waiting: Cluster page for the corresponding instance.

f) On the Active Sessions Waiting: Cluster page, you can see the most important wait events causing most of the waits in the Cluster wait class on the first instance. In the Top SQL: Cluster section, click the SQL identifier that uses most of the resources. This takes you to the SQL Details page for the corresponding statement. You will see that the script running on the first instance is executing a SELECT/UPDATE statement on table S that causes most of the Cluster waits.

8) Using Database Control, look at the Cluster Cache Coherency page. What are your conclusions?

a) On the Cluster Database Home page, click the Performance tab.

b) On the Performance page, click the Cluster Cache Coherency link in the Additional Monitoring Links section.

c) The Cluster Cache Coherency page clearly shows that there are lots of blocks transferred per second on the system. This represents more than 17% of the total logical reads. This is reflected in both the Global Cache Block Transfer Rate and the Global Cache Block Transfers and Physical Reads (vs. Logical Reads) graphics.

d) On the Cluster Cache Coherency page, you can also click Interconnects in the Additional Links section of the page to get more information about your private interconnect.

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

a) By using the drill-down method of Enterprise Manager, you can quickly identify the top waiting SQL statements and the top waiting sessions on both instances. Here it appears that a LOCK statement on table S is causing most of the waits for the Application wait class.

b) Go back to the Cluster Database Home page by clicking the Database tab located on the top right-end corner. On the Cluster Database Home page, click the Performance tab.

Practice 14-1: ADDM and RAC Part I (continued)

c) On the Performance page, make sure that the View Data field is set to Real Time: 15 Seconds Refresh. After a few seconds, the graphic should clearly show that the Cluster and Application wait classes are causing most waits. You will also notice that the transaction rate is about 100 per second.

d) In the Average Active Sessions graph, click the Application link on the right. This takes you to the Active Sessions By Instance: Application page.

e) On the Active Sessions By Instance: Application page, you must see that the number of active sessions is almost the same on all nodes. Click the link for the first instance (number 1) on the Summary Chart graph. This takes you to the Active Sessions Waiting: Application page of the first instance.

f) On the Active Sessions Waiting: Application page, you can see the most important wait events causing most of the waits in the Application wait class on the first instance. In the Top SQL: Application section, click the SQL identifier that uses most of the resources. This takes you to the SQL Details page for the corresponding statement. You must see that the script running on the first instance is executing a LOCK statement on table S that causes most of the Application waits.

g) After a while, you can see that all scripts are executed by looking at the Average Active Sessions graph as well as the Database Throughput graphics again. You should see the number of transactions per second going down.

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

```
$ ./create_snapshot.sh

PL/SQL procedure successfully completed.

$
```

11) Using Database Control, review the latest ADDM run. What are your conclusions?

a) On the Cluster Database Home page, click the Advisor Central link in the Related Links section.

b) On the Advisor Central page, make sure that the Advisory Type field is set to All Types, and that the Advisor Runs field is set to Last Run. Click Go.

c) In the Results table, select the latest ADDM run corresponding to Instance All. Then click View Result. This takes you to the Automatic Database Diagnostic Monitor (ADDM) page.

d) On the Automatic Database Diagnostic Monitor (ADDM) page, the ADDM Performance Analysis table shows you the consolidation of ADDM reports from all instances running in your cluster. This is your first entry point before drilling down to specific instances. From there, investigate the Top SQL Statements, Table Locks, and Global Cache Messaging findings.

e) Click the Top SQL Statements finding, which affects all instances, revealing `LOCK TABLE S` and `UPDATE S` commands as a possible problem to investigate. Click the Back button to return to the ADDM report.

f) Click the Table Locks finding, which affects all instances, revealing that you should investigate your application logic regarding the JFV.S object.

g) Click the Global Cache Messaging finding revealing again the `UPDATE S` command as responsible for approximately 30% of Cluster waits during the analysis period.

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

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

Practice 14-2: ADDM and RAC Part II

The goal of this lab is to show you how to manually discover performance issues by using the Enterprise Manager performance pages as well as ADDM. In this second part of the practice, you are going to correct the previously found issue by creating a sequence number instead of by using a table.

Note that all the necessary scripts for this lab are located in the `/home/oracle/labs/seq` directory on your first cluster node.

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

```
$ ./setupseq2.sh

PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


User dropped.


Tablespace dropped.


Tablespace created.


User created.


Grant succeeded.

 drop table s purge
            *
ERROR at line 1:
ORA-00942: table or view does not exist


 drop sequence s
               *
ERROR at line 1:
ORA-02289: sequence does not exist


 drop table t purge
            *
ERROR at line 1:
ORA-00942: table or view does not exist
```

Practice 14-2: ADDM and RAC Part II (continued)

```
Table created.


Index created.


Sequence created.


PL/SQL procedure successfully completed.

$
```

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

   a) Click the Performance tab from the Cluster Database Home page.

   b) On the Cluster Database Performance page, make sure Real Time: 15 Seconds Refresh is selected from the View Data drop-down list.

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

```
$ ./create_snapshot.sh

PL/SQL procedure successfully completed.

$
```

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

```
$ ./startseq2.sh
$ old   3:   insert into t values(s.nextval,'&1');
new   3:   insert into t values(s.nextval,'orcl1');
old   3:   insert into t values(s.nextval,'&1');
new   3:   insert into t values(s.nextval,'orcl3');
old   3:   insert into t values(s.nextval,'&1');
new   3:   insert into t values(s.nextval,'orcl2');

… Do not wait after this point and go to the next step.

PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.
$
```

5) While the scripts are still executing, look at the Average Active Sessions graphic. Then drill down to the Cluster wait class for the first node. What are your conclusions?
   a) By using the drill-down method of Enterprise Manager, you can quickly identify the top waiting SQL statements and the top waiting sessions on both instances. Here it appears that an `INSERT` statement on table T is causing most of the waits for the Cluster wait class.
   b) Click Cluster Database in the locator link at the top of the page to return to the Cluster Database Performance page.
   c) From there you can now see the Average Active Sessions graph. Make sure that the View Data field is set to Real Time:15 Seconds Refresh. After a few seconds, the graphic will clearly show that the Cluster and Application wait classes are causing most waits. Using the Throughput tabbed page graph underneath the Average Active Sessions graph, you should also notice that the transaction rate is about 320 per second (a better rate than in the previous practice).
   d) In the Average Active Sessions graph, click the Cluster link on the right. This takes you to the Active Sessions By Instance: Cluster page.
   e) On the Active Sessions By Instance: Cluster page, you must see that the number of active sessions is almost the same on all nodes. Click the first instance's link (instance number 1). This takes you to the Active Sessions Waiting: Cluster page for the corresponding instance.
   f) On the Active Sessions Waiting: Cluster page, you can see the most important wait events causing most of the waits in the Cluster wait class on the first instance. In the Top SQL: Cluster section, click the SQL identifier that uses most of the resources. This takes you to the SQL Details page for the corresponding statement. You will see that the script running on the first instance is executing an `INSERT` statement on table T that causes most of the Cluster waits.
   g) After a while you can see that all are executed by looking at the Average Active Sessions graphic again. The Database Throughput graphic tells you that this time, the number of transactions per second was a bit higher than in the previous lab for the same workload. Using the sequence number was a bit better in this case.
6) After the workload finishes, use PL/SQL to create a new AWR snapshot.

```
$ ./create_snapshot.sh

PL/SQL procedure successfully completed.

$
```

7) Using Database Control, review the latest ADDM run. What are your conclusions?
   a) On the Cluster Database Home page, click the Advisor Central link.
   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.

Practice 14-2: ADDM and RAC Part II (continued)

c) In the Results table, select the latest ADDM run corresponding to Instance All. Then click View Result. This takes you to the Automatic Database Diagnostic Monitor (ADDM) page.

d) On the Automatic Database Diagnostic Monitor (ADDM) page, the ADDM Performance Analysis table shows you the consolidation of ADDM reports from all instances running in your cluster. This is your first entry point before drilling down to specific instances. From there, investigate the Top SQL Statements, Sequence Usage, and Unusual "Concurrency" Wait Event findings.

e) The Top SQL Statements should reveal an `INSERT INTO T` command using sequence S as a possible problem to investigate.

f) The Sequence Usage finding reveals that you should use larger cache size for your hot sequences.

g) The Unusual "Concurrency" Wait Event finding asks you to investigate the cause for high "row cache lock" waits. Refer to the *Oracle Database Reference* for the description of this wait event.

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

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

Practice 14-3: ADDM and RAC Part III

The goal of this lab is to show you how to manually discover performance issues by using the Enterprise Manager performance pages as well as ADDM. This last part generates the same workload as in the previous lab but uses more cache entries for sequence number S.

Note that all the necessary scripts for this lab are located in the `/home/oracle/labs/seq` directory on your first cluster node.

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

```
$ ./setupseq3.sh

PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


User dropped.


Tablespace dropped.


Tablespace created.


User created.


Grant succeeded.

 drop table s purge
            *
ERROR at line 1:
ORA-00942: table or view does not exist


 drop sequence s
               *
ERROR at line 1:
ORA-02289: sequence does not exist


 drop table t purge
            *
ERROR at line 1:
ORA-00942: table or view does not exist
```

```
Table created.


Index created.


Sequence created.


PL/SQL procedure successfully completed.

$
```

2) Using Database Control, and connected as the `SYS` user, navigate to the Performance page of your Cluster Database.
   a) Click the Performance tab from the Cluster Database Home page.
   b) On the Cluster Database Performance page, make sure that Real Time: 15 Seconds Refresh is selected from the View Data drop-down list.
3) Use PL/SQL to create a new AWR snapshot.

```
$ ./create_snapshot.sh

PL/SQL procedure successfully completed.

$
```

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

```
$ ./startseq2.sh
$ old   3:   insert into t values(s.nextval,'&1');
new   3:   insert into t values(s.nextval,'orcl3');
old   3:   insert into t values(s.nextval,'&1');
new   3:   insert into t values(s.nextval,'orcl2');
old   3:   insert into t values(s.nextval,'&1');
new   3:   insert into t values(s.nextval,'orcl1');

… Do not wait after this point and go to the next step.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.
$
```

Practice 14-3: ADDM and RAC Part III (continued)

5) Until the scripts are executed, look at the Average Active Sessions graphic. What are your conclusions?
   a) This time, looking at the Average Active Sessions graphic, it is clear that there are no significant waits. The sequence has a big enough cache value to avoid the most significant waits.
   b) Click Cluster Database in the locator link at the top of the page to return to the Cluster Database Performance page.
   c) On the Performance page, make sure that the View Data field is set to Real Time:15 Seconds Refresh. After all the scripts have finished their execution, the Average Active Sessions graph will clearly show that there are no significant waits on your cluster. You must also notice that the transaction rate is now around 2400 per second.
6) After the workload finishes, use PL/SQL to create a new AWR snapshot.

```
$ ./create_snapshot.sh

PL/SQL procedure successfully completed.

$
```

7) Using Database Control, review the latest ADDM run. What are your conclusions?
   a) On the Cluster Database Home page, click the Advisor Central link.
   b) On the Advisor Central page, make sure that the Advisory Type field is set to All Types, and that the Advisor Runs field is set to Last Run. Click Go.
   c) In the Results table, select the latest ADDM run corresponding to Instance All. Then click View Result. This takes you to the Automatic Database Diagnostic Monitor (ADDM) page.
   d) On the Automatic Database Diagnostic Monitor (ADDM) page, the ADDM Performance Analysis table shows you the consolidation of ADDM reports from all instances running in your cluster. This is your first entry point before drilling down to specific instances. From there, investigate the Buffer Busy – Hot Block, Buffer Busy – Hot Objects, and Global Cache Busy findings. You should no longer see the Sequence Usage, nor specific instances impacted.
   e) The Buffer Busy – Hot Block finding should not reveal any particular object.
   f) The Buffer Busy – Hot Objects finding should not reveal any particular object.
   g) The Global Cache Busy finding should not reveal anything special.

Practices for Lesson 15

In these practices, you will create, manage, and monitor services.

Practice 15-1: Working with Services

In this practice, you will use Enterprise Manager to create one service called PROD1. You then observe what happens to your service when you terminate one of the instances on which it is running.

1) Use Enterprise Manager to create the PROD1 service. Make sure that you define your first and second instances (ORCL1 and ORCL2) as preferred, and the third instance (ORCL3) as available.
   a) Enter your EM address in a browser. It will look something like this:

      `https://your_host_name:1158/em`

   b) Log in using SYS credentials as SYSDBA.
   c) Click the Availability folder tab.
   d) Click the Cluster Managed Database Services link under the Services section.
   e) On the Cluster Managed Database Services: Cluster and Database Login page, provide the login credentials for the operating system user (`oracle/0racle`) and the SYSDBA credentials for the database (`sys/oracle_4U`) and click Continue.
   f) Click the Create Service button on the Cluster Managed Database Services page.
   g) On the Create Service page, enter `PROD1` for the service name. Verify that the "Start service after creation" check box is selected, and select the "Update local naming" check box. Under the High Availability Configuration section, set the service policy for `orcl1` and `orcl2` to Preferred and ORCL3 to Available. Leave the remaining fields with their default values and click the OK button.
   h) After the service has been created, you will be returned to the Cluster Managed Database Services page. Check the Running Instances column for PROD1, it should indicate the service running on `orcl1` and `orcl2`. Select PROD1 from the Services list and click the Test Connection button. It should test successfully. Click the Show All TNS Strings button and inspect the new entry to the `tnsnames.ora` file. It should look like this:

```
PROD1 = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)
(HOST = cluster01-scan.cluster01.example.com)
(PORT = 1521))(LOAD_BALANCE = YES)(CONNECT_DATA =
(SERVER = DEDICATED)(SERVICE_NAME = PROD1)))
```

   i) Click the Return button.

2) Use the `srvctl` command to check the status of the new service.

```
$ srvctl status service -d ORCL -s PROD1
Service PROD1 is running on instance(s) orcl1,orcl2

$
```

3) Use the `crsctl` command to view server pool relationships with the new service.

Practice 15-1: Working with Services (continued)

```
$ /u01/app/11.2.0/grid/bin/crsctl status serverpool -p

NAME=Free
IMPORTANCE=0
MIN_SIZE=0
MAX_SIZE=-1
SERVER_NAMES=
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r-x

NAME=Generic
IMPORTANCE=0
MIN_SIZE=0
MAX_SIZE=-1
SERVER_NAMES=host01 host02 host03
PARENT_POOLS=
EXCLUSIVE_POOLS=
ACL=owner:grid:r-x,pgrp:oinstall:r-x,other::r-x

NAME=myApache_sp
IMPORTANCE=0
MIN_SIZE=0
MAX_SIZE=-1
SERVER_NAMES=
PARENT_POOLS=Generic
EXCLUSIVE_POOLS=
ACL=owner:grid:rwx,pgrp:oinstall:rwx,other::r--

NAME=ora.orcl
IMPORTANCE=1
MIN_SIZE=0
MAX_SIZE=-1
SERVER_NAMES=host01 host02 host03
PARENT_POOLS=Generic
EXCLUSIVE_POOLS=
ACL=owner:oracle:rwx,pgrp:oinstall:rwx,other::r--

NAME=ora.orcl_PROD1
IMPORTANCE=0
MIN_SIZE=0
MAX_SIZE=-1
SERVER_NAMES=host01 host02 host03
PARENT_POOLS=ora.orcl
EXCLUSIVE_POOLS=
ACL=owner:oracle:rwx,pgrp:oinstall:rwx,other::r--

$
```

Practice 15-1: Working with Services (continued)

4) Connect to the service and look at the current value of the SERVICE_NAMES initialization parameter, and verify that it is set correctly. Query V$INSTANCE and determine what instance you are connected to.

```
$ /u01/app/oracle/product/11.2.0/dbhome_1/bin/sqlplus
sys/oracle_4U@PROD1 as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Fri Sep 4 11:10:29
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> show parameter service

NAME                   TYPE        VALUE
---------------------- ----------- --------------
service_names          string      PROD1

SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
orcl1

SQL> exit
```

5) From a terminal session as the oracle user, crash the instance on the first node. Find and kill the ora_pmon_orcl process. Use the pkill -9 –f pmon_orcl command to crash the database instance. The orcl1 instance will crash and the clusterware services will restart it very quickly

```
$ pkill -9 -f pmon_orcl1
```

6) Use the srvctl command to check the status of the PROD1 service. (It may take a few moments to show up on the orcl3)

```
$ srvctl status service -d ORCL -s PROD1
Service PROD1 is running on instance(s) orcl2,orcl3

$
```

Practice 15-1: Working with Services (continued)

7) Return to Enterprise Manager. Click the Availability folder tab. In the instance list under the Instances section, you should be able to verify that the first instance is indeed down.
8) Click the Cluster Managed Database Services link. On the Cluster Managed Database Service page, you can see `orcl2` and `orcl3` in the running instances column for PROD1. Select Manage from the Actions drop-down list and click Go.
9) Under the instances section, find the host that `orcl3` is running on and select the option in the Select column for that host. Click the Relocate button.
10) On the Relocate Service from Instance: orcl03 page, select the host name that `orcl1` is running on and click OK.
11) You should see a message indicating that the service was relocated successfully. Under the Instances section of the page, you should see the service running on `orcl1` and `orcl2` and stopped on `orcl3`.
    Note: The instance status shown in EM may still show that the instance is down. Click the browser Refresh button to see the actual status of the instance.

Practice 15-2: Monitoring Services

In this practice, you will use Database Control to determine the amount of resources used by sessions executing under a particular service.

1) As the `oracle` user, open a terminal session to your first node. Execute the `/home/oracle/labs/less_15/createuser.sh` script. This script creates a new user called FOO identified by the password `foo`. The default tablespace of this user is USERS, and its temporary tablespace is TEMP. This new user has the CONNECT, RESOURCE, and DBA roles.

```
$ cat /home/oracle/labs/less_15/createuser.sh

export ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
export ORACLE_SID=orcl1
/u01/app/oracle/product/11.2.0/dbhome_1/bin/sqlplus -s /NOLOG
<<EOF

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

EOF
$ /home/oracle/labs/less_15/createuser.sh
drop user FOO cascade
          *
ERROR at line 1:
ORA-01918: user 'FOO' does not exist



User created.


Grant succeeded.

$
```

2) Using SQL*Plus, connect to PROD1 as FOO. When connected, determine the instance on which your session is currently running. Then execute the following query:
`select count(*) from dba_objects,dba_objects,dba_objects`
Do not wait; instead, proceed with the next step.

```
$ sqlplus foo/foo@PROD1
SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
orcl1
```

Practice 15-2: Monitoring Services (continued)

```
SQL> select count(*) from dba_objects,dba_objects,dba_objects;
```

3) After a few moments, go to the Database Control Top Consumers page from the Cluster Database page. Connect as user SYS. Then check that PROD1 is using more and more resources.
   a. From the Cluster Database Home page, click the Performance tab.
   b. On the Performance page, click the Top Consumers link in the Additional Monitoring Links section.
   c. This takes you to the Top Consumers page with the Overview tab selected.
   d. On the Overview page, you can see the Top Services pie chart.
   e. Make sure that the View Data drop-down list is set to Real Time: 15 Second Refresh. Wait for the page to be refreshed a couple of times. Little by little, PROD1 is consuming almost all the resources (up to 100%).
   f. To have more details, click Top Services tab on the Top Consumers page.
   g. Make sure that the View Data drop-down list is set to Real Time: 15 Second Refresh, and View drop-down list is set to Active Services. You can click the "+" icon on the left of the PROD1 link to expand the service. This shows you the list of instances currently running the service. You can also click the PROD1 link itself to look at the detailed Statistics of the corresponding service.
4) In another terminal window as the `oracle` user, check statistics on your service with `gv$service_stats` from a SQL*Plus session connected as SYSDBA.

```
$ /u01/app/oracle/product/11.2.0/dbhome_1/bin/sqlplus
sys/oracle_4U@orcl as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Fri Sep 4 16:16:48
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> select stat_name, sum(value) from gv$service_stats where
service_name = 'PROD1' group by stat_name;

STAT_NAME                                                  SUM(VALUE)
---------------------------------------------------- ----------
user calls                                                         43
DB CPU                                                     1368469523
redo size                                                        1564
```

Practice 15-2: Monitoring Services (continued)

```
db block changes                                                  8
DB time                                                  1473281835
user rollbacks                                                    0
gc cr blocks received                                             2
gc cr block receive time                                          0
gc current blocks received                                        2
opened cursors cumulative                                        99
workarea executions - multipass                                   0

STAT_NAME                                                 SUM(VALUE)
--------------------------------------------------- -----------
session cursor cache hits                                        45
user I/O wait time                                             3540
parse count (total)                                              71
physical reads                                                    4
gc current block receive time                                     0
workarea executions - optimal                                    22
concurrency wait time                                         17361
parse time elapsed                                           110704
physical writes                                                   0
workarea executions - onepass                                     0
execute count                                                    96

STAT_NAME                                                  SUM(VALUE)
---------------------------------------------------- ----------
session logical reads                                          3825
cluster wait time                                              2161
application wait time                                         20622
logons cumulative                                                 2
sql execute elapsed time                                 1473090354
user commits                                                      0

28 rows selected.

SQL>
```

Practice 15-3: Services and Alert Thresholds

In this practice, you will set thresholds for service PROD1, and use Database Control to monitor the response time metric for this service. In this practice, you will set the Elapsed Time in seconds warning threshold at 4 and the critical threshold at 1. Preferred instances should be orcl1 and orcl2, and orcl3 should be available.

1) Set alert thresholds for your service PROD1 using Database Control.
   a) Log in as `sys` with SYSDBA privileges.
   b) On the Database Home page, click the Availability folder tab. Then click the Cluster Managed Database Services link.
   c) Select PROD1 from the Services list, select Edit Properties from the Actions drop-down list and click Go.
   d) Under the High Availability Configuration section, set the Service Policy for orcl1 to Preferred and Available for orcl2 and orcl3. Then click OK.
   e) Return to the Cluster Database home page, click the link corresponding to your first instance in the Instances table. This is the instance currently running PROD1.
   f) On the Database Instance page, click Metric and Policy settings in the Related Links section at the bottom of the page.
   g) On the Metric and Policy Settings page, select All metrics from the View drop-down list.
   h) Scroll down the Metric and Policy Settings page until you find the Service Response Time (per user call) (microseconds) metric.
   i) On the same line, click the corresponding multi-pens icon in the last column (Edit column).
   j) On the Edit Advanced Settings: Service Response Time (per user call) (microseconds) page, click Add.
   k) The Monitored Objects table should now show two entries.
   l) Enter `PROD1` in the Service Name field, `40000000` in the Warning Threshold field, and `100000000` in the Critical Threshold field. Make sure that the corresponding line is selected, and click Continue.
   m) On the Metric and Policy Settings page, you should see an Information warning explaining that your settings have been modified but not saved. Click OK to save the new settings.
   n) On the Confirmation page, you can see an Update succeeded message. Click OK.
   o) This takes you back to the Database Instance page.
2) Use Database Control to view the Service Response Time Metric Value graphic for PROD1.

a) From the Database Instance page, click All Metrics in the Related Links section at the bottom of the page.

b) On the All Metrics page, expand the Database Services link. On the All Metrics page, click the Service Response Time (per user call) (microseconds) link.

c) On the Service Response Time (per user call) (microseconds) page, click the PROD1 link in the Service Name column.

d) On the Service Response Time (per user call) (microseconds): Service Name PROD1: Last 24 hours page, select Real Time: 30 Second Refresh from the View Data drop-down list.

e) You should now see the Service Response Time (per user call) (microseconds): Service Name PROD1 page with your warning and critical thresholds set correctly.

3) Execute the `serv_wkload.sh` script to generate workload on your database. Looking at the Service Response time graphic for PROD1, what do you observe?

```
$ cd /home/oracle/labs/less_15
$ ./serv_wkload.sh
```

a) Still looking at the Service Response Time (per user call) (microseconds): Service Name PROD1 page on your first session, you should see the graphic crossing the warning threshold after few minutes. This will trigger a warning alert soon after the warning threshold is crossed.

b) You can see this alert propagated to your Database Instance Home page, and Cluster Database Home page.

c) To go back to your Database Instance Home page, click the Database Instance locator link on the Service Response Time page.

d) You should see the warning raised in the Alerts section of the Database Instance page.

e) On the Database Instance page, click the Cluster Database locator link of the page.

f) You should see the warning alert in the Problem Services line in the High Availability section of the page. Clicking this link takes you to the Cluster Home page. From there you can click the PROD1 link to directly go to the Cluster Managed Database Services: PROD1 page after you clicked Continue on the Login page. The PROD1 page shows you the alert with its details.

g) Soon after the script finishes its execution, you should not see the corresponding alert on your Cluster Database Home page anymore. You can go to the Alert History page on the first instance to look at the alert history for your services. You can go to the Database Instance Home page using the locator links at the top of any pages. From the Database Instance Home page, scroll down to the bottom of the page, and click Alert History in the Related Links section.

Practice 15-3: Services and Alert Thresholds (continued)

4) Use Database Control to remove the thresholds that you specified during this practice.
   a) From the Cluster Database Home page, click the link corresponding to the first instance of your cluster in the Instances section at the bottom of the page.
   b) On the Database Instance page, scroll down to the bottom of the page. Click Metric and Policy Settings in the Related Links section.
   c) On the Metric and Policy Settings page, scroll down the page until you see PROD1 in the Metric Thresholds table.
   d) On the line corresponding to the PROD1 entry, remove both the Warning Threshold and Critical Threshold values.
   e) Click OK.
   f) On the Confirmation page, you should see an Update succeeded message. Click OK.

B

DHCP and DNS Configuration For GNS

Objectives

After completing this lesson, you should be able to:

- Configure or communicate DHCP configuration needs in support of GNS.
- Configure or communicate DNS configuration needs in support of GNS.

GNS Overview

- In a static configuration all the addresses are assigned by administrative action.
- DHCP provides dynamic configuration of host IP addresses.
  – DHCP does not provide a good way to produce good names that are useful to external clients
  – Because of this, it is rarely used in server complexes.
- To configure GNS:
  – It is necessary configure the higher level DNS to forward or delegate a subdomain to the cluster.
  – The cluster must run GNS on an address known to the DNS.

ORACLE

GNS Overview

In a static configuration, all the addresses are assigned by administrative action and given names that resolve with whatever name service is provided for the environment. This is universal historic practice, as there has been no realistic alternative. One result is significant turn around time to obtain the address, and to make the name resolvable. This is undesirable for dynamic reassignment of nodes from cluster to cluster and function to function

DHCP provides for dynamic configuration of the hosts IP address but doesn't provide a good way to produce good names that are useful to external clients. As a result, it is rarely used in server complexes because the point of a server is to provide service, and the clients need to be able to find the server. This is solved in the current release by providing a service (GNS) for resolving names in the cluster, and defining this to the DNS service used by the clients.

To properly configure GNS to work for clients, it is necessary to configure the higher level DNS to forward or delegate a subdomain to the cluster and the cluster must run GNS on an address known to the DNS, by number. This GNS address is maintained as a VIP in the cluster, run on a single node, and a GNSD process that follows that VIP around the cluster and service names in the subdomain. To fully implement GNS, you need four things.

1. DHCP service for the public network in question;
2. A single assigned address in the public network for the cluster to use as the GNS VIP.
3. A forward from the higher level DNS for the cluster to the GNS VIP.
4. A running cluster with properly configured GNS

DHCP Service

- With DHCP, a host needing an address sends a broadcast message to the network.
- A DHCP server on the network responds to the request, and assigns an address, along with other information like:
  - What gateway to use
  - What DNS servers to use
  - What domain should be used
- In the request, a host typically sends a client identifier, usually the MAC address of the interface in question.
- The identifier sent by Clusterware is not a MAC address, but a VIP resource name like `ora.hostname.vip`
- Because the IP address is not bound to a fixed MAC address, Clusterware can move it between hosts as needed.

ORACLE

DHCP SERVICE

With DHCP, a host needing an address sends a broadcast message to the network. A DHCP server on the network can respond to the request, and give back an address, along with other information such as what gateway to use, what DNS server(s) to use, what domain should be used, what NTP server should be used, and so on.

In the request, a host typically identifies itself to the server by sending the MAC address of the interface in question. It can also send other values. Our uses for VIPs send a "client identifier" that is the name of the CRS resource associated. Therefore, instead of sending MAC address 00:04:23:A5:B2:C0, we'll send something like `ora.hostname.vip`. This lets Clusterware easily move the address from one physical host to another, because it is not bound to a particular hardware MAC address. There are three ways to get DHCP service:

- It can be there already, provided by the network administrator.
- You can provide it yourself from a host on the network.
- It can be provided with an appliance.

In production environments, the DHCP service would most likely already be configured. In test environments, you either setup your own server, or use a one that comes in a box. For the sake of example, we'll concentrate setting up our own DHCP server.

DHCP Configuration Example

Assumptions about the environment:

- The hosts have known addresses on the public network.
- DHCP provides one address per node plus three for the SCAN.
- The subnet and netmask on the interface to be serviced is `10.228.212.0/255.255.252.0`.
- The address range the DHCP server will serve is `10.228.212.10` through `10.228.215.254`.
- The gateway address is `10.228.212.1`.
- The name server is known to the cluster nodes.
- The domain your hosts are in is `us.example.com`.
- You have root access.
- You have the RPM for the DHCP server if it is not already installed.

DHCP Configuration Example

When we use DHCP for the public network, we will need two addresses per host (host address and VIP), plus three for cluster wide SCAN. The GNS VIP can not be obtained from DHCP, because it must be known in advance, so must be statically assigned.

For this example, let's make the following assumptions:

- The hosts have known addresses on the public network accessible by their hostname so they may be reached when Clusterware is not running.
- DHCP must provide four addresses, one per node plus three for the SCAN.
- You have a machine to use as your DHCP server. It is a single-point of failure, so it must always be up.
- The subnet and netmask on the interface to be serviced is `10.228.212.0/255.255.252.0`.
- The address range the DHCP server will serve are `10.228.212.10` through `10.228.215.254`.
- The gateway address is `10.228.212.1`.
- We have two name servers; M.N.P.Q and W.X.Y.Z.
- The domain your hosts are in for DNS search path purposes is `us.example.com`.
- You have root access.
- You have the RPM for the DHCP server if it is not already installed.

DHCP Configuration Example

The `/etc/dhcp.conf` file:

```
subnet 10.228.212.0 netmask 255.255.252.0
{
default-lease-time 43200;
max-lease-time 86400;
option subnet-mask 255.255.252.0;
option broadcast-address 10.228.215.255;
option routers 10.228.212.1;
option domain-name-servers  M.N.P.Q, W.X.Y.Z;
option domain-name "us.example.com";
Pool {
range 10.228.212.10 10.228.215.254;
  }
}
```

ORACLE

DHCP Configuration Example (Continued)

To install and configure DHCP, follow the steps below:

1. As the root user, install the DHCP rpm: `# rpm –Ivh dhcp-3.0.1-62.EL4.rpm`
2. The DHCP configuration file is `/etc/dhcp.conf`. Given our example, the minimal configuration for the public network will look something like this:

```
subnet 10.228.212.0 netmask 255.255.252.0
{
default-lease-time 43200;
max-lease-time 86400;
option subnet-mask 255.255.252.0;
option broadcast-address 10.228.215.255;
option routers 10.228.212.1;
option domain-name-servers  M.N.P.Q, W.X.Y.Z;
option domain-name "us.example.com";
Pool {
range 10.228.212.10 10.228.215.254;
} }
```

3. Start the DHCP service: `# /etc/init.d/dhcp start`

If you encounter any issues, check `/var/log/messages` for errors. You can adjust the lease time to suit your needs within the subnet.

DNS Concepts

- Host name resolution uses the `gethostbyname` family of library calls.
- These calls do a configurable search of name space providers, typically including:
  - Local `/etc/hosts` file
  - DNS service
  - Other directory services like NIS or LDAP
- DNS traffic is sent as UDP packets, usually to port 53.
- A query may ask for particular types of record, or all records for a name.
- Name resolution for Ipv4 uses `A` records for addresses, and `CNAME` records for aliases that must be re-resolved.

ORACLE

DNS Concepts

Host name resolution uses the `gethostbyname` family of library calls. These calls go through a configurable search path of name space providers, typically including a local `/etc/hosts` file, DNS service, and possibly other directories, such as NIS and/or LDAP. On Linux, these are usually defined in `/etc/nsswitch.conf`. For example, the line: `hosts: files dns nis` says to look in `/etc/hosts`, then consult DNS, then NIS.

When doing a lookup for a non-qualified name, the library will look for the unadorned name in all the name space providers above. If no answer is found, it will then successively apply domains from the search entry in the file `/etc/resolv.conf`. This also defines the DNS servers consulted when there is a `dns` entry in `/etc/nsswitch.conf` , for example:

```
search us.example.com example.com
nameserver M.N.P.Q
nameserver W.X.Y.Z
```

Usually the machine's domain is the first entry in the search path, followed by others of common use. DNS messages are sent as UDP packets, usually to the reserved port 53. A query may ask for particular types of record, or all records for a name. Name resolution for Ipv4 uses `A` records for addresses, and `CNAME` records for aliases (canonical names) that must be re-resolved. For IPv6, addresses are four times as large, and an `AAAA` record is used.

DNS Concepts (Continued)

`PTR` records are used for reverse lookups (ask for the name given an address).

`SRV` records are used to define service offerings, providing both address and port, and are used in the DNS-SD protocol we use for multicast discovery. `TXT` records contain arbitrary information, and are used in DNS-SD to carry attributes of a service.

DNS Forwarding For GNS

- To work properly, GNS needs to receive queries for all names in a sub-domain, from anywhere in the corporation.
- This is done by having the corporate DNS delegate the sub-domain to the cluster.
- DNS must know the full name of the sub-domain and the address where GNS will listen for queries.
- This clause from the `/etc/named.conf` file configures zone delegation for `cluster01.us.example.com`.

```
zone cluster01.us.example.com{
   type forward;
   forward only;
   forwarders {
      10.228.212.2 port 53;
   };
};
```


DNS Forwarding For GNS

For GNS to function properly, it needs to receive queries for all names in a sub-domain (zone), from anywhere in the corporation. This is done by having the corporate DNS delegate the sub-domain to the cluster. The two key pieces of data needed are the full name of the sub-domain being created and the address where GNS will listen for queries.

We typically use the clustername as the sub-domain. The GNS address must be statically assigned and on the public subnet of the cluster. It is virtual, and will be moved from host to host within the cluster to ensure the presence of the GNS service.

When using the common ISC BIND V8 name server, the network administrator sets up the zone delegation with an entry in the server configuration that looks like this:

```
zone cluster01.us.example.com{
   type forward;
   forward only;
   forwarders {
      10.228.212.2 port 53;
   };
};
```

DNS Forwarding For GNS (Continued)

Here, the sub-domain is cluster01.us.example.com. When the us.example.com DNS gets any query for anything under cluster01.us.example.com, it will send it to the server at that address and port, and return any results received. It is also likely to cache the returned result for the time-to-live (TTL) in the returned answer.

This does not establish any address for the name cluster01.us.example.com. Rather, it creates a way of resolving anything underneath, such as prod.cluster01.us.example.com.

In production sites, it is necessary to work with the network administrators to change the configuration of the DNS system, wherever it may reside.

DNS Configuration: Example

- We will make the following assumptions about our environment:
  - The cluster sub-domain is `cluster01.us.example.com`.
  - The address GNS will listen on is 10.228.212.2 port 53
  - The address for our new DNS server is 10.228.212.3.
  - The parent name servers are *M.N.P.Q* and *W.X.Y.Z*
- A summary of the steps is shown below:

1. **As root install the BIND (DNS) rpm:** `# rpm -Ivh bind-9.2.4-30.rpm`
2. **Configure delegation in** `/etc/named.conf`
3. **Populate the cache file:** `$ dig . ns > /var/named/db.cache`
4. **Populate** `/var/named/db.127.0.0` **to handle reverse lookups.**
5. **Start the name service (named):** `# /etc/init.d/named start`
6. **Modify** `/etc/resolv.conf` **on all nodes.**

DNS Configuration: Example

If you would like to set up something for testing, you can configure your own DNS service. Let's go through the exercise of configuring a local nameserver that all the other machines use for all addresses. Queries for the GNS domain will be sent to the GNS service, and all others will be sent up to the corporate DNS. This server will cache all intermediate results, so it is configured almost identically to a caching-only DNS server.

We will require the full name of the sub-domain being created and the address where GNS will listen for queries. Our GNS address will be 10.228.212.2. Other requirements are listed below.

- You must have a machine to use as your DHCP server. Remember, it is a single-point of failure so it must always be up. The address for our new DNS server is 10.228.212.3. Do not confuse this with our GNS address.
- The parent name servers are *M.N.P.Q* and *W.X.Y.Z*
- You have access to the root account.
- You have the RPM for the DNS server.

A summary of the steps required to install and configure the name server are listed below.

1. As root install the BIND (DNS) rpm:
   `# rpm -Ivh bind-9.2.4-30.rpm`
2. Configure delegation in `/etc/named.conf`

DNS Configuration: Example (Continued)

3. Populate the cache file:
   `$ dig . ns > /var/named/db.cache`
4. Populate `/var/named/db.127.0.0` to handle reverse lookups.
5. Start the name service (`named`):
   `# /etc/init.d/named start`
6. Modify `/etc/resolv.conf` to correctly configure the name space search order on all nodes.

DNS Configuration: Detail

1. As the root user, install the BIND DNS RPM.

```
# rpm -Ivh bind-9.2.4-30.rpm    ## Install the current RPM ##
```

2. Configure the behavior of the DNS server, named by editing the `/etc/named.conf` file. Define the:
   - Working directory
   - The nameserver cache file
   - Reverse lookup configuration
   - Delegation for your cluster sub-domain

DNS Configuration: Detail

1. The first action you will take is to install the most current BIND DNS rpm for your kernel version. This should be done as the root user. For example:
   `# rpm -Ivh bind-9.2.4-30.rpm`
2. Next, configure the behavior of the DNS server, `named` by editing the `/etc/named.conf` file. You will define the working directory, the root nameserver cache file, reverse lookup configuration, and delegation for your cluster sub-domain.

```
# vi /etc/named.conf

options {
 directory "/var/named";  # The named working directory (default value) #
 forwarders { M.N.P.Q; W.X.Y.Z; };  ## Where to resolve unknown addresses.
 forward only;      These are the same corporate nameservers used in the DHCP example ##
 };

 zone "." in {
 type hint;
 file "db.cache";   ## Defines the cache for root nameservers as /var/named/db.cache ##
 };
```

DNS Configuration: Detail (Continued)

```
zone "0.0.127.in-addr.arpa" in {
 type master;
 file "db.127.0.0"; ## localhost reverse lookup file ##
 };

zone cluster01.us.example.com{
type forward;
forward only;    ## This section defines the cluster GNS IP address to which requests for address
forwarders {        resolution for the cluster sub-domain cluster01.us.example.com are sent ##
10.228.212.2 port 53;
  };
};
```

DNS Configuration: Detail

3. Populate the cache file `/var/named/db.cache`.

```
# dig . ns > /var/named/db.cache
```

4. Populate `/var/named/db.127.0.0`.

```
$TTL 345600
@IN   SOA   localhost. root.localhost. (
00          ; Serial
86400       ; Refresh
7200        ; Retry
2592000     ; Expire
345600 )    ; Minimum
IN   NS     localhost.
1   IN   PTR   localhost.
```

3. Start the named server

```
# /etc/init.d/named start
```


DNS Configuration: Detail (Continued)

3. Populate the cache file `/var/named/db.cache`

   The zone "`.`" section of `named.conf` establishes root name servers. We need to populate this with data, and can do so with a simple lookup:

```
# dig . ns > /var/named/db.cache
```

   The output of this command looks something like this:

```
[root@host01 ~]# dig ns .
...
;.                          IN      NS
;; ANSWER SECTION:
.                   288     IN      NS      a.root-servers.net.
.                   288     IN      NS      b.root-servers.net.
.                   288     IN      NS      c.root-servers.net.
.                   288     IN      NS      d.root-servers.net.
.                   288     IN      NS      e.root-servers.net.
...
```

4.
```
;; ADDITIONAL SECTION:
a.root-servers.net. 459     IN      A       198.41.0.4
a.root-servers.net. 459     IN      AAAA    2001:503:ba3e::2:30
b.root-servers.net. 459     IN      A       192.228.79.201
c.root-servers.net. 459     IN      A       192.33.4.12
...
```

DNS Configuration: Detail (Continued)

4. Populate `/var/named/db.127.0.0`
   The “`zone 0.0.127.in-addr.arpa`” section of `named.conf` handles reverse lookups for the localhost. We need to create the `/var/named/db.127.0.0` data file for the 127.0.0.0 domain, which should contain:

```
 $TTL 345600
 @  IN  SOA  localhost. root.localhost. (
 00       ; Serial
 86400    ; Refresh
 7200     ; Retry
 2592000   ; Expire
 345600 )  ; Minimum
 IN  NS    localhost.
1  IN  PTR  localhost.
```

5. Start the Name server `named`.

```
# /etc/init.d/named start
```

Summary

In this lesson, you should have learned how to:

- Configure or communicate DHCP configuration needs in support of GNS.
- Configure or communicate DNS configuration needs in support of GNS.

ORACLE

High Availability of Connections

Objectives

After completing this lesson, you should be able to:

- Configure client-side, connect-time load balancing
- Configure client-side, connect-time failover
- Configure server-side, connect-time load balancing
- Use the Load Balancing Advisory (LBA)
- Describe the benefits of Fast Application Notification (FAN)
- Configure server-side callouts
- Configure Transparent Application Failover (TAF)

For more information see http://www.oracle.com/technology/products/database/clustering/pdf/awmrac11g.pdf

Note: Much of this appendix is geared toward Oracle Database 11*g* Release 1 connections.

Types of Workload Distribution

- Connection balancing is rendered possible by configuring multiple listeners on multiple nodes:
  - Client-side, connect-time load balancing
  - Client-side, connect-time failover
  - Server-side, connect-time load balancing
- Run-time connection load balancing is rendered possible by using connection pools:
  - Work requests automatically balanced across the pool of connections
  - Native feature of the JDBC implicit connection cache and ODP.NET connection pool

Types of Workload Distribution

With RAC, multiple listeners on multiple nodes can be configured to handle client connection requests for the same database service.

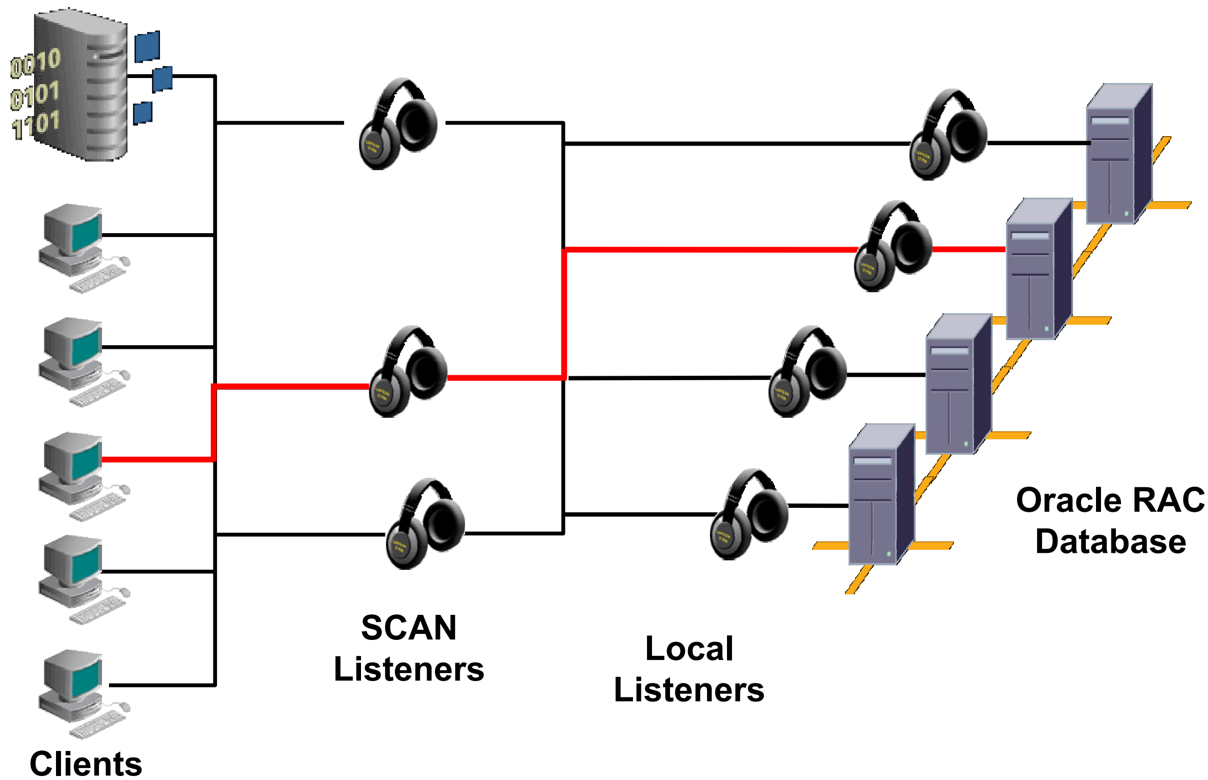
A multiple-listener configuration enables you to leverage the following failover and load-balancing features:

- Client-side, connect-time load balancing
- Client-side, connect-time failover
- Server-side, connect-time load balancing

These features can be implemented either one by one, or in combination with each other.

Moreover, if you are using connection pools, you can benefit from readily available run-time connection load balancing to distribute the client work requests across the pool of connections established by the middle tier. This possibility is offered by the Oracle JDBC implicit connection cache feature as well as Oracle Data Provider for .NET (`ODP.NET`) connection pool.

Client-Side Load Balancing



Client-Side, Connect-Time Load Balancing

Client-side load balancing is defined in your client connection definition by setting the parameter `LOAD_BALANCE=ON`. When you set this parameter to ON, Oracle Database randomly selects an address in the address list, and connects to that node's listener. This balances client connections across the available SCAN listeners in the cluster.

The SCAN listener redirects the connection request to the local listener of the instance that is least loaded and provides the requested service. When the listener receives the connection request, the listener connects the user to an instance that the listener knows provides the requested service. When using SCAN, Oracle Net automatically load balances client connection requests across the three IP addresses you defined for the SCAN, except when using EZConnect. To see what services a listener supports, run the `lsnrctl services` command.

Other Client-Side Connection Features

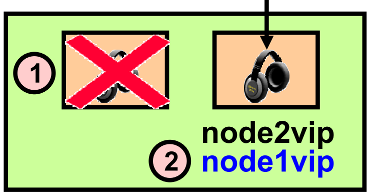
With the current release, we now have the ability to add the `connect_timeout` and `retry_count` parameters to individual tnsnames.ora connection strings.

```
(CONNECT_TIMEOUT=10)(RETRY_COUNT=3)
```

The granularity is seconds. Oracle Net waits for 10 seconds to receive a response, after which it assumes a failure. Oracle Net goes through the address list three times before it returns a failure to the client.

Client-Side, Connect-Time Failover

```
ERP =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (LOAD_BALANCE=ON)
      (FAILOVER=ON)
      (ADDRESS=(PROTOCOL=TCP)(HOST=node1vip)(PORT=1521))
      (ADDRESS=(PROTOCOL=TCP)(HOST=node2vip)(PORT=1521))
    )
    (CONNECT_DATA=(SERVICE_NAME=ERP)))
```



Client-Side, Connect-Time Failover

This feature enables clients to connect to another listener if the initial connection to the first listener fails. If an error is returned from the chosen address in the list, Oracle Net Services tries the next address in the list until it is either successful or it has exhausted all addresses in its list. For SCAN, Oracle Net Services tries all three addresses before returning a failure to the client. EZConnect with SCAN includes this connection failover feature.

To increase availability, you can specify a timeout that specifies how long Oracle Net waits for a response from the listener before returning an error. The method of setting this timeout parameter depends on the type of client access. As shown above, client-side, connect-time failover is enabled by setting `FAILOVER=ON` in the corresponding client-side TNS entry.

In the example, the client will randomly attempt connections to either `NODE1VIP` or `NODE2VIP`, because `LOAD_BALANCE` is set to `ON`. In the case where one of the nodes is down, the client cannot know this. If a connection attempt is made to a down node, the client must wait until notification is received that the node is not accessible before an alternate address in the `ADDRESS_LIST` is tried.

Using virtual host names in the `ADDRESS_LIST` of your connect descriptors is recommended. If a failure of a node occurs (1), the virtual IP address assigned to that node is failed over and brought online on another node in the cluster (2). Thus, all client connection attempts are still able to get a response from the IP address, without the need to wait for the operating system TCP/IP timeout (3). Therefore, clients get an immediate acknowledgement from the IP address, and are notified that the service on that node is not available.

Client-Side, Connect-Time Failover (Contnued)

The next address in the ADDRESS_LIST can then be tried immediately with no delay (4).

Note: If you use connect-time failover, do not set GLOBAL_DBNAME in your listener.ora file.

Server-Side Load Balancing

- Server-side load balancing:
  - Causes the listener to direct connection requests to the best instance currently providing the service
  - Uses connection information from the LBA
- When DBCA is used to create a RAC database:
  - Server-side load balancing is configured and enabled.
  - The `REMOTE_LISTENER` parameter is set to the SCAN listener.
- If DBCA is not used or listener ports other than 1521 are used, `LOCAL_LISTENER` and `REMOTE_LISTENER` parameters should point to *`scan_name:scan_port`*
- The LBA depends on an accurate configuration that includes setting the `CLB_GOAL` for the service

ORACLE

Server-Side Load Balancing

With server-side load balancing, the listener directs a connection request to the best instance currently providing the service by using information from the Load Balancing Advisory. When you create a RAC database with DBCA, it automatically configures and enables server-side load balancing. It also sets the remote listener parameter to the SCAN listener and creates a sample client-side load balancing connection definition in the `tnsnames.ora` file on the server. If you did not use DBCA, or if you are using listener ports other than the default of 1521, then you must configure the `LOCAL_LISTENER` and `REMOTE_LISTENER` database initialization parameters for your cluster database to point to *`scan_name:scan_port`*.

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

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

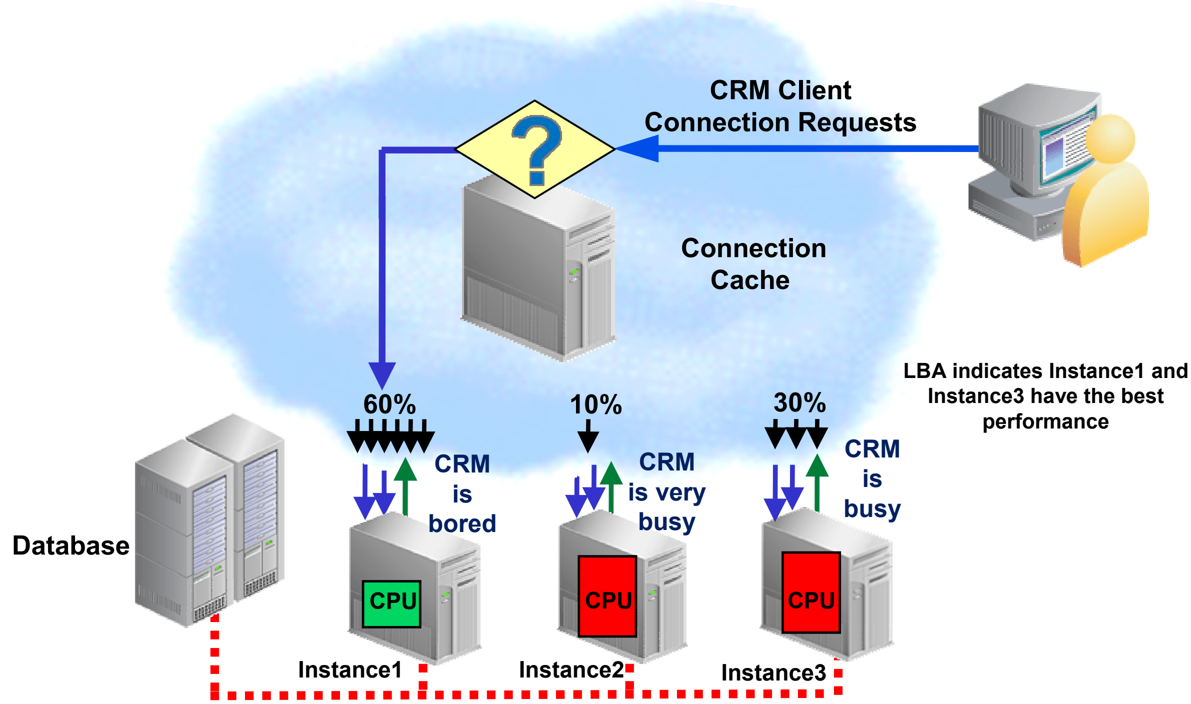
```
srvctl modify service -d db_unique_name -s BATCH -j LONG
```

Server-Side Load Balancing (Continued)

Use the `SHORT` connection load balancing method for applications that have short-lived connections. When using connection pools that are integrated with FAN, set the `CLB_GOAL` to `SHORT`. The following example modifies the service known as OLTP, using SRVTCL to set the connection load balancing goal to `SHORT`:

```
srvctl modify service -d db_unique_name -s OLTP -j SHORT
```

Runtime Connection Load Balancing and Connection Pools



Runtime Connection Load Balancing and Connection Pools

Runtime Connection Load Balancing is a feature of Oracle connection pools that can distribute client work requests across the instances in an Oracle RAC database based on the Load Balancing Advisory information. The connection allocation is based on the current performance level provided by the database instances as indicated by the Load Balancing Advisory FAN events. This provides load balancing at the transaction level, instead of load balancing at the time of the initial database connection.

With Runtime Connection Load Balancing, applications use LBA information to provide better performance to users. OCI Session pools and ODP.NET connection pools support Runtime Connection Load Balancing. For Java applications, Oracle recommends the Universal Connection Pool (UCP). The UCP is integrated to take advantage of LBA information.

You must enable the client data source for Runtime Connection Load Balancing with a service that has the following configuration:

- The Load Balancing Advisory is enabled and the service-level goal is set to either Service Time or Throughput.
- The service connection load balancing goal is set to Short.

The figure above illustrates Runtime Connection Load Balancing. In this illustration, the Oracle RAC database has three instances.

Runtime Connection Load Balancing and Connection Pools (Continued)

Suppose that the Load Balancing Advisory indicates that Instance1 and Instance3 have the best performance, while Instance2 currently has less than optimal performance. When Runtime Connection Load Balancing is enabled on the implicit connection cache, the following process occurs:

1. A client requests a connection from the connection pool.

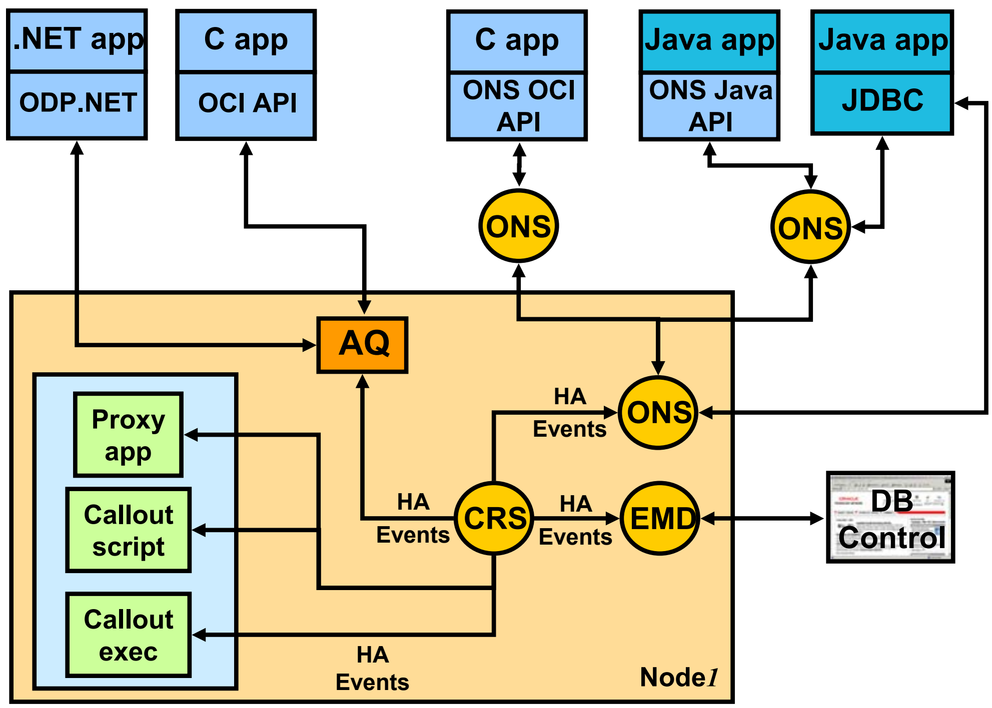
2. Runtime Connection Load Balancing selects the connection that belongs to the most efficient (best) instance from the connection pool. In the example, there are three possible nodes to which the connection can be routed. Instance1, which has the least amount of CPU workload, is currently being assigned about 60 percent of the incoming connections. Instance2, which is currently overloaded, is only being assigned around 10 percent of the incoming connections. Instance3, which has a high workload, is being assigned around 30 percent of the incoming connections. The best instance to handle the connection request in this case would be Instance1.

3. The client receives the connection that would process the work request with the best response time.

Oracle Database 11g introduces an additional flag in the load balancing advisory event called affinity hint. The affinity hint is automatic when load balancing advisory is turned on when setting the goal on the service. This flag is for temporary affinity that lasts for the duration of a web session. Web conversations often connect and disconnect many times during the entire session. During each of these connects, it may access the same or similar data, for example, a shopping cart, Siebel, and so on. Affinity can improve buffer cache efficiency, which lowers CPU usage and transaction latency. The Affinity hint is a flag that indicates if Affinity is active or inactive for a particular instance and service combination. Different instances offering the same service can have different settings for the Affinity hint.

Applications using Oracle Database 11g and UCP can take advantage of this new affinity feature. If the affinity flag is turned on in the Load Balancing Advisory event, then UCP creates an affinity context for the web session such that when that session does a get connection from the pool, the pool always tries to give it a connection to the instance it connected to the first time it acquired a session. The choice of instance for the first connection is based on the current load balancing advisory information.

Fast Application Notification: Overview



Fast Application Notification: Overview

Fast Application Notification (FAN) enables end-to-end, lights-out recovery of applications and load balancing based on real transaction performance in a RAC environment. With FAN, the continuous service built into Oracle Real Application Clusters is extended to applications and mid-tier servers. When the state of a database service changes, (for example, up, down, or not restarting), the new status is posted to interested subscribers through FAN events. Applications use these events to achieve very fast detection of failures, and rebalancing of connection pools following failures and recovery. The easiest way to receive all the benefits of FAN, with no effort, is to use a client that is integrated with FAN:

- Oracle Database JDBC
- Server-side callouts
- Oracle Notification Service (ONS) API
- Oracle Universal Connection Pool (UCP) for Java
- OCI Connection Pool or Session Pool
- Transparent Application Failover (TAF)
- ODP.NET Connection Pool

Note: The integrated Oracle clients must be Oracle Database 10*g* Release 2 or later to take advantage of the load balancing advisory FAN events.

Fast Application Notification: Benefits

- No need for connections to rely on connection timeouts
- Used by Load Balancing Advisory to propagate load information
- Designed for enterprise application and management console integration
- Reliable distributed system that:
  - Detects high-availability event occurrences in a timely manner
  - Pushes notification directly to your applications
- Tightly integrated with:
  - Oracle JDBC applications using connection pools
  - Enterprise Manager
  - Data Guard Broker

Fast Application Notification: Benefits

Traditionally, client or mid-tier applications connected to the database have relied on connection timeouts, out-of-band polling mechanisms, or other custom solutions to realize that a system component has failed. This approach has huge implications in application availability, because down times are extended and more noticeable.

With FAN, important high-availability events are pushed as soon as they are detected, which results in a more efficient use of existing computing resources, and a better integration with your enterprise applications, including mid-tier connection managers, or IT management consoles, including trouble ticket loggers and email/paging servers.

FAN is a distributed system that is enabled on each participating node. This makes it very reliable and fault tolerant because the failure of one component is detected by another. Therefore, event notification can be detected and pushed by any of the participating nodes.

FAN events are tightly integrated with Oracle Data Guard Broker, Oracle UCP, ODP.NET, TAF, and Enterprise Manager. For example, Oracle Database JDBC applications managing connection pools do not need custom code development. They are automatically integrated with the ONS if implicit connection cache and fast connection failover are enabled.

FAN Events

Event type	Description
SERVICE	Primary application service
SRV_PRECONNECT	Shadow application service event (mid-tiers and TAF using primary and secondary instances)
SERVICEMEMBER	Application service on a specific instance
DATABASE	Oracle database
INSTANCE	Oracle instance
ASM	Oracle ASM instance
NODE	Oracle cluster node
SERVICE_METRICS	Load Balancing Advisory

FAN Events

Before using some of the FAN features such as Server Side Callouts, you should understand FAN events. Oracle RAC FAN events consist of header and detail information delivered as a set of name-value pairs accurately describing the name, type and nature of the event. Based on this information, the event recipient can take concrete management, notification, or synchronization steps, such as shutting down the application connection manager, rerouting existing database connection requests, refreshing stale connection references, logging a trouble ticket, or sending a page to the database administrator.

FAN events are system events, sent during periods when cluster nodes may become unreachable and network interfaces slow or non-functional. There is an inherent reliance on minimal communication channel overhead to send, queue and receive notifications quickly. The objective is to deliver FAN events so that they precede regular connection timeouts or typical polling intervals. Three categories of events are supported in Oracle RAC FAN:

- Service events, which includes both application services and database services
- Node events, which includes cluster membership states and native join/leave operations
- Load Balancing Events sent from the RAC Load Balancing Advisory

RAC standardizes the generation, presentation, and delivery of events pertaining to managed cluster resources..

FAN Event Status

Event status	Description
`up`	Managed resource comes up.
`down`	Managed resource goes down.
`preconn_up`	Shadow application service comes up.
`preconn_down`	Shadow application service goes down.
`nodedown`	Managed node goes down.
`not_restarting`	Managed resource cannot fail over to a remote node.
`unknown`	Status of managed resource is unknown

ORACLE

FAN Event Status

This table describes the event status for each of the managed cluster resources seen previously.

FAN Event Reasons

Event Reason	Description
`user`	User-initiated commands, such as `srvctl` and `sqlplus`
`failure`	Managed resource polling checks for and detects a failure.
`dependency`	Dependency of another managed resource that triggered a failure condition
`autostart`	Initial cluster boot: Managed resource has profile attribute `AUTO_START=1`, and was offline before the last Oracle Clusterware shutdown.
`public_nw_down`	Public network is down
`member_leave`	Member node has left the cluster.

ORACLE

FAN Event Reasons

The event status for each managed resource is associated with an event reason. The reason further describes what triggered the event. The table in the slide gives you the list of possible reasons with a corresponding description.

FAN Event Format

```
<Event_Type>
VERSION=<n.n>
[service=<serviceName.dbDomainName>]
[database=<dbName>] [instance=<sid>]
[host=<hostname>]
status=<Event_Status>
reason=<Event_Reason>
[card=<n>]
timestamp=<eventDate> <eventTime>
```

```
SERVICE VERSION=1.0 service=ERP.oracle.com
database=ORCL status=up reason=user card=4
timestamp=21-Jul-2009 15:24:19
```

```
NODE VERSION=1.0 host=host01
status=nodedown timestamp=21-Jul-2009 12:41:02
```


FAN Event Format

In addition to its type, status, and reason, a FAN event has other payload fields to further describe the unique cluster resource whose status is being monitored and published:

- The event payload version, which is currently 1.0
- The name of the primary or shadow application service. This name is excluded from `NODE` events.
- The name of the RAC database, which is also excluded from `NODE` events
- The name of the RAC instance, which is excluded from `SERVICE`, `DATABASE`, and `NODE` events
- The name of the cluster host machine, which is excluded from `SERVICE` and `DATABASE` events
- The service cardinality, which is excluded from all events except for `SERVICE status=up` events
- The server-side date and time when the event is detected

The general FAN event format is described in the slide along with possible FAN event examples. Note the differences in event payload for each FAN event type.

Load Balancing Advisory: FAN Event

Parameter	Description
Version	Version of the event record
Event type	**`SERVICE, SERVICE_MEMBER, DATABASE, INSTANCE, NODE, ASM, SRV_PRECONNECT`**
Service	Matches the service in **`DBA_SERVICES`**
Database unique name	Unique DB name supporting the service
Time stamp	Date and time stamp (local time zone)
Instance	Instance name supporting the service
Percent	Percentage of work to send to this database and instance
Flag	**`GOOD`, `VIOLATING`, `NO DATA`, `UNKNOWN`**

Load Balancing Advisory: FAN Event

The Load Balancing Advisory FAN event is described in the slide. Basically, it contains a calculated percentage of work requests that should be sent to each instance. The flag indicates the behavior of the service on the corresponding instance relating to the thresholds set on that instance for the service.

Use the following example to monitor load balancing advisory events:

```
SET PAGES 60 COLSEP '|' LINES 132 NUM 8 VERIFY OFF FEEDBACK OFF
COLUMN user_data HEADING "AQ Service Metrics" FORMAT A60 WRAP
BREAK ON service_name SKIP 1

SELECT TO_CHAR(enq_time, 'HH:MI:SS') Enq_time, user_data
FROM sys.sys$service_metrics_tab
ORDER BY 1 ;
```

Implementation of Server-Side Callouts

- The callout directory:
  - **<*GRID_Home*>/racg/usrco**
  - Can store more than one callout
  - Grants execution on callouts and the callout directory to the Oracle Clusterware user
- The order in which callouts are executed is nondeterministic.
- Writing callouts involves:
  1. Parsing callout arguments: The event payload
  2. Filtering incoming FAN events
  3. Executing event-handling programs

ORACLE

Implementation of Server-Side Callouts

Each database event detected by the RAC High Availability (HA) framework results in the execution of each executable script or program deployed in the standard Oracle Clusterware callout directory. On UNIX, it is `GRID_Home/racg/usrco`. You must deploy each new callout on each RAC node.

The order in which these callouts are executed is nondeterministic. However, RAC guarantees that all callouts are invoked once for each recognized event in an asynchronous fashion. Thus, merging callouts whose executions need to be in a particular order is recommended.

You can install as many callout scripts or programs as your business requires, provided each callout does not incur expensive operations that delay the propagation of HA events. If many callouts are going to be written to perform different operations based on the event received, it might be more efficient to write a single callout program that merges each single callout.

Writing server-side callouts involves the steps shown in the slide. In order for your callout to identify an event, it must parse the event payload sent by the RAC HA framework to your callout. After the sent event is identified, your callout can filter it to avoid execution on each event notification. Then, your callout needs to implement a corresponding event handler that depends on the event itself and the recovery process required by your business.

Note: As a security measure, make sure that the callout directory and its contained callouts have write permissions only to the system user who installed Oracle Clusterware.

Server-Side Callout Parse: Example

```
#!/bin/sh
NOTIFY_EVENTTYPE=$1
for ARGS in $*; do
   PROPERTY=`echo $ARGS | $AWK -F"=" '{print $1}'`
   VALUE=`echo $ARGS | $AWK -F"=" '{print $2}'`
   case $PROPERTY in
     VERSION|version)      NOTIFY_VERSION=$VALUE ;;
     SERVICE|service)      NOTIFY_SERVICE=$VALUE ;;
     DATABASE|database)    NOTIFY_DATABASE=$VALUE ;;
     INSTANCE|instance)    NOTIFY_INSTANCE=$VALUE ;;
     HOST|host)            NOTIFY_HOST=$VALUE ;;
     STATUS|status)        NOTIFY_STATUS=$VALUE ;;
     REASON|reason)        NOTIFY_REASON=$VALUE ;;
     CARD|card)            NOTIFY_CARDINALITY=$VALUE ;;
     TIMESTAMP|timestamp) NOTIFY_LOGDATE=$VALUE ;;
     ??:??:??)             NOTIFY_LOGTIME=$PROPERTY ;;
   esac
done
```

ORACLE

Server-Side Callout Parse: Example

Unless you want your callouts to be executed on each event notification, you must first identify the event parameters that are passed automatically to your callout during its execution. The example in the slide shows you how to parse these arguments by using a sample Bourne shell script.

The first argument that is passed to your callout is the type of event that is detected. Then, depending on the event type, a set of `PROPERTY=VALUE` strings are passed to identify exactly the event itself.

The script given in the slide identifies the event type and each pair of `PROPERTY=VALUE` string. The data is then dispatched into a set of variables that can be used later in the callout for filtering purposes.

As mentioned in the previous slide, it might be better to have a single callout that parses the event payload, and then executes a function or another program on the basis of information in the event, as opposed to having to filter information in each callout. This becomes necessary only if many callouts are required.

Note: Make sure that executable permissions are set correctly on the callout script.

Server-Side Callout Filter: Example

```
if ((( [ $NOTIFY_EVENTTYPE = "SERVICE"       ] ||
       [ $NOTIFY_EVENTTYPE = "DATABASE"      ] ||    \
       [ $NOTIFY_EVENTTYPE = "NODE"          ]       \
     ) &&                                            \
     ( [ $NOTIFY_STATUS = "not_restarting"   ]       \
     )) &&                                           \
     ( [ $NOTIFY_DATABASE = "PROD"         ] ||    \
       [ $NOTIFY_SERVICE  = "ERP"            ]       \
     ))
then
     /usr/local/bin/logTicket $NOTIFY_LOGDATE  \
                              $NOTIFY_LOGTIME  \
                              $NOTIFY_SERVICE  \
                              $NOTIFY_DBNAME   \
                              $NOTIFY_HOST
fi
```

ORACLE

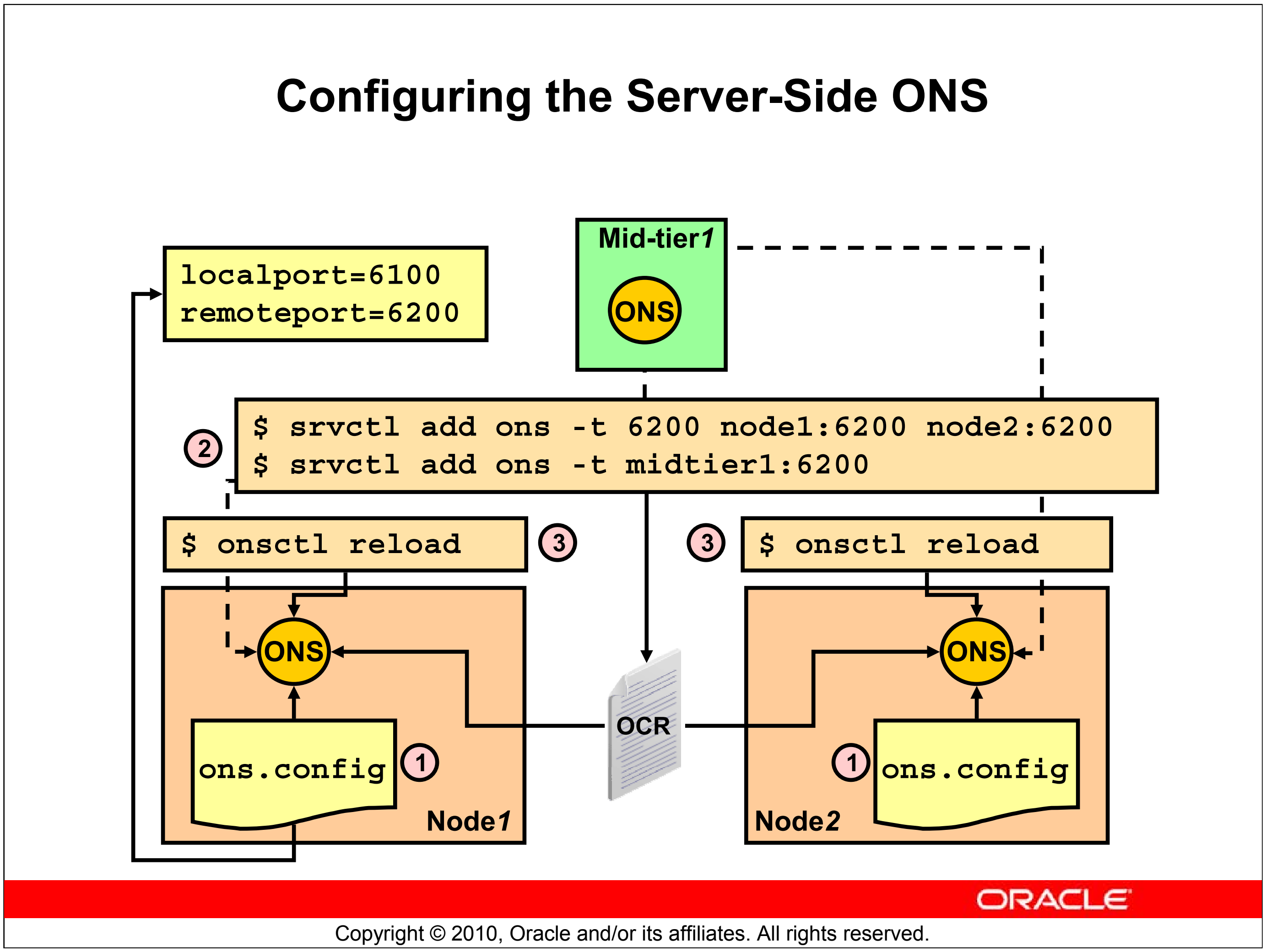
Server-Side Callout Filter: Example

The example in the slide shows you a way to filter FAN events from a callout script. This example is based on the example in the previous slide.

Now that the event characteristics are identified, this script triggers the execution of the trouble-logging program `/usr/local/bin/logTicket` only when the RAC HA framework posts a `SERVICE`, `DATABASE`, or `NODE` event type, with a status set to `not_restarting`, and only for the production `PROD` RAC database or the `ERP` service.

It is assumed that the `logTicket` program is already created and that it takes the arguments shown in the slide.

It is also assumed that a ticket is logged only for `not_restarting` events, because they are the ones that exceeded internally monitored timeouts and seriously need human intervention for full resolution.

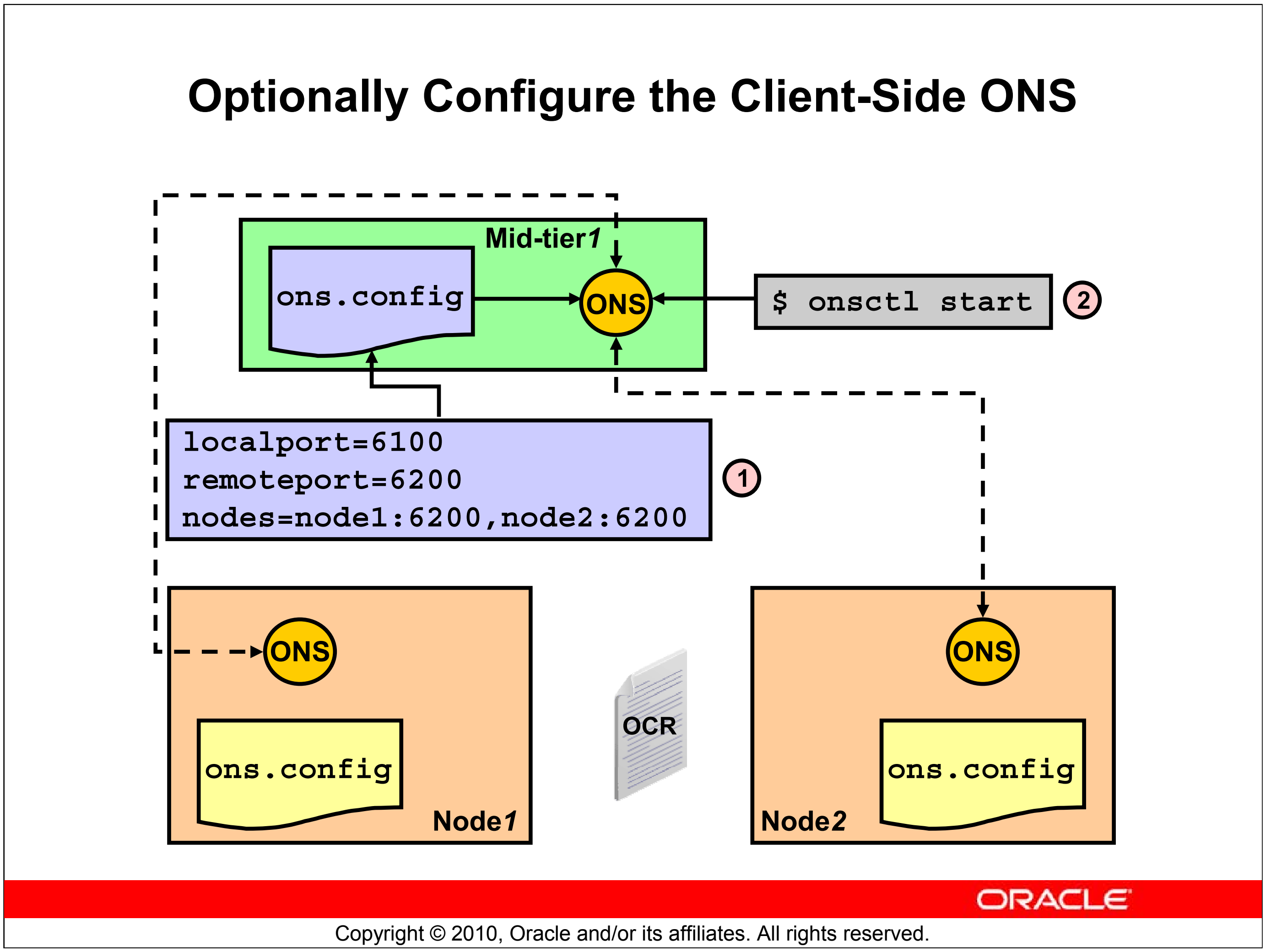


Configuring the Server-Side ONS

The ONS configuration is controlled by the `<GRID Home>/opmn/conf/ons.config` configuration file. This file is automatically created during installation. There are three important parameters that should always be configured for each ONS:

- The first is `localport`, the port that ONS uses to talk to local clients.
- The second is `remoteport`, the port that ONS uses to talk to other ONS daemons.
- The third parameter is called `nodes`. It specifies the list of other ONS daemons to talk to. This list should include all RAC ONS daemons, and all mid-tier ONS daemons. Node values are given as either host names or IP addresses followed by its `remoteport`.

In the slide, it is assumed that ONS daemons are already started on each cluster node. This should be the default situation after a correct RAC installation. However, if you want to use OCR, you should edit the `ons.config` file on each node, and then add the configuration to OCR before reloading it on each cluster node. This is illustrated in the slide.

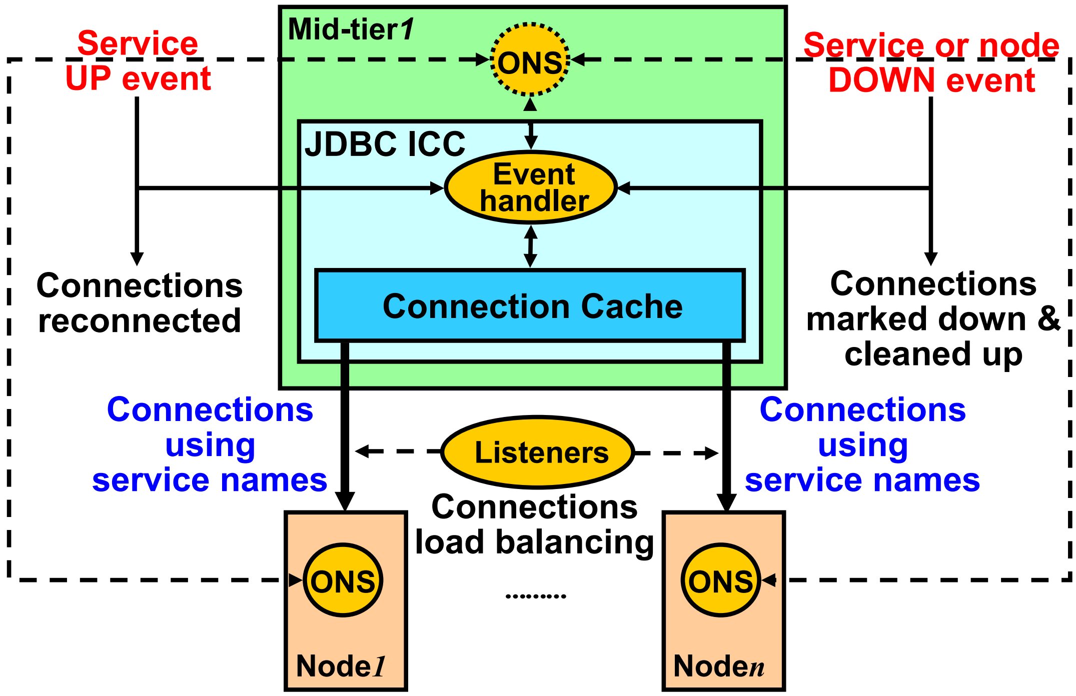


Optionally Configure the Client-Side ONS

Oracle Database FAN uses Oracle Notification Service (ONS) on the mid-tier to receive FAN events when you are using the Java Database Connectivity (JDBC) connection cache. To use ONS on the mid-tier, you need to install ONS on each host where you have client applications that need to be integrated with FAN. Most of the time, these hosts play the role of a mid-tier application server. Therefore, on the client side, you must configure all the RAC nodes in the ONS configuration file. A sample configuration file might look like the one shown in the slide.

After configuring ONS, you start the ONS daemon with the `onsctl start` command. It is your responsibility to make sure that an ONS daemon is running at all times. You can check that the ONS daemon is active by executing the `onsctl ping` command.

JDBC Fast Connection Failover: Overview



JDBC Fast Connection Failover: Overview

Oracle Application Server integrates Implicit Connection Cache (ICC) with the ONS API by having application developers enable Fast Connection Failover (FCF). FCF works in conjunction with the ICC to quickly and automatically recover lost or damaged connections. This automatic connection management results from FAN events received by the local ONS daemon, or by a remote ONS if a local one is not used, and handled by a special event handler thread. Both JDBC thin and JDBC OCI drivers are supported.

Therefore, if ICC and FCF are enabled, your Java program automatically becomes an ONS subscriber without having to manage FAN events directly.

Whenever a service or node down event is received by the mid-tier ONS, the event handler automatically marks the corresponding connections as down and cleans them up. This prevents applications that request connections from the cache from receiving invalid or bad connections.

Whenever a service up event is received by the mid-tier ONS, the event handler recycles some unused connections, and reconnects them using the event service name. The number of recycled connections is automatically determined by the connection cache. Because the listeners perform connection load balancing, this automatically rebalances the connections across the preferred instances of the service without waiting for application connection requests or retries.

For more information, refer to the *Oracle Database JDBC Developer's Guide and Reference*.

Note: Similarly, ODP.NET also allows you to use FCF using AQ for FAN notifications.

Using Oracle Streams Advanced Queuing for FAN

- Use AQ to publish FAN to ODP.NET and OCI.
- Turn on FAN notification to alert queue.

```
$ srvctl modify service -d crm -s gl.us.oracle.com -q TRUE
```

- View published FAN events using the `DBA_OUTSTANDING_ALERTS` or `DBA_ALERT_HISTORY` views

Using Oracle Streams Advanced Queuing for FAN

RAC publishes FAN events to a system alert queue in the database by using Oracle Streams Advanced Queuing (AQ). ODP.NET and OCI client integration uses this method to subscribe to FAN events.

To have FAN events for a service posted to that alert queue, the notification must be turned on for the service. You can do this using the Enterprise Manager interface.

To view FAN events that are published, you can use the `DBA_OUTSTANDING_ALERTS` or `DBA_ALERT_HISTORY` views.

JDBC/ODP.NET FCF Benefits

- Database connections are balanced across preferred instances according to LBA.
- Database work requests are balanced across preferred instances according to LBA.
- Database connections are anticipated.
- Database connection failures are immediately detected and stopped.

JDBC/ODP.NET FCF Benefits

By enabling FCF, your existing Java applications connecting through Oracle JDBC Universal Connection Pool (UCP) and application services, or your .NET applications using ODP.NET connection pools and application services benefit from the following:

- All database connections are balanced across all RAC instances that support the new service name, instead of having the first batch of sessions routed to the first RAC instance. This is done according to the Load Balancing Advisory algorithm you use (see the next slide). Connection pools are rebalanced upon service, instance, or node up events.
- The connection cache immediately starts placing connections to a particular RAC instance when a new service is started on that instance.
- The connection cache immediately shuts down stale connections to RAC instances where the service is stopped, or whose node goes down.
- Your application automatically becomes a FAN subscriber without having to manage FAN events directly by just setting up flags in your connection descriptors..

Note: For more information about how to subscribe to FAN events, refer to the *Oracle Database JDBC Developer's Guide*.

Load Balancing Advisory

- The Load Balancing Advisory (LBA) is an advisory for sending work across RAC instances.
- LBA advice is available to all applications that send work:
  – JDBC and ODP connection pools
  – Connection load balancing
- There are two types of service-level goals:
  – **SERVICE_TIME:** Directs work requests to instances according to response time.

```
$ srvctl modify service -d PROD -s OE -B SERVICE_TIME -j SHORT
```

  – **THROUGHPUT:** Directs requests based on the rate that work is completed in the service plus available bandwidth.

```
$ srvctl modify service -d PROD -s BATCH -B THROUGHPUT -j LONG
```

ORACLE

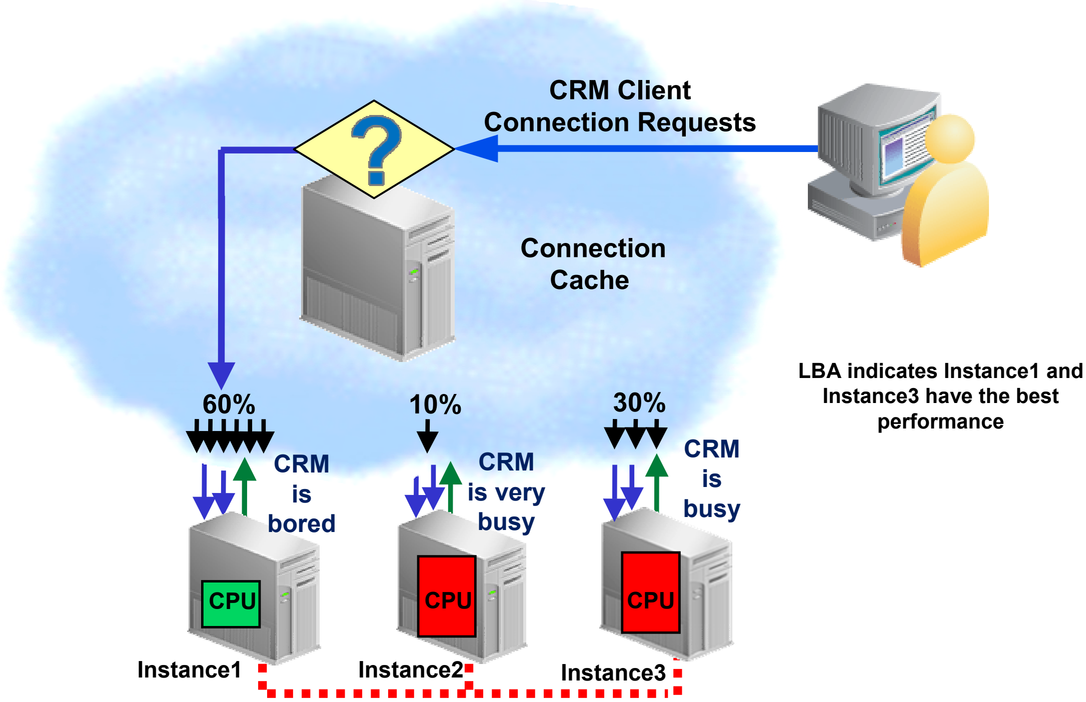
Load Balancing Advisory

Load balancing distributes work across all available database instances. The LBA provides advice about how to direct incoming work to the instances that provide the optimal quality of service for that work, minimizing the need to relocate the work later. By using the SERVICE_TIME or THROUGHPUT goals, feedback is built into the system.

- SERVICE_TIME: Attempts to direct work requests to instances according to response time. Load balancing advisory data is based on elapsed time for work done in the service plus available bandwidth to the service. An example for the use of SERVICE_TIME is for workloads such as Internet shopping where the rate of demand changes
- THROUGHPUT: Attempts to direct work requests according to throughput. The load balancing advisory is based on the rate that work is completed in the service plus available bandwidth to the service. An example for the use of THROUGHPUT is for workloads such as batch processes, where the next job starts when the last job completes:

Work is routed to provide the best service times globally, and routing responds gracefully to changing system conditions. In a steady state, the system approaches equilibrium with improved throughput across all of the Oracle RAC instances. The load balancing advisory is deployed with key Oracle clients, such as a listener, the JDBC universal connection pool, and the ODP.NET Connection Pool. The load balancing advisory is also open for third-party subscription by way of the JDBC and Oracle RAC FAN API or through callbacks with the Oracle Call Interface.

Runtime Connection Load Balancing and Connection Pools



Runtime Connection Load Balancing and Connection Pools

Runtime Connection Load Balancing is a feature of Oracle connection pools that can distribute client work requests across the instances in an Oracle RAC database based on the Load Balancing Advisory information. The connection allocation is based on the current performance level provided by the database instances as indicated by the Load Balancing Advisory FAN events. This provides load balancing at the transaction level, instead of load balancing at the time of the initial database connection.

With Runtime Connection Load Balancing, applications use LBA information to provide better performance to users. OCI Session pools and ODP.NET connection pools support Runtime Connection Load Balancing. For Java applications, Oracle recommends the Universal Connection Pool (UCP). The UCP is integrated to take advantage of LBA information.

You must enable the client data source for Runtime Connection Load Balancing with a service that has the following configuration:

- The Load Balancing Advisory is enabled and the service-level goal is set to either Service Time or Throughput.
- The service connection load balancing goal is set to Short.

The figure above illustrates Runtime Connection Load Balancing. In this illustration, the Oracle RAC database has three instances.

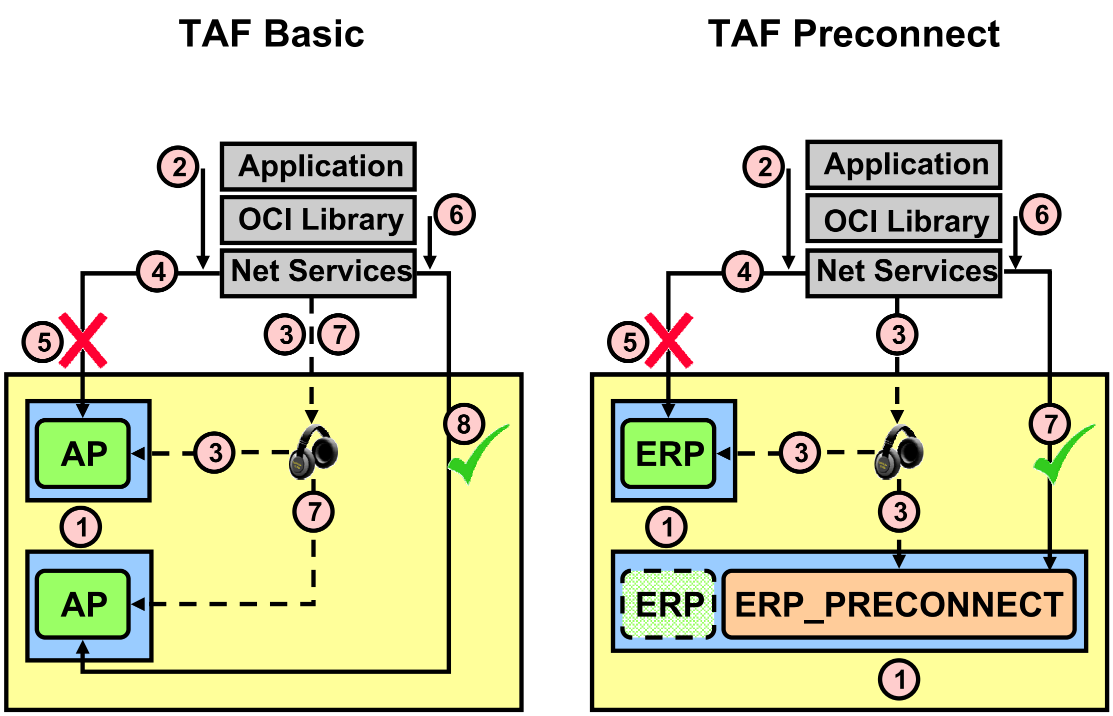
Monitor LBA FAN Events

```
SQL> SELECT TO_CHAR(enq_time, 'HH:MI:SS') Enq_time, user_data
  2  FROM sys.sys$service_metrics_tab
  3  ORDER BY 1 ;

ENQ_TIME USER_DATA
-------- -----------------------------------------------------
…
04:19:46 SYS$RLBTYP('JFSERV', 'VERSION=1.0 database=xwkE
         service=JFSERV { {instance=xwkE2 percent=50
         flag=UNKNOWN}{instance=xwkE1 percent=50 flag=UNKNOWN}
         } timestamp=2009-01-02 06:19:46')
04:20:16 SYS$RLBTYP('JFSERV', 'VERSION=1.0 database=xwkE
         service=JFSERV { {instance=xwkE2 percent=80
         flag=UNKNOWN}{instance=xwkE1 percent=20 flag=UNKNOWN}
         } timestamp=2009-01-02 06:20:16')
SQL>
```


Monitor LBA FAN Events

You can use the SQL query shown in the slide to monitor the Load Balancing Advisory FAN events for each of your services.

Transparent Application Failover: Overview



Transparent Application Failover (TAF): Overview

TAF is a run-time feature of the OCI driver. It enables your application to automatically reconnect to the service if the initial connection fails. During the reconnection, although your active transactions are rolled back, TAF can optionally resume the execution of a `SELECT` statement that was in progress. TAF supports two failover methods:

- With the `BASIC` method, the reconnection is established at failover time. After the service has been started on the nodes (1), the initial connection (2) is made. The listener establishes the connection (3), and your application accesses the database (4) until the connection fails (5) for any reason. Your application then receives an error the next time it tries to access the database (6). Then, the OCI driver reconnects to the same service (7), and the next time your application tries to access the database, it transparently uses the newly created connection (8). TAF can be enabled to receive FAN events for faster down events detection and failover.
- The `PRECONNECT` method is similar to the `BASIC` method except that it is during the initial connection that a shadow connection is also created to anticipate the failover. TAF guarantees that the shadow connection is always created on the available instances of your service by using an automatically created and maintained shadow service.

Note: Optionally, you can register TAF callbacks with the OCI layer. These callback functions are automatically invoked at failover detection and allow you to have some control of the failover process. For more information, refer to the *Oracle Call Interface Programmer's Guide*.

TAF Basic Configuration Without FAN: Example

```
$ srvctl add service -d RACDB -s AP -r I1,I2 \
> -P BASIC
$ srvctl start service -d RACDB -s AP
```

```
AP =
 (DESCRIPTION =(FAILOVER=ON)(LOAD_BALANCE=ON)
  (ADDRESS=(PROTOCOL=TCP)(HOST=cluster01-scan)(PORT=1521))
     (CONNECT_DATA =
      (SERVICE_NAME = AP)
      (FAILOVER_MODE =
        (TYPE=SELECT)
        (METHOD=BASIC)
        (RETRIES=180)
        (DELAY=5))))
```


TAF Basic Configuration Without FAN: Example

Before using TAF, it is recommended that you create and start a service that is used during connections. By doing so, you benefit from the integration of TAF and services. When you want to use `BASIC` TAF with a service, you should have the `-P BASIC` option when creating the service. After the service is created, you simply start it on your database.

Then, your application needs to connect to the service by using a connection descriptor similar to the one shown in the slide. The `FAILOVER_MODE` parameter must be included in the `CONNECT_DATA` section of your connection descriptor:

- `TYPE` specifies the type of failover. The `SELECT` value means that not only the user session is reauthenticated on the server side, but also the open cursors in the OCI can continue fetching. This implies that the client-side logic maintains the fetch-state of each open cursor. A `SELECT` statement is reexecuted by using the same snapshot, discarding those rows already fetched, and retrieving those rows that were not fetched initially. TAF verifies that the discarded rows are those that were returned initially, or it returns an error message.
- `METHOD=BASIC` is used to reconnect at failover time.
- `RETRIES` specifies the number of times to attempt to connect after a failover.
- `DELAY` specifies the amount of time in seconds to wait between connect attempts.

TAF Basic Configuration with FAN: Example

```
$ srvctl add service -d RACDB -s AP -r I1,I2
```

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

```
srvctl modify service -d RACDB -s AP -q TRUE -P BASIC \
-e SELECT -z 180 -w 5 -j LONG
```

```
AP =
 (DESCRIPTION =
  (ADDRESS=(PROTOCOL=TCP)(HOST=cluster01-scan)(PORT=1521))
    (LOAD_BALANCE = YES)
      (CONNECT_DATA =
       (SERVER = DEDICATED)
       (SERVICE_NAME = TEST)
       (FAILOVER_MODE =
   (TYPE = SELECT)(METHOD = BASIC)(RETRIES = 180)(DELAY = 5))))
```

ORACLE

TAF Basic Configuration with FAN: Example

Oracle Database 11*g* Release 2 supports server-side TAF with FAN. To use server-side TAF, create and start your service using `SRVCTL`, and then configure TAF in the RDBMS by using the `srvctl` command as shown in the slide. When done, make sure that you define a TNS entry for it in your `tnsnames.ora` file. Note that this TNS name does not need to specify TAF parameters as in the previous slide.

TAF Preconnect Configuration: Example

```
ERP =
 (DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = cluster01-scan)(PORT = 1521))
  (LOAD_BALANCE = YES)
  (CONNECT_DATA =
  (SERVER = DEDICATED)
  (SERVICE_NAME = ERP)
  (FAILOVER_MODE =
  (BACKUP = ERP_PRECONNECT)
  (TYPE = SELECT)(METHOD = PRECONNECT)(RETRIES = 180)(DELAY = 5))))

ERP_PRECONNECT =
  (DESCRIPTION =
   (ADDRESS = (PROTOCOL = TCP)(HOST = cluster01-scan)(PORT = 1521))
   (LOAD_BALANCE = YES)
   (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = ERP_PRECONNECT)
    (FAILOVER_MODE =
    (TYPE = SELECT)(METHOD = BASIC)(RETRIES = 180)(DELAY = 5)))
```


TAF Preconnect Configuration: Example

In order for the shadow service to be created and managed automatically by Oracle Clusterware, you must define the service with the `–P PRECONNECT` option. The shadow service is always named using the format `<service_name>_PRECONNECT`.

The main differences with the previous example are that `METHOD` is set to `PRECONNECT` and an additional parameter is added. This parameter is called `BACKUP` and must be set to another entry in your `tnsnames.ora` file that points to the shadow service.

Note: In all cases where TAF cannot use the `PRECONNECT` method, TAF falls back to the `BASIC` method automatically.

TAF Verification

```
SELECT   machine, failover_method, failover_type,
         failed_over, service_name, COUNT(*)
FROM     v$session
GROUP BY machine, failover_method, failover_type,
         failed_over, service_name;
```

First node

```
MACHINE FAILOVER_M FAILOVER_T FAI SERVICE_N COUNT(*)
------- ---------- ---------- --- --------  --------
node1   BASIC      SESSION    NO  AP        1
node1   PRECONNECT SESSION    NO  ERP       1
```

Second node

```
MACHINE FAILOVER_M FAILOVER_T FAI SERVICE_N COUNT(*)
------- ---------- ---------- --- --------- --------
node2   NONE       NONE       NO  ERP_PRECO 1
```

Second node after

```
MACHINE FAILOVER_M FAILOVER_T FAI SERVICE_N COUNT(*)
------- ---------- ---------- --- --------  --------
node2   BASIC      SESSION    YES AP        1
node2   PRECONNECT SESSION    YES ERP_PRECO 1
```

ORACLE

TAF Verification

To determine whether TAF is correctly configured and that connections are associated with a failover option, you can examine the `V$SESSION` view. To obtain information about the connected clients and their TAF status, examine the `FAILOVER_TYPE`, `FAILOVER_METHOD`, `FAILED_OVER`, and `SERVICE_NAME` columns. The example includes one query that you could execute to verify that you have correctly configured TAF.

This example is based on the previously configured `AP` and `ERP` services, and their corresponding connection descriptors.

The first output in the slide is the result of the execution of the query on the first node after two SQL*Plus sessions from the first node have connected to the `AP` and `ERP` services, respectively. The output shows that the `AP` connection ended up on the first instance. Because of the load-balancing algorithm, it can end up on the second instance. Alternatively, the `ERP` connection must end up on the first instance because it is the only preferred one.

The second output is the result of the execution of the query on the second node before any connection failure. Note that there is currently one unused connection established under the `ERP_PROCONNECT` service that is automatically started on the available `ERP` instance.

The third output is the one corresponding to the execution of the query on the second node after the failure of the first instance. A second connection has been created automatically for the `AP` service connection, and the original `ERP` connection now uses the preconnected connection.

FAN Connection Pools and TAF Considerations

- Both techniques are integrated with services and provide service connection load balancing.
- Do not use FCF when working with TAF, and vice versa.
- Connection pools that use FAN are always preconnected.
- TAF may rely on operating system (OS) timeouts to detect failures.
- FAN never relies on OS timeouts to detect failures.

ORACLE

FAN Connection Pools and TAF Considerations

Because the connection load balancing is a listener functionality, both FCF and TAF automatically benefit from connection load balancing for services.

When you use FCF, there is no need to use TAF. Moreover, FCF and TAF cannot work together.

For example, you do not need to preconnect if you use FAN in conjunction with connection pools. The connection pool is always preconnected.

With both techniques, you automatically benefit from VIPs at connection time. This means that your application does not rely on lengthy operating system connection timeouts at connect time, or when issuing a SQL statement. However, when in the SQL stack, and the application is blocked on a read/write call, the application needs to be integrated with FAN in order to receive an interrupt if a node goes down. In a similar case, TAF may rely on OS timeouts to detect the failure. This takes much more time to fail over the connection than when using FAN.

Summary

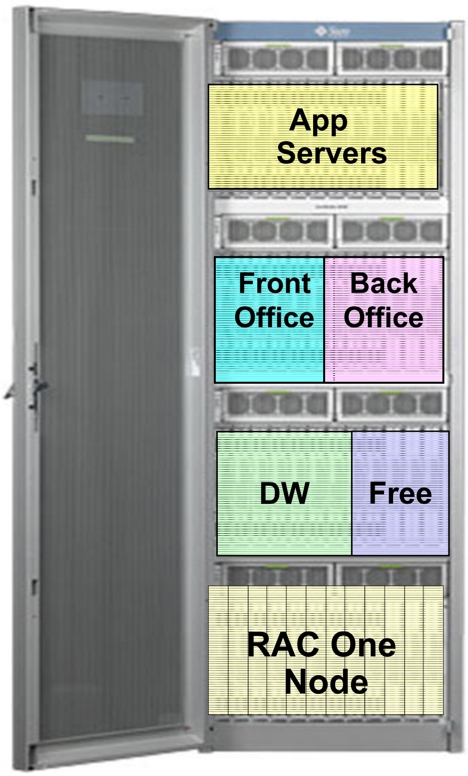
In this lesson, you should have learned how to:

- Configure client-side, connect-time load balancing
- Configure client-side, connect-time failover
- Configure server-side, connect-time load balancing
- Use the Load Balancing Advisory
- Describe the benefits of Fast Application Notification
- Configure server-side callouts
- Configure Transparent Application Failover

ORACLE

Oracle RAC One Node

Oracle RAC One Node



- Allows you to standardize Oracle database deployments across the enterprise
- Allows you to consolidate many databases into a single cluster with minimal overhead
- Supports live migration of instances across servers
- Simplifies rolling patches for single instance databases
- Employs built-in cluster failover for high availability
- Is supported on all platforms where Oracle Real Application Clusters is certified
- Complements virtual and physical servers

ORACLE

Oracle RAC One Node

Oracle RAC One Node allows you to standardize all Oracle Database deployments across the enterprise. Oracle RAC One Node is supported on all platforms where Oracle Real Application Clusters (RAC) is certified. Many databases can be consolidated into a single cluster with minimal overhead while providing the high availability benefits of failover protection, online rolling patch application, as well as rolling upgrades for the operating system and Oracle Clusterware.

Oracle Clusterware provides failover protection to Oracle RAC One Node. If the node fails, Oracle Clusterware will automatically restart the Oracle RAC One Node instance on another server in the cluster.

The Omotion Utility

- Omotion migrates an Oracle RAC One Node instance to another cluster node without application down time.
- Once the instance has been migrated, the node can be patched or upgraded.
- When the maintenance is complete, the Oracle RAC One Node instance can be migrated back to its original home.
- Omotion provides the same load balancing benefits of virtual machines (VMs) by allowing a migration of a database from a busy server to a server with spare capacity.

ORACLE

The Omotion Utility

The Omotion utility is provided to migrate an Oracle RAC One Node instance to another node in the cluster without down time to the application. Once the instance has been migrated, the node can be patched or upgraded. Once the maintenance is complete, the Oracle RAC One Node can be migrated back to its original home.

The Omotion feature provides the same load balancing benefits of VMs by allowing a migration of a database from a busy server to a server with spare capacity. Omotion leverages the ability of Oracle Real Application Clusters to simultaneously run multiple instances servicing a single database. In the figure above, the DB2 RAC One Node database on Server A is migrated to Server B. Oracle RAC One Node starts up a second DB2 instance on server B, and for a short period of time runs in an active-active configuration. As connections complete their transactions on server A, they are migrated to the instance on server B. Once all the connections have migrated, the instance on server A is shut down and the migration is complete.

Omotion does not require quiescing the environment even when the system is running at peak capacity. VMs generally require the environment to be quiesced in order for medium to heavy database workloads to be moved from one server to another. This requirement does not apply for light workloads or demos.

Oracle RAC One Node and OVM

- Using Oracle RAC One Node with Oracle VM increases the scalability and high availability benefits of Oracle VM.
- If a VM is sized too small, you can migrate the Oracle RAC One instance to another Oracle VM node in your cluster and resize the Oracle VM.
- Usually, migrating VMs requires quiescing the source VM, mirroring the memory, and then switching to the migrated VM.
- Using Omotion, highly loaded instances can be migrated as the work is split between the two servers during the migration.
- Omotion provides the ability to move between servers of different processor generations.

ORACLE

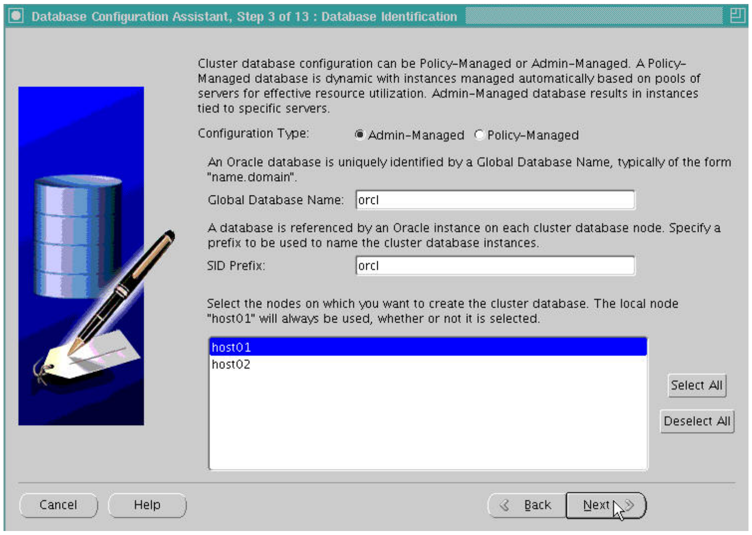
Oracle RAC One Node and OVM

Oracle VM is a free server virtualization and management solution that makes enterprise applications easier to deploy, manage, and support. Using Oracle RAC One Node with Oracle VM increases the benefit of Oracle VM with the high availability and scalability of Oracle RAC. If your VM is sized too small, then you can migrate the Oracle RAC One instance to another Oracle VM node in your cluster using Omotion, and then resize the Oracle VM. When you move the instance back to the newly resized Oracle VM node, you can dynamically increase any limits programmed with Resource Manger Instance Caging.

When migrating a VM, the VM must mirror its complete memory state across a network to the target host, recreating the state of that machine. If the database in question is highly loaded and is actively changing blocks in its database cache, it is very difficult for the memory mirroring function to keep up with the rate of changes. It becomes likely that the only way to successfully mirror the memory is to quiesce the source VM, mirror the memory, and then switch to the migrated VM. With Omotion, highly loaded instances pose no problem, as the work is actually split between two servers during the migration. Oracle RAC One Node can, therefore, easily migrate even heavy workloads to another server. VM migration normally requires that the processors be identical. Both processors must have exactly the same instruction set. Omotion provides the ability to move between servers of different processor generations. Omotion supports migration to new processors, or even between different vendors like Intel and AMD.

Creating and Managing RAC One Node

1. Create a RAC database on one node only.



2. Initialize the database with the `raconeinit` command.
3. Execute the `raconestatus` command to confirm that the initialization was successful.

Creating and Managing RAC One Node

Oracle RAC One Node is a single instance of Oracle RAC that runs on one node in a cluster. You can create a RAC One Node database on an existing cluster using DBCA. On the Database Identification page, you simply select one node only on which to create the database as shown above. After the database has been created, it must be initialized as a RAC One Node database using the `raconeinit` command. It will display a list of databases running in the cluster and allow you to choose which one to initialize:

```
[oracle@host01 ~]$ raconeinit
Candidate Databases on this cluster:
 #      Database          RACOne Node    Fix Required
===     ========        =============    ============
[1]     ORCL              NO                 N
Enter the database to initialize [1]: 1
Database orcl is now running on server host01.
Candidate servers that may be used for this DB:  host02
Enter the names of additional candidate servers where this DB may run (space
delimited): host02
Database configuration modified.
```

Execute the `raconestatus` command to confirm that the initialization was successful.

Creating and Managing RAC One Node (Continued)

```
[oracle@host01 ~]$ raconestatus
RAC One Node databases on this cluster:
 Database UP Fix Required   Current Server  Candidate Server Names
 ======== == ============   ==============  ======================
     orcl  Y       y               host01           host02 host01
```

One of the unique features of RAC One Node is the ability to online migrate a running database from one node in the cluster to another. This migration, called Omotion, is controlled by the Omotion command.

To continue with our example, move the database now running on host01 to host02 using the `Omotion` command. Choose 1 to select the orcl database. Hit return to accept the default maximum migration time of 30 minutes.

```
[oracle@host01 ~]$ Omotion
RAC One Node databases on this cluster:
 #  Database      Server          Fix Required
=== ========      ===========     ===================
[1] orcl          rac1                   N
Enter number of the database to migrate [1]: 1
Specify maximum time in minutes for migration to complete (max 30) [30]:
Available Target Server(s) :
 #      Server       Available
===     ===========  =========
[1]       host02          Y
Enter number of the target node [1]: 1
Omotion Started...
Starting target instance on host02...
Migrating sessions...
Stopping source instance on host01...
Omotion Completed...
=== Current Status ===
Database racone is running on node host02
```

Assume host01 has crashed. Running `raconestatus` shows the following:

```
[oracle@host01 ~]$ raconestatus
RAC One Node databases on this cluster:
 Database UP Fix Required   Current Server  Candidate Server Names
 ======== == ============   ==============  ======================
     orcl  Y       y               host02           host02 host01
```

After a failover, you must run the `raconefix` command to reset the RAC One metadata.

```
[oracle@host01 ~]$ raconefix
RAC One Node databases on this cluster:
 #      Database                    Server                        Fix Required
===     ========        ==============================           ============
[1]         orcl                                  host01                  Y
Enter number of the database to fix [1]: 1
If database is up, it will be checked and cleaned after a previous fail over.
```

E

Cloning Oracle Clusterware

Objectives

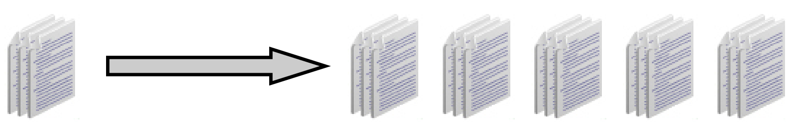
After completing this lesson, you should be able to:

- Describe the cloning procedure
- Prepare the software for cloning
- Describe the cloning script variables
- Clone Oracle Clusterware to create a new cluster
- Clone Oracle Clusterware to extend an existing cluster
- Examine cloning log files

What Is Cloning?

Cloning is the process of copying an existing Oracle Clusterware installation to a different location. It:

- Requires a successful installation as a baseline
- Can be used to create new clusters
- Cannot be used to remove nodes from the cluster
- Does not perform the operating system prerequisites to an installation
- Is useful to build many clusters in an organization



ORACLE

What Is Cloning?

Cloning is a process that allows the copying of an existing Oracle Clusterware installation to a different location and then updating the copied installation to work in the new environment. The cloned copy can be used to create a new cluster from a successfully installed cluster. To add or delete Oracle Clusterware from nodes in the cluster, use the `addNode.sh` and `rootcrs.pl` scripts. The cloning procedure cannot be used to remove a node from an existing cluster.

The cloning procedure is responsible for the work that would have been done by the Oracle Universal Installer (OUI) utility. It does not automate the prerequisite work that must be done on each node before installing the Oracle software.

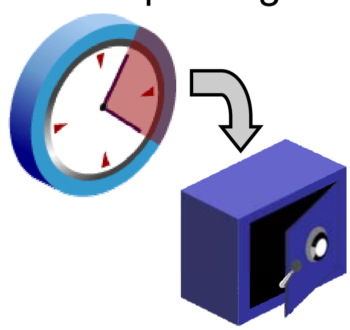
This technique is very useful if a large number of clusters need to be deployed in an organization. If only one or two clusters are being deployed, you should probably use the traditional installation program to perform the installations.

Note: The Oracle Enterprise Manager administrative tool with the Provisioning Pack feature installed has automated wizards to assist with cloning exercises.

Benefits of Cloning Oracle Clusterware

The following are some of the benefits of cloning Oracle Clusterware:

- Can be completed in silent mode from a Secure Shell (SSH) terminal session
- Contains all patches applied to the original installation
- Can be done very quickly
- Is a guaranteed method of repeating the same installation on multiple clusters

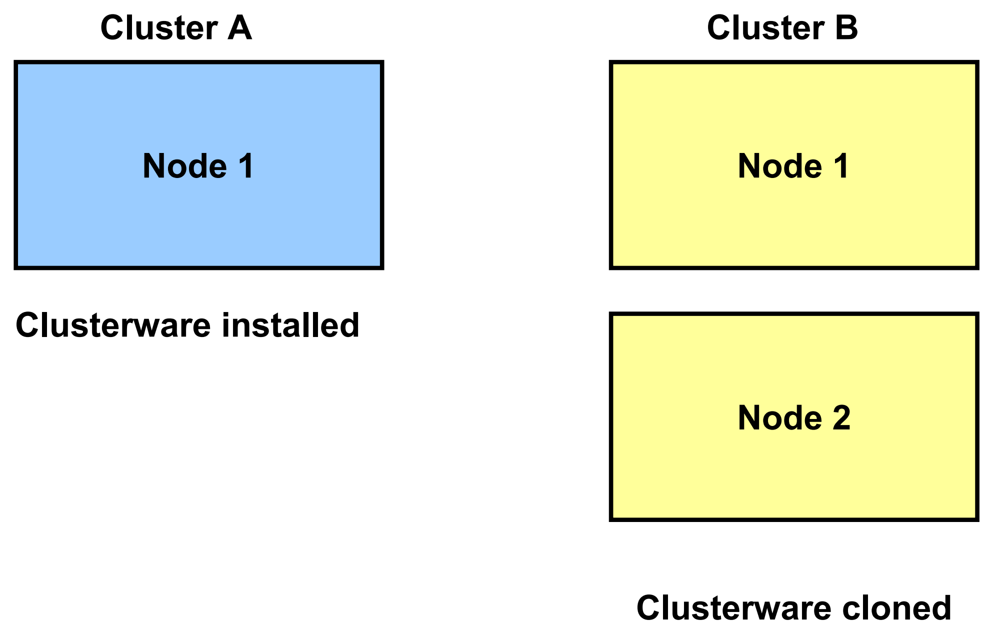


Benefits of Cloning Oracle Clusterware

Using the cloning procedure presented in this lesson has several benefits compared to the traditional Oracle Universal Installer (OUI) installation method. The OUI utility is a graphical program and must be executed from a graphical session. Cloning can be completed in silent mode from a command-line Secure Shell (SSH) terminal session without the need to load a graphical windows system. If the OUI program were to be used to install the software from the original installation media, all patches that have been applied since the first installation would have to be reapplied. The clone technique presented in this lesson includes all successfully applied patches, and can be performed very quickly to a large number of nodes. When the OUI utility performs the copying of files to remote servers, the job is executed serially to one node at a time. With cloning, simultaneous transfers to multiple nodes can be achieved. Finally, the cloning method is a guaranteed way of repeating the same installation on multiple clusters to help avoid human error.

The cloned installation acts the same as the source installation. It can be patched in the future and removed from a cluster if needed using ordinary tools and utilities. Cloning is an excellent way to instantiate test clusters or a development cluster from a successful base installation.

Creating a Cluster by Cloning Oracle Clusterware



Creating a Cluster by Cloning Oracle Clusterware

This example shows the result when you successfully clone an installed Oracle Clusterware environment to create a cluster. The environment on Node 1 in Cluster A is used as the source, and Cluster B, Nodes 1 and 2 are the destination. The clusterware home is copied from Cluster A, Node 1, to Cluster B, Nodes 1 and 2. When completed, there will be two separate clusters. The OCR and Voting disks are not shared between the two clusters after you successfully create a cluster from a clone.

Preparing the Oracle Clusterware Home for Cloning

The following procedure is used to prepare the Oracle Clusterware home for cloning:

1. Install Oracle Clusterware on the first machine.
   A. Use the Oracle Universal Installer (OUI) GUI interactively.
   B. Install patches that are required (for example, 11.1.0.*n*).
   C. Apply one-off patches, if necessary.
2. Shut down Oracle Clusterware.

```
# crsctl stop crs -wait
```

3. Make a copy of the Oracle Clusterware home.

```
# mkdir /stagecrs
# cp -prf /u01/app/11.2.0/grid /stagecrs
```


Preparing the Oracle Clusterware Home for Cloning

The cloning method requires that an existing, successful installation of Oracle Clusterware be already performed in your organization. Verify that all patch sets and one-off patches have been applied before starting the clone procedure to minimize the amount of work that will have to be performed for the cloning exercise.

To begin the cloning process, start by performing a shutdown of Oracle Clusterware on one of the nodes in the existing cluster with the `crsctl stop crs -wait` command. The other nodes in the existing cluster can remain active. After the prompt is returned from the shutdown command, make a copy of the existing Oracle Clusterware installation into a temporary staging area of your choosing. The disk space requirements in the temporary staging area will be equal to the current size of the existing Oracle Clusterware installation.

The copy of the Oracle Clusterware files will not include files that are outside the main installation directory such as `/etc/oraInst.loc` and the `/etc/oracle` directory. These files will be created later by running various `root` scripts.

Preparing the Oracle Clusterware Home for Cloning

4. Remove files that pertain only to the source node.

```
# cd /stagecrs/grid
# rm -rf /stagecrs/grid/log/<hostname>
# rm -rf root.sh*
# rm -rf gpnp/*
# find . -name '*.ouibak' -exec rm {} \;
# find . -name '*.ouibak.1' -exec rm {} \;
# rm –rf  \
  inventory/ContentsXML/oraclehomeproperties.xml
# cd ./cfgtoollogs
# find . -type f -exec rm -f {} \;
```

5. Create an archive of the source.

```
# cd /stagecrs/grid
# tar –zcvf /tmp/crs111060.tgz .
```


Preparing the Oracle Clusterware Home for Cloning (continued)

Each installation of Oracle Clusterware on local storage devices contains files and directories that are applicable to only that node. These files and directories should be removed before making an archive of the software for cloning. Perform the commands in the slide to remove node-specific information from the copy that was made in step 3. Then create an archive of the Oracle Clusterware copy. An example of using the Linux `tar` command followed by the `compress` command to reduce the file size is shown in the slide. On Windows systems, use the WinZip utility to create a zip file for the archive.

Note: Do not use the Java Archive (JAR) utility to copy and compress the Oracle Clusterware home.

Preparing the Oracle Clusterware Home for Cloning

6. Restart Oracle Clusterware.

```
# crsctl start crs
```

Cloning to Create a New Oracle Clusterware Environment

The following procedure uses cloning to create a new cluster:

1. Prepare the new cluster nodes. (See the lesson titled “Grid Infrastructure Installation” for details.)
   A. Check system requirements.
   B. Check network requirements.
   C. Install the required operating system packages.
   D. Set kernel parameters.
   E. Create groups and users.
   F. Create the required directories.
   G. Configure installation owner shell limits.
   H. Configure SSH and enable user equivalency.
   I. Use the Cluster Verify Utility (`cluvfy`) to check prerequisites.

ORACLE

Cloning to Create a New Oracle Clusterware Environment

The archive of the source files that was made when preparing the Oracle Clusterware home for cloning will become the source files for an installation on the new cluster system and the `clone.pl` script will be used to make it a working environment. These actions do not perform the varied prerequisite tasks that must be performed on the operating system before an Oracle Clusterware installation. The prerequisite tasks are the same as the ones presented in the lesson titled “Grid Infrastructure Installation” in the topic about installing Oracle Clusterware.

One of the main benefits of cloning is rapid instantiation of a new cluster. The prerequisite tasks are manual in nature, but it is suggested that a shell script be developed to automate these tasks. One advantage of the Oracle Enterprise Linux operating system is that an `oracle-validated-1.x.x.rpm` file can be obtained that will perform almost all the prerequisite checks including the installation of missing packages with a single command.

This cloning procedure is used to build a new distinct cluster from an existing cluster. In step 1, prepare the new cluster nodes by performing prerequisite setup and checks.

If `cluvfy` fails to execute because of user equivalence errors, the passphrase needs to be loaded with the following commands before executing `cluvfy`:

```
exec /usr/bin/ssh-agent $SHELL
ssh-add
```

Cloning to Create a New Oracle Clusterware Environment

2. Deploy Oracle Clusterware on each of the destination nodes.
   A. Extract the tar file created earlier.

```
# mkdir -p /u01/app/11.20/grid
# cd /u01/app/11.2.0
# tar -zxvf /tmp/crs111060.tgz
```

   B. Change the ownership of files, and create Oracle Inventory.

```
# chown -R crs:oinstall /u01/app/11.2.0/grid
# mkdir -p /u01/app/oraInventory
# chown grid:oinstall /u01/app/oraInventory
```

   C. Remove any network files from `/u01/app/11.2.0/grid/network/admin`

```
$ rm /u01/app/11.2.0/grid/network/admin/*
```

ORACLE

Cloning to Create a New Oracle Clusterware Environment (continued)

Step 2 is the deployment and extraction of the source archive to the new cluster nodes. If a shared Oracle Cluster home on a Cluster File System (CFS) is not being utilized, extract the source archive to each node's local file system. It is possible to change the operating system owner to a different one from that of the source with recursive `chown` commands as illustrated in the slide. If other Oracle products have been previously installed on the new nodes, the Central Oracle Inventory directory may already exist. It is possible for this directory to be owned by a different user than the Oracle Grid Infrastructure user; however, both should belong to the same primary group `oinstall`. Execute the `preupdate.sh` script on each target node, logged in as `root`. This script will change the ownership of the CRS home and `root`-owned files to the Oracle CRS user.

The `clone.pl` Script

Cloning to a new cluster and cloning to extend an existing cluster both use a PERL script. The `clone.pl` script is used, which:

- Can be used on the command line
- Can be contained in a shell script
- Accepts many parameters as input
- Is invoked by the PERL interpreter

```
# perl <CRS_home>/clone/bin/clone.pl $E01 $E02 $E03
$E04 $C01 $CO2
```


The `clone.pl` Script

The PERL-based `clone.pl` script is used in place of the graphical OUI utility to perform the installation on the new nodes so that they may participate in the existing cluster or become valid nodes in a new cluster. The script can be executed directly on the command line or at a DOS command prompt for Windows platforms. The `clone.pl` script accepts several parameters as input, typed directly on the command line. Because the `clone.pl` script is sensitive to the parameters being passed to it, including the use of braces, single quotation marks, and double quotation marks, it is recommended that a shell script be created to execute the PERL-based `clone.pl` script to input the arguments. This will be easier to rework if there is a syntax error generated. There are a total of seven arguments that can be passed as parameters to the `clone.pl` script. Four of them define environment variables, and the remaining three supply processing options. If your platform does not include a PERL interpreter, you can download one at:

`http://www.perl.org`

The clone.pl Environment Variables

The clone.pl script accepts four environment variables as input. They are as follows:

Symbol	Variable	Description
E01	ORACLE_BASE	The location of the Oracle base directory
E02	ORACLE_HOME	The location of the Oracle Grid Infrastructure home. This directory location must exist and must be owned by the Oracle operating system group: oinstall.
E03	ORACLE_HOME_NAME	The name of the Oracle Grid Infrastructure home
E04	INVENTORY_LOCATION	The location of the Oracle Inventory

The clone.pl Environment Variables

The clone.pl script accepts four environment variables as command-line arguments providing input. These variables would correlate to some of the questions presented by the OUI utility during an installation. Each variable is associated with a symbol that is used for reference purposes for developing a shell script. The choice of symbol is arbitrary. Each variable is case-sensitive. The variables are as follows:

- **ORACLE_BASE:** The location of the Oracle Base directory. A suggested value is /u01/app/grid. The ORACLE_BASE value should be unique for each software owner.
- **ORACLE_HOME:** The location of the Oracle Grid Infrastructure home. This directory location must exist and be owned by the Oracle Grid Infrastructure software owner and the Oracle Inventory group, typically grid:oinstall. A suggested value is /u01/app/<version>/grid.
- **ORACLE_HOME_NAME:** The name of the Oracle Grid Infrastructure home. This is stored in the Oracle inventory and defaults to the name Ora11g_gridinfrahome1 when performing an installation with OUI. Any name can be selected, but it should be unique in the organization.
- **INVENTORY_LOCATION:** The location of the Oracle Inventory. This directory location must exist and must initially be owned by the Oracle operating system group: oinstall. A typical location is /u01/app/oraInventory.

The `clone.pl` Command Options

The `clone.pl` script accepts two required command options as input. They are as follows:

#	Variable	Data Type	Description
C01	`CLUSTER_NODES`	String	This list of short node names for the nodes in the cluster
C02	`LOCAL_NODE`	String	The short name of the local node

The `clone.pl` Command Options

The `clone.pl` script accepts two command options as input. The command options are case-sensitive and are as follows:

- `CLUSTER_NODES`: This list of short node names for the nodes in the cluster
- `LOCAL_NODES`: The short name of the local node

Cloning to Create a New Oracle Clusterware Environment

3. Create a shell script to invoke `clone.pl` supplying input.

```
#!/bin/sh
ORACLE_BASE=/u01/app/oracle
GRID_HOME=/u01/app/11.2.0/grid
THIS_NODE=`hostname -s`

E01=ORACLE_BASE=${ORACLE_BASE}
E02=ORACLE_HOME=${ORACLE_HOME}
E03=ORACLE_HOME_NAME=OraGridHome1
E04=INVENTORY_LOCATION=${ORACLE_BASE}/oraInventory

#C00="-O'-debug'"
C01="-O'\"CLUSTER_NODES={node1,node2}\"'"
C02="-O'\"LOCAL_NODE=${THIS_NODE}\"'"

perl ${GRID_HOME}/clone/bin/clone.pl -silent $E01 $E02
$E03 $E04 $C01 $C02
```

Syntax per bug 8718041

ORACLE

Cloning to Create a New Oracle Clusterware Environment

The `clone.pl` script requires you to provide many setup values to the script when it is executed. You may enter the values interactively on the command line or create a script to supply the input values. By creating a script, you will have the ability to modify it and execute the script a second time if errors exist. The setup values to the `clone.pl` script are case-sensitive and sensitive to the use of braces, single quotation marks, and double quotation marks.

For step 3, create a shell script that invokes `clone.pl` supplying command-line input variables. All the variables should appear on a single line.

Cloning to Create a New Oracle Clusterware Environment

4. Run the script created in Step 3 on each node.

```
$ /tmp/my-clone-script.sh
```

5. Prepare the `crsconfig_params` file.

Cloning to Create a New Oracle Clusterware Environment (continued)

Step 4: Run the script you created in Step 3 as the operating system user that installed Oracle Clusterware. If you do not have a shared Oracle grid infrastructure home run this script on each node. The `clone.pl` command instantiates the `crsconfig_params` file in the next step.

Step 5: Prepare the `/u01/app/11.2.0/grid/install/crsconfig_params` file on all of the nodes in the cluster. You can copy the file from one node to all of the other nodes. More than 50 parameters are named in this file.

Notes: In 11.2.0.1 the `clone.pl` script does not propagate the `crsconfig_params` file.

Cloning to Create a New Oracle Clusterware Environment

6. Run the `orainstRoot.sh` script on each node.

```
# /u01/app/oraInventory/orainstRoot.sh
```

7. Run the `<CRS_home>/root.sh` and `rootcrs.pl` scripts on each node.

```
# /u01/app/11.2.0/grid/root.sh
# /u01/app/11.2.0/grid/perl/bin/perl \
   -I/u01/app/11.2.0/grid/perl/lib \
   -I/u01/app/11.2.0/grid/crs/install \
   /u01/app/11.2.0/grid/crs/install/rootcrs.pl
```


Cloning to Create a New Oracle Clusterware Environment (continued)

When the shell script containing `clone.pl` successfully completes, it is necessary to execute the `orainstRoot.sh` script for step 6 and the `root.sh` script for step 7 on each node, both as the `root` user.

Note

Step 4 must be run to completion before you start Step 5. Similarly, Step 5 must be run to completion before you start Step 6.

You can perform Step 4, Step 5, and Step 6 simultaneously on different nodes. Step 6 must be complete on all nodes before you can run Step 7.

Step 7: Ensure that the `root.sh` and `rootcrs.pl` scripts have completed on the first node before running them on the second node and subsequent nodes.

Cloning to Create a New Oracle Clusterware Environment

8. Run the configuration assistants on each new node.

```
$ /u01/app/11.2.0/grid/bin/netca \
    /orahome /u01/app/11.2.0/grid \
    /orahnam OraGridHome1 /instype typical \
    /inscomp client,oraclenet,javavm,server\
    /insprtcl tcp /cfg local \
    /authadp NO_VALUE \
    /responseFile \
 /u01/app/11.2.0/grid/network/install/netca_typ.rsp \
    /silent
```

9. Run the `cluvfy` utility to validate the installation.

```
$ cluvfy stage -post crsinst -n all -verbose
```


Cloning to Create a New Oracle Clusterware Environment (continued)

Step 8: You should execute the command shown in the slide, on the first node.

If you are using ASM, then run the following command:

```
$ /u01/app/11.2.0/grid/bin/asmca -silent -postConfigureASM \
 -sysAsmPassword oracle -asmsnmpPassword oracle
```

If you a plan to run a pre-11*g* Release 2 (11.2) database on this cluster, then you should run `oifcfg` as described in the Oracle Database 11*g* Release 2 (11.2) documentation.

To use IPMI, use the `crsctl` command to configure IPMI on each node:

```
# crsctl set css ipmiaddr ip_address
```

At this point, Oracle Clusterware is fully installed and configured to operate in the new cluster. The `cluvfy` utility can be invoked in the post-CRS installation stage to verify the installation with the following syntax:

```
cluvfy stage -post crsinst -n all -verbose
```

If `cluvfy` fails to execute because of user equivalence errors, the passphrase needs to be loaded with the following commands before executing `cluvfy` again:

```
exec /usr/bin/ssh-agent $SHELL
ssh-add
```

Log Files Generated During Cloning

The following log files are generated during cloning to assist with troubleshooting failures.

- Detailed log of the actions that occur during the OUI part:

```
/u01/app/oraInventory/logs/cloneActions<timestamp>.log
```

- Information about errors that occur when OUI is running:

```
/u01/app/oraInventory/logs/oraInstall<timestamp>.err
```

- Other miscellaneous messages generated by OUI:

```
/u01/app/oraInventory/logs/oraInstall<timestamp>.out
```

ORACLE

Log Files Generated During Cloning

Several log files are generated when the `clone.pl` script is executed and are useful in diagnosing any errors that may occur. For a detailed log of the actions that occur during the OUI part of the cloning, examine the log file:

```
<Central_Inventory>logs/cloneActions/<timestamp>.log
```

If errors occurred during the OUI portion of the cloning process, examine the log file:

```
<Central_Inventory>logs/oraInstall/<timestamp>.err
```

Other miscellaneous messages generated by OUI can be found in the output file:

```
<Central_Inventory>logs/oraInstall/<timestamp>.out
```

Log Files Generated During Cloning

The following log files are generated during cloning to assist with troubleshooting failures:

- Detailed log of the actions that occur before cloning as well as during cloning operations:

```
/u01/app/grid/clone/logs/clone-<timestamp>.log
```

- Information about errors that occur before cloning as well as during cloning operations:

```
/u01/app/grid/clone/logs/error-<timestamp>.log
```


Log Files Generated During Cloning (continued)

Several log files are generated when the `clone.pl` script is executed and are useful in diagnosing errors that may occur. For a detailed log of the actions that occur before cloning as well as during the cloning process, examine the log file:

```
<CRS home>/clone/logs/clone-<timestamp>.log
```

For a detailed log of the errors that occur before cloning as well as during the cloning process, examine the log file:

```
<CRS home>/clone/logs/error-<timestamp>.log
```

Cloning to Extend Oracle Clusterware to More Nodes

The procedure is very similar to cloning to create new clusters.

1. Prepare the new cluster nodes. (See the lesson titled “Oracle Clusterware Installation” for details. Suggest that this be made into a shell script for reuse.)
   A. Check system requirements.
   B. Check network requirements.
   C. Install the required operating system packages.
   D. Set kernel parameters.
   E. Create groups and users.
   F. Create the required directories.
   G. Configure installation owner shell limits.
   H. Configure SSH and enable user equivalency.
   I. Use the `cluvfy` utility to check prerequisites.

ORACLE

Cloning to Extend Oracle Clusterware to More Nodes

The cloning procedure using the `clone.pl` script can also be used to extend Oracle Clusterware to more nodes within the same cluster with steps similar to the procedure for creating a new cluster. The first step shown here is identical to the first step performed when cloning to create a new cluster. These steps differ depending on the operating system used.

When you configure Secure Shell (SSH) and enable user equivalency, remember that the `authorized_keys` and `known_hosts` files exist on each node in the cluster. Therefore, it will be necessary to update these files on the existing nodes with information about the new nodes that Oracle Clusterware will be extended to.

Note: Not supported in 11.2.0.1

Cloning to Extend Oracle Clusterware to More Nodes

2. Deploy Oracle Clusterware on the destination nodes.
   A. Extract the tar file created earlier.

```
# mkdir -p /u01/app/crs
# cd /u01/app/crs
# tar -zxvf /tmp/crs111060.tgz
```

   B. Change the ownership of files and create Oracle Inventory.

```
# chown -R crs:oinstall /u01/app/crs
# mkdir -p /u01/app/oraInventory
# chown crs:oinstall /u01/app/oraInventory
```

   C. Run the `preupdate.sh` script on each target node.

```
# /u01/app/crs/install/preupdate.sh \
-crshome /u01/app/crs -crsuser crs -noshutdown
```

ORACLE

Cloning to Extend Oracle Clusterware to More Nodes (continued)

Step 2 is the deployment and extraction of the source archive to the new cluster nodes. If a shared Oracle Cluster home on a CFS is not being used, extract the source archive to each node's local file system. Because this node will participate with the existing nodes in a cluster, it is not possible to use different account names for the Oracle Clusterware software owner. If other Oracle products have been previously installed on the new nodes, the Central Oracle Inventory directory may already exist. It is possible for this directory to be owned by a different user than the Oracle Clusterware user; however, both should belong to the same primary group `oinstall`. Execute the `preupdate.sh` script on each target node, logged in as `root`. This script will change the ownership of the CRS home and `root`-owned files to the Oracle CRS user if needed.

Cloning to Extend Oracle Clusterware to More Nodes

3. Create a shell script to invoke `clone.pl` supplying input.

```
#!/bin/sh
# /tmp/my-clone-script.sh
E01=ORACLE_BASE=/u01/app
E02=ORACLE_HOME=/u01/app/crs
E03=ORACLE_HOME_NAME=OraCrs11g
C01="-O'sl_tableList={node3:node3-priv:node3-vip:N:Y}'"
C02="-O'INVENTORY_LOCATION=/u01/app/oraInventory'"
C03="-O'-noConfig'"

perl /u01/app/crs/clone/bin/clone.pl $E01 $E02 $E03 $C01 $C02 $C03
```

4. Run the shell script created in step 3 on each new node.

```
$ /tmp/my-clone-script.sh
```


Cloning to Extend Oracle Clusterware to More Nodes (continued)

For step 3, the quantity of input variables to the `clone.pl` procedure is greatly reduced because the existing cluster has already defined many of the settings that are needed. Creating a shell script to provide these input values is still recommended. For step 4, run the shell script that you developed to invoke the `clone.pl` script.

Cloning to Extend Oracle Clusterware to More Nodes

5. Run the `orainstRoot.sh` script on each new node.

```
# /u01/app/oraInventory/orainstRoot.sh
```

6. Run the `addNode` script on the source node.

```
$ /u01/app/crs/oui/bin/addNode.sh -silent \
"CLUSTER_NEW_NODES={new_nodes}" \
"CLUSTER_NEW_PRIVATE_NODE_NAMES={new_node-priv}" \
"CLUSTER_NEW_VIRTUAL_HOSTNAMES={new_node-vip}" \
-noCopy
```

7. Run the `rootaddnode.sh` script on the source node.

```
# /u01/app/crs/install/rootaddnode.sh
```


Cloning to Extend Oracle Clusterware to More Nodes (continued)

For step 5, run the `orainstRoot.sh` script on each new node as the `root` user. For step 6, run the `addNode.sh` script on the source node, not on the destination node. Because the `clone.pl` scripts have already been run on the new nodes, this script only updates the inventories of the existing nodes. For step 7, again on the source node, run the `rootaddnode.sh` script as `root` to instantiate the node.

Cloning to Extend Oracle Clusterware to More Nodes

8. Run the `<CRS_home>/root.sh` script on the new node.

```
# /u01/app/crs/root.sh
```

9. Run the configuration assistants on each node that is listed in the `configToolAllCommands` file.

```
$ cat /u01/app/crs/cfgtoollogs/configToolAllCommands
```

```
$ onsconfig add_config node3:6501
```

10. Run `cluvfy` to validate the installation.

```
$ cluvfy stage -post crsinst -n all -verbose
```

ORACLE

Cloning to Extend Oracle Clusterware to More Nodes (continued)

Next, in step 8, run the `root.sh` script on each new node joining the cluster as the `root` user. For step 9, additional commands for the configuration assistants need to be run on the new nodes joining the cluster as the Oracle Clusterware software owner. The list of commands can be found in the `<CRS home>/cfgtoollogs/configToolAllCommands` file. Finally, for step 10, which is the last step, run `cluvfy` to verify the success of the Oracle Clusterware installation.

Quiz

An Oracle Clusterware home that was created with cloning techniques can be used as the source for additional cloning exercises.

1. True
2. False

Answer: 1

Quiz

Which scripting language is used for the cloning script?

1. Java
2. PERL
3. Shell
4. Python

Answer: 2

Summary

In this lesson, you should have learned how to:

- Describe the cloning process
- Describe the `clone.pl` script and its variables
- Perform a clone of Oracle Clusterware to a new cluster
- Extend an existing cluster by cloning

ORACLE

F

Clusterware Concepts

Objectives

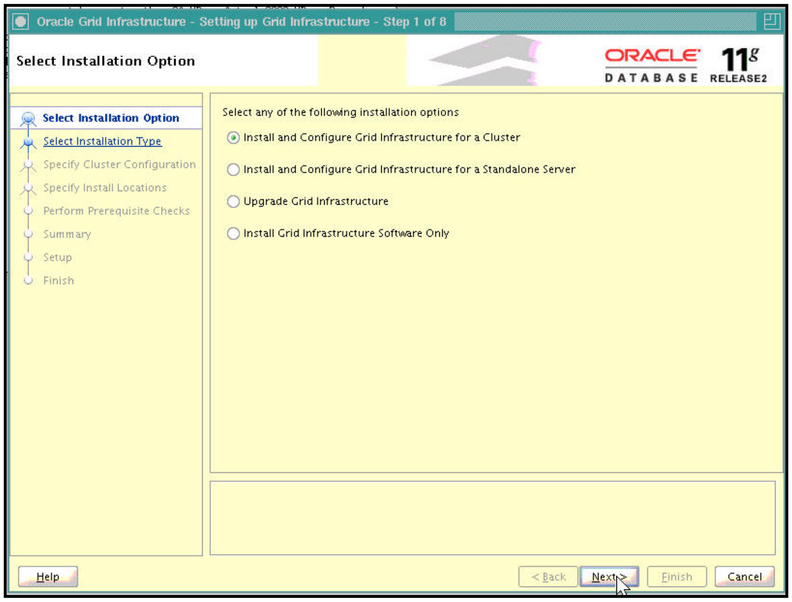
After completing this lesson, you should be able to:

- Explain the principles and purposes of clusters
- Describe the Oracle Clusterware architecture
- Describe how Grid Plug and Play affects Clusterware

ORACLE

Oracle Grid Infrastructure

- ASM and Oracle Clusterware are installed into a single home directory called Oracle Grid Infrastructure 11*g* Release 2.
- This directory is referred to as the Grid Infrastructure home.

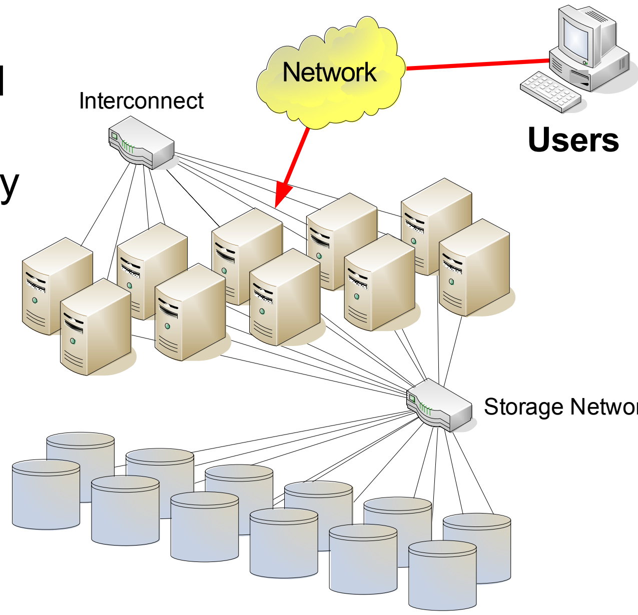


Oracle Grid Infrastructure

With Oracle Database 11*g* Release 2, Automatic Storage Management (ASM) and Oracle Clusterware are installed into a single home directory, collectively called Oracle Grid Infrastructure. This directory is referred to as the Grid Infrastructure home. Configuration assistants start after the Oracle Universal Installer interview process and binary installation that configure ASM and Oracle Clusterware. Although the installation is called Oracle Grid Infrastructure, Oracle Clusterware and Automatic Storage Manager remain separate components.

What Is a Cluster?

- A group of independent, but interconnected, computers that act as a single system
- Usually deployed to increase availability and performance or to balance a dynamically changing workload



What Is a Cluster?

A cluster consists of a group of independent but interconnected computers whose combined resources can be applied to a processing task. A common cluster feature is that it should appear to an application as though it were a single server. Most cluster architectures use a dedicated network (cluster interconnect) for communication and coordination between cluster nodes.

A common cluster architecture for data-intensive transactions and computations is built around shared disk storage. Shared-nothing clusters use an alternative architecture where storage is not shared and data must be either replicated or segmented across the cluster. Shared-nothing clusters are commonly used for workloads that can be easily and predictably divided into small units that can be spread across the cluster in parallel. Shared disk clusters can perform these tasks but also offer increased flexibility for varying workloads. Load balancing clusters allow a single application to balance its workload across the cluster. Alternatively, in a failover cluster, some nodes can be designated as the primary host for an application, whereas others act as the primary host for different applications. In a failover cluster, the failure of a node requires that the applications it supports be moved to a surviving node. Load balancing clusters can provide failover capabilities but they can also run a single application across multiple nodes providing greater flexibility for different workload requirements. Oracle supports a shared disk cluster architecture providing load balancing and failover capabilities. In an Oracle cluster, all nodes must share the same processor architecture and run the same operating system.

What Is Clusterware?

Software that provides various interfaces and services for a cluster. Typically, this includes capabilities that:

- Allow the cluster to be managed as a whole
- Protect the integrity of the cluster
- Maintain a registry of resources across the cluster
- Deal with changes to the cluster
- Provide a common view of resources

What Is Clusterware?

Clusterware is a term used to describe software that provides interfaces and services that enable and support a cluster.

Different cluster architectures require clusterware that delivers different services. For example, in a simple failover cluster the clusterware may monitor the availability of applications and perform a failover operation if a cluster node becomes unavailable. In a load balancing cluster, different services are required to support workload concurrency and coordination.

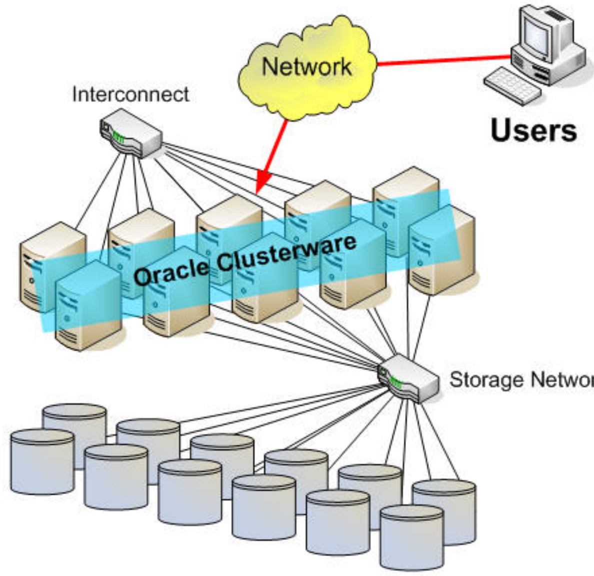
Typically, clusterware includes capabilities that:

- Allow the cluster to be managed as a single entity (not including OS requirements), if desired
- Protect the integrity of the cluster so that data is protected and the cluster continues to function even if communication with a cluster node is severed
- Maintain a registry of resources so that their location is known across the cluster and so that dependencies between resources is maintained
- Deal with changes to the cluster such as node additions, removals, or failures
- Provide a common view of resources such as network addresses and files in a file system

Oracle Clusterware

Oracle Clusterware is:

- A key part of Oracle Grid Infrastructure
- Integrated with Oracle Automatic Storage Management (ASM)
- The basis for ASM Cluster File System (ACFS)
- A foundation for Oracle Real Application Clusters (RAC)
- A generalized cluster infrastructure for all kinds of applications



Oracle Clusterware

Oracle Clusterware is a key part of Oracle Grid Infrastructure, which also includes Automatic Storage Management (ASM) and the ASM Cluster File System (ACFS).

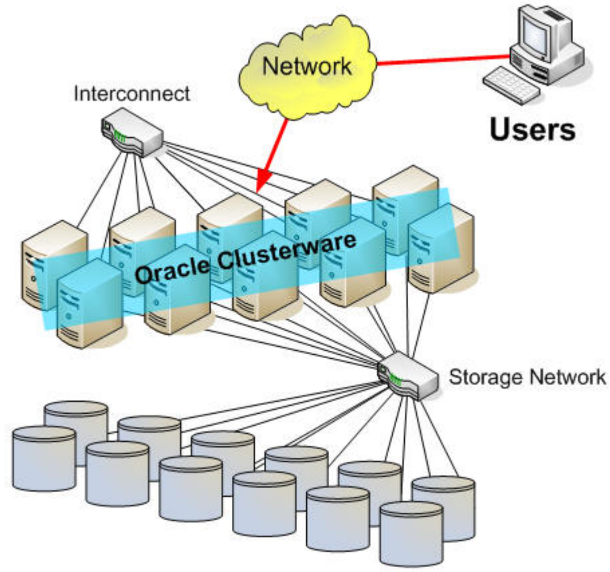
In Release 11.2, Oracle Clusterware can use ASM for all the shared files required by the cluster. Oracle Clusterware is also an enabler for the ASM Cluster File System, a generalized cluster file system that can be used for most file-based data such as documents, spreadsheets, and reports.

The combination of Oracle Clusterware, ASM, and ACFS provides administrators with a unified cluster solution that is not only the foundation for the Oracle Real Application Clusters (RAC) database, but can also be applied to all kinds of other applications.

Note: Grid Infrastructure is the collective term that encompasses Oracle Clusterware, ASM, and ACFS. These components are so tightly integrated that they are often collectively referred to as Oracle Grid Infrastructure.

Oracle Clusterware Architecture and Services

- Shared disk cluster architecture supporting application load balancing and failover
- Services include:
  - Cluster management
  - Node monitoring
  - Event services
  - Time synchronization
  - Network management
  - High availability



Oracle Clusterware Architecture and Services

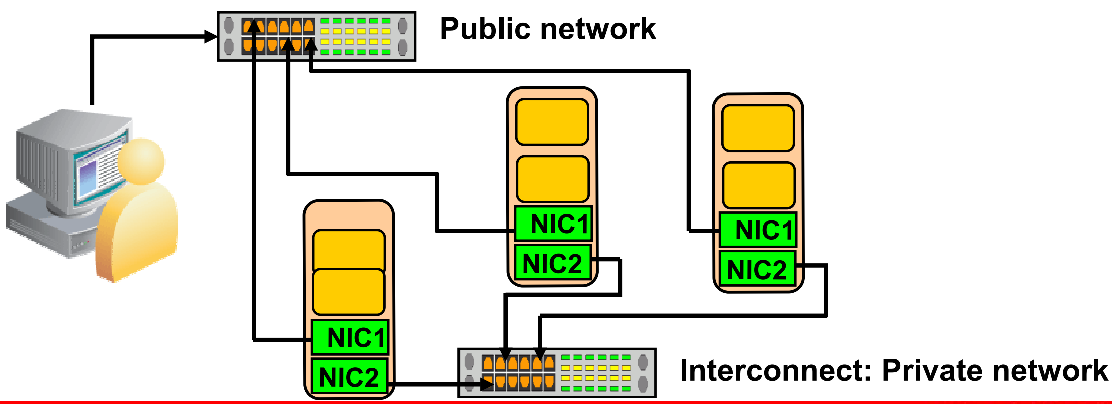
Oracle Clusterware provides a complete set of cluster services to support the shared disk, load balancing cluster architecture of the Oracle Real Application Cluster (RAC) database. Oracle Clusterware can also be used to provide failover clustering services for single instance Oracle databases and other applications.

The services provided by Oracle Clusterware include:

- Cluster Management, which allows cluster services and application resources to be monitored and managed from any node in the cluster
- Node Monitoring, which provides real-time information regarding which nodes are currently available and the resources they support. Cluster integrity is also protected by evicting or fencing unresponsive nodes.
- Event Services, which publishes cluster events so that applications are aware of changes in the cluster
- Time Synchronization, which synchronizes the time on all nodes of the cluster
- Network Management, which provisions and manages Virtual IP (VIP) addresses that are associated with cluster nodes or application resources to provide a consistent network identity regardless of which nodes are available. In addition, Grid Naming Service (GNS) manages network naming within the cluster.
- High Availability, which services, monitors, and restarts all other resources as required

Oracle Clusterware Networking

- Each node must have at least two network adapters.
- Each public network adapter must support TCP/IP.
- The interconnect adapter must support:
  - User Datagram Protocol (UDP) or Reliable Data Socket (RDS) for UNIX and Linux for database communication
  - TCP for Windows platforms for database communication
- All platforms use Grid Interprocess Communication (GIPc)



Oracle Clusterware Networking

Each node must have at least two network adapters: one for the public network interface and the other for the private network interface or interconnect. In addition, the interface names associated with the network adapters for each network must be the same on all nodes. For example, in a two-node cluster, you cannot configure network adapters on node1 with eth0 as the public interface, but on node2 have eth1 as the public interface. Public interface names must be the same, so you must configure eth0 as public on both nodes. You should configure the private interfaces on the same network adapters as well. If eth1 is the private interface for node1, then eth1 should be the private interface for node2.

Before starting the installation, on each node, you must have at least two interfaces to configure for the public and private IP addresses. You can configure IP addresses with one of the following options:

- Oracle Grid Naming Service (GNS) using one static address defined during installation, which dynamically allocates VIP addresses using Dynamic Host Configuration Protocol (DHCP), which must be running on the network. You must select the Advanced Oracle Clusterware installation option to use GNS.
- Static addresses that network administrators assign on a network domain name server (DNS) or each node. To use the Typical Oracle Clusterware installation option, you must use static addresses.

Oracle Clusterware Networking (continued)

For the public network, each network adapter must support TCP/IP.

For the private network, the interconnect must support UDP or RDS (TCP for Windows) for communications to the database. Grid Interprocess Communication (GIPc) is used for Grid (Clusterware) interprocess communication. GIPC is a new common communications infrastructure to replace CLSC/NS. It provides a full control of the communications stack from the operating system up to whatever client library uses it. The dependency on network services (NS) prior to 11.2 is removed, but there is still backwards compatibility with existing CLSC clients (primarily from 11.1). GIPC can support multiple communications types: CLSC, TCP, UDP, IPC and of course the communication type GIPC.

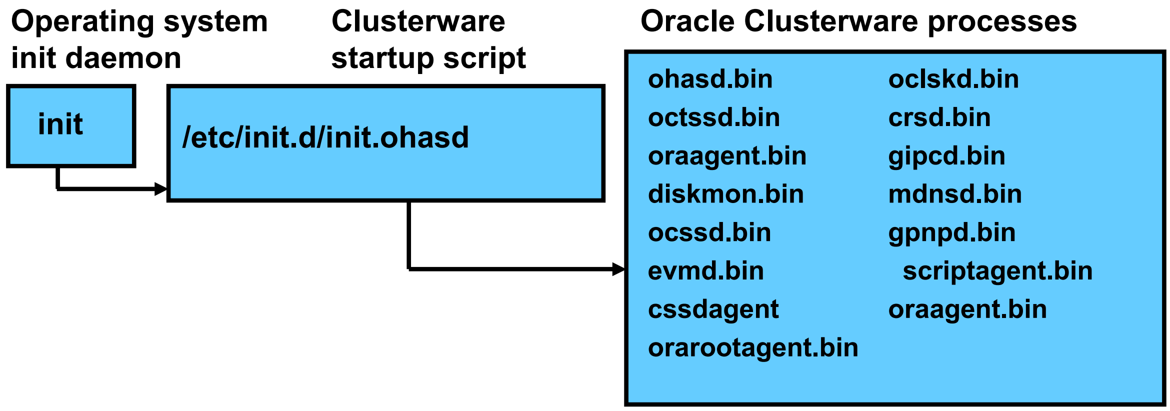
Use high-speed network adapters for the interconnects and switches that support TCP/IP. Gigabit Ethernet or an equivalent is recommended.

Each node in a cluster requires a supported interconnect protocol to support Cache Fusion and TCP/IP to support Clusterware polling. Token Ring is not supported for cluster interconnects on IBM AIX. Your interconnect protocol must be certified by Oracle for your platform.

Note: Cross-over cables are not supported for use with Oracle Clusterware interconnects.

Oracle Clusterware Startup

- Oracle Clusterware is started by the OS init daemon.



- Oracle Clusterware installation modifies the `/etc/inittab` file to restart `ohasd` in the event of a crash.

```
# cat /etc/inittab
..
h1:35:respawn:/etc/init.d/init.ohasd run >/dev/null 2>&1 </dev/null
```

ORACLE

Oracle Clusterware Startup

During the installation of Oracle Clusterware, the `init.ohasd` startup script is copied to `/etc/init.d`. The wrapper script is responsible for setting up environment variables and then starting the Oracle Clusterware daemons and processes.

The Oracle High Availability Services daemon (`ohasd`) is responsible for starting in proper order, monitoring, and restarting other local Oracle daemons, up through the `crsd` daemon, which manages clusterwide resources. When `init` starts `ohasd` on Clusterware startup, `ohasd` starts `orarootagent`, `cssdagent`, and `oraagent`. These processes then carry out the following tasks:

- `orarootagent` starts `crsd`.
  - `crsd` starts another `orarootagent` process responsible for root-owned CRS resources including the SCAN VIPS.
- `cssdagent` starts `cssd` (`ocssd`).
- `oraagent` starts `mdnsd`, `evmd`, ASM, `ctssd`, and `gpnpd`. `oraagent` also starts `gsd`, Oracle Notification Service (ONS), and the listeners.

Some of the high availability daemons will be running under the `root` user with real-time priority, and others will be running under the Clusterware owner with user-mode priorities after they are started. When a command is used to stop Oracle Clusterware, the daemons will be stopped, but the `ohasd` process will remain running.

Oracle Clusterware Process Architecture

Clusterware processes are organized into several component groups. They include:

Component	Processes	Owner
Cluster Ready Service (CRS)	`crsd`	`root`
Cluster Synchronization Service (CSS)	`ocssd,cssdmonitor, cssdagent`	`grid owner, root, root`
Event Manager (EVM)	`evmd, evmlogger`	`grid owner`
Cluster Time Synchronization Service (CTSS)	`octssd`	`root`
Oracle Notification Service (ONS)	`ons, eons`	`grid owner`
Oracle Agent	`oraagent`	`grid owner`
Oracle Root Agent	`orarootagent`	`root`
Grid Naming Service (GNS)	`gnsd`	`root`
Grid Plug and Play (GPnP)	`gpnpd`	`grid owner`
Multicast domain name service (mDNS)	`mdnsd`	`grid owner`

ORACLE

Oracle Clusterware Process Architecture

Oracle Clusterware comprises several processes that facilitate cluster operations. The Cluster Ready Service (CRS), Cluster Synchronization Service (CSS), Event Management (EVM), and Oracle Notification Service (ONS) components communicate with other cluster component layers in the same cluster database environment. These components are also the main communication links between Oracle Database, applications, and the Oracle Clusterware high availability components. In addition, these background processes monitor and manage database operations. The following list describes some major Oracle Clusterware background processes. The list includes components that are processes on Linux and UNIX, or services on Windows.

- **Cluster Ready Service (CRS):** Is the primary program for managing high availability operations in a cluster. The CRS process manages two types of CRS resources:
  - **Cluster resources:** A cluster resource is an Oracle Clusterware resource. Cluster resources are viewed, added, modified, or removed using the `crsctl` command.
  - **Local resources:** A local resource runs on every node in the cluster (no failover) and can be, for example, a listener, ASM, a disk group, or Oracle Notification Service (ONS).
- The CRS daemon (`crsd`) manages cluster resources based on configuration information that is stored in Oracle Cluster Registry (OCR) for each resource. This includes start, stop, monitor, and failover operations. The `crsd` process generates events when the status of a resource changes.

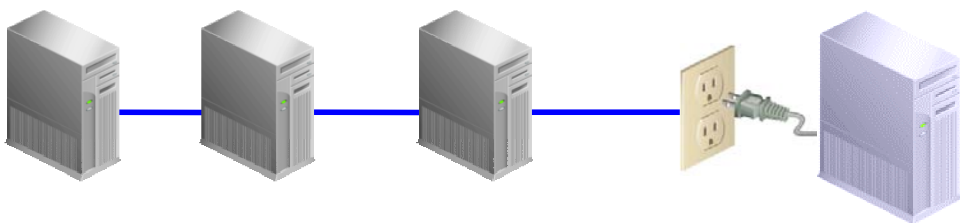
Oracle Clusterware Process Architecture (continued)

When you have Oracle RAC installed, the `crsd` process monitors the Oracle database instance, listener, and so on, and automatically restarts these components when a failure occurs. When a CRS resource fails, the CRS daemon attempts to restart it, if the resource is so configured. CRS fails the resource over to another node (again, if it is configured to do so) after exhausting restart attempts.

- **Cluster Synchronization Service (CSS):** Manages the cluster configuration by controlling which nodes are members of the cluster and by notifying members when a node joins or leaves the cluster. If you are using certified third-party clusterware, then CSS processes interfaces with your clusterware to manage node membership information. CSS has three separate processes: the CSS daemon (`ocssd`), the CSS Agent (`cssdagent`), and the CSS Monitor (`cssdmonitor`). The `cssdagent` process monitors the cluster and provides input/output fencing. This service formerly was provided by Oracle Process Monitor daemon (`oprocd`), also known as OraFenceService on Windows. A `cssdagent` failure results in Oracle Clusterware restarting the node.
- **Disk Monitor daemon (**`diskmon`**):** Monitors and performs input/output fencing for Oracle Exadata Storage Server. As Exadata storage can be added to any Oracle RAC node at any point in time, the `diskmon` daemon is always started when `ocssd` is started.
- **Event Manager (EVM):** Is a background process that publishes Oracle Clusterware events
- **Multicast domain name service (mDNS):** Allows DNS requests. The mDNS process is a background process on Linux and UNIX, and a service on Windows.
- **Oracle Grid Naming Service (GNS):** Is a gateway between the cluster mDNS and external DNS servers. The GNS process performs name resolution within the cluster.
- **Oracle Notification Service (ONS):** Is a publish-and-subscribe service for communicating Fast Application Notification (FAN) events
- `oraagent`**:** Extends clusterware to support Oracle-specific requirements and complex resources. It runs server callout scripts when FAN events occur. This process was known as RACG in Oracle Clusterware 11*g* Release 1 (11.1).
- **Oracle root agent** (`orarootagent`)**:** Is a specialized `oraagent` process that helps CRSD manage resources owned by root, such as the network, and the Grid virtual IP address
- **Cluster kill daemon** (`oclskd`)**:** Handles instance/node evictions requests that have been escalated to CSS
- **Grid IPC daemon** (`gipcd`)**:** Is a helper daemon for the communications infrastructure

Grid Plug and Play

- In previous releases, adding or removing servers in a cluster required extensive manual preparation.
- In Oracle Database 11*g* Release 2, GPnP allows each node to perform the following tasks dynamically:
  – Negotiating appropriate network identities for itself
  – Acquiring additional information from a configuration profile
  – Configuring or reconfiguring itself using profile data, making host names and addresses resolvable on the network
- To add a node, simply connect the server to the cluster and allow the cluster to configure the node.



Grid Plug and Play

With past releases, adding or removing servers in a cluster required extensive manual preparation. With the release of Oracle Database 11*g* Release 2, Grid Plug and Play (GPnP) reduces the costs of installing, configuring, and managing server nodes by using a Grid Naming Service within the cluster to allow each node to perform the following tasks dynamically:

- Negotiating appropriate network identities for itself
- Acquiring additional information it needs to operate from a configuration profile
- Configuring or reconfiguring itself using profile data, making host names and addresses resolvable on the network

As servers perform these tasks dynamically, adding and removing nodes simply requires an administrator to connect the server to the cluster and to allow the cluster to configure the node. Using Grid Plug and Play, and using best practices recommendations, adding a node to the database cluster is part of the normal server restart, and removing a node happens when a server is turned off. This removes many manual operations, reduces opportunity for error, and encourages configurations that can be changed more easily than those requiring fixed per-node configuration.

The best case uses ASM and Automatic Undo Management so there is no particular policy decision to make if an undo tablespace needs to be allocated for a newly identified database instance.

GPnP Domain

- The GPnP domain is a collection of nodes belonging to a single cluster served by the GPnP service:
  - Cluster name: **`cluster01`**
  - Network domain: **`example.com`**
  - GPnP domain: **`cluster01.example.com`**
- Each node participating in a GPnP domain has the following characteristics:
  - Must have at least one routable interface with connectivity outside of the GPnP domain for the public interface
  - A unique identifier that is unique within the GPnP domain
  - A personality affected by the GPnP profile, physical characteristics, and software image of the node

GPnP Domain

The GPnP domain is a collection of nodes served by the GPnP service. In most cases, the size of the domain is limited to the domain served by the multicast domain. The nodes in a multicast domain implicitly know which GPnP domain they are in by the public key of their provisioning authority. A GPnP domain is defined by the nodes in a multicast domain that recognize a particular provisioning authority, as determined by the certificate in their software images. A GPnP node refers to a computer participating in a GPnP domain. It must have the following:

IP Connectivity: The node must have at least one routable interface with connectivity outside of the GPnP domain. If the node has varying connectivity (multiple interfaces, subnets, and so on), the required binding (public, private, storage) needs to be identified in the GPnP profile. The physical connectivity is controlled by some outside entity and is not modified by GPnP.

Unique identifier: Associated with a node is a unique identifier, established through OSD. The ID is required to be unique within the GPnP domain, but is usefully unique globally within the largest plausible domain.

Personality: Characteristics of the node personality include:

- Cluster name
- Network classifications (public/private)
- Storage to be used for ASM and CSS
- Digital signatures
- The software image for the node, including the applications that may be 14

GPnP Components

Software image

- A software image is a read-only collection of software to be run on nodes of the same type.
- At a minimum, the image must contain:
  - An operating system
  - The GPnP software
  - A security certificate from the provisioning authority
  - Other software required to configure the node when it starts up

GPnP Components

A software image is a read-only collection of software to be run on nodes of the same type. At a minimum, it must contain an operating system, the GPnP software, the security certificate of the provisioning authority, and other software required to configure the node when it starts up. In the fullest form, the image may contain all application software to be run on the node, including the Oracle Database, iAS, and customer applications. Under GPnP, an image need not be specific to a particular node and may be deployed to as many nodes as needed.

An image is created by a provisioning authority through means not defined by GPnP. It may involve doing an installation on an example machine, then “scraping” the storage to remove node-dependent data.

An image is distributed to a node by the provisioning authority in ways outside the scope of GPnP—it may be a system administrator carrying CDs around, a network boot, or any other mechanism.

GPnP Profile

The `profile.xml` file:

```
$ cat GRID_HOME/gpnp/profiles/peer/profiles.xml
<?xml version="1.0" encoding="UTF-8"?><gpnp:GPnP-Profile Version="1.0"
xmlns="http://www.grid-pnp.org/2005/11/gpnp-profile" ...
xsi:schemaLocation="http://www.grid-pnp.org/2005/11/gpnp-profile gpnp-profile.xsd"
ProfileSequence="4" ClusterUId="2deb88730e0b5f1bffc9682556bd548e" ClusterName="cluster01"
PALocation=""><gpnp:Network-Profile><gpnp:HostNetwork id="gen" HostName="*"><gpnp:Network
id="net1" IP="192.0.2.0" Adapter="eth0" Use="public"/><gpnp:Network id="net2"
IP="192.168.1.0" Adapter="eth1"
Use="cluster_interconnect"/></gpnp:HostNetwork></gpnp:Network-Profile><orcl:CSS-Profile
id="css" DiscoveryString="+asm" LeaseDuration="400"/><orcl:ASM-Profile id="asm"
DiscoveryString="/dev/sd*" SPFile="+data/spfile.ora"/><ds:Signature
  <ds:SignedInfo>
    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
    <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <ds:Reference URI="">
      <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
          <InclusiveNamespaces xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"
PrefixList="gpnp orcl xsi"/>
        ...
      <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue>gIBakmtUNi9EVW/XQoE1mym3Bnw=</ds:DigestValue>
    ...
<ds:SignatureValue>cgw3yhP/2oEm5DJzdachtfDMbEr2RSfFFUlZujLemnOgsM...=</ds:SignatureValue>
```


GPnP Profile

A GPnP profile is a small XML file that is used to establish the correct global personality for otherwise identical images. A profile may be obtained at boot time from a remote location, or it may be burned permanently in an image. It contains generic GPnP configuration information and may also contain application-specific configuration data. The generic GPnP configuration data is used to configure the network environment and the storage configuration during bootup. Application parts of the profile may be used to establish application-specific personalities. The profile is security sensitive. It might identify the storage to be used as the root partition of a machine. A profile must be digitally signed by the provisioning authority, and must be validated on each node before it is used. Therefore, the image must contain the public certificate of the authority to be used for this validation. Profiles are intended to be difficult to change. They may be burned into read-only images or replicated to machines that are down when a change is made. Profile attributes defining node personality include:

- Cluster name
- Network classifications (public/private)
- Storage to be used for ASM and CSS
- Digital signature information

Changes made to a cluster and therefore the profile are replicated by the `gpnpd` daemon during installation, system boot, or when updated. These updates can be triggered by making changes with configuration tools such as `oifcfg`, `crsctl`, `asmca`, and so on.

Grid Naming Service

- GNS is an integral component of Grid Plug and Play.
- The only static IP address required for the cluster is the GNS virtual IP address.
- The cluster subdomain is defined as a delegated domain.

```
[root@my-dns-server ~]# cat /etc/named.conf
// Default initial "Caching Only" name server configuration
...
# Delegate to gns on cluster01
cluster01.example.com #cluster sub-domain# NS cluster01-
gns.example.com
# Let the world know to go to the GNS vip
cluster01-gns.example.com 192.0.2.155 # cluster GNS Address
```

- A request to resolve cluster01-scan.cluster01.example.com would be forwarded to the GNS on 192.0.2.155.
- Each node in the cluster runs a multicast DNS (mDNS) process.

ORACLE

Grid Naming Service

Employing Grid Naming Service (GNS) assumes that there is a DHCP server running on the public network with enough addresses to assign to the VIPs and single client access name (SCAN) VIPs. With GNS, only one static IP address is required for the cluster, the GNS virtual IP address. This address should be defined in the DNS domain. GNS sets up a multicast DNS (mDNS) server within the cluster, which resolves names in the cluster without static configuration of the DNS server for other node IP addresses.

The mDNS server works as follows: Within GNS, node names are resolved using link-local multicast name resolution (LLMNR). It does this by translating the LLMNR ".local" domain used by the multicast resolution to the subdomain specified in the DNS query. When you select GNS, an mDNS server is configured on each host in the cluster. LLMNR relies on the mDNS that Oracle Clusterware manages to resolve names that are being served by that host.

To use GNS, before installation, the DNS administrator must establish domain delegation to the subdomain for the cluster. Queries to the cluster are sent to the GNS listener on the GNS virtual IP address. When a request comes to the domain, GNS resolves it using its internal mDNS and responds to the query.

Single Client Access Name

- The single client access name (SCAN) is the address used by clients connecting to the cluster.
- The SCAN is a fully qualified host name located in the GNS subdomain registered to three IP addresses.

```
# dig @192.0.2.155 cluster01-scan.cluster01.example.com
...
;; QUESTION SECTION:
;cluster01-scan.cluster01.example.com. IN A
;; ANSWER SECTION:
cluster01-scan.cluster01.example.com. 120 IN A  192.0.2.244
cluster01-scan.cluster01.example.com. 120 IN A  192.0.2.246
cluster01-scan.cluster01.example.com. 120 IN A  192.0.2.245
;; AUTHORITY SECTION:
cluster01.example.com.  10800   IN      A       192.0.2.155
;; SERVER: 192.0.2.155#53(192.0.2.155)
```

- The SCAN provides a stable, highly available name for clients to use, independent of the nodes that make up the cluster.

ORACLE

Single Client Access Name

The single client access name (SCAN) is the address used by clients connecting to the cluster. The SCAN is a fully qualified host name (host name + domain) registered to three IP addresses. If you use GNS, and you have DHCP support, then the GNS will assign addresses dynamically to the SCAN.

If you do not use GNS, the SCAN should be defined in the DNS to resolve to the three addresses assigned to that name. This should be done before you install Oracle Grid Infrastructure. The SCAN and its associated IP addresses provide a stable name for clients to use for connections, independent of the nodes that make up the cluster.

SCANs function like a cluster alias. However, SCANs are resolved on any node in the cluster, so unlike a VIP address for a node, clients connecting to the SCAN no longer require updated VIP addresses as nodes are added to or removed from the cluster. Because the SCAN addresses resolve to the cluster, rather than to a node address in the cluster, nodes can be added to or removed from the cluster without affecting the SCAN address configuration.

During installation, listeners are created on each node for the SCAN IP addresses. Oracle Clusterware routes application requests to the cluster SCAN to the least loaded instance providing the service.

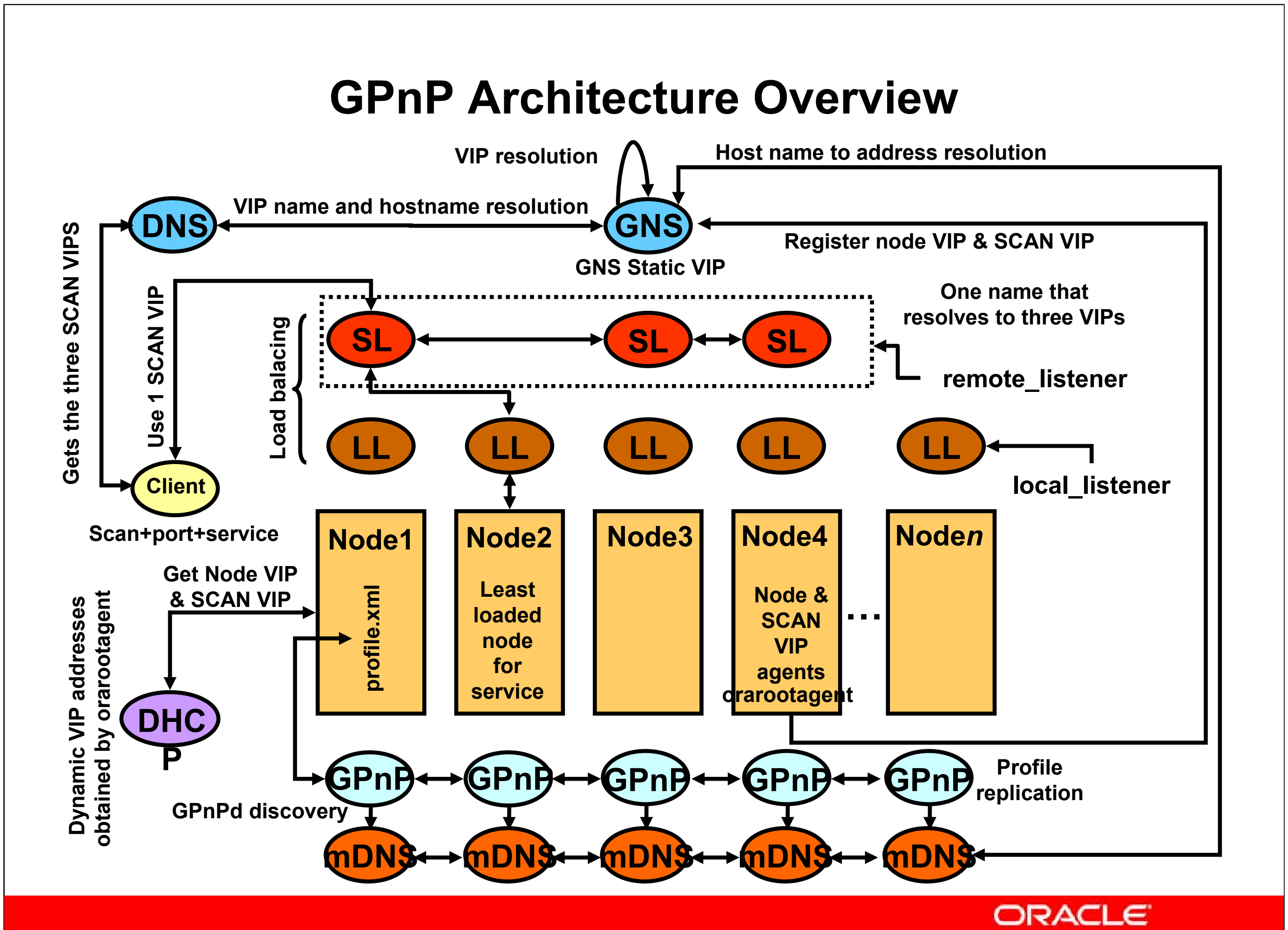
Single Client Access Name (continued)

SCAN listeners can run on any node in the cluster. SCANs provide location independence for the databases so that client configuration does not have to depend on which nodes run a particular database.

Oracle Database 11*g* Release 2 and later instances register with SCAN listeners only as remote listeners. Upgraded databases register with SCAN listeners as remote listeners, and also continue to register with all other listeners.

If you specify a GNS domain during installation, the SCAN defaults to *`clustername-scan.GNS_domain`*. If a GNS domain is not specified at installation, the SCAN defaults to *`clustername-scan.current_domain`*.

Note: `dig`: Domain Information Groper



GPnP Architecture Overview

GPnP Service

The GPnP service is collectively provided by all the GPnP agents. It is a distributed method of replicating profiles. The service is instantiated on each node in the domain as a GPnP agent. The service is peer-to-peer, there is no master process. This allows high availability because any GPnP agent can crash and new nodes will still be serviced. GPnP requires standard IP multicast protocol (provided by mDNS), to locate peer services. Using multicast discovery, GPnP locates peers without configuration. This is how a GPnP agent on a new node locates another agent that may have a profile it should use.

Name Resolution

A name defined within a GPnP domain is resolvable in the following cases:

- Hosts inside the GPnP domain use normal DNS to resolve the names of hosts outside of the GPnP domain. They contact their regular DNS service and proceed. They may get the address of their DNS server by global configuration, or by having been told by DHCP.
- Within the GPnP domain, host names are resolved using mDNS. This requires an mDNS responder on each node that knows the names and addresses used by this node, and operating system client library support for name resolution using this multicast protocol. Given a name, a client executes `gethostbyname`, resulting in an mDNS query. If the name exists, the responder on the node that owns the name will respond with the IP address. The client software may cache the resolution for the given time-to-live value.

GPnP Architecture Overview (continued)

- Machines outside the GPnP domain cannot resolve names in the GPnP domain by using multicast. To resolve these names, they use their regular DNS. The provisioning authority arranges the global DNS to delegate a sub-domain (zone) to a known address that is in the GPnP domain. GPnP creates a service called called GNS to resolve the GPnP names on that fixed address.
  The node on which the GNS server is running listens for DNS requests. On receipt, they translate and forward to mDNS, collect responses, translate, and send back to the outside client. GNS is "virtual" because it is stateless. Any node in the multicast domain may host the server. The only GNS configuration is global:
  - The address on which to listen on standard DNS port 53;
  - The name(s) of the domains to serviced.

  There may be as many GNS entities as needed for availability reasons. Oracle-provided GNS may use CRS to ensure availability of a single GNS provider.

SCAN and Local Listeners

When a client submits a connection request, the SCAN listener listening on a SCAN IP address and the SCAN port is contracted on the client's behalf. Because all services on the cluster are registered with the SCAN listener, the SCAN listener replies with the address of the local listener on the least-loaded node where the service is currently being offered. Finally, the client establishes connection to the service through the listener on the node where service is offered. All of these actions take place transparently to the client without any explicit configuration required in the client.

During installation, listeners are created on nodes for the SCAN IP addresses. Oracle Net Services routes application requests to the least loaded instance providing the service. Because the SCAN addresses resolve to the cluster, rather than to a node address in the cluster, nodes can be added to or removed from the cluster without affecting the SCAN address configuration.

How GPnP Works: Cluster Node Startup

1. IP addresses are negotiated for public interfaces using DHCP:
   – VIPs
   – SCAN VIPs
2. A GPnP agent is started from the nodes Clusterware home.
3. The GPnP agent either gets its profile locally or from one of the peer GPnP agents that responds.
4. Shared storage is configured to match profile requirements.
5. Service startup is specified in the profile, which includes:
   – Grid Naming Service for external names resolution
   – Single client access name (SCAN) listener

ORACLE

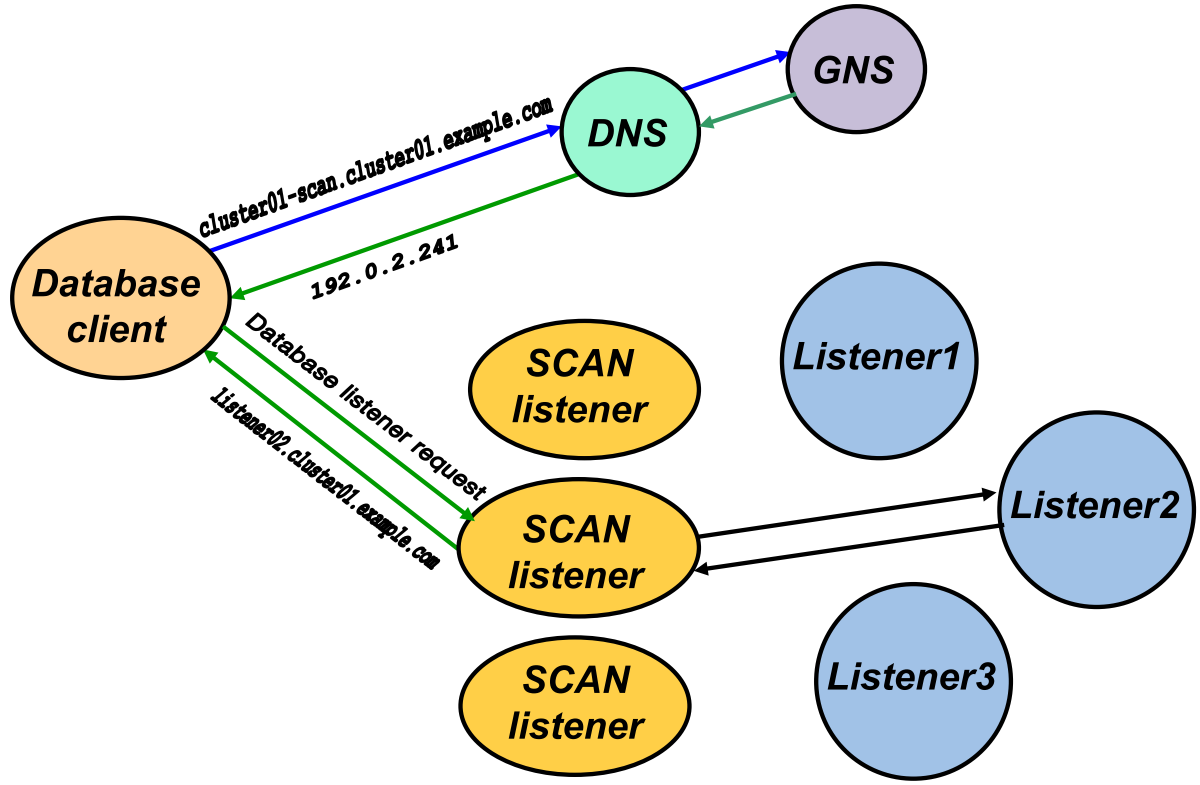
How GPnP Works: Cluster Node Startup

When a node is started in a GPnP environment:

- Network addresses are negotiated for all interfaces using DHCP
- The Clusterware software on the starting node starts a GPnP agent
- The GPnP agent on the starting node gets its profile locally or uses resource discovery (RD) to discover the peer GPnP agents in the grid. If RD is used, it gets the profile from one of the GPnP peers that responds.

  The GPnP agent acquires the desired network configuration from the profile. This includes creation of reasonable host names. If there are static configurations, they are used in preference to the dynamic mechanisms. Network interfaces may be reconfigured to match the profile requirements.
- Shared storage is configured to match the profile requirements
- System and service startup is done as configured in the image. In the case of RAC, the CSS and CRS system will then be started, which will form the cluster and bring up appropriate database instances. The startup of services may run down their own placeholder values, or may dynamically negotiate values rather than rely on fixed-up configurations. One of the services likely to be started somewhere is the GNS system for external name resolution. Another of the services likely to be started is an Oracle SCAN listener.

How GPnP Works: Client Database Connections



How GPnP Works: Client Database Connections

In a GPnP environment, the database client no longer has to use the TNS address to contact the listener on a target node. Instead, it can use the EZConnect method to connect to the database. When resolving the address listed in the connect string, the DNS will forward the resolution request to the GNS with the SCAN VIP address for the chosen SCAN listener and the name of the database service that is desired. In EZconnect syntax, this would look like:

`scan-name.cluster-name.company.com/ServiceName`, where the service name might be the database name. The GNS will respond to the DNS server with the IP address matching the name given; this address is then used by the client to contact the SCAN listener. The SCAN listener uses its connection load balancing system to pick an appropriate listener, whose name it returns to the client in an OracleNet Redirect message. The client reconnects to the selected listener, resolving the name through a call to the GNS.

The SCAN listeners must be known to all the database listener nodes and clients. The database instance nodes cross-register only with the known SCAN listeners, also sending them per-service connection metrics. The SCAN known to the database servers may be profile data or stored in OCR.

Summary

In this lesson, you should have learned how to:

- Explain the principles and purposes of clusters
- Describe the Oracle Clusterware architecture
- Describe how Grid Plug and Play affects Clusterware

G

RAC Concepts

Objectives

After completing this lesson, you should be able to:

- Explain the necessity of global resources
- Describe global cache coordination
- Explain object affinity and dynamic remastering

Benefits of Using RAC

- High availability: Surviving node and instance failures
- Scalability: Adding more nodes as you need them in the future
- Pay as you grow: Paying for only what you need today
- Key grid computing features:
  - Growth and shrinkage on demand
  - Single-button addition of servers
  - Automatic workload management for services

Benefits of Using RAC

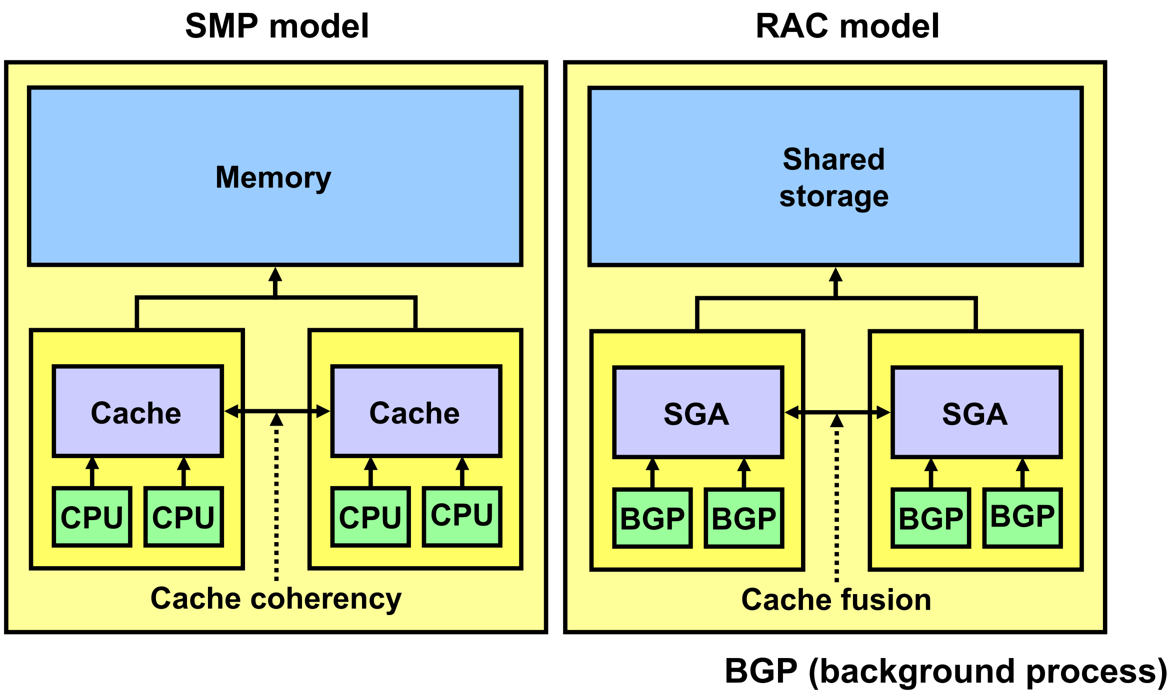
Oracle Real Application Clusters (RAC) enables high utilization of a cluster of standard, low-cost modular servers such as blades.

RAC offers automatic workload management for services. Services are groups or classifications of applications that comprise business components corresponding to application workloads. Services in RAC enable continuous, uninterrupted database operations and provide support for multiple services on multiple instances. You assign services to run on one or more instances, and alternate instances can serve as backup instances. If a primary instance fails, the Oracle server moves the services from the failed instance to a surviving alternate instance. The Oracle server also automatically load-balances connections across instances hosting a service.

RAC harnesses the power of multiple low-cost computers to serve as a single large computer for database processing, and provides the only viable alternative to large-scale symmetric multiprocessing (SMP) for all types of applications.

RAC, which is based on a shared-disk architecture, can grow and shrink on demand without the need to artificially partition data among the servers of your cluster. RAC also offers a single-button addition of servers to a cluster. Thus, you can easily provide or remove a server to or from the database.

Clusters and Scalability



Clusters and Scalability

If your application scales transparently on SMP machines, then it is realistic to expect it to scale well on RAC, without having to make any changes to the application code.

RAC eliminates the database instance, and the node itself, as a single point of failure, and ensures database integrity in the case of such failures.

Following are some scalability examples:

- Allow more simultaneous batch processes.
- Allow larger degrees of parallelism and more parallel executions to occur.
- Allow large increases in the number of connected users in online transaction processing (OLTP) systems.

Levels of Scalability

- Hardware: Disk input/output (I/O)
- Internode communication: High bandwidth and low latency
- Operating system: Number of CPUs
- Database management system: Synchronization
- Application: Design

ORACLE

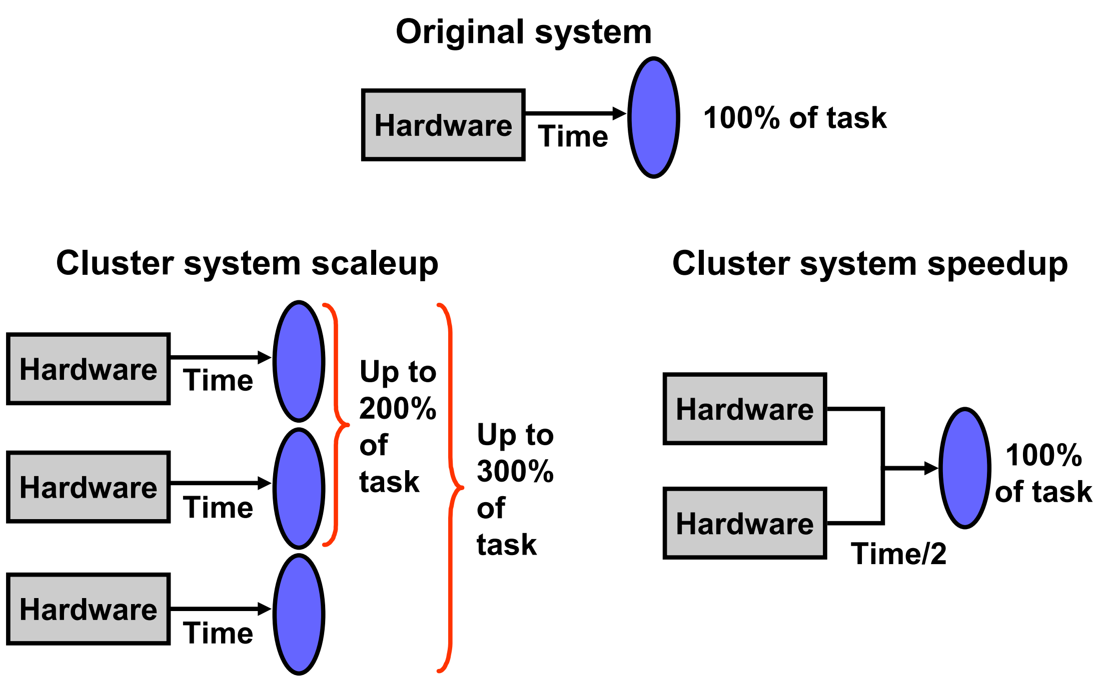
Levels of Scalability

Successful implementation of cluster databases requires optimal scalability on four levels:

- **Hardware scalability:** Interconnectivity is the key to hardware scalability, which greatly depends on high bandwidth and low latency.
- **Operating system scalability:** Methods of synchronization in the operating system can determine the scalability of the system. In some cases, potential scalability of the hardware is lost because of the operating system's inability to handle multiple resource requests simultaneously.
- **Database management system scalability:** A key factor in parallel architectures is whether the parallelism is affected internally or by external processes. The answer to this question affects the synchronization mechanism.
- **Application scalability:** Applications must be specifically designed to be scalable. A bottleneck occurs in systems in which every session is updating the same data most of the time. Note that this is not RAC specific and is true on single-instance system too.

It is important to remember that if any of the areas above are not scalable (no matter how scalable the other areas are), then parallel cluster processing may not be successful. A typical cause for the lack of scalability is one common shared resource that must be accessed often. This causes the otherwise parallel operations to serialize on this bottleneck. A high latency in the synchronization increases the cost of synchronization, thereby counteracting the benefits of parallelization. This is a general limitation and not a RAC-specific limitation.

Scaleup and Speedup



Scaleup and Speedup

Scaleup is the ability to sustain the same performance levels (response time) when both workload and resources increase proportionally:

Scaleup = (volume parallel) / (volume original)

For example, if 30 users consume close to 100 percent of the CPU during normal processing, then adding more users would cause the system to slow down due to contention for limited CPU cycles. However, by adding CPUs, you can support extra users without degrading performance.

Speedup is the effect of applying an increasing number of resources to a fixed amount of work to achieve a proportional reduction in execution times:

Speedup = (time original) / (time parallel)

Speedup results in resource availability for other tasks. For example, if queries usually take ten minutes to process and running in parallel reduces the time to five minutes, then additional queries can run without introducing the contention that might occur were they to run concurrently.

Speedup/Scaleup and Workloads

Workload	Speedup	Scaleup
OLTP and Internet	No	Yes
DSS with parallel query	Yes	Yes
Batch (mixed)	Possible	Yes

ORACLE

Speedup/Scaleup and Workloads

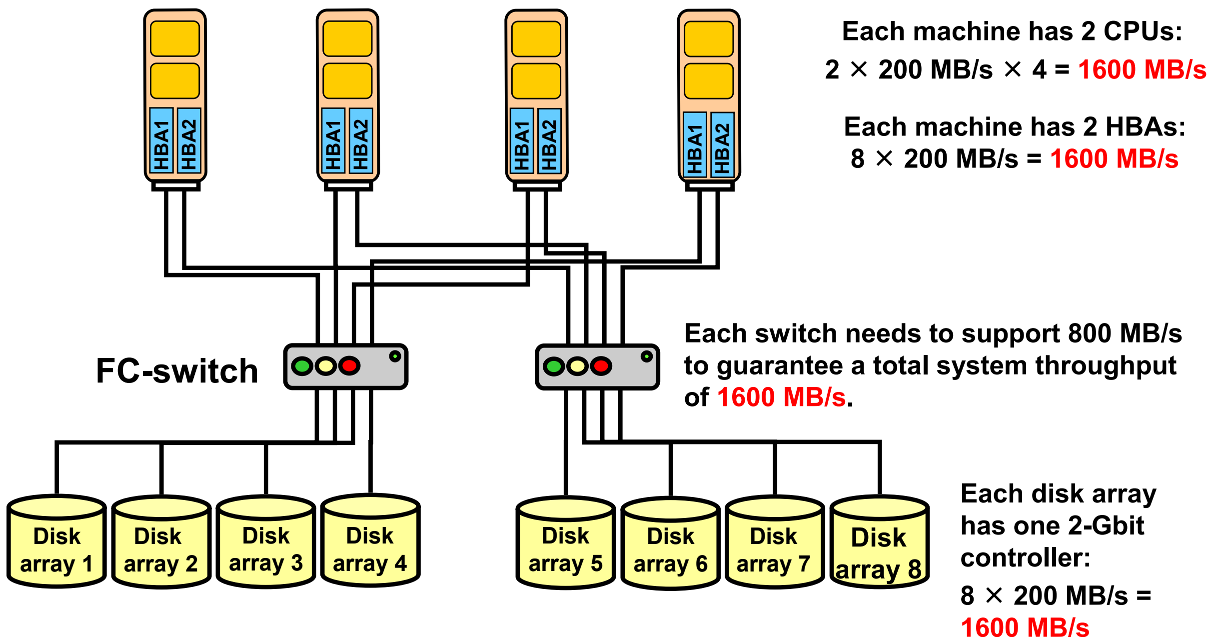
The type of workload determines whether scaleup or speedup capabilities can be achieved using parallel processing.

Online transaction processing (OLTP) and Internet application environments are characterized by short transactions that cannot be further broken down and, therefore, no speedup can be achieved. However, by deploying greater amounts of resources, a larger volume of transactions can be supported without compromising the response.

Decision support systems (DSS) and parallel query options can attain speedup, as well as scaleup, because they essentially support large tasks without conflicting demands on resources. The parallel query capability within the Oracle database can also be leveraged to decrease overall processing time of long-running queries and to increase the number of such queries that can be run concurrently.

In an environment with a mixed workload of DSS, OLTP, and reporting applications, scaleup can be achieved by running different programs on different hardware. Speedup is possible in a batch environment, but may involve rewriting programs to use the parallel processing capabilities.

I/O Throughput Balanced: Example



I/O Throughput Balanced: Example

To make sure that a system delivers the IO demand that is required, all system components on the IO path need to be orchestrated to work together.

The weakest link determines the IO throughput.

On the left, you see a high-level picture of a system. This is a system with four nodes, two Host Bus Adapters (HBAs) per node, two fibre channel switches, which are attached to four disk arrays each. The components on the IO path are the HBAs, cables, switches, and disk arrays. Performance depends on the number and speed of the HBAs, switch speed, controller quantity, and speed of disks. If any one of these components is undersized, the system throughput is determined by this component. Assuming you have a 2-GBit HBA, the nodes can read about 8 × 200 MB/s = 1.6 GBytes/s. However, assuming that each disk array has one controller, all 8 arrays can also do 8 × 200 MB/s = 1.6 GBytes/s. Therefore, each of the fibre channel switches also need to deliver at least 2 GBit/s per port, to a total of 800 MB/s throughput. The two switches will then deliver the needed 1.6 GBytes/s.

Note: When sizing a system, also take the system limits into consideration. For instance, the number of bus slots per node is limited and may need to be shared between HBAs and network cards. In some cases, dual port cards exist if the number of slots is exhausted. The number of HBAs per node determines the maximal number of fibre channel switches. And the total number of ports on a switch limits the number of HBAs and disk controllers.

Performance of Typical Components

Component	Throughput Performance: Theory (Bit/s)	Throughput Performance: Maximal Byte/s
HBA	½ Gbit/s	100/200 Mbytes/s
16 Port Switch	8 × 2 Gbit/s	1600 Mbytes/s
Fibre Channel	2 Gbit/s	200 Mbytes/s
Disk Controller	2 Gbit/s	200 Mbytes/s
GigE NIC	1 Gbit/s	80 Mbytes/s
Infiniband	10 Gbit/s	890 Mbytes/s
CPU		200–250 MB/s

ORACLE

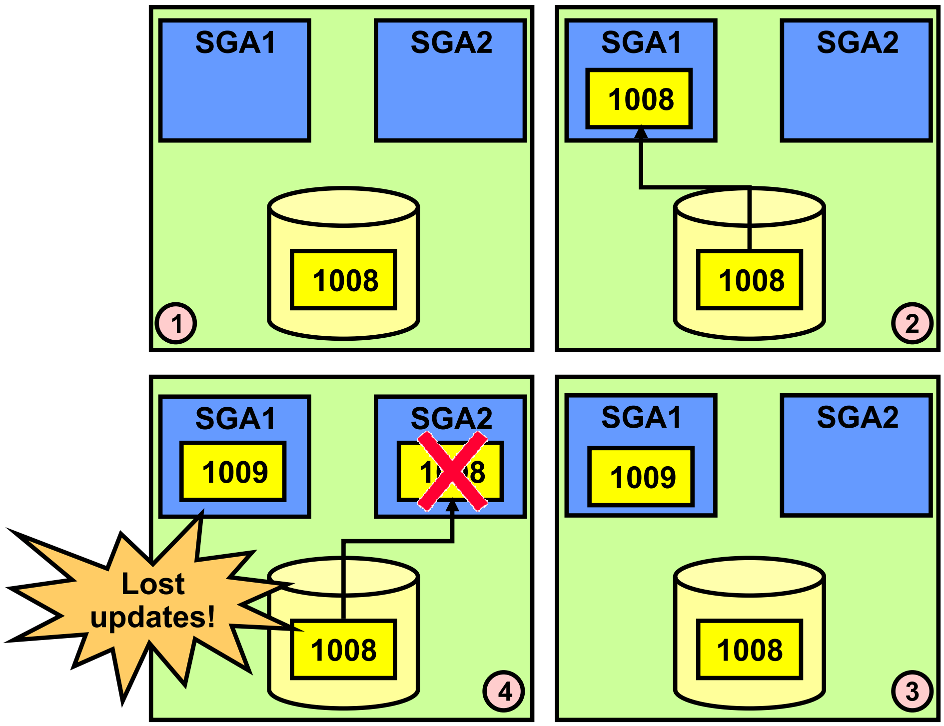
Performance of Typical Components

While discussing, people often confuse bits with bytes. This confusion originates mainly from the fact that hardware vendors tend to describe component's performance in bits/s whereas database vendors and customers describe their performance requirements in bytes/s.

The following is a list of common hardware components with their theoretical performance in bits/second and typical performance in bytes/second:

- HBAs come in 1 or 2 GBit per second with a typical throughput of 100 or 200 MB/s.
- A 16 Port Switch comes with sixteen 2-GBit ports. However, the total throughput is 8 times 2 Gbit, which results in 1600 Mbytes/s.
- Fibre Channel cables have a 2-GBit/s throughput, which translates into 200 MB/s.
- Disk Controllers come in 2-GBit/s throughput, which translates into about 200 MB/s.
- GigE has a typical performance of about 80 MB/s whereas Infiniband delivers about 160 MB/s.

Necessity of Global Resources



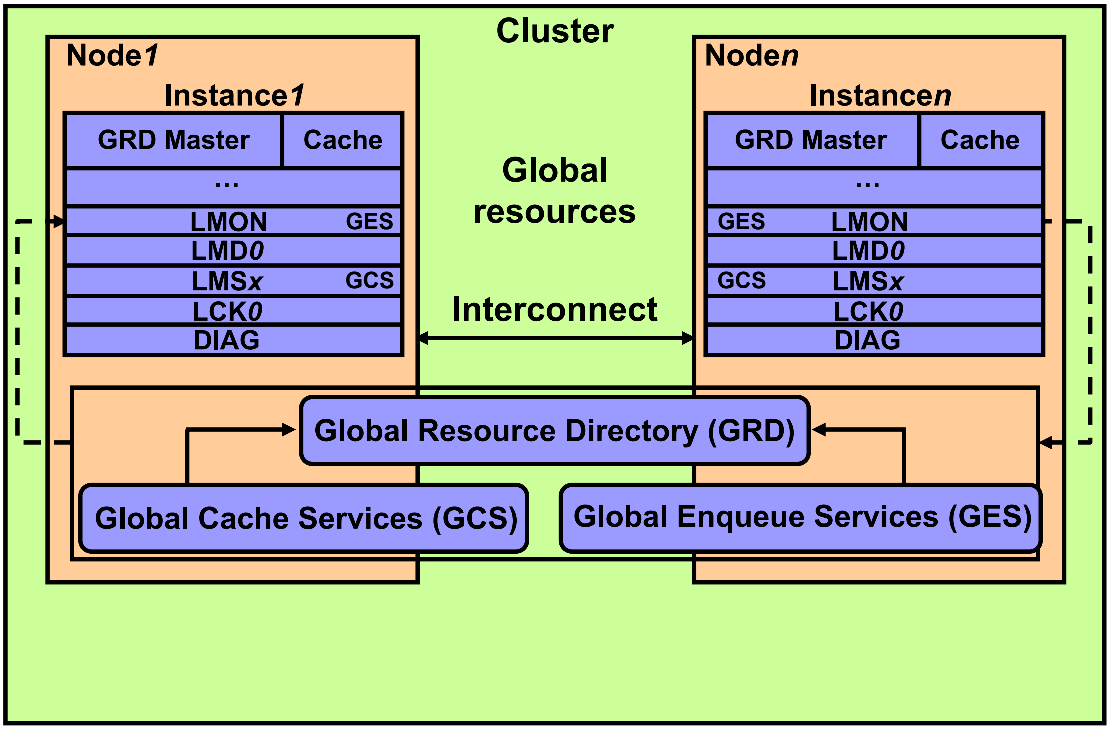
Necessity of Global Resources

In single-instance environments, locking coordinates access to a common resource such as a row in a table. Locking prevents two processes from changing the same resource (or row) at the same time.

In RAC environments, internode synchronization is critical because it maintains proper coordination between processes on different nodes, preventing them from changing the same resource at the same time. Internode synchronization guarantees that each instance sees the most recent version of a block in its buffer cache.

Note: The slide shows you what would happen in the absence of cache coordination. RAC prohibits this problem.

Global Resources Coordination



Global Resources Coordination

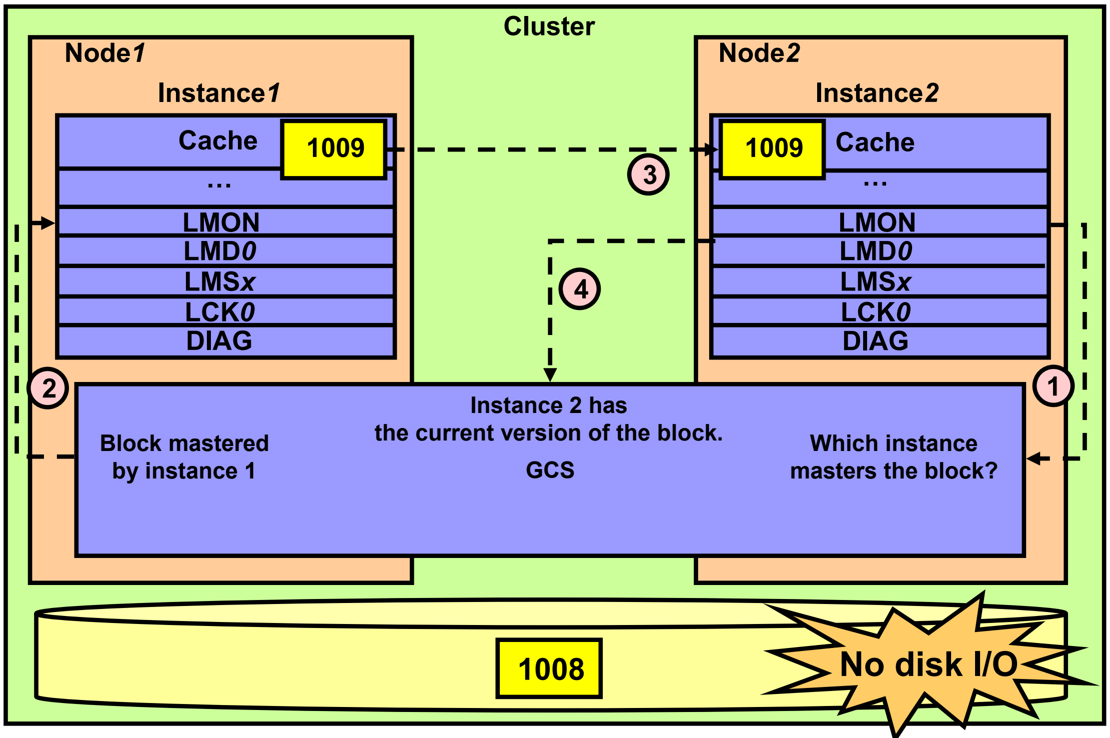
Cluster operations require synchronization among all instances to control shared access to resources. RAC uses the Global Resource Directory (GRD) to record information about how resources are used within a cluster database. The Global Cache Services (GCS) and Global Enqueue Services (GES) manage the information in the GRD.

Each instance maintains a part of the GRD in its System Global Area (SGA). The GCS and GES nominate one instance to manage all information about a particular resource. This instance is called the resource master. Also, each instance knows which instance masters which resource.

Maintaining cache coherency is an important part of a RAC activity. Cache coherency is the technique of keeping multiple copies of a block consistent between different Oracle instances. GCS implements cache coherency by using what is called the Cache Fusion algorithm.

The GES manages all non–Cache Fusion interinstance resource operations and tracks the status of all Oracle enqueuing mechanisms. The primary resources of the GES controls are dictionary cache locks and library cache locks. The GES also performs deadlock detection to all deadlock-sensitive enqueues and resources.

Global Cache Coordination: Example



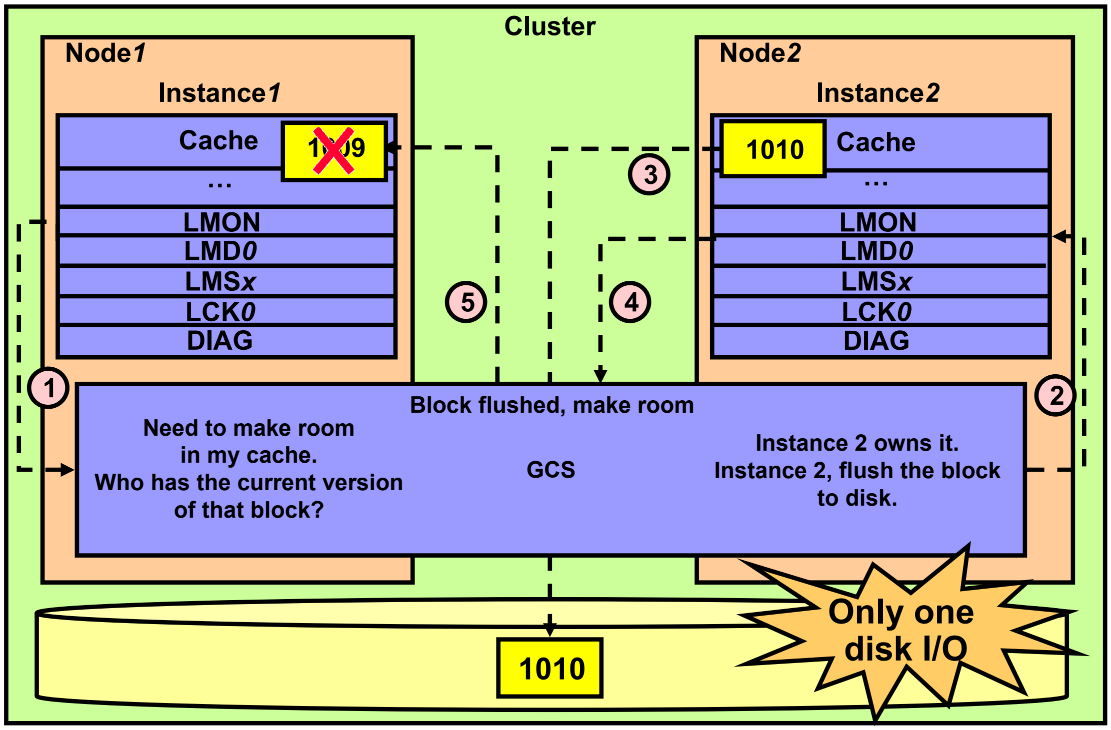
Global Cache Coordination: Example

The scenario described in the slide assumes that the data block has been changed, or dirtied, by the first instance. Furthermore, only one copy of the block exists clusterwide, and the content of the block is represented by its SCN.

1. The second instance attempting to modify the block submits a request to the GCS.
2. The GCS transmits the request to the holder. In this case, the first instance is the holder.
3. The first instance receives the message and sends the block to the second instance. The first instance retains the dirty buffer for recovery purposes. This dirty image of the block is also called a past image of the block. A past image block cannot be modified further.
4. On receipt of the block, the second instance informs the GCS that it holds the block.

Note: The data block is not written to disk before the resource is granted to the second instance.

Write to Disk Coordination: Example



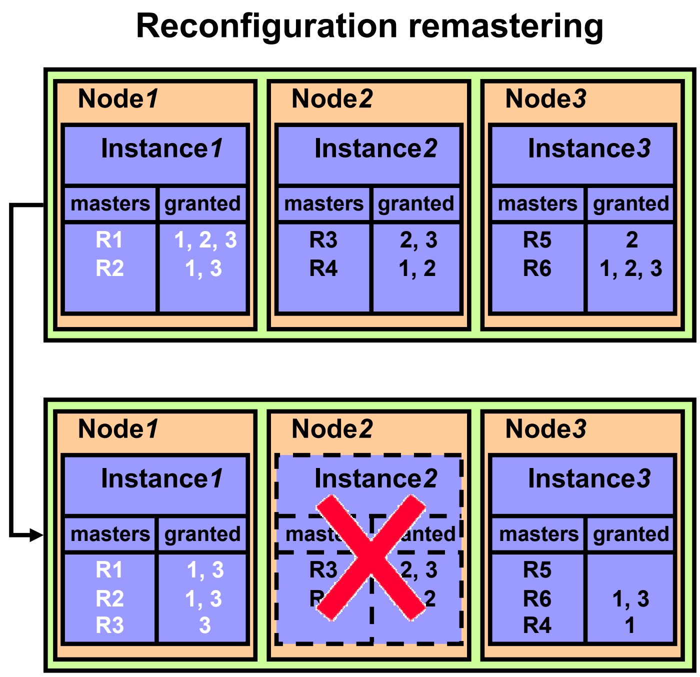
Write to Disk Coordination: Example

The scenario described in the slide illustrates how an instance can perform a checkpoint at any time or replace buffers in the cache as a response to free buffer requests. Because multiple versions of the same data block with different changes can exist in the caches of instances in the cluster, a write protocol managed by the GCS ensures that only the most current version of the data is written to disk. It must also ensure that all previous versions are purged from the other caches. A write request for a data block can originate in any instance that has the current or past image of the block. In this scenario, assume that the first instance holding a past image buffer requests that the Oracle server writes the buffer to disk:

1. The first instance sends a write request to the GCS.
2. The GCS forwards the request to the second instance, which is the holder of the current version of the block.
3. The second instance receives the write request and writes the block to disk.
4. The second instance records the completion of the write operation with the GCS.
5. After receipt of the notification, the GCS orders all past image holders to discard their past images. These past images are no longer needed for recovery.

Note: In this case, only one I/O is performed to write the most current version of the block to disk.

Dynamic Reconfiguration

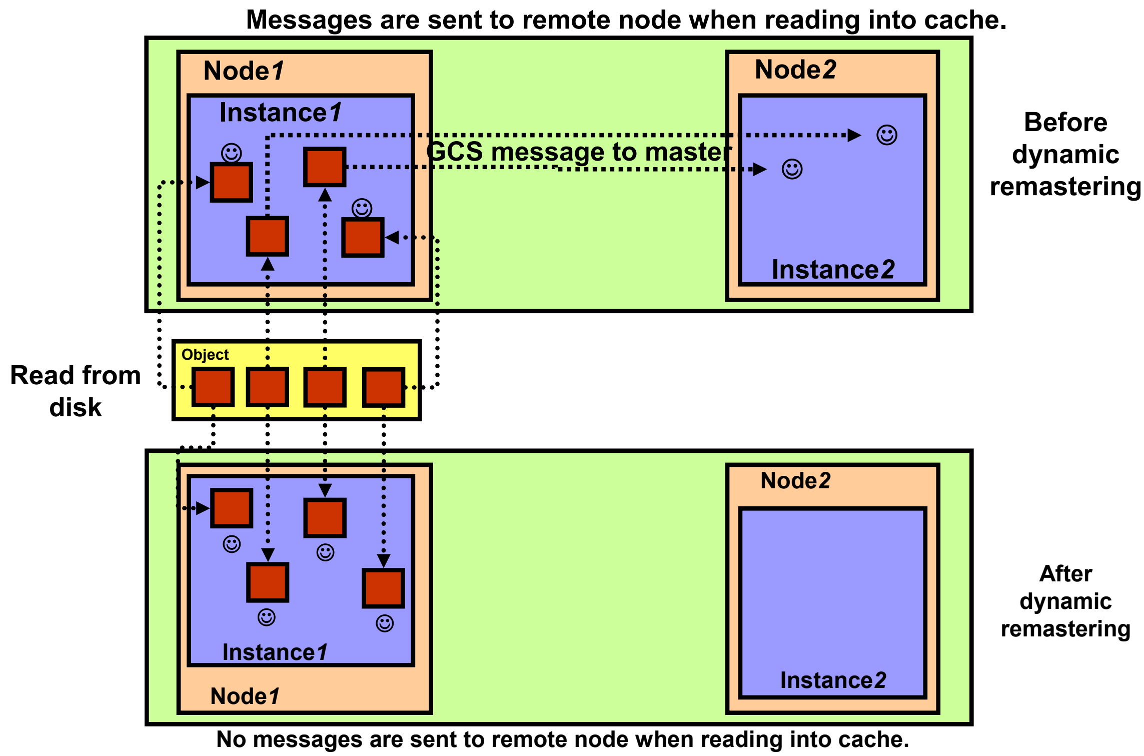


Dynamic Reconfiguration

When one instance departs the cluster, the GRD portion of that instance needs to be redistributed to the surviving nodes. Similarly, when a new instance enters the cluster, the GRD portions of the existing instances must be redistributed to create the GRD portion of the new instance.

Instead of remastering all resources across all nodes, RAC uses an algorithm called lazy remastering to remaster only a minimal number of resources during a reconfiguration. This is illustrated on the slide. For each instance, a subset of the GRD being mastered is shown along with the names of the instances to which the resources are currently granted. When the second instance fails, its resources are remastered on the surviving instances. As the resources are remastered, they are cleared of any reference to the failed instance.

Object Affinity and Dynamic Remastering



Object Affinity and Dynamic Remastering

In addition to dynamic resource reconfiguration, the GCS, which is tightly integrated with the buffer cache, enables the database to automatically adapt and migrate resources in the GRD. This is called dynamic remastering. The basic idea is to master a buffer cache resource on the instance where it is mostly accessed. In order to determine whether dynamic remastering is necessary, the GCS essentially keeps track of the number of GCS requests on a per-instance and per-object basis. This means that if an instance, compared to another, is heavily accessing blocks from the same object, the GCS can take the decision to dynamically migrate all of that object's resources to the instance that is accessing the object most.

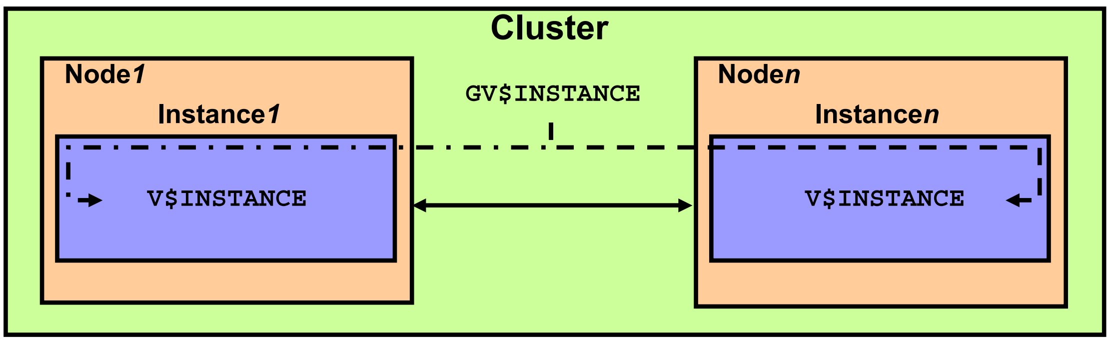
The upper part of the graphic shows you the situation where the same object has master resources spread over different instances. In that case, each time an instance needs to read a block from that object whose master is on the other instance, the reading instance must send a message to the resource's master to ask permission to use the block.

The lower part of the graphic shows you the situation after dynamic remastering occurred. In this case, blocks from the object have affinity to the reading instance which no longer needs to send GCS messages across the interconnect to ask for access permissions.

Note: The system automatically moves mastership of undo segment objects to the instance that owns the undo segments.

Global Dynamic Performance Views

- Retrieve information about all started instances
- Have one global view for each local view
- Use one parallel slave on each instance



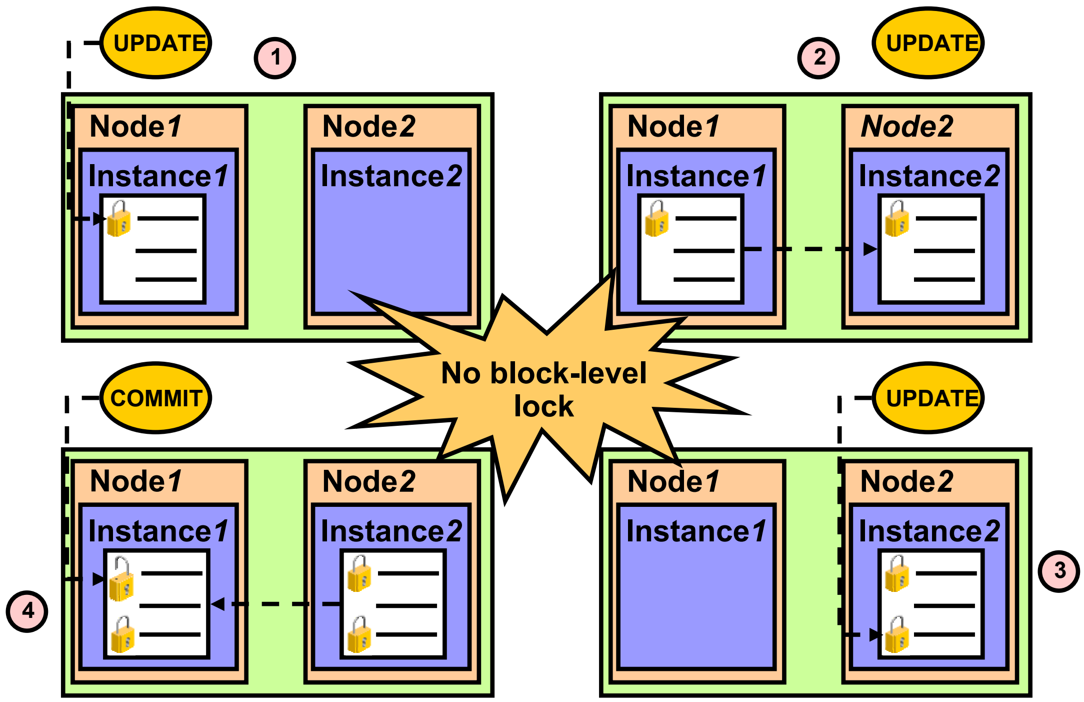
Global Dynamic Performance Views

Global dynamic performance views retrieve information about all started instances accessing one RAC database. In contrast, standard dynamic performance views retrieve information about the local instance only.

For each of the `V$` views available, there is a corresponding `GV$` view except for a few exceptions. In addition to the `V$` information, each `GV$` view possesses an additional column named `INST_ID`. The `INST_ID` column displays the instance number from which the associated `V$` view information is obtained. You can query `GV$` views from any started instance.

`GV$` views use a special form of parallel execution. The parallel execution coordinator runs on the instance that the client connects to, and one slave is allocated in each instance to query the underlying `V$` view for that instance.

Efficient Internode Row-Level Locking



Efficient Internode Row-Level Locking

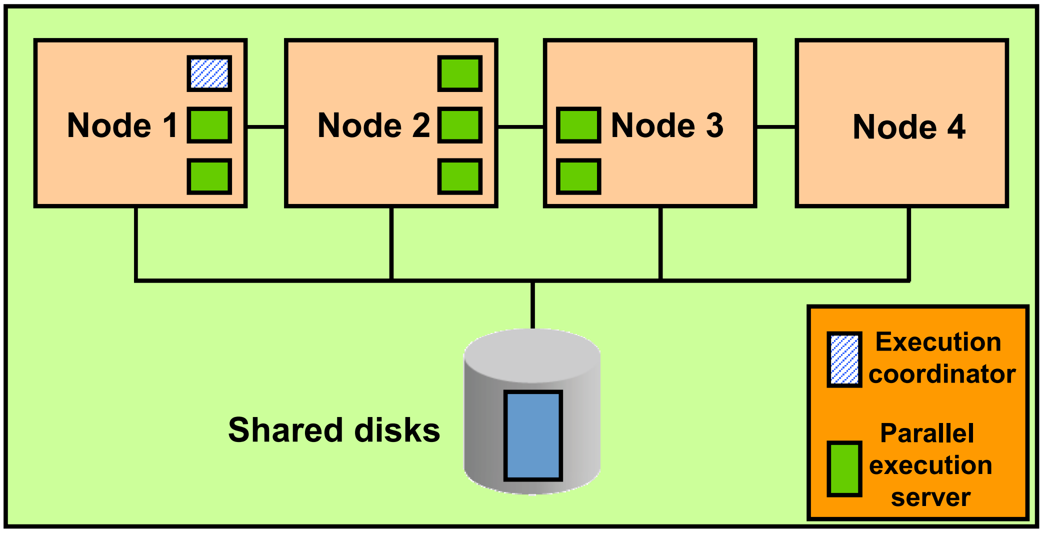
Oracle supports efficient row-level locks. These row-level locks are created when data manipulation language (DML) operations, such as `UPDATE`, are executed by an application. These locks are held until the application commits or rolls back the transaction. Any other application process will be blocked if it requests a lock on the same row.

Cache Fusion block transfers operate independently of these user-visible row-level locks. The transfer of data blocks by the GCS is a low-level process that can occur without waiting for row-level locks to be released. Blocks may be transferred from one instance to another while row-level locks are held.

GCS provides access to data blocks allowing multiple transactions to proceed in parallel.

Parallel Execution with RAC

Execution slaves have node affinity with the execution coordinator but will expand if needed.



Parallel Execution with RAC

Oracle's cost-based optimizer incorporates parallel execution considerations as a fundamental component in arriving at optimal execution plans.

In a RAC environment, intelligent decisions are made with regard to intranode and internode parallelism. For example, if a particular query requires six query processes to complete the work and six parallel execution slaves are idle on the local node (the node that the user connected to), then the query is processed by using only local resources. This demonstrates efficient intranode parallelism and eliminates the query coordination overhead across multiple nodes. However, if there are only two parallel execution servers available on the local node, then those two and four of another node are used to process the query. In this manner, both internode and intranode parallelism are used to speed up query operations.

In real-world decision support applications, queries are not perfectly partitioned across the various query servers. Therefore, some parallel execution servers complete their processing and become idle sooner than others. The Oracle parallel execution technology dynamically detects idle processes and assigns work to these idle processes from the queue tables of the overloaded processes. In this way, the Oracle server efficiently redistributes the query workload across all processes. Real Application Clusters further extends these efficiencies to clusters by enabling the redistribution of work across all the parallel execution slaves of a cluster.

Summary

In this lesson, you should have learned how to:

- Explain the necessity of global resources
- Describe global cache coordination
- Explain object affinity and dynamic remastering