Topic	**: Real Application Clusters**
Version	**: Oracle 10g Release 2**
Platform	**: RHEL AS 4 Update 4**

CONTENTS

Oracle 10g RAC

=========== * ============ * ============ * ============ * ============ * ==========

Oracle 10g - REAL APPLICATION CLUSTERS

----

1. REQUIREMENTS

HARDWARE:

Servers / Nodes	: Min 2 nodes
Processor	: PIV and above
RAM	: Min 1 GB
Hard Disk	: 15 GB for Operating System and 4 GB for Oracle Software
Network Cards	: 2 cards in each node (1 for Public IP, 1 for Private IP) : 1 Cross cable / Network switch for Private Interconnect
Shared Disk	: Shared Disk (FireWire ohci 1394 Compatible)/Any External Shared Disk(SAN,DAS,NAS)# : FireWire cables to connect shared disk from each node : One FireWire card in each node to connect to FireWire port

#The 1394 Open Host Controller Interface (Open HCI) Revision 1.1 is the latest implementation specification of the link layer protocol of the 1394 Serial Bus for use in PC systems. The link layer provides additional features to support the transaction and bus management layers. The 1394 Open HCI also includes DMA engines for high-performance data transfer and a host bus interface.

SOFTWARE:

Operating System	: RedHat Linux AS 4 Update 4
Cluster Software	: Oracle 10g Release 2 Clusterware

Database Software : Oracle 10g Release 2 Enterprise Edition for Linux
Modules : OCFS Modules:
ocfs2-2.6.9-42.ELsmp-1.2.7-1.el4.i686.rpm (292.2KB)
ocfs2console-1.2.7-1.el4.i386.rpm (169KB)
ocfs2-tools-1.2.7-1.el4.i386.rpm (1019KB)
These modules are required to configure OCFS (oracle clustered file system) on the shared storage
Network card drivers : To recognize network cards.

Note:
Details of installation can also be obtained from http://otn.oracle.com
OCFS modules can be downloaded from: http://oss.oracle.com/projects/ocfs/

---

2. OPERATING SYSTEM INSTALLATION

Partitions (select manual disk partitioning):

/	- 2 GB	- Fixed Size	- for root
/boot	- 100M	- Fixed Size	- for Boot Loader
/usr	- 6 to 8 GB	- Fixed Size	- for selected packages [select all packages while installation]
swap	- Twice of RAM	- Fixed Size	- swap / virtual memory
/tmp	- 1GB	- Fixed Size	- temporary files area
/var	- 500 M	- Fixed Size	- O/S log files
/opt	- 500 M	- Fixed Size	- for optional packages
/home	- Fill to maximum available size		- for storing user files

Host Name : rac1 and rac2 respectively for both nodes
Public IP : Configure now or Configure using DHCP
Firewall : Select "nofirewall"
SE Linux : Select "Disable"
Packages : Select "Customize" and select "Everything"

---

3. CONFIGURING OPERATING SYSTEM

---

a. Hostname and IP address
Use 'neat' command to assign IP address and Hostname

```
# neat
```

RAC1 **RAC2**

Public IP   : 192.9.200.144   192.9.200.146 Mask: 255.255.255.0
Private IP   : 10.0.0.4   10.0.0.5   Mask: 255.0.0.0
[Activate both network cards]

Verify your configurations by (on both nodes):

```
# ifconfig      --> for ip addresses
# hostname      --> for hostname
```

b. Setup the Hosts file (all nodes)

```
# vi /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       localhost.localdomain localhost

#PUBLIC IP
192.9.200.144   rac1    rac1.ambasa.com
192.9.200.146   rac2    rac2.ambasa.com

#PRIVATE IP
10.0.0.4        racp1   racp1.ambasa.com
10.0.0.5        racp2   racp2.ambasa.com

#VIRTUAL IP
192.9.200.70   racvip1  racvip1.ambasa.com
192.9.200.71   racvip2  racvip2.ambasa.com
```

Note: Ensure that the node names are **not** included for the loopback address in the `/etc/hosts` file. If the machine name is listed in the in the loopback address entry as below:

```
127.0.0.1 rac1 localhost.localdomain localhost
```

it will need to be removed as shown below:

```
127.0.0.1 localhost.localdomain localhost
```

If the RAC node name is listed for the loopback address, you will receive the following error during the RAC installation:

```
ORA-00603: ORACLE server session terminated by fatal error
```

Ping each other node to check the connectivity:

[rac**1**]# ping rac2
[rac**1**]# ping racp2

[rac**2**]# ping rac1
[rac**2**]# ping racp1

Note: Virtual IP will not ping until the clusterware is installed

c. Setup the kernel parameters (all nodes)

Edit /etc/sysctl.conf

```
[root@rac1 ~]# vi /etc/sysctl.conf
kernel.shmmax=570000000
kernel.sem=250 32000 100 128
net.core.rmem_default=262144
net.core.rmem_max=262144
net.core.wmem_default=262144
net.core.wmem_max=262144
fs.file-max=65536
net.ipv4.ip_local_port_range=1024 65000
kernel.hostname = rac1          # change the hostname in other nodes
kernel.domainname       = ambasa.com
```

Load the sysctl settings (without reboot) – on all nodes

```
[root@rac1 ~]# sysctl -p
```

Check the hostname and domain name (on all nodes):

```
[root@rac1 ~]# hostname
rac2
[root@rac1 ~]# domainname
ambasa.com
```

d. Check Firewall is disabled

```
[root@rac1 ~]# /etc/rc.d/init.d/iptables status
Firewall is stopped.
```

e. Disable SE Linux (if enabled) – on all nodes:

```
[root@rac1 root]# /usr/bin/system-config-securitylevel  &
```

f. Enable/Disable services (both nodes)

```
# chkconfig sendmail off        --> turn off the sendmail configuration
# chkconfig cups off            --> turn off the printer service (optional)
# chkconfig xinetd on--> for telnet service
# chkconfig telnet on           --> enable telnet
# chkconfig vsftpd on--> for ftp service
# service xinetd restart        --> restart the services
# service vsftpd restart
```

g. Apply compat patches (all nodes)

```
[root@rac1 root]# ls
compat-gcc-7.3-2.96.128.i386.rpm
compat-gcc-c++-7.3-2.96.128.i386.rpm
compat-libstdc++-7.3-2.96.128.i386.rpm
compat-libstdc++-devel-7.3-2.96.128.i386.rpm

[root@rac1 root]# rpm -ivh *.rpm --force
```

h. Configure Shared Storage: (For SAN)

```
# ping 192.9.200.133
# vi /etc/iscsi.conf
discovery address = 192.9.200.133 ----->>uncomment this line and specify the IP
:wq
# chkconfig iscsi on              -->for install & load iscsi Modules
# service iscsi restart (generate initiatorname file)
#cat /etc/initiatorname.iscsi          --->>>Do not edit this file
provide macid of shared storage


# fdisk -l                        --> list the disks and partitions
```

---

4. CONFIGURING USER and SHARED DISK

---

a. Create Oracle user and Directories

You will be using OCFS2 to store the files required to be shared for the Oracle Clusterware software. When using OCFS2, the UID of the UNIX user `oracle` and GID of the UNIX group `dba` should be identical on all machines in the cluster. If either the UID or GID are different, the files on the OCFS file system may show up as "unowned" or may even be owned by a different user.

Execute following commands in all node:

```
# groupadd -g 2000 dba
# groupadd -g 1000 oinstall
# useradd -u 600 -g oinstall -G dba -md /u01/oracle10g oracle10g
# passwd oracle10g
```

Create mount point for cluster files in all nodes:

```
# mkdir /home/cluster                          --> mount point for OCFS2
# chown -R oracle10g:oinstall /home/cluster       --> change ownership to Oracle user
```

b. Create partitions in the shared disk [FROM ONE NODE ONLY]

FileSystem	partition	size	mountpoint	
ocfs2	/dev/sdb1	1GB	/home/cluster	--> Cluster registry and Voting disk
ASM	/dev/sdb2	20G	+ORCL_DATA1	--> Oracle database files
ASM	/dev/sdb3	20G	+ORCL_DATA1	--> Oracle database files
ASM	/dev/sdb4	40G	+FLASH_RECOVERY_AREA	-->

Flash recovery area

```
# fdisk /dev/sdb                --> give the name of the device detected [ /dev/sda]

Command (m for help): p                          --> Type 'p' to print partitions

Disk /dev/sdb: 500 GB, 500000000 bytes
255 heads, 63 sectors/track, 16709 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start       End    Blocks   Id  System
                                                   --> Currently there are no
partitions

Command (m for help): n                          --> Type 'n' for creating new
partition
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-16709, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-16709, default 16709):
+1G
```

Create 3 Primary partitions and remaining as Extended partitions

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
e
Selected partition 4
First cylinder (34-16709, default 34):
Using default value 34
Last cylinder or +size or +sizeM or +sizeK (34-16709, default
16709):
Using default value 16709

Command (m for help): p

Disk /dev/sdb: 500 GB, 500000000 bytes
255 heads, 63 sectors/track, 16709 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sdb1             1        13    104391   83  Linux
/dev/sdb2            14        26    104422+  83  Linux
/dev/sdb3            27        33     56227+  83  Linux
/dev/sdb4            34     16709 133949970    5  Extended

Command (m for help): n                          --> Type 'n' for creating extended
```

partition

```
First cylinder (34-16709, default 34):
Using default value 34
Last cylinder or +size or +sizeM or +sizeK (34-16709, default
16709):+40G

Command (m for help): n
First cylinder (47-16709, default 47):
Using default value 47
Last cylinder or +size or +sizeM or +sizeK (47-16709, default
16709): +40G

....
....
Command (m for help): p                        --> Type 'p'  to print all partitions

Disk /dev/sdb: 500 GB, 500000000 bytes
255 heads, 63 sectors/track, 16709 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1         123      987966   83  Linux
/dev/sdb2             124        2556    19543072+  83  Linux
/dev/sdb3            2557        4989    19543072+  83  Linux
/dev/sdb4            4990       16709    94140900    5  Extended
/dev/sdb5            4990        9853    39070048+  83  Linux
/dev/sdb6            9854       14717    39070048+  83  Linux

Command (m for help): w                        --> Type 'w' to save and quit
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

After creating all required partitions, you should now update the kernel of the partition changes using the following syntax as the `root` user account:

[root@rac1 ~]# partprobe --> *issue this command from all nodes*

[root@rac1 root]# **fdisk -l** /dev/sdb --> *To check the updated list of partitions*

```
Disk /dev/sda: 500 GB, 500000000 bytes
255 heads, 63 sectors/track, 16709 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1               1         123      987966   83  Linux
/dev/sda2             124        2556    19543072+  83  Linux
/dev/sda3            2557        4989    19543072+  83  Linux
/dev/sda4            4990       16709    94140900    5  Extended
/dev/sda5            4990        9853    39070048+  83  Linux
/dev/sda6            9854       14717    39070048+  83  Linux
```

c. Setting Shell limits for Oracle User

To improve the performance of the software on Linux systems, Oracle recommends you increase the following shell limits for the oracle user:

Maximum number of open file descriptors	-> nofile	-> 65536 (hard limit)
Maximum number of processes available to a single user	-> nproc	-> 16384 (hard limit)

Execute the following from all nodes:

```
[root@rac1 ~]# vi /etc/security/limits.conf
 oracle10g soft nproc 2047
 oracle10g hard nproc 16384
 oracle10g soft nofile 1024
 oracle10g hard nofile 65536
```

d. Setting correct date and time

During the installation of Oracle Clusterware, the Database, and the Companion CD, the Oracle Universal Installer (OUI) first installs the software to the local node running the installer (i.e. linux1). The software is then copied remotely to all of the remaining nodes in the cluster (i.e. linux2). During the remote copy process, the OUI will execute the UNIX "`tar`" command on each of the remote nodes to extract the files that were archived and copied over. If the date and time on the node performing the install is greater than that of the node it is copying to, the OUI will throw an error from the "`tar`" command indicating it is attempting to extract files stamped with a time in the future. Ensure that the date and time of the all nodes are same (unless you are using Network Time). To set the date and time now, you can execute the following commands:

```
rac1# date -s "9/13/2007 23:00:00"
rac2# date -s "9/13/2007 23:00:20"
```

--> node 2 is greater than node 1 for safety

e. Configuring "hangcheck-timer"

Starting with Oracle9*i* Release 2 (9.2.0.2), the watchdog daemon has been deprecated by a Linux kernel module named `hangcheck-timer` which addresses availability and reliability problems much better. The `hang-check` timer is loaded into the Linux kernel and checks if the system hangs. It will set a timer and check the timer after a certain amount of time. There is a configurable threshold to `hang-check` that, if exceeded will reboot the machine.

The hangcheck-timer was normally shipped only by Oracle, however, this module is now included with Red Hat Linux AS starting with kernel versions 2.4.9-e.12 and higher

```
[root@rac1 ~]# find /lib/modules -name "hangcheck-timer.ko"
```

--> check the module presence

```
/lib/modules/2.6.9-22.EL/kernel/drivers/char/hangcheck-timer.ko
```

`hangcheck-tick`*: This parameter defines the period of time between checks of system health. The default value is 60 seconds; Oracle recommends setting it to 30 seconds.*
`hangcheck-margin`*: This parameter defines the maximum hang delay that should be tolerated before hangcheck-timer resets the RAC node. It defines the margin of error in seconds. The default value is 180 seconds; Oracle recommends setting it to 180 seconds.*

Set the hangcheck-timer settings in /etc/modprobe.conf (all nodes)

```
[root@rac1 ~]# vi /etc/modprobe.conf
options hangcheck-timer hangcheck_tick=30 hangcheck_margin=180
```

Add hangcheck-timer module in /etc/rc.local to probe it at every startup:

```
[root@rac1 ~]# vi /etc/rc.local
/sbin/modprobe hangcheck-timer
```

To test the hangcheck-timer module manually (before reboot):

```
[root@rac1 ~]# modprobe hangcheck-timer
[root@rac1 ~]# grep Hangcheck /var/log/messages | tail -2
May 29 11:40:35 rac1 kernel: Hangcheck: starting hangcheck timer 0.5.0 (tick is
30 seconds, margin is 180 seconds).
```

f. Configure racnodes for remote access:

Before you can install and use Oracle Real Application clusters, you must configure either secure shell (SSH) or remote shell (RSH) for the "oracle" UNIX user account on all cluster nodes. The goal here is to setup *user equivalence* for the "oracle" UNIX user account. User equivalence enables the "oracle" UNIX user account to access all other nodes in the cluster (running commands and copying files) without the need for a password.

`Using` **`RSH:`**

```
# vi /etc/hosts.equiv
+rac1
+rac2
+racp1
+racp2

# chkconfig rsh on
# chkconfig rlogin on

# which rsh
```

```
/usr/kerberos/bin/rsh


# mv /usr/kerberos/bin/rsh /usr/kerberos/bin/rsh.bak
# mv /usr/kerberos/bin/rlogin /usr/kerberos/bin/rlogin.bak
# mv /usr/kerberos/bin/rcp /usr/kerberos/bin/rcp.bak

# which rsh
/usr/bin/rsh

$ rcp a.txt rac2:
```

Using **SSH**:

Perform all the above steps (1 to 6) in the **all the nodes**.

1. su – oracle10g
2. mkdir .ssh
3. chmod -R 700 .ssh
4. cd .ssh
5. ssh-keygen -t rsa
6. ssh-keygen -t dsa

And perform the following steps from the **first node only.**

7. ssh rac1 cat /home/oracle10g/.ssh/id_rsa.pub >> authorized_keys
8. ssh rac1 cat /home/oracle10g/.ssh/id_dsa.pub >> authorized_keys
9. ssh rac2 cat /home/oracle10g/.ssh/id_rsa.pub >> authorized_keys
10. ssh rac2 cat /home/oracle10g/.ssh/id_dsa.pub >> authorized_keys
11. scp authorized_keys rac2:/home/oracle10g/.ssh/
12. scp known_hosts rac2:/home/oracle10g/.ssh/

Note: The steps 12 should not prompt for password

**--
---------------**

5. CONFIGURING OCFS2 (Oracle Clustered File System)

**--
---------------**

a. Check necessary packages for Oracle software

The following packages must be installed for Oracle 10g:
make-3.80-5
glibc-2.3.4-2.9
glibc-devel-2.3.4-2.9
glibc-headers-2.3.4-2.9
glibc-kernheaders-2.4-9.1.87

```
cpp-3.4.3-22.1
compat-db-4.1.25-9
compat-gcc-32-3.2.3-47.3
compat-gcc-32-c++-3.2.3-47.3
compat-libstdc++-33-3.2.3-47.3
compat-libstdc++-296-2.96-132.7.2
openmotif-2.2.3-9.RHEL4.1
setarch-1.6-1
```

Use “rpm” command to check the availability of package:

```
[root@rac1 ~]# rpm -qa | grep make-3.80-5
make-3.80-5
```

b. Install and Configure OCFS

OCFS Release 1 was released in 2002 to enable Oracle RAC users to run the clustered database without having to deal with RAW devices. The filesystem was designed to store database related files, such as data files, control files, redo logs, archive logs, etc.

Install the OCFS rpms (all nodes):

```
[root@rac1 OCFS_AS4U2]# ls
ocfs2-2.6.9-42.ELsmp-1.2.7-1.el4.i686.rpm  ocfs2-tools-1.2.7-1.el4.i386.rpm
 ocfs2console-1.2.7-1.el4.i386.rpm
[root@rac1 OCFS_AS4U4]# rpm -ivh *.rpm
Preparing...                ########################################### [100%]
   1:ocfs2-tools            ########################################### [ 33%]
   2:ocfs2-2.6.9-42.ELsmp   ########################################### [
67%]
   3:ocfs2console           ########################################### [100%]
```

Configure Cluster Nodes to OCFS (on all nodes):

[root@rac1 ~]# ocfs2console &
Cluster -> Configure Nodes -> “Node Configuration” window -> Add -> *Enter the hostname and ip address of all nodes (keeping port number unchanged)* -> “Apply” -> Close -> File -> Quit

[root@rac1 ~]# cat /etc/ocfs2/cluster.conf --> *to verify the configuration*

Understanding O2CB Service:

Before we can do anything with OCFS2 like formatting or mounting the file system, we need to first have OCFS2's cluster stack, O2CB, running (which it will be as a result of the configuration process performed above). The stack includes the following services:

- ***NM:*** *Node Manager that keep track of all the nodes in the cluster.conf*
- ***HB:*** *Heart beat service that issues up/down notifications when nodes join or leave the cluster*
- ***TCP:*** *Handles communication between the nodes*
- ***DLM:*** *Distributed lock manager that keeps track of all locks, its owners and status*
- ***CONFIGFS:*** *User space driven configuration file system mounted at /config*

- ***DLMFS:*** *User space interface to the kernel space DLM*

All of the above cluster services have been packaged in the *o2cb* system service (`/etc/init.d/o2cb`)
You can use the following commands to manage the o2cb services:

```
/etc/init.d/o2cb status                --> check status
/etc/init.d/o2cb load                  --> Load all ocfs modules
/etc/init.d/o2cb online ocfs2          --> Online the cluster we created: ocfs
/etc/init.d/o2cb offline ocfs2    --> Offline the cluster we created: ocfs
/etc/init.d/o2cb unload                --> Unload all ocfs modules
```

Configure O2CB to Start on Boot:

Before attempting to configure the on-boot properties:
REMOVE the following lines in `/etc/init.d/o2cb` (all nodes)

```
### BEGIN INIT INFO
# Provides: o2cb
# Required-Start:
# Should-Start:
# Required-Stop:
# Default-Start: 2 3 5
# Default-Stop:
# Description: Load O2CB cluster services at system boot.
### END INIT INFO
```

Reconfigure to implement the change (all nodes):

```
# chkconfig --del o2cb
# chkconfig --add o2cb
# chkconfig --list o2cb
o2cb          0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

Unload modules from all nodes (all nodes):

```
# /etc/init.d/o2cb offline ocfs2
# /etc/init.d/o2cb unload
```

Start configuration (all nodes):

```
#/etc/init.d/o2cb configure
Load O2CB driver on boot (y/n) [n]: y
Cluster to start on boot (Enter "none" to clear) [ocfs2]:
Specify heartbeat dead threshold (>=7) [7]: 600
```

Format the drive with OCFS2 File system (from Only ONE node):

Format from GUI utility `ocfs2console`, use the menu `[Tasks] ->` Select appropriate settings `-> [Format]`.
OR Command line [mkfs]:

```
# mkfs.ocfs2 -b 4K -C 32K -N 4 -L oradatafiles /dev/sdb1
```

where b -> Block size
C -> Cluster
N -> Nodes
L -> Label

Mount the formated slice (on all nodes) to the directory '/home/cluster' :

```
# mount  -t ocfs2 -o datavolume,nointr  /dev/sdb1 /home/cluster
```

Execute “mount” command to verify the mounted file system (all nodes):

```
[root@rac1 ~]# mount | grep /home/cluster
/dev/sdb1 on /home/cluster type ocfs2 (rw,_netdev,datavolume,nointr,heartbeat=local)
```

Add entry in /etc/fstab for auto mount at startup (all nodes):

```
/dev/sdb1               /home/cluster           ocfs2   _netdev,datavolume,nointr  0 0
```

Check permissions on the new file system (all nodes):

```
# ls -ld /home/cluster/
drwxr-xr-x  3 root root 4096 May 25 21:29 /home/cluster/
```

Change it to Oracle user's access (from both nodes) - all nodes:

```
# chown oracle10g:oinstall /home/cluster; chmod 775 /home/cluster/
# ls -ld /home/cluster/
drwxrwxr-x  3 oracle10g oinstall 4096 May 25 21:29 /home/cluster/
```

REBOOT ALL THE NODES

After reboot verify whether ocfs partition is mounted(all nodes):

```
[root@rac1 ~]# mount | grep /home/cluster
/dev/sdb1 on /home/cluster type ocfs2 (rw,_netdev,datavolume,nointr,heartbeat=local)
```

---

6. INSTALL AND CONFIGURE ORACLE CLUSTERWARE

Logout from root user (GUI) and Login to Oracle user

a. Download and extract the Oracle 10g Release 2 Clusterware (only ONE node)

```
[oracle10g@rac1 10gRAC]$ unzip 10201_clusterware_linux32.zip
[oracle10g@rac1 10gRAC]$ ls
10201_clusterware_linux32.zip
clusterware
[oracle10g@rac1 10gRAC]$ cd clusterware/
[oracle10g@rac1 clusterware]$ ls
cluvfy  doc  install  response  rpm  runInstaller  stage  upgrade  welcome.html
```

a. Use `cluvfy` utility to verify the prerequisites before installation

```
# cd clusterware
# ls
cluvfy  doc  install  response  rpm  runInstaller  stage  upgrade  welcome.html
# cd rpm
```

# rpm -ivh cluvqdisk-1.0.1-1.rpm --> *apply patch to use cluvfy utility in* ***all the nodes***

Verify the node connectivity (only ONE node):

```
[oracle10g@rac1 cluvfy]$
[oracle10g@rac1 cluvfy]$ ./runcluvfy.sh comp nodecon -n rac1,rac2 -verbose
```

Verify the prerequisites for CRS installation (only ONE node):

```
[oracle10g@rac1 cluvfy]$ ./runcluvfy.sh stage -pre crsinst -n rac1,rac2 -verbose
```

If all the above verifications complete successfully, then you can proceed with the CRS installation.

c. Invoke the Oracle Universal Installer (only ONE node)

```
[oracle10g@rac1 clusterware]$ ./runInstaller
```

1. Click Next
2. Choose path for OraInventory "/home/oracle10g/oraInventory", Choose Oracle group as "oinstall"
3. Home Name: OraCrs10g_home
   Path: /home/oracle10g/crs
4. Verify requirements and click "next"
5. Specify cluster configuration details:
   Cluster Name: crs
   Cluster nodes: rac1 racp1 racvip1
   rac2 racp2 racvip2
6. Verify the network interface usage:
   eth0 192.9.200.0 Public
   eth1 10.0.0.0 Private
7. Specify Oracle Cluster Registry Location:
   Location: /home/cluster/OCR1
   Mirror Location: /home/cluster/OCR2
8. Specify Voting Disk Location:
   Location: /home/cluster/Vote1
   Mirror1: /home/cluster/Vote2
   Mirror2: /home/cluster/Vote3
9. Click Install to start installation
10. Execute Configuration Scripts:
    Execute "orainstRoot.sh" - on all nodes as root user only

```
[root@rac1 ~]# /home/oracle10g/oraInventory/orainstRoot.sh
Changing permissions of /home/oracle10g/oraInventory to 770.
Changing groupname of /home/oracle10g/oraInventory to dba.
The execution of the script is complete

[root@rac1 ~]# /home/oracle10g/crs/root.sh
```

While executing root.sh in any of the remote node, if you get a message "eth0 is not public" or any

similar error, you need to execute the VIPCA (Virtual IP Configuration Assistant) manually.

Running vipca manually:

```
[root@rac2 ~]# sh /home/oracle10g/crs/bin/vipca
```

Enter the proper IP address of your VIP and its alias names, then click Finish to complete the configuration.

You can verify the pinging of Virtual IP address now:

```
[oracle10g@rac2 ~]$ ping racvip1

[oracle10g@rac1 ~]$ ping racvip2
```

Return to the " Execute Configuration Scripts" Screen and Click "OK"

11. Once the configurations are run successfully, click "Exit" to exit the installation

d. Post install verification (all nodes)

List the cluster nodes:

```
[oracle10g@rac1 ~]$ /home/oracle10g/crs/bin/olsnodes -n
rac1    1
rac2    2
```

Check oracle cluster auto-startup scripts:

```
[oracle10g@rac1 ~]$ ls -l /etc/init.d/init.*
-r-xr-xr-x  1 root root  1951 May 29 21:30 /etc/init.d/init.crs
-r-xr-xr-x  1 root root  4716 May 29 21:30 /etc/init.d/init.crsd
-r-xr-xr-x  1 root root 35396 May 29 21:30 /etc/init.d/init.cssd
-r-xr-xr-x  1 root root  3192 May 29 21:30 /etc/init.d/init.evmd
```

Check cluster ready services:

```
[oracle10g@rac1 ~]$ ps -ef | grep crs
```

Check cluster synchronization services:

```
[oracle10g@rac1 ~]$ ps -ef | grep css
```

Check the pinging of Virtual IP:

```
[oracle10g@rac1 ~]$ ping racvip2

[oracle10g@rac2 ~]$ ping racvip1
```

---

7. INSTALL ORACLE DATABASE SOFTWARE

Verify the prerequisites for RDBMS installation (only ONE node):

```
[oracle10g@rac1 cluvfy]$ cd clusterware/cluvfy

[oracle10g@rac1 cluvfy]$ ./runcluvfy.sh stage -pre dbinst -n rac1,rac2 -verbose
```

If all the above verifications complete successfully, then you can proceed with the CRS installation.

a. Download and extract the Oracle 10g Release 2 Database Software (one NODE only)

```
[oracle10g@rac1 10gRAC]$ unzip Ora10gSetup.zip
[oracle10g@rac1 10gRAC]$ cd database/
[oracle10g@rac1 database]$ ls
doc  install  response  runInstaller  stage  welcome.html
```

b. Invoke the Oracle Universal Installer (one NODE only)

```
[oracle10g@rac1 database]$ ./runInstaller
```

1. You can verify the cluster installation by clicking “installed products”. Click “next”
2. Choose “Enterprise Edition”
3. Choose Home details:
   Name: OraDb10g_home1
   Path: /home/oracle10g/oracle/product/10.2.0/db_1
4. Click “Select all” for installing in all clustered nodes
5. Verify the requirements and click “next”
6. Choose “Install database Software only”
7. Click “install” to start installation
8. Execute Configuration Scripts
   Execute “root.sh” in all nodes (one at a time) as root user only
   `[root@rac1 ~]# /home/oracle10g/oracle/product/10.2.0/db_1/root.sh`
   Once the scripts are run successfully, return to “Execute Configuration Scripts” window
   and click “ok”
9. Click “Exit” to exit the installation

c. Set the Oracle Environment

Edit the .bash_profile of oracle user (all nodes):

```
export ORACLE_BASE=/home/oracle10g
export ORACLE_HOME=$ORACLE_BASE/oracle/product/10.2.0/db_1
export ORA_CRS_HOME=$ORACLE_BASE/crs
export ORACLE_SID=orcl1                              --> change sid in other nodes
export PATH=$ORACLE_HOME/bin:$ORA_CRS_HOME/bin:$PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$ORA_CRS_HOME/lib
```

Execute the .bash_profile (all nodes):

```
[oracle10g@rac1 ~]$ .  .bash_profile
```

Verify the environment:

```
[oracle10g@rac1 ~]$ echo $ORACLE_HOME
/home/oracle10g/oracle/product/10.2.0/db_1
```

----

8. CONFIGURING DISKS FOR ASM

----

a. Configure disks for ASM with Standard Linux I/O

Edit /etc/sysconfig/rawdevices to map the devices with Linux raw files (all nodes)

```
[root@rac1 ~]# vi /etc/sysconfig/rawdevices
/dev/raw/raw2  /dev/sdb2     --> sda1 is not mapped as it used for OCFS2
/dev/raw/raw3  /dev/sdb3
/dev/raw/raw5  /dev/sdb5     --> sda4 is not mapped as it is extended partition
```

Restart the rawdevices service (all nodes):

```
[root@rac1 ~]# service rawdevices restart
```

```
#raw -qa        --->>to check the raw devices mapped with shared device
```

Change permissions to raw devices, so that the Oracle user has read and write access (all nodes):

```
[root@rac1 ~]# chown oracle10g:dba /dev/raw/raw2; chmod 600 /dev/raw/raw2
[root@rac1 ~]# chown oracle10g:dba /dev/raw/raw3; chmod 600 /dev/raw/raw3
[root@rac1 ~]# chown oracle10g:dba /dev/raw/raw4; chmod 600 /dev/raw/raw4
[root@rac1 ~]# chown oracle10g:dba /dev/raw/raw5; chmod 600 /dev/raw/raw5
```

Edit /etc/rc.local to add permission details to be assigned at every startup (all nodes):

```
[root@rac1 ~]# vi /etc/rc.local
chown oracle10g:dba /dev/raw/raw2; chmod 600 /dev/raw/raw2
chown oracle10g:dba /dev/raw/raw3; chmod 600 /dev/raw/raw3
chown oracle10g:dba /dev/raw/raw4; chmod 600 /dev/raw/raw4
chown oracle10g:dba /dev/raw/raw5; chmod 600 /dev/raw/raw5
```

----

9. CREATE ORACLE DATABASE

----

a. Configure Listeners on all nodes (as it would be required for creating database by DBCA)

Use “netca” and configure listeners and start the listener (from only ONE node)

```
[oracle10g@rac1 ~]$ netca
```

1. Choose “RAC” option
2. Choose “Listener configuration”
3. Choose “Add”
4. Add the name, protocol, and port details
5. Wait for listener to start in all nodes
   If listener does not start, you can manually start the listeners by using LSNRCTL command line utility
6. Click finish to exit

b. Configure ASM instance (using DBCA)

Invoke Database Configuration Assistant (DBCA) - from only ONE node

```
[oracle10g@rac1 ~]$ dbca
```

1. Choose “RAC” Options
2. Choose “Configure Automatic Storage Management”
3. Click “Select All” to select all the nodes to be configured
4. Specify the password for SYS user of ASM instance.
   Choose “Spfile” for creating parameter file. Specify the location of OCFS2 file system:
   /home/cluster/spfile+ASM.ora
5. Click “Ok” to create ASM instance
6. Initially, you'll not have any diskgroups created. Click “create new” to created diskgroups.
7. Give the diskgroup name, select disk paths required, specify the fail group name, and click “OK”
8. Please wait until the disk group is created.
9. Now, you can see the list the diskgroups created.
10. Similarly, you can create many diskgroups with the existing disks
11. Finally, click “finish” to exit ASM configuration

You can verify the asm instance (all nodes):

```
[oracle10g@rac1 admin]$ ps -ef | grep asm
```

You can login to asm instance:

```
[oracle10g@rac1 admin]$ export ORACLE_SID=+ASM1
[oracle10g@rac1 admin]$ sqlplus
SQL*Plus: Release 10.2.0.1.0 - Production on Wed May 30 09:47:57 2007
Copyright (c) 1982, 2005, Oracle. All rights reserved.
Enter user-name: /as sysdba

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, Real Application Clusters, OLAP and Data Mining options
```

```
SQL> select name from v$asm_diskgroup;
SQL> select group_number, disk_number, mount_status, header_status, state, name, failgroup from v$asm_disk;
SQL> select group_number, name, state, free_mb, offline_disks from v$asm_diskgroup;
```

c. Create database using DBCA

Invoke Database Configuration Assistant (DBCA) - from only ONE node

```
[oracle10g@rac1 ~]$ dbca
```

1. Choose “RAC” Options
2. Choose “Create database”
3. Click “Select All” to select all the nodes
4. Choose the type of database “General Purpose”
5. Specify the Global Database Name as “orcl”
6. Choose “Configure database with enterprise manager”
7. Specify the passwords for user accounts
8. Choose “Automatic storage management”
9. Select the Disk groups where the database has to be created
10. Choose “Use Oracle Managed Files” and edit the diskgroup name if you want
11. Choose whether to use Flash Recovery Area and Archiving to your database
12. Select whether to create sample schemas
13. Review the services
14. Choose Automatic or custom for memory management. Ensure that you have enough space for shared pool.
15. Review the files
16. Click “Finish”
17. Review the Initialization variables and click “OK” to start the database creation
18. Please wait until the database is created successfully
19. A summary is shown at the end for your information. Click “exit”
    (you can note the SID, SPFile path and OEM address)
20. The database gets restarted after clicking “exit” button
    You can visit OEM Home page by using the address
    (Like: http://rac1:1158/em)

---

10. COMMAND LINE UTILITIES

<u>Using command line utilities:</u>

Manual configuration of OEM dbconsole:

1. Ensure that you have listeners and password files created for your database and asm instances in all instances
2. Configure the repository for Enterprise Manager

```
$ emca -repos create
```

3. Configure the EM dbconsole

```
$ emca -config dbcontrol db
```

4. For any help in the syntax, you can use the following command

```
$ emca help=y
```

5. Start EM DBConsole during next startup:

```
$ emctl start dbconsole
```

Manual configuration of database services with srvctl:

Add ASM details:

```
$ srvctl add asm -n rac1 -i +ASM1 -o $ORACLE_HOME
$ srvctl add asm -n rac2 -i +ASM2 -o $ORACLE_HOME
```

```
$ srvctl enable asm -n rac1
$ srvctl enable adm -n rac2
```

Add database details:

```
$ srvctl add database -d orcl -o /home/oracle9i/OraHome1
$ srvctl add instance -d orcl -i orcl1 -n rac1
$ srvctl add instance -d orcl -i orcl2 -n rac2
```

Check the configuration:

```
$ srvctl config database -d orcl
```

Start or Stop database:

```
$ srvctl { start | stop } database -d orcl [ -o normal ]
```

Start or Stop instance:

```
$ srvctl { start | stop } instance -d orcl -i orcl1
```

Check configuration in the OCR:

```
$ crs_stat -t
```

Follow the steps (manual) to shutdown your servers:

```
$ emctl stop dbconsole               --> to stop the Database Console
$ srvctl stop database -d orcl       --> to stop the database in all the instances
$ srvctl status database -d orcl     --> to check the status
$ srvctl stop asm -n rac1            --> to stop the ASM in each node
$ srvctl stop asm -n rac2            --> to stop the ASM in each node
$ srvctl stop nodeapps -n rac1       --> to stop other utilities in each node
# crsctl stop crs                    --> to stop cluster ready services
```

Checklist at next startup:

- Switch on the shared disks
- Switch on the Servers
- Check for CSS and CRS Processes (all nodes)

  ```
      [oracle10g@rac1 ~]$ crsctl check crs
  ```
- Check the Pinging between nodes

  ```
      [oracle10g@rac2 ~]$ ping rac1
      [oracle10g@rac2 ~]$ ping racp1
      [oracle10g@rac2 ~]$ ping racvp1
  ```
- Check whether OCFS is mounted

  ```
      [oracle10g@rac2 ~]$ mount | grep /home/cluster
  ```
- Start the Node applications in all nodes

  ```
      [oracle10g@rac1 ~]$ srvctl status nodeapps -n rac1
              VIP is running on node: rac1
              GSD is not running on node: rac1
              Listener is not running on node: rac1
              ONS daemon is running on node: rac1

      [oracle10g@rac1 ~]$ srvctl start nodeapps -n rac1
      [oracle10g@rac1 ~]$ srvctl start nodeapps -n rac2
  ```

- Start the ASM instance in all nodes

  ```
    [oracle10g@rac1 ~]$ srvctl start asm -n rac1
    [oracle10g@rac1 ~]$ srvctl start asm -n rac2
  ```

- Start the Database instance from one node

  ```
    [oracle10g@rac1 ~]$ srvctl start database -d orcl
  ```

- Start the Enterprise Manager DB Console on all nodes

  ```
    [oracle10g@rac1 ~]$ emctl start dbconsole
    [oracle10g@rac2 ~]$ emctl start dbconsole
  ```

Configure Network interfaces:

```
$ oifcfg getif -node rac1
$ oifcfg getif -global

$ oifcfg setif <interface-name>/<subnet>:<cluster_interconnect|public>
```

Note on Parameter file:

Parameter Types:

Identical across instances

- db_name
- compatible
- cluster_database
- control_file
- db_block_size

Unique across instances

- instance_name
- instance_number
- rollback_segments
- thread
- undo_tablespace

Multi-Valued Parameters

- fast_start_mttr_target
- instance_groups (for parallel query operations)

Spfile Features:

- Oracle recommends to use spfile
- Easier to use and manage
- Single, central location
- Parameters are applicable to all instances
- It permits dynamic changes
- Persistent
- Can specify common values and specific values
  - *.open_cursors=300
  - racins1.open_cursors=200

Tuning considerations:

Statistics / events to look for:

```
SQL> select * from gv$sysstat where name like '%gcs %';
```

This will give you a result set with specific attention to GCS messages sent across the nodes. If this value is inconsistent across nodes or if huge differences are apparent then it might be time to investigate.

===
=========

The following query can also be run to monitor the average cr block receive time since the last startup (should be less than 15):

```
set numwidth 20
column "AVG CR BLOCK RECEIVE TIME (ms)" format 9999999.9
select b1.inst_id, b2.value "GCS CR BLOCKS RECEIVED",
b1.value "GCS CR BLOCK RECEIVE TIME",
((b1.value / b2.value) * 10) "AVG CR BLOCK RECEIVE TIME (ms)"
from gv$sysstat b1, gv$sysstat b2
where b1.name = 'global cache cr block receive time' and
b2.name = 'global cache cr blocks received' and b1.inst_id = b2.inst_id ;
```

===
=========

When a consistent read buffer cannot be found in the local cache, an attempt is made to find a usable version in another instance. There are 3 possible outcomes, depending on whether any instance in the cluster has the requested data block cached or not:

a) A cr block is received (i.e. another instance found or managed to produce the wanted version). The "global cache cr blocks received" statistic is incremented.

b) No other instance has the block cached and therefor the requesting instance needs to read from disk, but a shared lock will be granted to the requestor The "global cache gets" statistic is incremented

c) RAC Only --> A current block is received (the current block is good enough for the query). The " global cache current blocks received" statistic is incremented.

===
=========

Querying the gv$ges_traffic_controller or
gv$dlm_traffic_controller views, you may find that the TCKT_AVAIL shows '0'. To
find out the available network buffer space we introduce the concepts of tickets. The maximum number of tickets available is a function of the network send buffer size. In the case of lmd and lmon, they always buffer their messages in case of ticket unavailability. A node relies on messages to come back from the remote node to release tickets for reuse.

===
=========

1. Tuning UDP
Linux (edit files)

```
        /proc/sys/net/core/rmem_default
        /proc/sys/net/core/rmem_max
```

```
/proc/sys/net/core/wmem_default
/proc/sys/net/core/wmem_max
```

2. Use netstat command to check the if the interconnect is slow, busy, or faulty, you can look for dropped packets,retransmits, or cyclic redundancy check errors (CRC)

3. Use of Backup interconnect by setting parameter:
CLUSTER_INTERCONNECTS

4. Choose Locally managed tablespaces with Automatic Segment Space Management