Using Network Appliance^{TM} Snapshot^{TM} and FlexClone® with Oracle® Cluster File System in an Oracle Environment

Eesha Pathak, Network Appliance Inc.

December 2007 | TR-3605

Table of Contents

1. Introduction

Oracle Cluster File System 2 (OCFS2) is an Oracle open-source cluster file system, which presents a consistent file image across all servers in the cluster. The OCFS2 is specifically designed for Linux™ to alleviate the need for managing raw devices.

This technical report documents installation and configuration of OCFS2 in an Oracle Database Real Application Clusters (RAC) environment and the use of Network Appliance Snapshot and FlexClone technology for backing up and creating Clone Databases on an OCFS2 file system.

2. Assumptions

This technical report assumes readers are familiar with the:

- Oracle Database 10*g* Release 2 RAC concepts
- Operation of Red Hat Linux operating systems
- Operation of Network Appliance storage systems, operation of storage area networks (SANs) over Fiber Channel
- General knowledge in networking.

3.Snapshot Copies and FlexClone Overview

A Snapshot copy is an online read-only copy of a volume. Typically a Snapshot copy only takes a few seconds to create (usually less than one second) regardless of the size of the volume or the level of activity on the NetApp storage system. After a Snapshot copy is created, changes to data objects are reflected in updates to the current version of the objects, as if Snapshot copies did not exist. Meanwhile, the Snapshot version of the data remains completely stable. A NetApp Snapshot copy incurs no performance overhead.

A Snapshot copy can be used as an online backup capability, allowing users to recover their own files. A Snapshot copy also simplifies backup to tape. Since a Snapshot copy is a read-only copy of the entire file system, it allows self-consistent backup from an active system. Instead of taking the system offline, the system administrator can make a backup to tape of a recently created Snapshot copy.

The process of creating Snapshot backups in the SAN environment differs from the NAS environment in one very fundamental way: In the SAN environment, the storage controller does not control the state of the file system. For this reason, Snapshot must be initiated from the host after the appropriate operations have been performed to ensure that a consistent file system image is obtained in the Snapshot backup.

Starting with Data ONTAP® 7G, storage administrators have access to a powerful new feature that instantly creates clones of a flexible volume (FlexVol® volume). A FlexClone volume is a writable point-in-time image of a FlexVol volume or another FlexClone volume. FlexClone volumes add a new level of agility and efficiency to storage operations. They take only a few seconds to create, and are created without interrupting access to the parent FlexVol volume. FlexClone volumes use space very efficiently, leveraging the Data ONTAP architecture to store only data that changes between the parent and clone. This is cost efficient, saves space and energy. In addition to these benefits, clone volumes have the same high performance as other kinds of volumes.

Conceptually, FlexClone volumes are advantageous for situations where testing or development occurs, situations where progress is made by locking in incremental improvements, and situations where data is required to be distributed in a changeable form without endangering the integrity of the original.

4.Oracle Cluster File System (OCFS2) Overview

OCFS2 is the Oracle open-source file system available on Linux platforms. This is an extent-based (an extent is a variable contiguous space) file system that is currently intended for Oracle data files and Oracle RAC. Unlike the previous release (OCFS), OCFS2 is a general purpose file system that can be used for shared Oracle home installations, making management of Oracle RAC installations even easier. In terms of the file interface provided to applications, OCFS2 balances performance and manageability by providing functionality that is in between the functionality provided by raw devices and typical file systems. OCFS2 provides higher-order, more manageable file interfaces, while retaining the performance of raw devices. In this respect, the OCFS2 service can be thought of as a file system-like interface to raw devices. At the same time, the cluster features of OCFS go well beyond the functionality of a typical file system.

OCFS2 files can be shared across multiple nodes on a network so that the files are simultaneously accessible by all the nodes, which is essential in RAC configurations. For example, sharing data files allows media recovery in case of failures, as all the data files (archive log files) are visible from the nodes that constitute the RAC cluster. Beyond clustering features and basic file service, OCFS2 provides a number of manageability benefits (for example, easy resizing data files and partitions) and comes with a set of tools to manage OCFS2 files.

5. The Server Environment

In this report and testing environment, the servers run the Red Hat Enterprise Linux 4 Update 5 operating system. The Oracle Cluster File System 2 version used is 1.5.2-1 (hereafter referred to as OCFS2). This is a certified configuration, and as such the components presented in this document have to be used in the same combination to support all parties involved. The only exception is the application of certain patches (as defined and required by all the vendors in this configuration). Two NetApp storage systems are configured in cluster to operate in a SAN (FCP) environment.

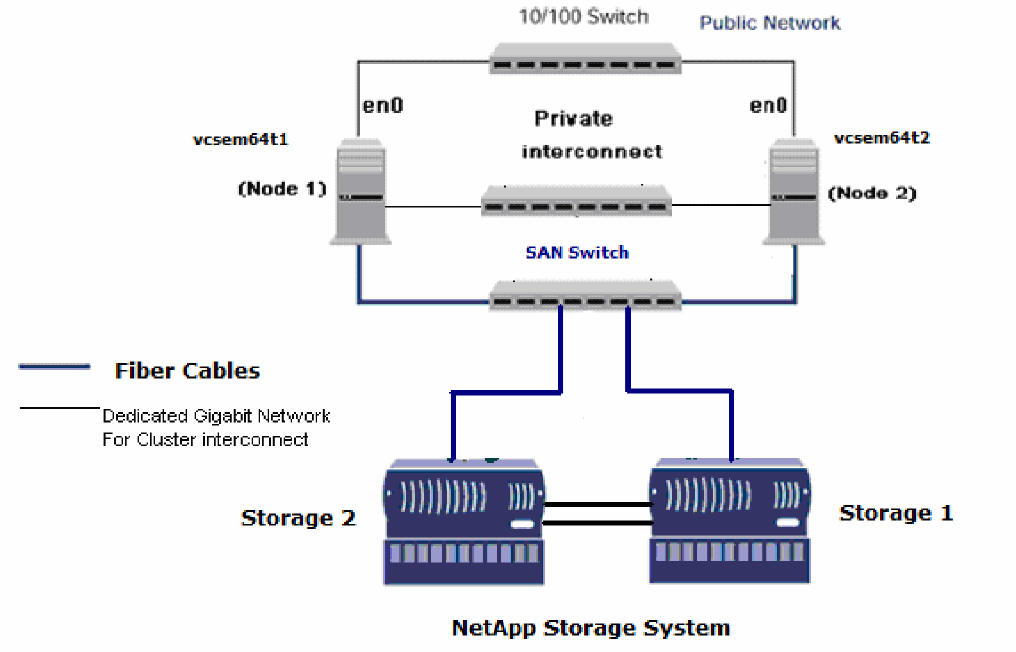


Figure 1) Oracle Database 10 Release 2 Cluster of Two Nodes Utilizing Network Appliance Storage Cluster

Figure 1 illustrates a typical configuration of a two-node Oracle10*g*™ Release 2 RAC utilizing the NetApp storage cluster in a SAN environment over Fibre Channel protocol. This is a scalable configuration and allows users to scale horizontally and internally in terms of processor, memory, and storage.

Each Linux host is connected to both the NetApp storage systems using fiber cables, so in the event of failure of one storage system, the greater reliability is achieved through a cluster failover (CFO).

6. Hardware and Software Requirements

6.1 Hardware Requirements

Cluster Nodes

- Three servers running RHEL4 U5 (two used as cluster nodes and one for hosting clone database)
- Two 10/100/1000Base-TX Ethernet PCI adapters per server (for private interconnect and VIP)
- Dual-port 2GB per second host-based adapter (HBA) per server

Storage Infrastructure

- Two Network Appliance FAS2xx/F7xx/F8xx/FASF9x/FAS30xx systems with Data ONTAP 7.2.2
- Two dual-port 2GB per second host-based adapter (HBA) per system
- One or more disk shelves based on the disk space requirements

6.2 Software Requirements

For two nodes in the participating cluster unless specified otherwise:

- Red Hat Enterprise Linux 4 update 5
- Oracle Database 10*g* Release 2 with RAC and 10.2.0.3 patch-set software
- OCFS2 version 1.2.5-1
- Data ONTAP 7.2.2

7. NetApp Storage Cluster Setup

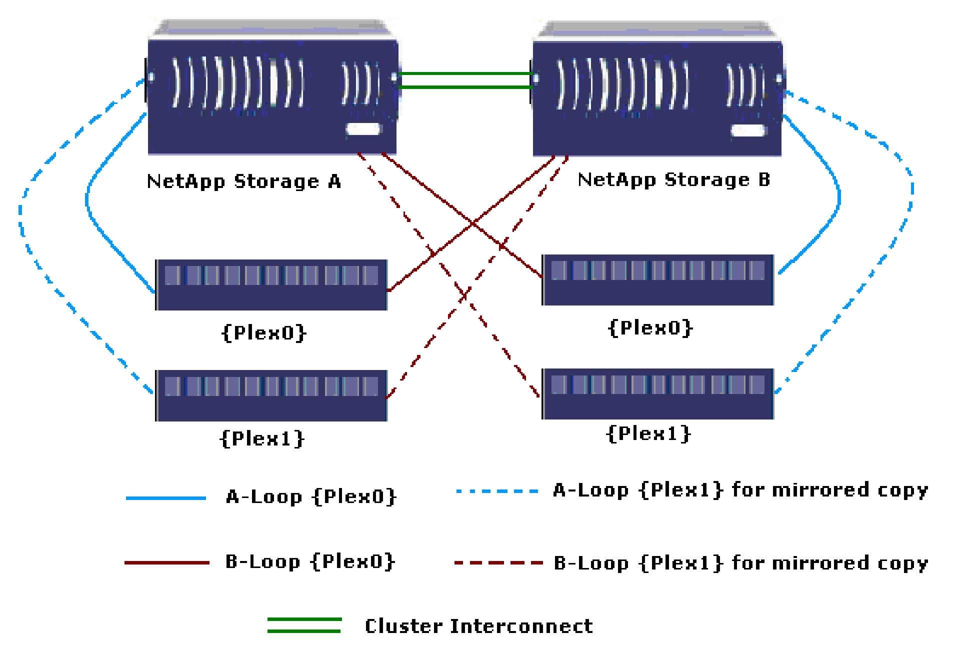


Figure 2) Hardware Setup on Mirrored Active-Active Controllers

For more information, refer to the Network Appliance storage system installation and setup guides at http://now.netapp.com.

The storage configuration described in this document is a mirrored Active-Active controller configuration of NetApp FAS6070 systems. The words failover and takeover, failback and giveback are also used interchangeably throughout the document. The word Partner described in a cluster pair refers to a storage controller.

When one partner fails or becomes impaired, a takeover occurs, and the partner storage system continues to serve the failed storage system's data.

When the failed storage system is functioning again, the administrator initiates a giveback command that transfers resources (failed over resources) back to original partner storage system to resume normal operation, serving its own data.

It is recommended that not both NetApp storage systems be configured for automatic giveback. Giveback should be initiated manually by the administrator during planned downtime because the giveback process takes longer than the takeover process.

1. Configure a NetApp storage system running Data ONTAP 7.2.2 and with cluster, fcp, SnapMirror®, Snapmirror_sync, syncmirror_local, flex_clone and SnapRestore® license keys.

2. The cluster failover parameters on both NetApp storage systems should have following values:

`CF.GIVEBACK.AUTO.ENABLE`	`OFF`
`CF.GIVEBACK.CHECK.PARTNER`	`ON`
`CF.TAKEOVER.DETECTION.SECONDS`	`15`
`CF.TAKEOVER.ON_FAILURE`	`ON`
`CF.TAKEOVER.ON_NETWORK_INTERFACE_FAILURE`	`ON`
`CF.TAKEOVER.ON_PANIC`	`ON`
`CF.TAKEOVER.ON_SHORT_UPTIME`	`ON`

3. Create and export volumes for storing shared database files on the storage:

   Create flexible volumes on the storage as follows:

   The following three volumes will be created:

   - `oradata`
   - `oralogs`
   - `ora10g`

   To create flexible volumes, at the NetApp storage console, type:

```
Storage> vol create oradata 3
Storage> vol create oralogs 3
Storage> vol create ora10g 3
```

   Note: We created all the flexible volumes with three disks each. You can create your volumes based on your workload and application needs.

4. Create LUNs to be accessed over FCP by Linux hosts.

   OCFS2 is a file system that requires disks or partitions to be available on Linux hosts. We will create LUNs under the flexible volumes created above.

   This section describes how to set up the NetApp storage system and configure Linux nodes to access LUNs over FCP.

I. The following software is required to be installed on both Linux nodes:

   - The NetApp FCP Linux host attached utilities kit
   - QLogic HBA drivers for Linux

   The FCP host attach kit can be downloaded from the NetApp NOW^{TM} Web site at http://now.netapp.com. If you do not have access to the above Web site, contact your NetApp sales representative.

   For the latest drivers for QLogic HBA, visit http://support.qlogic.com/support.

II. On the NetApp storage system, to enable FCP service, do the following:

a) Type:

```
Storage> fcp start
```

Create Linux igroups using the Linux hosts HBA WWPN number.

```
Storage> igroup create -f -t linux linux_host1 21:00:00:e0:8b:9b:97:6c

Storage> igroup create -f -t linux linux_host2 21:00:00:e0:8b:9b:cc:6c
```

Where `21:00:00:e0:8b:9b:97:6c` and `21:00:00:e0:8b:9b:cc:6c` are the WWPN numbers of the HBA cards on two Linux hosts through which hosts connect to storage.

b) Create the LUNs as follows:

```
Storage> lun create -s 20g -t linux /vol/oradata/one

Storage> lun create -s 10g -t linux /vol/oralogs/two

Storage> lun create -s 1g -t linux /vol/ora10g/three
```

c) Map the LUNs to both the igroups created above as follows.

```
Storage> lun map <lun name> <igroup name> <lun-id>
```

The `lun-id` should be same for both the igroups for a single LUN.

```
Storage> lun map /vol/aix/one linux_host1 10

Storage> lun map /vol/aix/two linux_host1 11

Storage> lun map /vol/aix/three linux_host1 12

Storage> lun map /vol/aix/one linux_host2 10

Storage> lun map /vol/aix/two linux_host2 11

Storage> lun map /vol/aix/three linux_host2 12
```

III. Accessing storage LUNs from Linux hosts over FCP:

a) To discover the LUNs on Linux hosts, type:

```
vcsem64t1#> /opt/netapp/santools/qla2xxx_lun_rescan all
```

b) Use the sanlun command on the Linux host to view the discovered LUNs and associated device names:

```
vcsem64t1#> sanlun lun show all

filer   lun-pathname    device filename   adapter   protocol     lun
size      lun state

Storage: /vol/oradata/one   /dev/sda         host1     FCP
20g(21474836480)  GOOD Storage: /vol/oradata/one   /dev/sdb
     host1     FCP      10g(10737418240)  GOOD Storage:
/vol/oradata/one   /dev/sdc     host1     FCP       1g(1073741824)
GOOD

vcsem64t2#> sanlun lun show all

filer   lun-pathname    device filename   adapter   protocol     lun
size      lun state

Storage: /vol/oradata/one   /dev/sda         host2     FCP
20g(21474836480)  GOOD Storage: /vol/oradata/one   /dev/sdb
     host2     FCP      10g(10737418240)  GOOD Storage:
/vol/oradata/one   /dev/sdc           host2     FCP
1g(1073741824)   GOOD
```

We will use the above shown devices for OCFS2 file system mountpoints.

8. Operating System Configuration

8.1 Patches

Before you install Oracle10g Release 2 RAC, the following RPMs must be applied to the Linux hosts. Some of these RPMs may have already been applied to your system. Ensure to verify if they already exist before applying them.

To determine whether the required RPMs are installed and committed, type:

```
# rpm –qa | grep compat
```

The following list of patches is required. If any of the patches are not installed or committed, install them.

- binutils-2.15.92.0.2-22
- compat-libstdc++-33-3.2.3-47.3
- gcc-3.4.6-8
- gcc-c++-3.4.6-8
- gcc-objc-3.4.6-8
- glib-1.2.10-15
- glib2-2.4.7-1
- glibc-2.3.4-2.36
- glib2-devel-2.4.7-1
- libaio-0.3.105-2
- libaio-devel-0.3.105-2
- libgcc-3.4.6-8
- libgcj-3.4.6-8
- libgcj-devel-3.4.6-8
- libobjc-3.4.6-8
- libstdc++-3.4.6-8
- libstdc++-devel-3.4.6-8
- openmotif-2.2.3-10.1.el4
- openmotif-devel-2.2.3-10.1.el4
- openmotif21-2.1.30-11.RHEL4.6
- perl-5.8.5-36.RHEL4
- tar-1.14-12.RHEL4
- tcl-8.4.7-2
- unzip-5.51-9.EL4.5
- zip-2.3-27

8.2 Operating System Settings

On Red Hat Linux systems, the default limits for individual users are set in `/etc/security/limits.conf`.

1. As a root user, add following entries in /etc/security/limits.conf to specify oracle user's limits.

```
# Oracle specific settings
oracle soft nofile  4096
oracle hard nofile  65536
oracle soft nproc   2047
oracle hard nproc   16384
oracle soft memlock 3145728
oracle hard memlock 3145728
```

This procedure must be performed on all nodes of the cluster. The server must be restarted to activate updated limits. After modifying the settings, the `ulimit –a` command should display the following:

```
# ulimit –a
  core file size        (blocks, -c)      0
  data seg size         (kbytes, -d)      unlimited
  file size             (blocks, -f)      unlimited
  max locked memory     (kbytes, -l)      unlimited
  max memory size       (kbytes, -m)      unlimited
  open files                    (-n)      1024
  pipe size          (512 bytes, -p)      8
  stack size            (kbytes, -s)      unlimited
  cpu time             (seconds, -t)      unlimited
  max user processes            (-u)      15168
  virtual memory        (kbytes, -v)      unlimited
```

2. As a root user, add the following parameters for the shared memory and semaphores to the `/etc/sysctl.conf` file:

```
kernel.shmall       =     2097152
kernel.shmmax       =     2147483648
kernel.shmmni       =     4096
kernel.sem          =     250 32000 100 1024
fs.file-max         =     65536
net.ipv4.ip_local_port_range = 1024 65000
net.core.rmem_default =   1048576
net.core.wmem_default =   262144
net.core.rmem_max   =     1048576
net.core.wmem_max   =     262144
```

9. OCFS2 Installation and Configuration

9.1 Configuring OCFS2 Cluster

The OCFS2 distribution comprises of two sets of packages, the kernel module and tools. The kernel module can be downloaded from http://oss.oracle.com/projects/ocfs2/files/ and the tools from http://oss.oracle.com/projects/ocfs2-tools/files/.

For the kernel module, download the one that matches the distribution, platform, kernel version, and kernel flavor (hugemem, smp, psmp, and so on). For tools, simply match the platform and distribution.

In the testing environment the following packages are used:

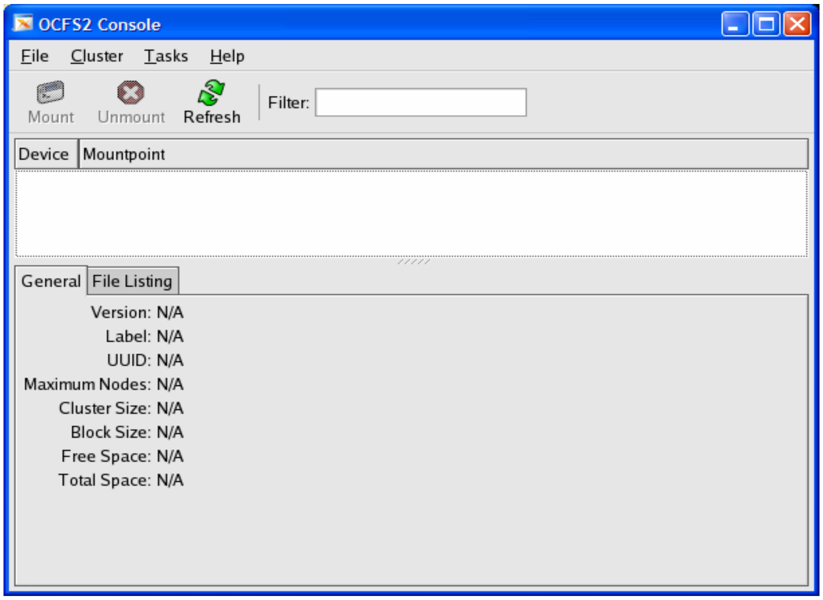
```
Ocfs2-tools-devel-1.2.7-1.el4
Ocfs2-2.6.9-55.EL-debuginfo-1.2.7-1.el4
Ocfs2-2.6.9-55.EL-1.2.7-1.el4
Ocfs2-tools-debuginfo-1.2.7-1.el4
Ocfs2-tools-1.2.7-1.el4
Ocfs2console-1.2.7-1.el4
Ocfs2-2.6.9-55.ELsmp-1.2.7-1.el4
```

Install the packages using the `rpm –ih` command on each Linux node that will be part of an OCFS2 cluster.

OCFS2 has a configuration file named `/etc/ocfs2/cluster.conf`. In this configuration file, all nodes participating in the cluster must be specified. This file should be the same on all the nodes in the cluster. Whereas one can add new nodes to the cluster dynamically, any other changes, like name or IP address, requires the cluster to be restarted for the changes to take effect.

OCFS2 tools provide a GUI utility named `ocfs2console` to set up and propagate the `cluster.conf` to all the nodes in the cluster. This needs to be done only on one of the nodes in the cluster. After this performing this step, you can see the same `/etc/ocfs2/cluster.conf` on all nodes in the cluster.

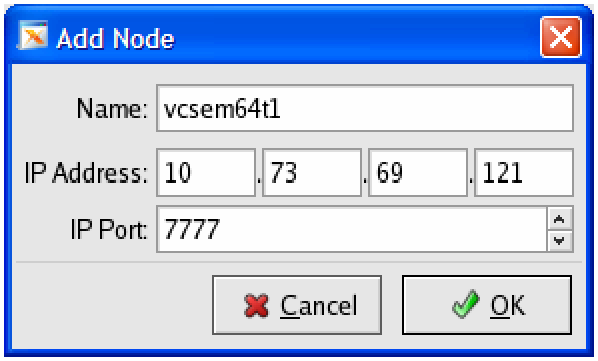
Start the `ocfs2console` and click the `Cluster` menu item, and click `Configure Nodes`. The console will create a cluster with default name `ocfs2`. See Figure 3.



Version: N/A
Label: N/A
UUID: N/A
Maximum Nodes: N/A
Cluster Size: N/A
Block Size: N/A
Free Space: N/A
Total Space: N/A

Figure 3) OCFS2 Console

Click Add to add nodes to the cluster. Enter the node name (same as hostname) and the IP address (same as the private IP address). See Figure 4.



Name: vcsem64t1
IP Address: 10 . 73 . 69 . 121
IP Port: 7777

Figure 4) Add Node

After both nodes are added, propagate the configuration to both the nodes by clicking the `Cluster` menu item, and then clicking `Propagate Configuration`.

The console uses SSH to propagate the configuration file.

Refer to the Appendix for an example of `cluster.conf` file.

9.2 Configuring O2CB Cluster Service

OCFS2 comes bundled with its own cluster stack, O2CB. The stack includes:

- NM: Node manager that keeps track of all the nodes in the `cluster.conf`
- HB: Heartbeat 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 the cluster services is packaged in the `o2cb` system service. OCFS2 operations, such as format, mount, and so on, require the O2CB cluster service to be at least started in the node where the operation will be performed.

To check the status of the cluster, do as follows:

```
# /etc/init.d/o2cb status
Module "configfs": Not loaded
Filesystem "configfs": Not mounted
Module "ocfs2_nodemanager": Not loaded
Module "ocfs2_dlm": Not loaded
Module "ocfs2_dlmfs": Not loaded
Filesystem "ocfs2_dlmfs": Not mounted
```

Perform the following steps on all the nodes:

1. Enable the cluster stack on all nodes by typing:

   ```
   /etc/init.d/o2cb enable
   ```

2. Stop the o2cb service on all the nodes by typing:

   ```
   /etc/init.d/o2cb stop
   ```

3. Edit the following parameters in the /etc/sysconfig/o2cb:

   a) `O2CB_HEARTBEAT_THRESHOLD=81`

   `O2CB_HEARTBEAT_THRESHOLD` value is set based on the timeout value of the I/O layer and a node is deemed dead if it does not update its timestamp for O2CB_HEARTBEAT_THRESHOLD loops.

   b) `O2CB_IDLE_TIMEOUT_MS=60000`

   `O2CB_IDLE_TIMEOUT_MS` describes the value in milliseconds before a network connection is considered dead. It can be set based on the amount of network traffic and the expected delay because of the traffic.

   c) `O2CB_KEEPALIVE_DELAY_MS=4000`

   `O2CB_KEEPALIVE_DELAY_MS` specifies the maximum delay in milliseconds before a keep-alive packet is sent. This done between the nodes if the network connection between them is silent for `O2CB_KEEPALIVE_DELAY_MS` duration. If the node to which the keep-alive packet is sent is alive and connected, then it is expected to respond.

   d) `O2CB_RECONNECT_DELAY_MS=4000`

   `O2CB_RECONNECT_DELAY_MS` specifies the minimum delay in milliseconds before the reconnection attempts.

   These parameters can be configured based on the failover time of the underlying hardware, multipath I/O, and parameters like network bonding.

The O2CB cluster should be now started, and is ready for OCFS2 operations such as format and other O2CB operations that are described below.

4. To start the O2CB cluster, type:

   ```
   /etc/init.d/o2cb start
   ```

5. To offline cluster ocfs2, do as follows:

   ```
   # /etc/init.d/o2cb offline ocfs2
   Cleaning heartbeat on ocfs2: OK
   Stopping cluster ocfs2: OK
   ```

6. To unload the modules, do as follows:

   ```
   # /etc/init.d/o2cb unload
   Unmounting ocfs2_dlmfs filesystem: OK
   Unloading module "ocfs2_dlmfs": OK
   Unmounting configfs filesystem: OK
   Unloading module "configfs": OK
   ```

7. To configure O2CB to start on boot, do as follows:

   ```
   # /etc/init.d/o2cb configure
   Configuring the O2CB driver.
   This will configure the on-boot properties of the O2CB driver.
   The following questions will determine whether the driver is loaded on
   boot. The current values will be shown in brackets ('[]'). Hitting
   <ENTER> without typing an answer will keep that current value. Ctrl-C
   will abort.
   Load O2CB driver on boot (y/n) [n]: y
   Cluster to start on boot (Enter "none" to clear) []: ocfs2
   Writing O2CB configuration: OK
   ```

8. If the cluster is set up to load on boot, start and stop cluster ocfs2 as follows:

   ```
   # /etc/init.d/o2cb start
   Loading module "configfs": OK
   Mounting configfs filesystem at /config: OK
   Loading module "ocfs2_nodemanager": OK
   Loading module "ocfs2_dlm": OK
   Loading module "ocfs2_dlmfs": OK
   Mounting ocfs2_dlmfs filesystem at /dlm: OK
   Starting cluster ocfs2: OK
   ```

9.3 Formatting LUNS and Mounting OCFS2 Partitions

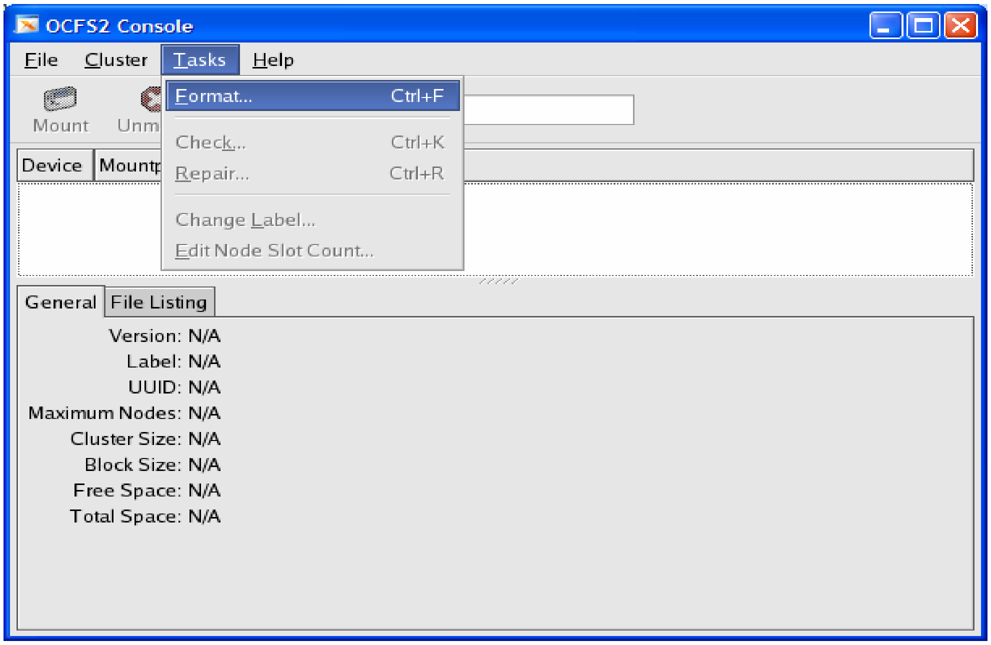
As explained in Section 6 we have three LUNs discovered for both the Linux nodes. This section explains how to format a LUN device and mount it as an OCFS2 partition:

A disk or LUN can be formatted using the OCFS2 console or using the command line tool such as `mkfs.ocfs2`. This needs to be done from one node in the cluster and does not need to be repeated for the same LUN from other nodes.

To format using the console, do the following:

1. Start the `OCFS2 Console`.

2. Click the `Tasks > Format`.



Version: N/A
Label: N/A
UUID: N/A
Maximum Nodes: N/A
Cluster Size: N/A
Block Size: N/A
Free Space: N/A
Total Space: N/A

Figure 5) OCFS2 Console

3. From the `Available devices` drop-down list, select a `device` to format.

   Wherever possible, the console will list the existing file system type.

4. Enter a label. It is recommended to use one label for the device for ease of management.

   The `label` is changeable after the format.

5. Select a cluster size.

   The sizes supported range from 4K to 1M. For a data file's volume or large files, a cluster size of 128K or larger is appropriate.

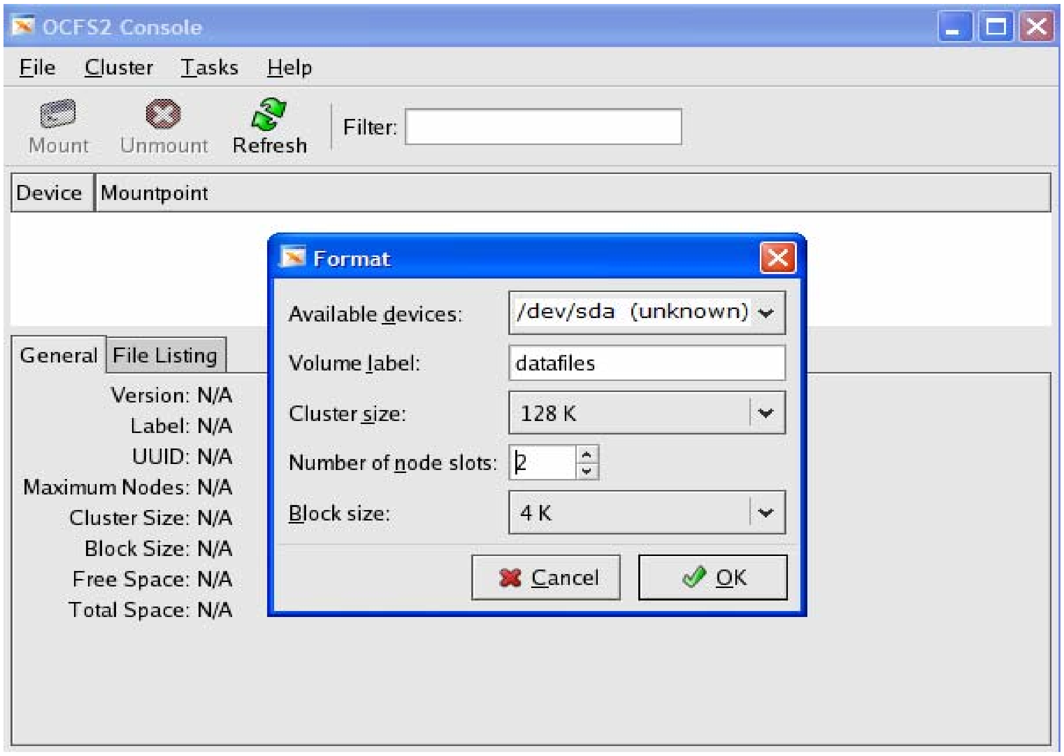
6. Select a `block size`.

   The sizes supported range from 512 bytes to 4K. As OCFS2 does not allocate a static node area on format, a 4K block size is most recommended for most disk sizes. On the other hand, even though it supports 512 bytes, that small a block size is never recommended.

   Note: Both the `cluster` and `blocks sizes` are not changeable after the format.

7. Enter the `number of node slots`. This number determines the number of nodes that can concurrently mount the volume. This number can be increased, but not decreased, at a later date.

8. Click `OK` to format the volume (Figure 6).



Available devices: /dev/sda (unknown)

Volume label: datafiles

Cluster size: 128 K

Number of node slots: 2

Block size: 4 K

Version: N/A
Label: N/A
UUID: N/A
Maximum Nodes: N/A
Cluster Size: N/A
Block Size: N/A
Free Space: N/A
Total Space: N/A

Figure 6) Format Volume

9. To format a volume with a 4K `block size`, `32K cluster size`, and 2 `node slots` using the `mkfs.ocfs2`command line tool, do as follows:

```
# mkfs.ocfs2 -b 4K -C 32K -N 2 -L oracle_home /dev/sda

mkfs.ocfs2 1.2.0

Overwriting existing ocfs2 partition.

Proceed (y/N): y

Filesystem label=oracle_home

Block size=4096 (bits=12)

Cluster size=32768 (bits=15)

Volume size=21474820096 (655359 clusters) (5242872 blocks)

21 cluster groups (tail covers 10239 clusters, rest cover 32256 clusters)

Journal size=33554432

Initial number of node slots: 2

Creating bitmaps: done

Initializing superblock: done

Writing system files: done

Writing superblock: done

Writing lost+found: done

mkfs.ocfs2 successful
```

10. If the O2CB cluster service is offline, start it. The mount operation requires the cluster to be online. To mount from the command line, do as follows:

    ```
    # mount -t ocfs2 /dev/sda /oradata
    ```

11. To unmount a volume, one can select the desired volume in the console and click Unmount or do as follows:

    ```
    # umount /oradata
    ```

Oracle Database users must mount the volumes containing the voting disk file (CRS), cluster registry (OCR), data files, redo logs, archive logs, and control files with the `datavolume, nointr` mount options. The `datavolume` mount option ensures that the Oracle processes open the files with the `O_DIRECT` flag. The `nointr` mount option ensures that the reads and writes on that device are not interrupted by signals. All other volumes, including Oracle home, should be mounted without these mount options.

12. To mount a volume containing Oracle data files, voting disk, and so on, do as follows:

    ```
    # mount -t ocfs2 -o datavolume,nointr /dev/sda /oradata

    # mount

    /dev/sda on /oradata type ocfs2 (rw,datavolume,nointr)
    ```

13. To mount OCFS2 volumes on boot, one needs to enable both the o2cb and ocfs2 services using `chkconfig`, configure `o2cb` to load on boot, and add the mount entries into `/etc/fstab` as follows:

    ```
    # cat /etc/fstab

    ...

    /dev/sda /oradata ocfs2 _netdev,datavolume,nointr 0 0

    /dev/sdb /oralogs ocfs2 _netdev,datavolume,nointr 0 0

    /dev/sdc /ora10g ocfs2 _netdev,datavolume,nointr 0 0

    ...
    ```

The `_netdev` mount option is a must for OCFS2 volumes. This mount option indicates that the volume is to be mounted after the network is started and dismounted before the network is shut down. (The `datavolume` and `nointr` mount options are only required for Oracle data files and so on.) The `ocfs2` service can be used to mount and unmount OCFS2 volumes. It should always be enabled to ensure that the OCFS2 volumes are unmounted before the network is stopped during shutdown.

```
# chkconfig --add ocfs2

ocfs2 0:off 1:off 2:on 3:on 4:off 5:on 6:off

# chkconfig --add o2cb

o2cb 0:off 1:off 2:on 3:on 4:off 5:on 6:off

# /etc/init.d/o2cb configure

...

Load O2CB driver on boot (y/n) [n]: y

Cluster to start on boot (Enter "none" to clear) []: ocfs2

Writing O2CB configuration: OK
```

10. Installation Procedure

10.1 Installing the Oracle RAC 10g Release 2 Cluster Ready Services (CRS)

For detailed information on installing Oracle Cluster Ready Services on Linux, refer to the Oracle Real Application Clusters Installation and Configuration Guide 10*g* Release 2 (10.2.0.1) for UNIX® systems at http://otn.oracle.com/docs/content.html. This section briefly describes the procedures for using Oracle Universal Installer (OUI) to install Cluster Ready Services (CRS).

Note: The CRS home that you identify in this phase of the installation is only for CRS software; this home cannot be the same home as the Oracle Database 10*g* RAC home. That is, `ORACLE_HOME` and CRS HOME must be different locations.

1. Run the `runInstaller` command from the `/crs` subdirectory on the Oracle Cluster Ready Services Release 2 (10.2.0.1) CD-ROM or from the staging area where Oracle CRS software has been dumped. This is a separate CD that contains the Cluster Ready Services software. This document assumes that `OUI` is started from node 1 `(vcsem64t1).` When `OUI` displays the `Welcome` page, click `Next.`
2. On the Specify Inventory page, enter a nonshared location for Oracle Inventory. This is the only part of Oracle Database 10g that should not be shared. For the testing environment, we used /home/oracle/oraInventory for the Oracle Inventory information. Click `Next`.
3. The `Specify File Location`s page contains predetermined information for the source of the installation files and the target destination information. Specify the destination path for the shared CRS home. The path should be on a shared file system and different from `$ORACLE_HOME`. In this exercise, the shared CRS home was `/orahome/ora10g/product/10.2.0/crs_1`.
4. On the next screen, specify the cluster name, public interface names (hostnames), private interface names, and virtual interface hostnames to be used for the cluster interconnect. In this case, the public names are `vcsem64t1` and `vcsem64t2`, the private names are `vcsem64t1-i` and `vcsem64t2-i`, and the virtual hostnames are `vcsem64t1-v` and `vcsem64t2-v`. Click `Next` to continue.
5. On the `Network Interface Usage` page, specify the private network to be used for the cluster interconnect. This is a very important step. Do not leave it set to the default, which is `Do Not Use`. In this case, `eth1 (vcsem64t1-i)` was used as the private interconnect and `eth0 (vcsem64t2`) was used as the public interface. Select the interface and click the `Edit` button to modify it. Click `Next`.
6. On the `Oracle Cluster Registry` page, specify the OCR (Oracle Cluster Registry) file. Be sure to specify the full path to a shared location along with the name of the file. Do the same for a mirror file if you want normal redundancy. In our case, we used `/ora10g OCFS2` partition for creating the OCR file. Click `Next`.
7. On the `Voting Disk` page, specify the CSS (Cluster Synchronization Services) voting disk file location. We used `/ora10`g, OCFS2 partition as the CSS voting disk location. In case of normal redundancy specify the path along with the name. Click `Next` to install the CRS.
8. When prompted, run the following script as root user starting from primary node when it is prompted:

   ```
   /orahome/ora10g/orainventory/orainstRoot.sh
   ```

   ```
   /oarhome/ora10g/product/10.2.0/crs_1/root.sh
   ```
9. In the Configuration Assistant window you may see some warnings. Click OK to continue.
10. Run the vipca utility from the `$ORA_CRS_HOME/bin` directory as root user on the master node (vcsem64t1). Click `Next`.
11. Select the Public Interface. Click `Next`.
12. Specify the Virtual IP address and Subnet Mask of each node. Click `Next`.
13. Click `Finish` to continue vipca.

14. Click `OK` and then `Exit` to finish VIPCA.
15. To verify your CRS installation, execute the olsnodes command from the `$CRS_HOME/bin` directory.

    The `olsnodes` command syntax is:

    ```
    olsnodes [-n] [-l] [-v] [-g]
    Where:
    ```

    `-n` displays the member number with the member name

    `-l` displays the local node name

    `-v` activates verbose mode

    `-g` activates logging

    The output from this command should be a list of the nodes on which CRS was installed.

10.2. Installing Oracle Database 10G Release 2 Software

1. After making sure that Oracle Cluster Ready Services have started on the cluster nodes, start runInstaller from Disk1 of the Oracle Database 10g Release 2 CDs or from the staging area where you have kept the Oracle Database 10g downloads.
2. On the Specify File Locations screen, enter the destination path for the shared ORACLE_HOME. This should be a different location than the shared CRS Home. For this exercise, the shared `ORACLE_HOME` was `/orahome/ora10g/product/10.2.0/db_1`.
3. On the next screen, select `Cluster Installation` and then select all the nodes in the cluster. For our exercise, the two cluster nodes were `vcsem64t1` and `vcsem64t2`. Click Next.

    Note: If the nodes are not displayed in the cluster node selection, then Oracle Cluster Ready Services are not configured or started on those cluster nodes.
4. For installation type, select `Enterprise Edition` and click `Next`.
5. On the `Select Database Configuration` screen, select `Do not create a starter database`. We used dbca to create a database later. Click `Next`.
6. Run the following scripts as root user starting from master node when prompted.

    ```
    ./$ORACLE_HOME/root.sh
    ```
7. Click `Exit` to finish the database installation.

    Note: Install Oracle Database 10*g* Release 2 Patch 3 on both `CRS_HOME` and `ORACLE_HOME` using `OUI Oracle Universal Installer`. For more details about the patch installation, refer to the Oracle Patch Installation Guide.

11. Creating Snapshot Copies and FlexClone Volumes

This section describes how to create a Snapshot copy of the volume, create a FlexClone volume, and start the clone database on a different Linux host.

Before creating a Snapshot copy of any Oracle Database volume, we need to put all the tablespaces into hot backup mode. Refer to the Appendix for the sample script of putting tablespaces into hot backup mode. Also use the `sync` command on Linux host to force any changed blocks to disks.

In our setup we are using three flexible volumes: `oradata`, `oralogs,` and `ora10g`. Since we will only be creating the clone of the database, we would create Snapshot copies of only data files, control files, and online redo log files which reside in `oradata` and `oralogs` volumes.

1. Enable RSH or SSH between the Linux host and the NetApp storage system.

2. To create a Snapshot copy of a volume, use the following commands:

```
rsh <storage-name> “snap create <volume-name> <snap-name>;”

rsh Storage “snap create oradata oradata_snap1;”

rsh Storage “snap create oralogs oralogs_snap1;”
```

3. The Snapshot copy created can be checked by typing using the following commands:

```
rsh Storage “snap list oradata”

Volume oradata

working...

  %/used         %/total             date                name

----------              ----------          ------------        ------------

  0% ( 0%)       0% ( 0%)           Jul 16 17:10         oradata_snap1

rsh Storage “snap list oralogs”

Volume oralogs

working...

  %/used         %/total             date                name

----------              ----------          ------------        ------------

  0% ( 0%)       0% ( 0%)            Jul 16 17:12        oralogs_snap1
```

4. After creating Snapshot copies, put the tablespaces into normal mode. Refer to the Appendix for the sample script.

5. Before we create a FlexClone volume, check the size of the aggregate on which the parent volume resides as follows:

```
rsh Storage “ df –Ag aggr1;”

Aggregate              total         used          avail         capacity

aggr1                  109GB         44GB          65GB          41%

aggr1/.snapshot  5GB           0GB           5GB           0%
```

6. To create a FlexClone volume, type:

```
rsh <storage-name> “ vol clone create <clone-volume-name> -s none –b
<parent-vol-name> <parent-snap-name>

rsh Storage “ vol clone create oradata_clone –s none –b oradata
oradata_snap1;”

rsh Storage “ vol clone create oralogs_clone –s none –b oralogs
oralogs_snap1;”
```

While executing the `vol clone` command, Data ONTAP displays the following message:

```
Reverting volume GadgetData to a previous snapshot.
```

For those not familiar with Data ONTAP, this is the standard message when a Snapshot copy is used to restore a volume to a previous state. Since FlexClone volumes leverage Snapshot technology to get a point-in-time image of the parent FlexVol volume, the same mechanism and message are used. The volume mentioned in the message is the new FlexClone volume. Although the word `“revert”` implies that it is going back to a previous version, it is not actually “reverted,” since it has just come into existence.

7. Check the status of FlexClone volumes by typing:

```
rsh Storage “vol status oradata_clone;”
rsh Storage “vol status oralogs_clone;”
```

8. Check the space of the aggregate after creating a clone, by typing:

```
rsh Storage “ df –Ag aggr1;”

Aggregate               total       used        avail       capacity

aggr1                   109GB       44GB        65GB        41%

aggr1/.snapshot  5GB          0GB         5GB         0%
```

Note that the amount of space used in the aggregate has not increased. This is because space reservations are disabled for FlexClone volumes in Data ONTAP 7G.

9. Check the LUN status, by typing:

```
rsh Storage “lun show;”
    /vol/oradata/one       20g (21474836480)      (r/w, online, mapped)
    /vol/oralogs/two       10g (10737418240)      (r/w, online, mapped)
    /vol/ora10g/three       1g (1073741824)       (r/w, online, mapped)
    /vol/oradata_clone/one 20g (21474836480)      (r/w, offline)
    /vol/oralogs_clone/two 10g (10737418240)      (r/w, offline)
```

The newly created clone volumes have the same LUNs as that of the parent volume, but they are offline and not mapped to any igroup. A new igroup with the WWPN number of Linux nodes that will host the clone database must be created and then the new LUNs mapped to it.

10. Use the following commands:

```
rsh Storage “igroup create -f -t linux linux_host3 21:00:00:e0:8b:9b:97:6b;”
rsh Storage “lun online /vol/oradata_clone/one;”
rsh Storage “lun online /vol/oralogs_clone/two;”
rsh Storage “lun map /vol/oradata_clone/one linux_host3 10;”
rsh Storage “lun map /vol/oralogs_clone/two linux_host3 11;”
```

11. Ensure that the Linux node is connected to the same NetApp storage through fiber cables and follow all preinstallation OS activities for this node to host the clone database.

12. Log into the Linux node (which hosts the clone database) as a root.

13. To discover the LUNs mapped to this Linux node, type:

```
vcsem64t3#> /opt/netapp/santools/qla2xxx_lun_rescan all
```

14. Check the newly discovered LUNs by typing:

```
vcsem64t3#> sanlun lun show all
```

```
filer    lun-pathname       device filename  adapter  protocol    lun size
lun state
Storage: /vol/oradata_clone/one  /dev/sda  host3     FCP    20g(21474836480)
GOOD Storage: /vol/oralogs_clone/two  /dev/sdb    host3     FCP
10g(10737418240)  GOOD
```

15. For mounting these LUN devices as OCFS2 partitions, perform the steps mentioned in Section 9, OCFS2 Installation and Configuration. The only difference would be to configure the OCFS2 cluster with a single node. Mount the OCFS2 partitions by typing:

```
mount -t ocfs2 -o datavolume,nointr /dev/sda /oradata
mount -t ocfs2 -o datavolume,nointr /dev/sdb /oralogs
```

16. Refer to Oracle10g documentation to install Oracle Database 10g Release 2 database software on this Linux node. Copy the init<sid>.ora file and create the dump directories in respective folders same as primary RAC database. Since a non-RAC clone database created from the RAC database, it is required to create a new controlfile. Also it is required to remove the parameters related to cluster database from the init<sid>.ora file.

17. After starting the clone instance in the nomount stage, create a new control file, recover the database, and then open the database.

12. Appendix

1. Script to put tablespaces into hot backup mode

```
set head off;
spool /tmp/backup.sql;
select distinct 'alter tablespace ' || tablespace_name || ' begin
backup;' from dba_data_files;
spool off;
@@/tmp/backup.sql;
exit;
```

2. Script to put tablespaces into normal mode

```
set head off;
spool /tmp/backupend.sql;
select distinct 'alter tablespace ' || tablespace_name || ' end
backup;' from dba_data_files;
spool off;
@@/tmp/backupend.sql;
exit;
```

3. .bash_profile for oracle user

```
# .bash_profile
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi
# User specific environment and startup programs
export ORACLE_BASE=/orahome/ora10g
export ORACLE_PRODUCT=$ORACLE_BASE/product/10.2.0
export ORACLE_HOME=$ORACLE_PRODUCT/db_1
export ORACLE_CRS=$ORACLE_PRODUCT/crs_1
export ORA_CRS_HOME=$ORACLE_PRODUCT/crs_1
LD_LIBRARY_PATH=$ORACLE_HOME/lib:$ORACLE_HOME/lib32:$ORACLE_HOME/rdbms/li
b:$ORACLE_HOME/rdbms/lib32:$ORACLE_CRS/lib:$ORACLE_CRS/lib32:$ORACLE_CRS/
rdbms/lib:$ORACLE_CRS/rdbms/lib32:$LD_LIBRARY_PATH:/usr/lib64:/usr/lib:/l
ib:$ORACLE_HOME/oracm/lib:/usr/local/lib
export LD_LIBRARY_PATH

LIBPATH=$ORACLE_HOME/lib:$ORACLE_HOME/lib32:$ORACLE_HOME/rdbms/lib:$ORACL
E_HOME/rdbms/lib32:$ORACLE_CRS/lib:$ORACLE_CRS/lib32:$ORACLE_CRS/rdbms/li
b:$ORACLE_CRS/rdbms/lib32:$LIBPATH:/usr/lib64:/usr/lib:/lib:$ORACLE_HOME/
oracm/lib:/usr/local/lib
export LIBPATH
ORACLE_PATH=$ORACLE_BASE/common/oracle/sql:.:$ORACLE_HOME/rdbms/admin
Export ORACLE_PATH
export ORACLE_TERM=xterm
export TNS_ADMIN=$ORACLE_HOME/network/admin
export ORA_NLS10=$ORACLE_HOME/nls/data
PATH=/usr/bin:$PATH:$HOME/bin:$ORACLE_HOME/bin:$ORACLE_CRS/bin:/orahome_1
1g/ora11g/product/11.1.0/crs_1/bin
export PATH
unset USERNAME
```

13. Revision History

Creation Date	Created by	Modification Date	Modified by
July 2007	Sushant Desai		
		Dec 2007	Eesha Pathak

14. Acknowledgement

The author would like to thank the following individuals for their contribution to the testing process and technical report:

Network Appliance, Inc.

Uday Shet, Sanjay Gulabani

15. Disclaimer

Each environment has its own specific set of requirements, and no guarantees can be given that the results presented in this report will work as expected on other platforms. This paper should assist in the research and troubleshooting that may be required in a particular case and serve as a checklist of items to be aware of. Please forward any errors, omissions, differences, new discoveries, and comments about this paper to eesha.pathak@netapp.com.