Technical report:

Oracle Cluster File System and Oracle RAC on IBM System Storage N series

Best-practice installation and configuration

Document NS3605-0

May 22, 2008

Table of contents

Abstract

This technical report documents installation and configuration of an open-source Oracle cluster file system in an Oracle database cluster environment on an IBM System Storage N series. The solution uses IBM System Storage N series with Snapshot and IBM System Storage N series with FlexClone technologies for backing up and creating clone databases on the Oracle file system.

Introduction

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

This technical report documents best-practice installation and configuratione OCFS2 in an Oracle Database Real Application Clusters (RAC) environmentand use IBM® System Storage™ N series with Snapshot™ and IBM System Storage N series with FlexClone™ technologies for backing up and creatinge clone databases on an OCFS2 file system.

Assumptions

This technical report assumes readers are familiar with Oracle Database RAC 10*g*™ Release 2 (Oracle RAC10gR2) concepts, the operation of Red Hat Linux operating systems, the operation of IBM N series storage systems, operation of storage area networks (SANs) over fibre channel (FC), and general knowledge in networking.

Overview of snapshot copies and Flexclone

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 N series storage system. After a snapshot copy has been 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. An N series 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 network area storage (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, a 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 IBM System Storage N series with Data ONTAP® 7.1, storage administrators have access to a powerful new feature that allows them to instantly create clones of a flexible volume via IBM System Storage N series with 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 a huge potential saving in dollars, space, and energy. In addition to all these benefits, clone volumes have the same high performance as other kinds of volumes.

Conceptually, FlexClone volumes are great for any situation where testing or development occurs, any situation where progress is made by locking in incremental improvements, and any situation where there is a desire to distribute data in changeable form without endangering the integrity of the original.

Overview of OCFS2

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 it provides to applications, OCFS2 balances performance and manageability by providing functionality that is in between the functionality provided by raw devices and typical file systems. While retaining the performance of raw devices, OCFS2 provides higher-order, more manageable file interfaces. 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, resizing data files and partitions is easy) and comes with a set of tools to manage OCFS2 files.

The server environment

In this report and testing environment, the servers are running the Red Hat Enterprise Linux 4 Update 5 (RHEL4U5) operating system. The OCFS2 version is 1.5.2-1 (hereafter again 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 gain support from all parties involved. The only exception to this is the application of certain patches (as defined and required by all the vendors in this configuration). Two N series storage systems are configured in cluster to operate in a SAN FC protocol (FCP) environment.

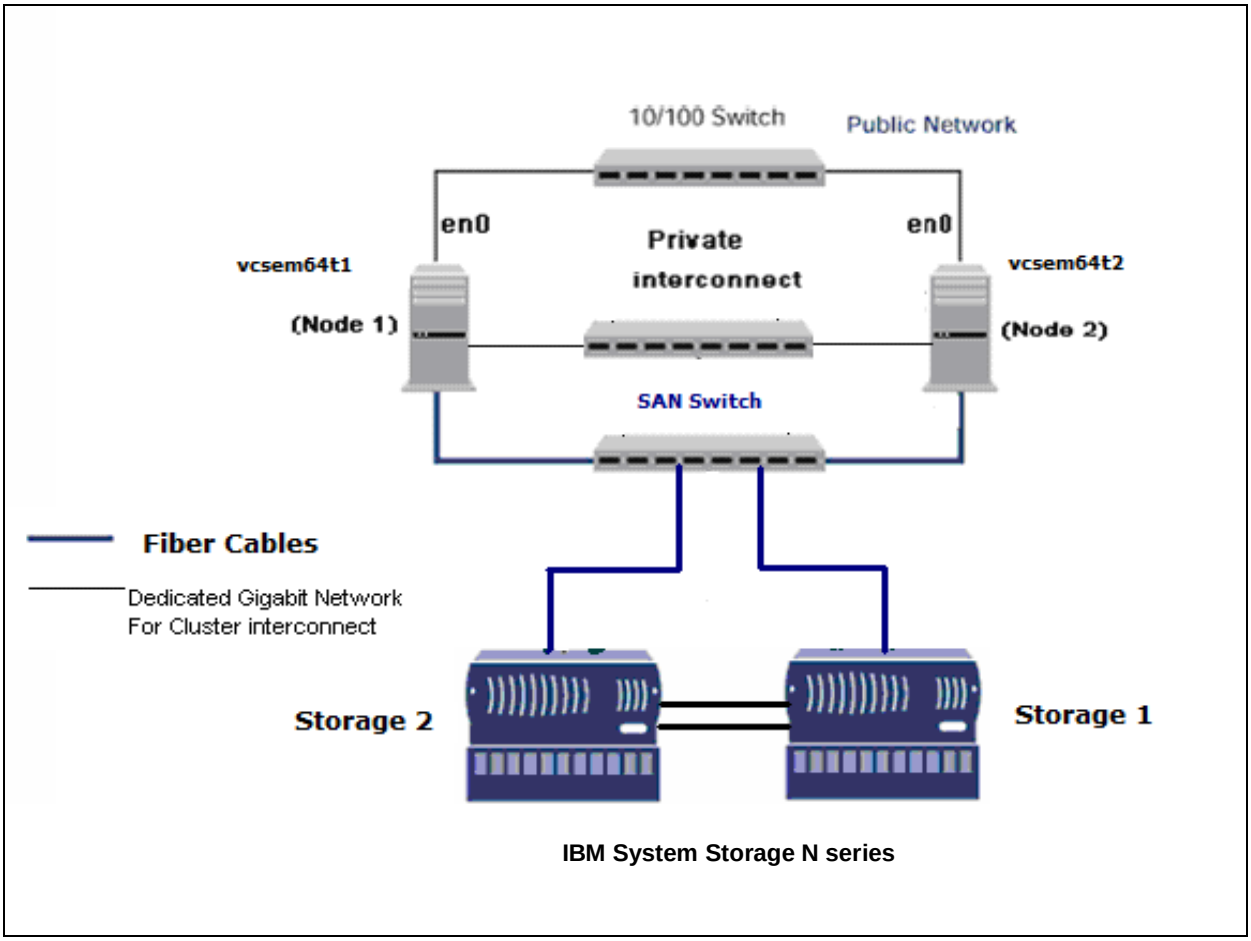


Figure 1) Oracle RAC10gR2 cluster of two nodes utilizing an N series storage cluster.

Figure 1 illustrates a typical configuration of a two-node OracleRAC10*g*R2 infrastructure utilizing the N series storage cluster in a SAN environment over FCP. 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 N series 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).

Requirements

Hardware

Cluster nodes

- Three servers running RHEL4U5 (two used as cluster nodes and one for hosting a 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 IBM System Storage N series systems with Data ONTAP 7.2.2
- Two dual-port 2GB per second HBA per system
- One or more disk shelves based on the disk space requirements.

Software

For all two nodes in the participating cluster unless specified otherwise:

- RHEL4U5
- OracleRAC10*g*R2 and 10.2.0.3 patch-set software
- OCFS2 version 1.2.5-1
- Data ONTAP 7.2.2.

Setup for IBM N series storage cluster

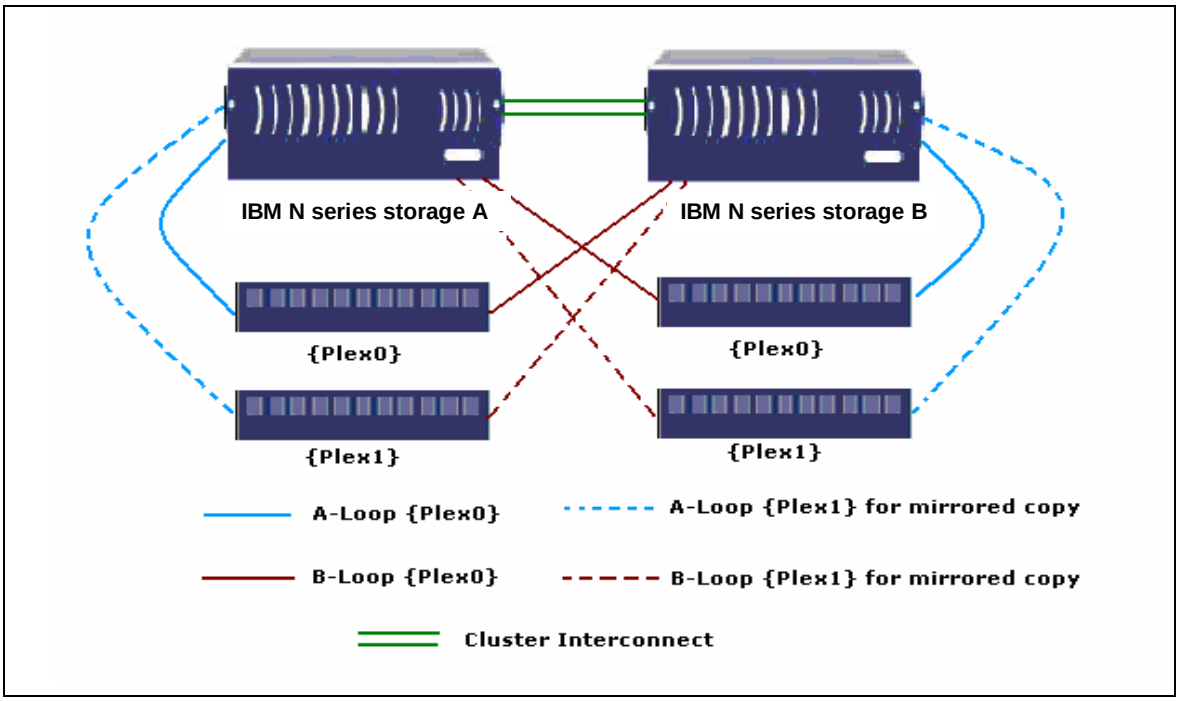


Figure 2) N series hardware setup on mirrored active-active controllers.

The storage configuration described in this document is a mirrored active-active controller configuration of N series N5200 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 N series 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. Please configure a N series storage system running Data ONTAP 7.2.2 and with cluster, FCP, IBM System Storage N series with SnapMirror®, IBM System Storage N series with SyncMirror, FlexClone, and IBM System Storage N series with SnapRestore® license keys.
2. The cluster failover parameters on both N series 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 listed below.

   We will create three volumes, namely `(1) oradata, (2) oralogs, and 3) ora10g.`

   To create flexible volumes, use the following command at the N series storage console:

```
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 logical unit numbers (LUNs) to be accessed over FCP by Linux hosts.

Creating LUNs

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

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

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

- The N series FCP Linux host attachment utilities kit
- QLogic HBA drivers for Linux.

The FCP host attachment utilities kit is available at the IBM support website. At the time of this report writing, the latest drivers for QLogic HBA can be downloaded from http://support.qlogic.com/support.

1. On the N series storage system, enter the following command to enable FCP service.

```
Storage> fcp start
```

2. 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.

3. Create the LUNs.

```
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
```

4. Map the LUNs to both the igroups created above.

```
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
```

Accessing storage LUNs from Linux hosts over FCP

1. Run the following command to discover the LUNs on Linux hosts:
   ```
   vcsem64t1#> /opt/ibmn/santools/qla2xxx_lun_rescan all
   ```
2. Use the `sanlun` command on the Linux host to view 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.

Operating system configuration

Patches

Before you install OracleRAC10*g*R2, the following RPMs must be applied to the Linux hosts. Some of these RPMs may have already been applied to your system. Be sure to verify whether they already exist before applying them.

To determine if the required RPMs are installed and committed, enter a command similar to the following:

```
# rpm –qa | grep compat
```

Here is a list of required patches. 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

Operating-system settings

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

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 must be done on all nodes of the cluster. A server reboot is required to activate updated limits. After you modify 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
```

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
```

OCFS2 installation and configuration

OCFS2 cluster configuration

The OCFS2 distribution comprises of two sets of packages, the kernel module and tools. The kernel module and tools are available for download from Oracle (at the time of this report writing at http://oss.oracle.com/projects/ocfs2/files/ and http://oss.oracle.com/projects/ocfs2-tools/files/ respectively). 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 were used:

```
ocfs2-2.6.9-55.ELsmp-1.2.5-1.x86_64.rpm
ocfs2console-1.2.4-1.x86_64.rpm
ocfs2-tools-1.2.4-1.x86_64.rpm
```

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

OCFS2 has a configuration file, `/etc/ocfs2/cluster.conf`. In it, one needs to specify all the nodes participating in the cluster. 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 change, like name or IP address, requires the cluster to be restarted for the changes to take effect.

OCFS2 tools provide a graphical user interface (GUI) utility, OCFS2 Console, to set up and propagate the file `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 step, users will be able to see the same `/etc/ocfs2/cluster.conf` on all nodes in the cluster.

Start the OCFS2 Console and click the menu item “Cluster” followed by “Configure Nodes.” The console will create a cluster with default name `ocfs2`.

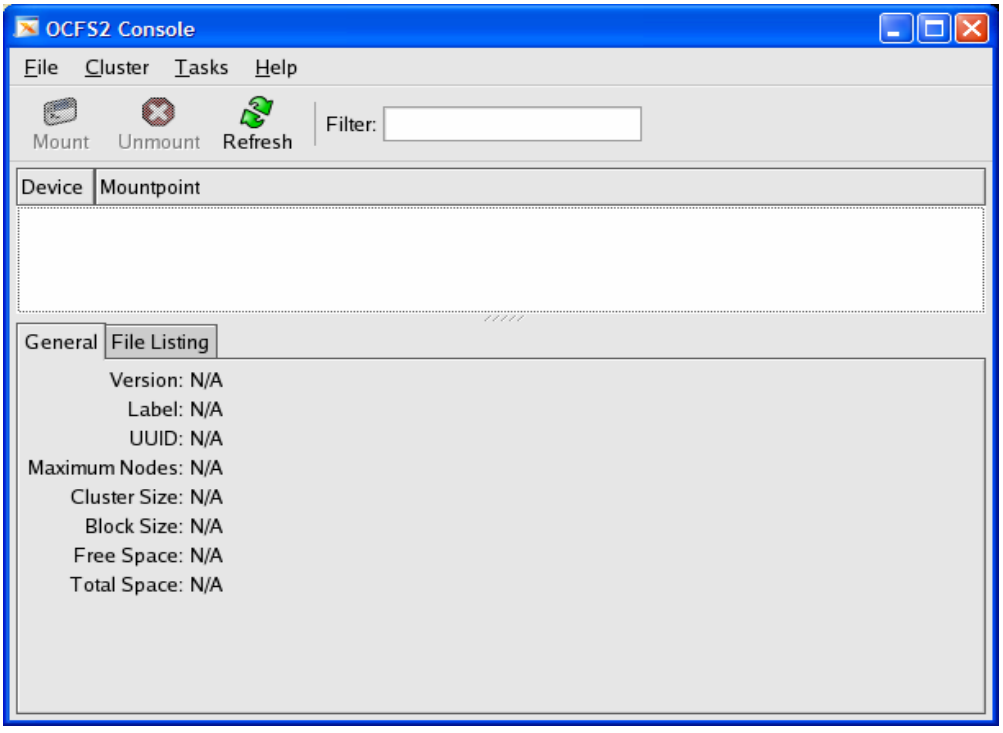


Figure 3) Cluster configuration on OCFS2 Console.

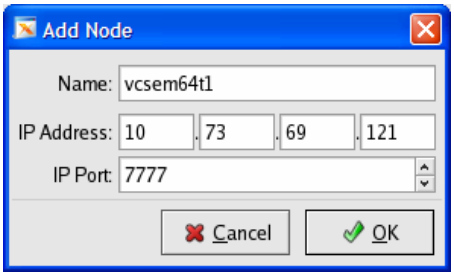


Figure 4) Node addition to OCFS2 cluster.

Then click “Add” to add nodes to the cluster. Enter the node name (same as hostname) and the IP address as shown in the preceding screen shot.

Once both nodes are added, one can propagate the configuration to both the nodes by clicking the menu item “Cluster,” followed by “Propagate Configuration.” The console uses the Secure Shell (SSH) network protocol to propagate the configuration file.

Refer to the document appendix for a sample `cluster.conf` file.

O2CB cluster service configuration

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

NM: Node manager that keeps track of all the nodes in the `cluster.conf` file
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 have been packaged in the O2CB system cluster 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:

```
# /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
```

To load the modules, do:

```
# /etc/init.d/o2cb load
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
```

To online cluster OCFS2, do:

```
# /etc/init.d/o2cb online ocfs2
Starting cluster ocfs2: OK
```

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

To offline cluster OCFS2, do:

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

To unload the modules, do:

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

To configure O2CB to start on boot, do:

```
# /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
```

If the cluster is set up to load on boot, one could start and stop cluster OCFS2 as follows:

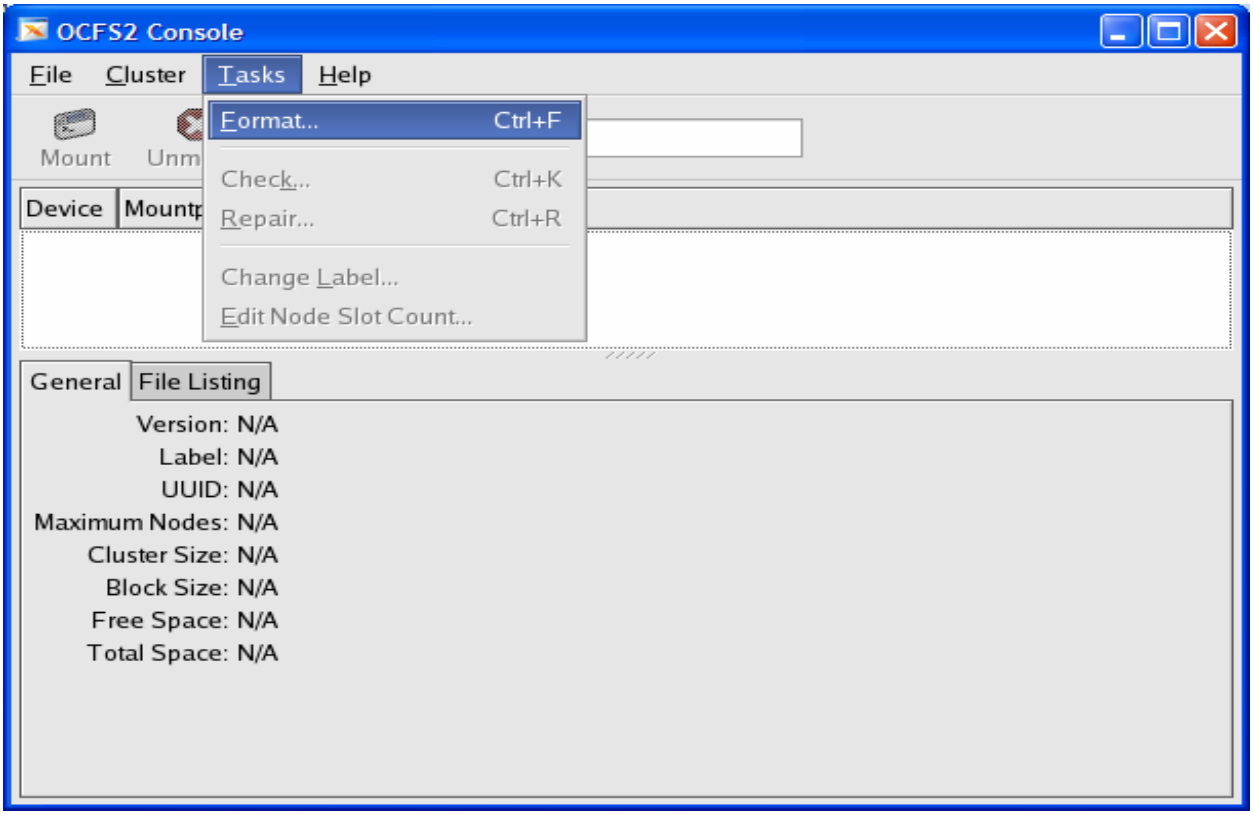
```
# /etc/init.d/o2cb start
Loading module "configfs": OK
Mounting configfs filesystem at /config: OK
Loading module "ocfs2_nodemanager": OK
Loading module "ocfs2_dlm": OK
Loading module "ocfs2_dlmfs": OK
Mounting ocfs2_dlmfs filesystem at /dlm: OK
Starting cluster ocfs2: OK
```

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:

One can format a disk or LUN using the OCFS2 Console or using the command line tool `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, start OCFS2 Console and click the menu item “Tasks.” followed by “Format.”



Then continue with the below steps.

1. Select a device to format in the drop-down list, “Available Devices.” Wherever possible, the console will list the existing file system type.
2. Enter a label. It is recommended to use one label for the device for ease of management. The label is changeable after the format.
3. Select a cluster size. The sizes supported range from 4 KB to 1 MB. For a data file’s volume or large files, a cluster size of 128 KB or larger is appropriate. The cluster size is not changeable after the format.
4. Select a block size. The sizes supported range from 512 bytes to 4 KB. As OCFS2 does not allocate a static node area on format, a 4 KB 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. The block size is not changeable after the format.

5. 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.
6. Click “OK” to format the volume.

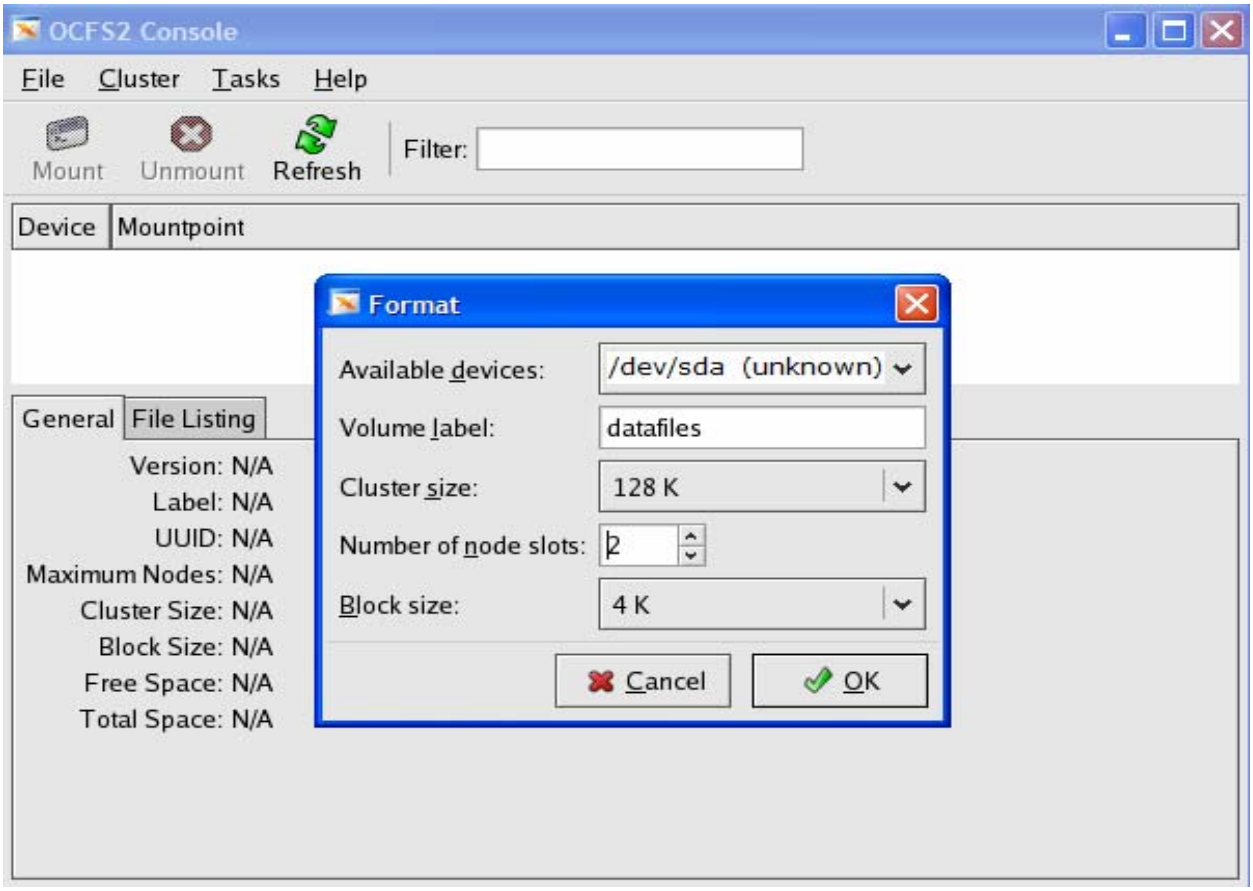


Figure 5) Formatting LUNs in the OCFS2 cluster.

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

```
# 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
```

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:

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

To unmount a volume, one can select the desired volume in the console and click `Unmount` or do:

```
# umount /oradata
```

Oracle database users must mount the volumes containing the Oracle voting disk file – Cluster Ready Services (CRS), the Oracle Cluster Registry (OCR) file, 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.

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

```
# mount -t ocfs2 -o datavolume,nointr /dev/sda /oradata
# mount
/dev/sda on /oradata type ocfs2 (rw,datavolume,nointr)
```

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
```

Installation procedure

Installing OracleRAC10*g*R2 CRS

For detailed information on installing Oracle CRS on Linux, refer to the appropriate Oracle RAC installation and configuration guide. At the time of this report writing, the guide for OracleRAC10*g*R2 (10.2.0.1) for UNIX® systems could be found at http://otn.oracle.com/docs/content.html. This section briefly describes the procedures for using Oracle Universal Installer (OUI) to install 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 OracleRAC10*g* home. That is, ORACLE_HOME and CRS HOME must be different locations.

1. Run the `runInstaller` command from the `/crs` subdirectory on the Oracle CRS 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 CRS 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 Oracle10*g* 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 Locations" 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 for use in the cluster interconnect. Here, 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."
5. On the "Network Interface Usage" page, specify the private network for use in the cluster interconnect. This is an important step. Do not leave it at the default, which is `Do Not Use`. Here, `eth1 (vcsem64t1-i)` was the private interconnect and `eth0 (vcsem64t2)` was 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 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 Cluster Synchronization Services (CSS) voting disk file location. We used `/ora10g`, 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 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 `$ORA_CRS_HOME/bin` directory as root user on the master node (`vcsem64t1`). Click “Next.”
11. Select “Public Interface.” Click “Next.”
12. Specify the VIP address and subnet mask of each node. Click “Next.”
13. Click “Finish” to continue the VIPCA utility.
14. Click “OK” and then “Exit” to finish the VIPCA utility.
15. To verify your CRS installation, execute the `olsnodes` command from the `$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.

Installing Oracle Database software

The following steps install the Oracle10*g* Release 2 (Oracle10*g*R2) software.

1. After making sure that Oracle CRS has started on the cluster nodes, start `runInstaller` from Disk 1 of the Oracle10*g*R2 CDs or from the staging area where you have kept the Oracle10*g* 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 file 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 CRS is 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 DCBA 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 Oracle10*g*R2 Patch 3 on both `CRS_HOME` and `ORACLE_HOME` using OUI. For more details about the patch installation, refer to the available Oracle patch installation guide.

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. Please 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.

Enable Remote Shell (RSH) or SSH between the Linux host and the N series storage system. To create a snapshot copy of a volume, following commands can be used:

```
rsh <storage-name> “snap create <volume-name> <snap-name>;”
rsh Storage “snap create oradata oradata_snap1;”
rsh Storage “snap create oralogs oralogs_snap1;”
```

One can check the snapshot copy created 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
```

After creating snapshot copies, put the tablespaces into normal mode. See the document appendix for a sample script.

Before creating a FlexClone volume, we check the size of the aggregate where the parent volume resides:

```
rsh Storage “ df –Ag aggr1;”

Aggregate               total        used         avail        capacity
aggr1                   109GB        44GB         65GB         41%
aggr1/.snapshot         5GB          0GB          5GB          0%
```

To create a FlexClone volume, use the following commands:

```
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;”
```

During the `vol clone` command, Data ONTAP prints an informational message saying “`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.

Check the status of FlexClone volumes using the following command:

```
rsh Storage "vol status oradata_clone;"
rsh Storage "vol status oralogs_clone;"
```

We will check the space of the aggregate after creating a clone:

```
rsh Storage " df –Ag aggr1;"

Aggregate              total        used         avail        capacity
aggr1                  109GB        44GB         65GB         41%
aggr1/.snapshot        5GB          0GB          5GB          0%
```

The amount of space used in the aggregate did not increase — because space reservations are disabled for FlexClone volumes in Data ONTAP 7.1 and later. Also, check the LUN status using the following command:

```
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 the parent volume, but they are offline and not mapped to any igroup. We must create a new igroup with the WWPN number of Linux nodes that will host the clone database and then map new LUNs to it.

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;"
```

Make sure that the Linux node is connected to the same N series storage through fiber cables and follow all preinstallation OS activities for this node to host the clone database.

Log into the Linux node that will host the clone database as a root.

To discover the LUNs mapped to this Linux node, run the following command:

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

Check the newly discovered LUNs using following command:

```
vcsem64t3#> sanlun lun show all

filer    lun-pathname       device filename  adapter  protocol    lun size     lun state
Storage: /vol/oradata_clone/one  /dev/sdahost3     FCP    20g(21474836480)  GOOD
Storage: /vol/oralogs_clone/two  /dev/sdbhost3     FCP    10g(10737418240)  GOOD
```

For mounting these LUN devices as OCFS2 partitions, follow the steps mentioned in the section of this report about OCFS2 installation and configuration. The only difference would be to configure the OCFS2 cluster with a single node. Mount the OCFS2 partitions using the following command:

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

Refer to Oracle10*g* documentation to install Oracle10*g*R2 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 we will create a non-RAC clone database from the RAC database, we must create a new `controlfile`. Also, we must remove the parameters related to cluster database from the `init<sid>.ora` file. After starting the clone instance in the `nomount` stage, create a new control file, recover the database, and then open the database.

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:$ORACLE_HOME/rdbms/lib:$ORA
CLE_HOME/rdbms/lib32:$ORACLE_CRS/lib:$ORACLE_CRS/lib32:$ORACLE_CRS/rdbms/lib:$O
RACLE_CRS/rdbms/lib32:$LD_LIBRARY_PATH:/usr/lib64:/usr/lib:/lib:$ORACLE_HOME/or
acm/lib:/usr/local/lib
export LD_LIBRARY_PATH

LIBPATH=$ORACLE_HOME/lib:$ORACLE_HOME/lib32:$ORACLE_HOME/rdbms/lib:$ORACLE_HOME
/rdbms/lib32:$ORACLE_CRS/lib:$ORACLE_CRS/lib32:$ORACLE_CRS/rdbms/lib:$ORACLE_CR
S/rdbms/lib32:$LIBPATH:/usr/lib64:/usr/lib:/lib:$ORACLE_HOME/oracm/lib:/usr/loc
al/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_11g/ora
11g/product/11.1.0/crs_1/bin
export PATH
unset USERNAME
```

Trademarks and special notices

FlexClone, FlexVol, Data ONTAP, Network Appliance, the Network Appliance logo SnapMirror, SnapRestore and Snapshot are trademarks or registered trademarks of Network Appliance, Inc., in the U.S. and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

Information is provided "AS IS" without warranty of any kind.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.