VIRIDENT FLASHMAX II AND FLASHMAX CONNECT HIGH AVAILABILTY ORACLE REFERENCE ARCHITECTURE

Growing businesses increasingly rely on enterprise-class software applications such as databases for their long-term success. Logically, this leads to large investments in complex clustering and high availability technologies to stave off the business consequences of application downtime. Clustering at the software level can help, but it often requires specialized engineers and additional software licenses, which can increase costs. Clustering at the hardware level can mitigate these issues but still requires expensive storage arrays and associated storage area networks (SANs).

Viridient answers these concerns with the Viridient FlashMAX II PCIe storage device, and the FlashMAX Connect software suite, to offer low latency, highly available cluster technology using commodity server hardware, without the need for external storage arrays or SANs. When combined with Red Hat Enterprise Linux, Viridient FlashMAX II and FlashMAX Connect help businesses enjoy the benefits of a highly available solution, without the high cost of premium storage arrays and expensive high availability software features.

The purpose of this paper is threefold: to present the Viridient FlashMAX II and FlashMAX Connect, to show you the steps necessary to achieve a highly available Oracle architecture, and to discuss the failover capabilities that we verified in our labs.

AUGUST 2013
A PRINCIPLED TECHNOLOGIES TEST REPORT
Commissioned by Viridient

INTRODUCTION

Virident FlashMAX II & FlashMAX Connect

Virident FlashMAX II is an internal PCIe flash storage card that combines memory-class storage with high capacity. Businesses can choose from different models varying in capacity from 550 GB to 2,200 GB to meet their specific needs.

The FlashMAX Connect software suite works in conjunction with FlashMAX II PCIe storage to deliver a highly available storage solution for your critical applications. The three parts of the FlashMAX Connect suite:

- vHA, which conducts synchronous replication across servers
- vCache, which brings caching for SAN to FlashMAX II
- vShare, which allows shared access to FlashMAX II devices on other server nodes

With the combination of FlashMAX II and FlashMAX Connect, server-side storage can scale up or out, giving businesses maximum flexibility and enterprise-class availability.

Failover reliability in Flash storage

When the hardware systems that host your key business applications fail, any extended downtime or data loss can seriously affect business. Depending on the application and the environment, just a few minutes of downtime can lead to tens of thousands of dollars of lost revenue. The costs of data loss can go even higher. Virident FlashMAX II and vHA eliminate the risk of data loss and minimize downtime through synchronous data replication between primary and secondary servers ensuring that your data is always available on both sides of the cluster.

When used with an Oracle database, the Virident FlashMAX II and vHA cooperate to provide a highly available active/passive cluster configuration. The Oracle database runs from the primary FlashMAX II card, which is operating in Read/Write mode. The FlashMAX Connect software then almost instantaneously copies data to the FlashMAX II card in the secondary server. During a failover event, the system acts like any other clustering technology except that instead of using a single copy of the data on a shared SAN, the active server always points to the synchronized copy of the data on its own internal Virident FlashMAX II card. The switch between servers happens almost instantly, with only a brief pause of service while the database on the secondary server comes live. This process leaves all your data intact, now operating on the opposite cluster node.

Figure 1 shows a basic replication flow between two FlashMAX II cards.

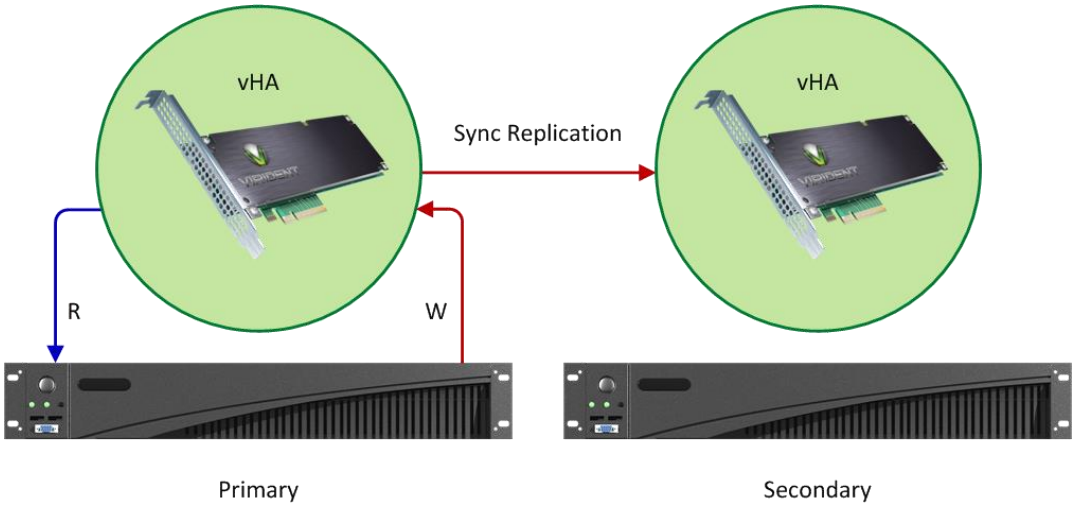

Figure 1: Overview of data replication using FlashMAX Connect vHA with FlashMAX II.

TESTING FAILOVER RECOVERY WITH THE VIRIDENT FLASHMAX II

We tested the vHA recovery software using Oracle Database 11*g* R2 and running a TPC-C-like workload using Dell Quest Benchmark Factory. This benchmark measures transactions per second (TPS) in an online transaction processing (OLTP) simulation. Our goal was not to develop a peak TPS number but instead to generate a reasonable database load to test the failover capabilities of the Virident FlashMAX II solution. We initialized a TPC-C database consisting of 3,000 warehouses, which resulted in approximately a 300GB database.

For our tests, we used two Dell™ PowerEdge™ R720 servers with dual Intel® Xeon® E5-2680 processors, each of which ran Red Hat Enterprise Linux 6. We attached one 550GB FlashMAX II card to each server and one 40Gbps InfiniBand® adapter in each server connected back-to-back. We did not install InfiniBand switches since they are not required for this solution. However, InfiniBand switches are supported. For detailed information on the systems we tested, see Appendix A.

Figure 2 shows our test run with the failover denoted mid-test. We experienced no application or data loss because the application was designed to automatically reconnect, and the workload experienced only a slight interruption during failover. The secondary server node picked up the workload of the first, and application performance continued with no degradation after the failover occurred.

For detailed testing information, see Appendix B.

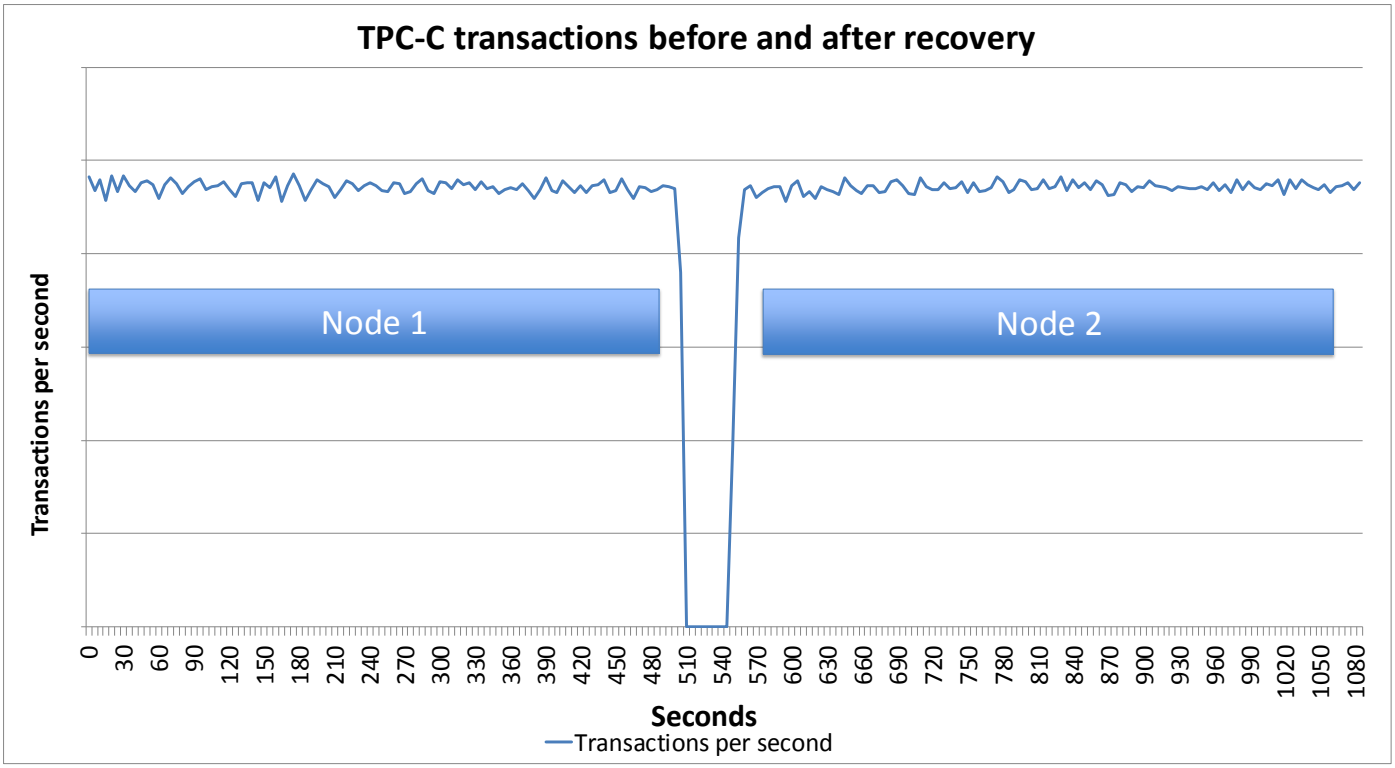

Figure 2: Performance before and after recovery.

Figure 3 shows the CPU utilization on each server before and after the recovery event. Again, there was no application or data loss, and the test continued through the failover simulation. Note the recovery server operated at the same utilization level as the primary server before the recovery event.

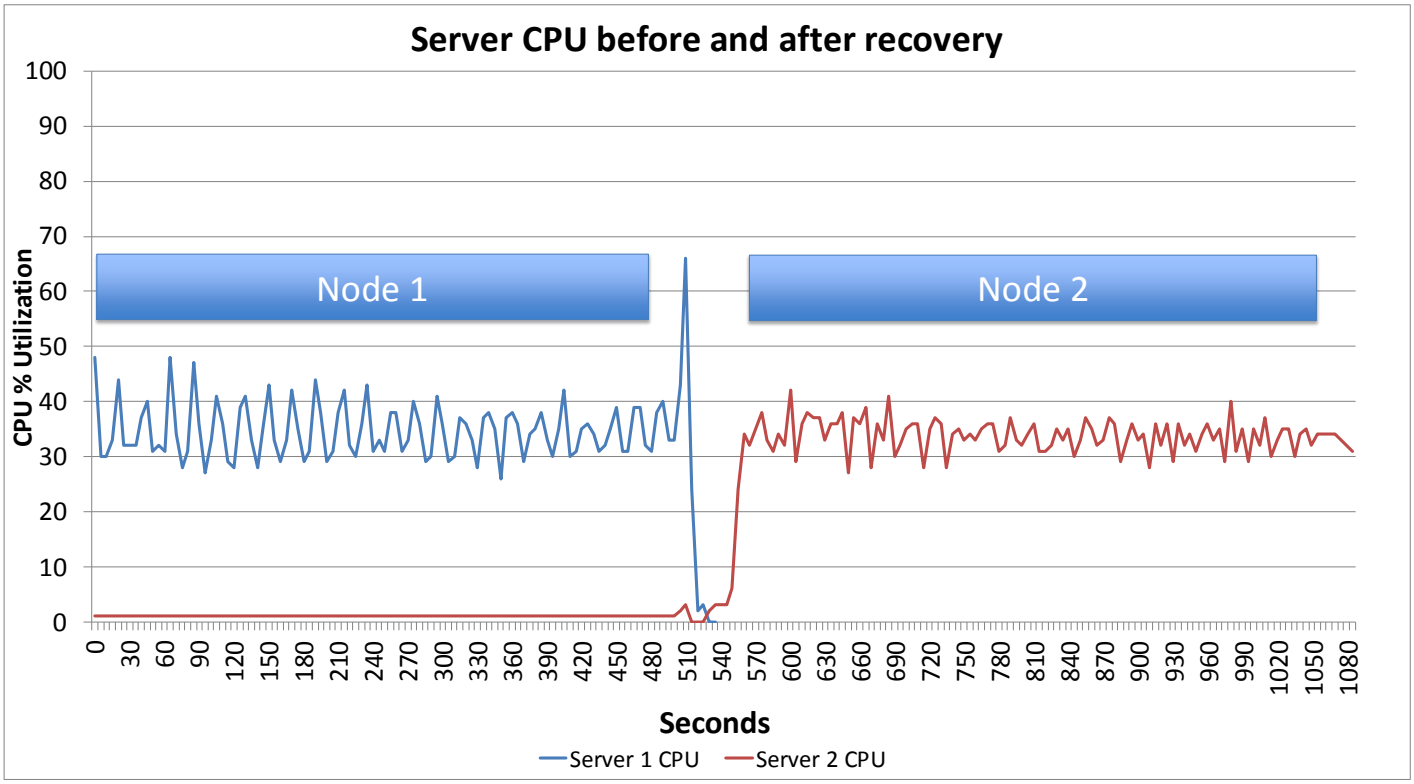

Figure 3: Processor utilization before and after recovery.

CONFIGURING VIRIDENT FLASHMAX II WITH ORACLE

In this section, we provide a general overview of the hardware and software installation process. FlashMAX II cards differ in storage capability, so choose which devices will best suit your business needs. Follow normal hardware safety precautions before installing the FlashMAX II devices.

For detailed installation instructions, see Appendix B.

Installing and configuring hardware

For a failover-capable setup, you need a minimum of two FlashMAX II cards and two QDR InfiniBand cards. In addition, you must have QDR-capable cabling. FlashMAX Connect vHA uses approximately 5GB of memory for every TB of Flash storage, so make sure that there is enough DRAM for your application and the caching driver. Uninstall any pre-existing FlashMAX II cards and their drivers before installation.

Installing Virident hardware

Installation of the hardware is identical for both servers. In both servers, install the same model of FlashMAX II cards into Gen 2 PCIe slots rated at least x8 or greater.

For best practices, also install the same model of InfiniBand card in both servers. FlashMAX Connect vHA requires the ability to connect between both servers via TCP/IP.

Installing InfiniBand

1. Connect to the InfiniBand network. FlashMAX Connect vHA requires an InfiniBand network connection between the two servers.
2. Install OpenFabrics Enterprise Distribution (OFED) InfiniBand drivers. Refer to the user's guide for instructions on downloading and installing the Mellanox OFED drivers.
3. Use OpenSMD service for back-to-back or unmanaged InfiniBand networks.
4. Disable firewall or allow vHA IP port communication.

For detailed installation of the InfiniBand device, see Appendix B.

Installing and configuring software

SELinux can cause issues that are hard to debug, so consider setting it to permissive or disabling it until you are comfortable with the setup and can properly configure access parameters for it.

Installing FlashMAX Connect requires root access and takes approximately 10 minutes to complete. FlashMAX Connect installation and installation of the base FlashMAX II drivers are similar.

Corosync and Pacemaker

Corosync and Pacemaker handle the cluster management in data replication. These tools, available in the Red Hat Enterprise Linux 6 High Availability channel, integrate with FlashMAX Connect vHA and Oracle to handle server failover. The Oracle database runs on the primary server. Data replication occurs on the secondary server. Corosync and Pacemaker automatically detect the primary server failure and start Oracle on the secondary server, which then becomes the active primary. Applications connect to this highly available Oracle instance through a virtual IP (VIP), which Corosync and Pacemaker manage and direct to the active primary server. See Figure 4 for general system architecture and see Appendix B for Corosync and Pacemaker installation. Note that currently, Pacemaker cluster resource manager software is offered as Technology Preview in Red Hat Enterprise Linux 6. Technology Preview features are unsupported, may not be functionally complete, and are not suitable for deployment in production.

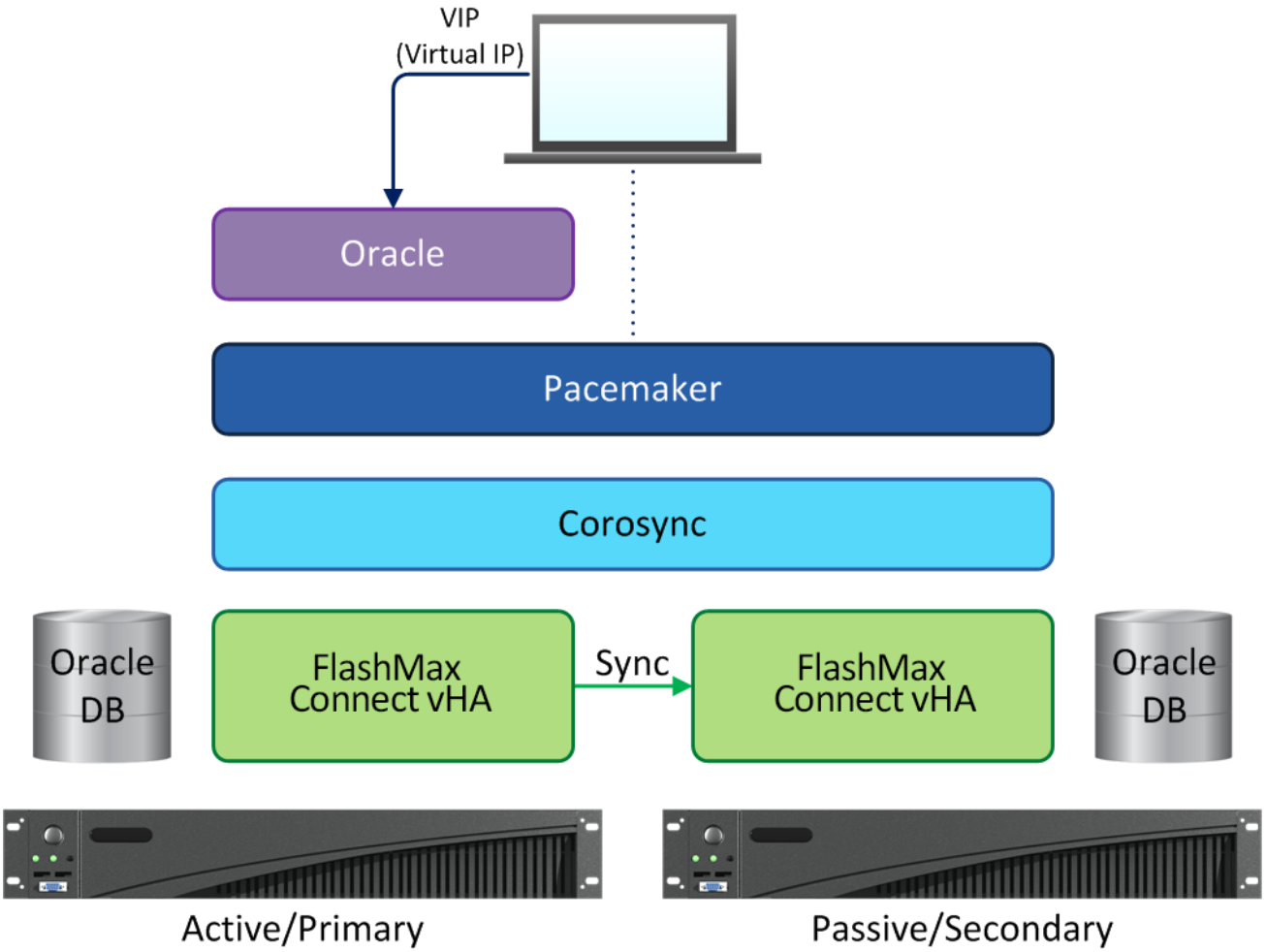

Figure 4: General system architecture.

Drivers and vHA namespace

Before setting up the Oracle database, your servers need the current FlashMAX Connect drivers and a vHA namespace on each. Follow these steps before installing the Oracle database:

1. Install the driver RPMs on both server nodes.
2. Verify that the FlashMAX II devices are installed and detected properly.
3. Load the drivers on the primary and secondary servers.
4. Create and configure a vHA namespace between primary and secondary servers.
5. After installing the drivers and prerequisites, you can begin the process to create a replicated vHA namespace.
6. Enable vHA on the partition.

7. Create the vHA namespace on the primary server. The vHA namespace contains the Oracle filesystem. A namespace is simply a portion of the FlashMAX II that is referenced by its name and UUID (universally unique identifier).
8. Create the vHA namespace on the secondary server and connect it. Use the UUID from the primary namespace.

For detailed installation of drivers and the vHA namespace setup, see Appendix B.

Installing and configuring Oracle Database 11*g* R2

Creating a highly available Oracle database ensures minimal or no I/O disruption. This requires configuration of Oracle involving cluster manager Config after cluster creation and failover to the secondary server. Do not configure the cluster manager before Oracle database has been set up and do not skip any steps. Before configuring High Availability for the database, follow these steps:

1. Set up the environment for Oracle on both servers.
2. Create a base Oracle directory on the primary server.
3. Install the Oracle database on the primary server.

For more details on Oracle installation, see "Deploying Oracle Database 11g R2 on Red Hat Enterprise Linux 6 – Best Practices."[1] See Appendix B for detailed information on how we installed and configured Oracle.

Configuring high availability for Oracle Database

Configure the Corosync cluster manager to manage both the Oracle instance for the primary server and the FlashMAX Connect cluster. For this configuration, three things must occur: replication of the secondary node, validation of Oracle functionality, and creation of primitives. When completed, Corosync will automatically mount the filesystem, migrate the system virtual IP, and start Oracle listener and Oracle Database on the cluster's primary server.

1. To create the cluster manually, the following steps are required to allow Corosync to manage the High Availability aspects. Complete the first step on the initial primary server, and complete the remaining steps on the secondary server (which becomes the new primary server after failover).

[1] www.redhat.com/rhecm/rest-rhecm/jcr/repository/collaboration/jcr:system/jcr:versionStorage/ee6fe0000a0526020f35498ae39e9939/14/jcr:frozenNode/rh:resourceFile

2. Create a FlashMAX Connect vHA namespace primitive.
3. Failover the cluster.
4. The secondary server is promoted to the primary server (the data is already replicated over).
5. Manually demote the former primary server.
6. Create and mount the base Oracle directory.
7. Create primitives on the new primary (formerly secondary) server.
    - Filesystem primitive
    - Virtual IP primitive
    - Oracle listener primitive
    - Oracle database primitive
8. Set up a Pacemaker management group to detect service failure.

Create the dependencies between all these so data migrates in the proper order. After completing these steps, Corosync will have enough information to manage failover and service continuity for Oracle on the FlashMAX Connect vHA namespace. See Appendix B for detailed information on configuring high availability for Oracle.

ORACLE HA CLUSTER FAILOVER (WITH CRM)

In normal operation, a client connects to the Oracle database currently active on the cluster using the virtual IP, which will route to the primary server. When this server suffers a failure, reboots, or is detected as dead, the cluster management daemon on the secondary server will perform a series of operations to promote it to the new primary. Service will be restored and the client will reconnect to Oracle over the same virtual IP and access all the same data, up to and including the last completed Oracle transaction.

In the 1.0 release of FlashMAX Connect vHA, manual intervention is required to reconnect it to the FlashMAX Connect vHA namespace and begin the resynchronization process once the failed machine restarts. In the 1.1 release of the software, this reconnection and automatic resynchronization happens without user intervention. Once begun, the resynchronization will proceed in the background without user intervention.

Updating FlashMAX Connect software in a running cluster

Periodically ensure that your servers have the most recent versions of the FlashMAX Connect software installed. Install any necessary updates on the secondary

server first, and then initiate a failover. When the remaining server becomes the secondary server after failover, update the FlashMAX Connect software on it.

IN CONCLUSION

Businesses looking to utilize highly available Oracle databases and protect their datacenters from loss of data during failover have a solution in Virident FlashMAX II and FlashMAX Connect software featuring vHA. The detailed steps in this report demonstrate the simplicity of installation for any organization implementing this hardware and software into their datacenter running Red Hat Enterprise Linux 6. Virident FlashMAX II with Red Hat Enterprise 6 and Oracle brings premium high performance and ultimate reliability for your database applications, without requiring an expensive storage array or costly high availability software.

APPENDIX A – SYSTEM CONFIGURATION INFORMATION

Figure 5 provides detailed configuration information for the test systems.

System	Dell PowerEdge R720
Power supplies	
Total number	2
Vendor and model number	Dell E1100E-S0
Wattage of each (W)	1100
Cooling fans	
Total number	6
Vendor and model number	AVC DBTC0638B2U
Dimensions (h x w) of each	2.5” x 2.5”
Volts	12
Amps	1.2
General	
Number of processor packages	2
Number of cores per processor	8
Number of hardware threads per core	2
System power management policy	Performance
CPU	
Vendor	Intel
Name	Xeon
Model number	E5-2680
Stepping	6
Socket type	FCLGA2011
Core frequency (GHz)	2.70
Bus frequency	8 GT/s
L1 cache	32 KB + 32 KB (per core)
L2 cache	256 KB (per core)
L3 cache	20 MB
Platform	
Vendor and model number	Dell PowerEdge R720
Motherboard model number	00W9X3
BIOS name and version	Dell 1.6.0
BIOS settings	Defaults
Memory module(s)	
Total RAM in system (GB)	64
Vendor and model number	Hynix HMT31GR7BFR4A-H9
Type	PC3-10600R
Speed (MHz)	1,333
Speed running in the system (MHz)	1,333
Timing/Latency (tCL-tRCD-tRP-tRASmin)	9-9-9-36
Size (GB)	8

System	Dell PowerEdge R720
Number of RAM module(s)	8
Chip organization	Double-sided
Rank	Dual
Operating system	
Name	Red Hat Enterprise Linux 6.2
File system	ext4
Kernel	2.6.32-220.el6.x86_64
Graphics	
Vendor and model number	Matrox G200eR
Graphics memory (MB)	8
Driver	Matrox Graphics Inc. 2.4.1.0 (09/08/2011)
RAID controller	
Vendor and model number	PERC H710P Mini
Firmware version	3.130.05-1796
Driver version	DELL 5.2.220.64 (06/18/2012)
Cache size (MB)	1 GB
Hard drives	
Vendor and model number	Dell MK3001GRRB
Number of drives	2
Size (GB)	73
RPM	15,000
Type	SAS 6 Gbps
Ethernet adapters	
Vendor and model number	Intel Gigabit 4P I350-t rNDC
Type	Integrated
Driver	igb 3.0.6-k
Optical drive(s)	
Vendor and model number	TEAC DV-28SW
Type	DVD-ROM
InfiniBand adapters	
Vendor and model number	Mellanox MCX353A-QCBT
Driver	OFED_1.5.3-3.1.0

Figure 5: General hardware configuration for our test systems.

APPENDIX B - HOW WE TESTED

We used the name R720-1 for the primary server and R720-2 for the secondary server in testing.

Software components

Figure 6 provides software component information for the test systems.

Software component	Version
Operating system	
Name	Red Hat Enterprise Linux 6.2
Kernel	2.6.32-220.el6.x86_64
Database	
Name	Oracle Database 11g Release 2 (x64)
Version	11.2.0.1
Cluster	
Pacemaker	1.1.6-3.el6.x86_64
Corosync	1.4.1-4.el6.x86_64
Virident	
FlashMax Connect	1.0-56277.V3.x86_54
InfiniBand adapters	
Mellanox	OFED 1.5.3-3.1.0 x86_64

Figure 6: General software configuration for our test systems.

Remove any existing FlashMAX II cards

Uninstall their drivers using the standard methods detailed in the FlashMAX II User's Guide (service vgcd stop; rpm -qa | grep vgc | xargs rpm -e).

Connect the InfiniBand network

Install the Mellanox Connect-X 2 or 3 cards in both the primary and secondary servers if you have not already done so.

Back-to-back connections (one primary server connected to another server) can exist directly between the two InfiniBand cards, or through an InfiniBand switch. If you are running a back-to-back connection, note the configuration of services after the OpenFabrics Enterprise Distribution (OFED) and InfiniBand driver installation.

OFED (Infiniband) driver installation

Refer to the user guide for instructions on downloading and installing the Mellanox OFED drivers.

1. To install a new system of Mellanox OFED drivers:

```
root@(Primary,Secondary) # yum -y install tk tcsh
```

```
...
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : 1:tcl-8.5.7-6.el6.x86_64                                    1/3
  Installing : 1:tk-8.5.7-5.el6.x86_64                                     2/3
  Installing : tcsh-6.17-19.el6_2.x86_64                                   3/3
Installed:
  tk.x86_64 1:8.5.7-5.el6 tcsh.x86_64 0:6.17-19.el6_2
Dependency Installed:
  tcl.x86_64 1:8.5.7-6.el6
Complete!
```

2. Download the appropriate ISO at www.mellanox.com/page/products_dyn?product_family=26.

```
root@(Primary,Secondary) # wget
http://www.mellanox.com/downloads/ofed/MLNX_OFED_LINUX-1.5.3-3.1.0-rhel6.2-
x86_64.iso
```

```
Resolving www.mellanox.com... 72.3.194.0
Connecting to www.mellanox.com|72.3.194.0|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 210358272 (201M) [application/octet-stream]
Saving to: "MLNX_OFED_LINUX-1.5.3-3.1.0-rhel6.2-x86_64.iso"
...
```

3. Mount it, and install the OFED drivers using the included script.

```
root@(Primary,Secondary) # mkdir /tmp/iso
root@(Primary,Secondary) # mount -o loop MLNX_OFED_LINUX-1.5.3-3.1.0-rhel6.2-
x86_64.iso /tmp/iso
root@(Primary,Secondary) # cd /tmp/iso
```

```
root@(Primary,Secondary) # ./mlnxofedinstall
```

```
This program will install the MLNX_OFED_LINUX package on your machine.
Note that all other Mellanox, OEM, OFED, or Distribution IB packages will be
removed.
Do you want to continue?[y/N]: y
Starting MLNX_OFED_LINUX-1.5.3-3.0.0 installation ...
Installing mlnx-ofa_kernel RPM
Preparing...                ##################################################
mlnx-ofa_kernel             ##################################################
Installing kmod-mlnx-ofa_kernel RPM
Preparing...                ##################################################
kmod-mlnx-ofa_kernel        ##################################################
Installing mlnx-ofa_kernel-devel RPM
Preparing...                ##################################################
mlnx-ofa_kernel-devel       ##################################################
...
INFO: updating ...
Installation finished successfully.

Configuring /etc/security/limits.conf.
Please reboot your system for the changes to take effect.
```

```
root@(Primary,Secondary) # cd /root
root@(Primary,Secondary) # umount /tmp/iso
root@(Primary,Secondary) # reboot
```

Enable opensmd service for back-to-back or unmanaged Infiniband networks

For back-to-back Infiniband connections, use `chkconfig` and `service` to enable the InfiniBand subnet manager, or opensmd server, on both the primary and secondary servers. The `service` command is required only once.

```
root@(Primary, Secondary) # chkconfig opensmd on
root@(Primary, Secondary) # service opensmd start
```

```
Starting IB Subnet Manager.                                [  OK  ]
```

Disable firewall or allow vHA IP port communication

FlashMAX Connect vHA requires the ability to connect between the primary and secondary servers via TCP/IP. Consider completely disabling the firewalls by stopping iptables and ensuring it does not start up at boot time. Open

port 30000 for vHA, and 5405 and 4000 (for Corosync) on both the primary and secondary servers for UDP and TCP connections.

```
root@(Primary,Secondary) # service iptables stop
```

```
iptables: Flushing firewall rules:                         [  OK  ]
iptables: Setting chains to policy ACCEPT: filter          [  OK  ]
iptables: Unloading modules:                               [  OK  ]
```

```
root@(Primary,Secondary) # chkconfig iptables off
root@(Primary,Secondary) # service ip6tables stop
```

```
ip6tables: Flushing firewall rules:                        [  OK  ]
ip6tables: Setting chains to policy ACCEPT: filter         [  OK  ]
ip6tables: Unloading modules:                              [  OK  ]
```

```
root@(Primary,Secondary) # chkconfig ip6tables off
```

Install Corosync and Pacemaker

Corosync and Pacemaker are part of the Red Hat Enterprise Linux High Availability Add-On. The systems must be subscribed to the Red Hat Enterprise Linux High Availability/Clustering channel in order to install these tools.

1. Execute `yum install pacemaker` to install both applications and their prerequisites. This will install corosync as well.

```
root@(Primary,Secondary) # yum install pacemaker
```

Create and configure corosync.conf

After installation, configure the services on both servers starting with the primary server.

1. Copy /etc/corosync/corosync.conf.example to /etc/corosync/corosync.conf, and then configure the appropriate network and multicast addresses.

```
# Please read the corosync.conf.5 manual page
compatibility: whitetank
totem {
        version: 2
        secauth: off
        threads: 0
       token: 10000
        token_retransmits_before_loss_const: 10

        interface {
                ringnumber: 0
                bindnetaddr: 172.16.72.0
                mcastaddr: 226.94.1.1
                mcastport: 5405
                ttl: 1
        }
}

logging {
        fileline: off
        to_stderr: no
        to_logfile: yes
        to_syslog: yes
        logfile: /var/log/cluster/corosync.log
        debug: off
        timestamp: on
        logger_subsys {
                subsys: AMF
                debug: off
        }
}

amf {
        mode: disabled
}
```

Set up the Corosync service

Pacemaker does not install into the Corosync configuration.

1. Create the file /etc/corosync/service.d/pcmk and add:

```
service {
      # Load the Pacemaker Cluster Resource Manager
      name: pacemaker
      ver: 1
}
```

2. After configuring the files /etc/corosync/corosync.conf and /etc/corosync/service.d/pcmk, replicate them from the primary server to the secondary server using SCP or another file transfer.

Enable Corosync and Pacemaker

After configuration, enable the two services on both the primary server and the secondary server. The server that has them enabled first will become the primary.

```
root@(Primary,Secondary) #  chkconfig corosync on
root@(Primary,Secondary) #  service corosync start
root@(Primary,Secondary) #  chkconfig pacemaker on
root@(Primary,Secondary) #  service pacemaker start
```

Disable SELinux

Disable SELinux until you are comfortable with the setup and can properly configure access parameters for it:

```
root@(Primary,Secondary) # vim /etc/selinux/config
```

```
...
SELINUX=disabled
```

```
root@(Primary,Secondary) # echo 0 >/selinux/enforce
```

Install the driver RPMs on the primary and secondary nodes

Install the driver and utilities RPMs using RPM standard commands. For best practices, include --nodeps on the command line:

```
root@(Primary,Secondary) # rpm -ivh --nodeps *.rpm
```

```
Preparing...                ########################################### [100%])
   1:vgc-utils              ########################################### [ 33%])
   2:vgc-tools              ########################################### [ 67%])
   3:kmod-vgc-2.6.32-220.el########################################### [100%])
   4: vgc_rdma-2.6.32-220.el6########################################## [100%]
```

Ignore warnings about missing kernel requirements.

Verify installation of the FlashMAX II device

1. After the initial booting of the primary system with the installed device, use `lspci` to verify that the PCI subsystem detects the device and initializes it properly. There should be one device listed for each device installed.
2. If a device is not detected, power down the server, and ensure that it is seated properly.

```
root@(Primary,Secondary) # lspci -d 1a78:
```

```
42:00.0 FLASH memory: Virident Systems Inc. Device 0040 (rev 01)
```

Load the drivers on the primary and secondary servers

1. Run `vgcd start` to load drivers that allow configuring and using FlashMAX Connect.
2. Reset the server.

```
root@(Primary,Secondary) # service vgcd start
```

3. Load the drivers on both the primary and secondary servers.
4. Run `vgc-monitor -d /dev/vgca` to check the status of the card. vHA are listed as disabled. Before using the partition as vHA, a special configuration step is required.

```
root@(Primary,Secondary) # vgc-monitor -d /dev/vgca
```

```
vgc-monitor: FlashMAX Connect Software Suite 1.0(53313.V3)

Driver Uptime: 3:01
Card_Name      Num_Partitions      Card_Type          Status
/dev/vgca      1                   VIR-M2-LP-2200-2A Good

  Serial Number      : SJT00124
  Card Info          : Part: SJT00124
                       Rev : FlashMax V2 47955, module 47956, x8 Gen2
  Temperature        : 52 C (Safe)
  Card State Details : Normal
  Action Required    : None

  Partition      Usable_Capacity     RAID      vCache    vHA       vShare
  /dev/vgca0     2222 GB             enabled   disabled  disabled  disabled

    Mode                          : maxcapacity
    Total Flash Bytes             : 936312772315136 (936.31TB) (reads)
                                    1338814879399936 (1.34PB) (writes)
    Remaining Life                : 96.54%
    Partition State               : READY
    Flash Reserves Left           : 99.97%
```

Enable vHA on the partition

Enable vHA on the partition via `vgc-config`. Only `--enable-vha` is required.

```
root@Primary,Secondary # vgc-config -f -p /dev/vgca0 --enable-vha
```

```
vgc-config: FlashMAX Connect Software Suite 1.0(52876.V3)

*** Formatting drive.  Please wait... ***
root@Primary,Secondary # vgc-monitor -d /dev/vgca
vgc-monitor: FlashMAX Connect Software Suite 1.0(53313.V3)

Driver Uptime: 3:45
Card_Name       Num_Partitions      Card_Type          Status
/dev/vgca       1                   VIR-M2-LP-2200-2A Good

  Serial Number       : SJT00124
  Card Info           : Part: SJT00124
                        Rev : FlashMax V2 47955, module 47956, x8 Gen2
  Temperature         : 52 C (Safe)
  Card State Details : Normal
  Action Required     : None


  Partition       Usable_Capacity     RAID       vCache     vHA        vShare
  /dev/vgca0      2222 GB             enabled    disabled   enabled    disabled

    Mode                           : maxcapacity
    Total Flash Bytes              : 936312786937856 (936.31TB) (reads)
                                     1338814948589568 (1.34PB) (writes)
    Remaining Life                 : 96.53%
    Partition State                : READY
    Flash Reserves Left            : 99.96%
```

Create the vHA namespace on the primary server

The configuration and management utility for vHA is `vgc-vha-config`. Use this utility to create, assign roles to, delete, and modify vHA namespaces. The active I/O will run on the primary server at all times, with synchronous replication to the secondary server handled by FlashMAX Connect. The listed UUID is required to create the secondary peer for vha.

Run `vgc-vhamonitor –list` to verify the creation of the vHA namespace.

```
root@Primary,Secondary # vgc-vha-config –h
```

```
vgc-vha-config: FlashMAX Connect Software Suite 1.0(53313.V3)

Usage: vgc-vha-config <command> [command options]
           --create --role <role> --peer <peername> --size <vha-dev-size in
GB>
                    --uuid <vha-uuid> <vha-name> <backing-dev>
           --start <vha-name> <backing-dev>
           --stop <vha-name> <backing-dev>
           --promote <vha-name> <backing-dev>
           --demote <vha-name> <backing-dev>
           --replace-peer --peer <peername> <vha-name> <backing-dev>
           --connect <vha-name> <backing-dev>
           --disconnect <vha-name> <backing-dev>
           --delete <vha-name> <backing-dev>
           --delete-all
           --help
```

The role can be either primary or secondary. Since creation of the primary generates the UUID , use the same UUID to create the secondary. The `--replace-peer` option is currently not supported.

Create the vHA namespace on the secondary server and connect it

On the secondary system, use the previous UUID and `vgc-vha-config` to create a connection.

1. Specify the hostname of the primary server (or its IPv4 dot address), the UUID shown, and the vHA name.

   The size specified must also match that of the primary server.

2. Use `vgc-vha-monitor --list` to verify vHA creation and connection.

A successful creation and connection will show on the secondary as Connected. The vHA namespaces now provide synchronous mirroring, allowing the cluster configuration to continue.

```
root@Primary # vgc-vha-config --create --role primary --peer R720-2 --size 2222
vha1 /dev/vgca0
```

```
vgc-vha-config: FlashMAX Connect Software Suite 1.0(55607.V3)

Hostname = R720-2, Address = 172.16.72.124
Namespace creation successful : 1
vHA creation success: vha (250:0)
Done.
root@primary #vgc-vha-monitor
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(55607.V3)
root@Primary # vgc-vha-monitor --list
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(55607.V3)
--------------------------------------------------------------------------------
----
 vHA Device          Local Device      Role          State          UUID
--------------------------------------------------------------------------------
----
 /dev/vgca0_vha1     /dev/vgca0        primary       Disconnected
7ebb5240-...
```

Examine the namespace in more detail

1. Use the `--detail` option on the primary or secondary server to report the current state and configuration. The vHA State is slave for both the primary and the secondary and will remain in that state until the crm is configured.

```
root@(Primary,Secondary) # vgc-vha-monitor --detail /dev/vgca0_vha1
```

```
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53313.V3)

 vHA Device         Local Device       Role           State
 /dev/vgca0_vha1    /dev/vgca0         primary        Disconnected

  vHA Configuration:
    Cluster UUID                : 7ebb5240-…
    Capacity                    : 1800 GB
    Peer IP addr                : 172.16.72.120
    Peer hostname               : R720-1
  IO Statistics:
    Total Bytes Written         : 0 (0 GB)
    Total Bytes Read            : 135168 (0 GB)
    Total Read Errors           : 0
    Total Write Errors          : 0
  vHA State:
    Master-Slave                : slave
    Connection Score            : 3
    Nodes in sync               : 1
  vHA Statistics:
    Total bytes mirrored        : 0 (0 GB)
    Total bytes resynced        : 0 (0 GB)
    Bytes resynced last         : 0 (0 GB)
```

2. Use the --uuid option of vgc-vha-monitor to retrieve only the UUID of the cluster for use in scripted applications.

```
root@(Primary,Secondary) # vgc-vha-monitor --get-uuid /dev/vgca0_vha1
```

```
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53313.V3)

7ebb5240-b565-4330-b74b-97298ab1e28d
```

Configure the primary Oracle instance on the primary server

Setting up the environment for Oracle (on both the primary and the secondary servers)

There are several RPMs needed to run the Oracle Universal Installer, an X Window application.

1. Run these commands on both the primary and secondary server to install RPMs.

```
# yum install tuned
```

```
# chkconfig tuned on
# tuned-adm profile enterprise-storage

# yum install –y binutils-2*x86_64* glibc-2*x86_64* glibc-2*i686* nss-softokn-
freebl-3*x86_64* nss-softokn-freebl-3*i686* compat-libstdc++-33*x86_64* glibc-
common-2*x86_64* glibc-devel-2*x86_64* glibc-devel-2*i686* glibc-headers-
2*x86_64* elfutils-libelf-0*x86_64* elfutils-libelf-devel-0*x86_64* gcc-4*x86_64*
gcc-c++-4*x86_64* ksh-*x86_64* libaio-0*x86_64* libaio-devel-0*x86_64* libaio-
0*i686* libaio-devel-0*i686* libgcc-4*x86_64* libgcc-4*i686* libstdc++-4*x86_64*
libstdc++-4*i686* libstdc++-devel-4*x86_64* make-3.81*x86_64* numactl-devel-
2*x86_64* sysstat-9*x86_64* compat-libstdc++-33*i686* compat-libcap*
# yum install -y unixODBC unixODBC-devel unixODBC.i686 unixODBC-devel.i686
```

2. Set the Hugepages and shared memory. These are recommended settings for Oracle installation.

```
cat > sysctl.add <<EOF
####ORACLEBEGIN
vm.nr_hugepages=512
vm.swappiness=0
vm.dirty_background_ratio=3
vm.dirty_ratio=15
vm.dirty_expire_centisecs=500
vm.dirty_writeback_centisecs=100
kernel.shmall=25165824
kernel.shmmax=51539607552
kernel.shmmni=4096
kernel.sem = 250 32000 100 128
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
fs.file-max = 6815744
fs.aio-max-nr = 1048576
####ORACLEEND
EOF
```

3. Edit the /etc/security/limits.conf, /etc/sysctl.conf and /etc/hosts, and load the settings.

```
cat > limits.add <<EOF
####ORACLEBEGIN
oracle soft nproc 2047
```

```
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
oracle soft stack 10240
oracle hard stack 32768
####ORACLEEND
EOF
cat sysctl.add >> /etc/sysctl.conf
cat limits.add >> /etc/security/limits.conf
sysctl –p
```

4. Create the required Oracle groups for proper installation and configuration.

```
# groupadd -g 1001 oinstall ## software inventory
# groupadd -g 1002 dba ## database
# groupadd -g 1003 oper ## database
# groupadd -g 1004 asmadmin ## ASM, if needed
# groupadd -g 1005 asmdba ## ASM, if needed
# groupadd -g 1006 asmoper ## ASM, if needed
# useradd -u 1002 -g oinstall -G dba,oper,asmadmin,asmdba,asmoper oracle
# passwd oracle
```

5. Stop updating the secondary server, and continue to setup the primary server.

Create base Oracle directory on the primary server

By default, this location is /u01/.

```
# mkdir –p /u01
```

Create the filesystem for Oracle's datadir on the primary server

6. Use the standard `mkfs utility` on the /dev/vgca0_vha device node to create a FlashMAX Connect vHA namespace filesystem on the primary node.

```
root@Primary # mkfs -t ext4 /dev/vgca0_vha
```

```
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
109871104 inodes, 439453125 blocks
21972656 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
13412 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
        102400000, 214990848

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

This filesystem will be automatically checked every 21 mounts or 180 days. Use `tune2fs -c` or `-i` to override.

Create the rest of the base directories for Oracle on the primary server

1. Mount the created file system to /u01/ using the standard mount command.
2. After mounting, change the owner and group to Oracle. This is required for Oracle installation and access.

```
root@Primary # mount /dev/vgca0_vha /u01/

root@Primary # mount /dev/vgca0_vha on /u01/ type ext4 (rw)
root@Primary # mkdir -p /u01/app/oracle

root@Primary # chown -R oracle:oinstall /u01

root@Primary # chmod -R 775 /u01
```

Install the Oracle database on the primary server

Install and launch the vnc server to run the Oracle installer as an Oracle user.

1. Run the command:

```
yum install -y twm tigervnc-server pixman pixman-devel libXfont xterm xorg-x11-utils.
```

2. Start the vnc session as the Oracle user, and launch the Oracle 11g installer.
3. Choose the options stand-alone, enterprise server and filesysytem.

After installation, validate that the install is functional.

4. Log into the primary server as `root` and then export the Oracle paths. Add $ORACLE_HOME/bin to the path.

```
export ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
export ORACLE_SID=orcl
PATH=$PATH:$ORACLE_HOME/bin
```

5. For best practices, save this to .bash_profile.
6. Test access to the database, and shut down the vnc server.
7. Edit the /etc/oratab file, and set Y.

The database needs this to start. Oracle utilities uses this file. It is created by root.sh. Multiple entries with the same ORACLE_SID are not allowed.

```
orcl:/u01/app/oracle/product/11.2.0/dbhome_1:Y
```

8. Use `lsnrctl status` to check if the listener is running.
9. If it is not running, start it with `lsnrctl start`, and check the status again.

```
LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 24-APR-2013 09:15:47

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

Starting /u01/app/oracle/product/11.2.0/dbhome_1//bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 11.2.0.1.0 - Production
System parameter file is
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/tm07/listener/alert/log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=127.0.0.1)(PORT=1521)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 11.2.0.1.0 - Production
Start Date                24-APR-2013 09:15:49
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/tm07/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=127.0.0.1)(PORT=1521)))
The listener supports no services
The command completed successfully
```

10. Check the log files for errors.
11. Start the database with dbstart.

```
ORACLE_HOME_LISTNER is not SET, unable to auto-start Oracle Net Listener
Usage: /u01/app/oracle/product/11.2.0/dbhome_1/bin/dbstart ORACLE_HOME
Processing Database instance "orcl": log file
/u01/app/oracle/product/11.2.0/dbh
ome_1/startup.log

[root@tm07 ~]# vi /u01/app/oracle/product/11.2.0/dbhome_1/startup.log

/u01/app/oracle/product/11.2.0/dbhome_1/bin/dbstart: Starting up database
"orcl"
Wed Apr 24 09:11:42 PDT 2013

SQL*Plus: Release 11.2.0.1.0 Production on Wed Apr 24 09:11:42 2013

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

SQL> Connected to an idle instance.
SQL> ORACLE instance started.

Total System Global Area 1085640704 bytes
Fixed Size                  2212536 bytes
Variable Size             654314824 bytes
Database Buffers          419430400 bytes
Redo Buffers                9682944 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 11g Enterprise Edition Release
11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

/u01/app/oracle/product/11.2.0/dbhome_1/bin/dbstart: Database instance "orcl"
warm started.
```

12. Note: If you receive an error that the memory on the server is not supported: Check the /dev/shm directory and make sure it is empty.
13. Create a user for accessing the created database.

To verify that the client can access the database, use an ODBC client. Red Hat ODBC support documentation can be located here:
https://access.redhat.com/site/documentation/en-US/JBoss_Enterprise_Data_Services/5/html/Data_Services_Client_Developer_Guide/chap-ODBC_Support.html

Validation of the connection is client (using ODBC) using the primary IP

1. Add the primary node IP and SID to tnsnames.ora, and configure ODBC to access the database from the client.
2. Edit the tnsnames.ora.

```
ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 172.16.72.120)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )
```

Create an ODBC entry for the database

1. Consult Red Hat documentation on creating ODBC client configurations and test the connection.
2. Once validated, shut down the database, and unmount /u01/.
3. If the volume is busy, run `lsof (8)`, kill the Oracle processes, and unmount the /u01/.

```
root@primary# dbshut
```

```
Processing Database instance "orcl": log file
/u01/app/oracle/product/11.2.0/dbhome_1/shutdown.log
[root@primary ~]# vi /u01/app/oracle/product/11.2.0/dbhome_1/shutdown.log

SQL*Plus: Release 11.2.0.1.0 Production on Wed Apr 24 09:43:53 2013

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

SQL> Database instance "orcl" shut down.
root@primary# umount /u01/
```

Configure high availability for Oracle

Create a FlashMAX Connect vHA cluster and namespace primitive

The `vgc-cm-config` utility will add the two nodes and the vHA namespace to the Corosync cluster, in addition to the vHA namespace. Along with the creation and connection of the cluster, this will create the namespace primitive.

1. Configure the cluster using your cluster's UUID.

```
root@Primary # vgc-cm-config -c 7ebb5240-b565-4330-b74b-97298ab1e28d
```

2. Create a cluster setup with the parameters:

`cluster_uuid - 7ebb5240-b565-4330-b74b-97298ab1e28d, node1 - R720-1, node2 - R720-2a`

```
Proceed? [y/n]: y
Successfully created the cluster
Successfully started the cluster
```

```
root@Primary # vgc-cm-config -l
 Master/Slave Set: ms-7ebb5240-b565-4330-b74b-97298ab1e28d \
                          [vha-7ebb5240-b565-4330-b74b-97298ab1e28d]
     Masters: [ R720-1 ]
     Slaves: [ R720-2 ]
```

3. Use `vgc-cm-config` to validate the state and use `crm_mon` to validate the state of the cluster.

```
root@Primary # crm_mon
```

```
============
Last updated: Mon Mar 18 07:29:29 2013
Last change: Mon Mar 18 00:18:51 2013 via cibadmin on R720-1
Stack: openais
Current DC: R720-1 - partition with quorum
Version: 1.1.7-6.el6-148fccfd5985c5590cc601123c6c16e966b85d14
2 Nodes configured, 2 expected votes
5 Resources configured.
============
Online: [ R720-2 R720-1 ]
Master/Slave Set: ms-7ebb5240-b565-4330-b74b-97298ab1e28d
                        [vha-7ebb5240-b565-4330-b74b-97298ab1e28d]
     Masters: [ R720-1 ]
     Slaves: [ R720-2 ]
```

vHA cluster failover

Once the failed machine recovers, manually reconnect it to the FlashMAX Connect vHA namespace and begin the resynchronization process. Resynchronization will then run in the background without user intervention.

The process of manual reconnection: Immediately and automatically, the secondary server becomes the new primary. After the former primary reboots, demote it using `vgc-vha-config --demote` to restore HA functionality to the cluster. After demotion of the former primary to secondary, re-synchronization starts automatically. The cluster is back in a highly available state once resynchronization is complete.

Base state of the primary and secondary before failover

1. Use `crm_mon` to check the state of the services on the cluster with both hosts online.

```
root@Primary  # crm_mon
```

```
============
Last updated: Thu Mar 21 14:29:28 2013
Last change: Thu Mar 21 10:02:48 2013 via crmd on R720-2
Stack: openais
Current DC: R720-2 - partition with quorum
Version: 1.1.7-6.el6-148fccfd5985c5590cc601123c6c16e966b85d14
2 Nodes configured, 2 expected votes
5 Resources configured.
============
Online: [ R720-2 R720-1 ]
Master/Slave Set: ms-7ebb5240-b565-4330-b74b-97298ab1e28d [vha-7ebb5240-b565-
4330-b74b-97298ab
1e28d]
     Masters: [ R720-2 ]
     Slaves: [ R720-1 ]
```

2. Use `vgc-vha-monitor` to check that the base state of the primary is connected and that the FlashMAX Connect vHA namespace state is secondary and connected.

```
root@Primary(R720-1)#  vgc-vha-monitor
```

```
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53741.V3)

  ------------------------------------------------------------------------------
  vHA Device          Local Device        Role           State          UUID
  ------------------------------------------------------------------------------
  /dev/vgca0_vha      /dev/vgca0          primary        Connected      7ebb...
```

```
root@Secondary(R720-2)# vgc-vha-monitor
```

```
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53741.V3)

  ------------------------------------------------------------------------------
  vHA Device          Local Device        Role           State          UUID
  ------------------------------------------------------------------------------
  /dev/vgca0_vha     /dev/vgca0          secondary      Connected      7ebb...
```

Initiate the failover, reboot from the primary

1. Reboot on the current primary server to force failover from the primary server to the secondary.

The former primary must be manually demoted using `vgc-vha-config` with `--demote` option to restore the cluster back to the HA connected state.

```
root@Primary(R720-1) # reboot
```

```
Broadcast message from root@R720-1
        (/dev/pts/1) at 8:33 ...

The system is going down for reboot NOW!
```

2. Use Connection Score reported by `vgc-vha-monitor –detail` to verify which server should be demoted.
3. Demote the server with the lower connection score.

This connection score is a monotonically increasing value kept by the FlashMAX Connect driver on every connection and promotion or demotion.

```
root@newPrimary (R720-2) # vgc-vha-monitor --detail /dev/vgca0_vha
```

```
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53741.V3)

  vHA Device          Local Device        Role           State
  /dev/vgca0_vha      /dev/vgca0          primary        Disconnected

vHA Configuration:
    vHA UUID                    : 7ebb5240-b565-4330-b74b-97298ab1e28d
    Capacity                    : 2222 GB
    Peer IP addr                : 172.16.72.120
    Peer hostname               : R720-1
  IO Statistics:
    Total Bytes Written         : 241664 (0 GB)
    Total Bytes Read            : 8902656 (0 GB)
    Total Read Errors           : 0
    Total Write Errors          : 0
  vHA State:
    Master-Slave                : master
    Connection Score            : 21
    Nodes in sync               : 1
  vHA Statistics:
    Total bytes mirrored        : 0 (0 GB)
    Total bytes resynced        : 0 (0 GB)
    Bytes resynced last         : 0 (0 GB)
```

4. Repeat this process for the other server to note its connection score:

```
root@oldPrimary (R720-1) # vgc-vha-monitor --detail /dev/vgca0_vha
```

```
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53741.V3)

  vHA Device          Local Device        Role           State
  /dev/vgca0_vha      /dev/vgca0          primary        Disconnected

  vHA Configuration:
    vHA UUID                    : 7ebb5240-b565-4330-b74b-97298ab1e28d
    Capacity                    : 2222 GB
    Peer IP addr                : 172.16.72.124
    Peer hostname               : R720-2
  IO Statistics:
    Total Bytes Written         : 0 (0 GB)
    Total Bytes Read            : 414720 (0 GB)
    Total Read Errors           : 0
    Total Write Errors          : 0
  vHA State:
    Master-Slave                : slave
    Connection Score            : 20
    Nodes in sync               : 0
  vHA Statistics:
    Total bytes mirrored        : 0 (0 GB)
    Total bytes resynced        : 0 (0 GB)
    Bytes resynced last         : 0 (0 GB)
```

Manual demotion of the new secondary

Use `vgc-vha-config --demote` on the new secondary (former primary) to connect the FlashMAX Connect vHA namespace and begin automatic resynchronization.

```
root@newSecondary (R720-1) # vgc-vha-config --demote --force /dev/vgca0_vha
```

```
vgc-vha-config: FlashMAX Connect Software Suite 1.0(53741.V3)

Modify role success
Done.
```

Secondary resynchronization progresses automatically

The secondary server will resynchronize with the primary to get the most recent data. Only updated portions of the data will resynchronize. During this process, the data and services on the primary are fully accessible.

Use `vgc-vha-monitor` to see the resynchronization state.

```
root@newSecondary (R720-1) # vgc-vha-monitor
```

```
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53741.V3)
  ------------------------------------------------------------------------------
   vHA Device        Local Device             Role            State          UUID
  ------------------------------------------------------------------------------
  /dev/vgca0_vha      /dev/vgca0              secondary       Resync         7ebb...
```

Validate that the cluster is now in the HA connected state

Time to complete synchronization depends on the amount of data that updated during the server downtime.

1. Run `vgc-vha-monitor` to verify that the vHA cluster shows the primary/secondary connected state.

```
root@newSecondary (R720-1) # vgc-vha-monitor
```

```
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53741.V3)
  ------------------------------------------------------------------------------
  vHA Device         Local Device         Role            State          UUID
------------------------------------------------------------------------------
  /dev/vgca0_vha     /dev/vgca0           secondary        Connected      7ebb...
```

```
root@newPrimary (R720-2) # vgc-vha-monitor
```

```
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53741.V3)
  ------------------------------------------------------------------------------
  vHA Device         Local Device          Role            State          UUID
  ------------------------------------------------------------------------------
  /dev/vgca0_vha     /dev/vgca0          primary         Connected      7ebb...
```

2. Use crm_mon to validate the state of the nodes.

The output should reflect the new primary and secondary.

```
root@Primary  # crm_mon
```

```
============
Last updated: Thu Mar 21 14:29:28 2013
Last change: Thu Mar 21 10:02:48 2013 via crmd on R720-2
Stack: openais
Current DC: R720-1 - partition with quorum
Version: 1.1.7-6.el6-148fccfd5985c5590cc601123c6c16e966b85d14
2 Nodes configured, 2 expected votes
5 Resources configured.
============
Online: [ R720-2 R720-1 ]
Master/Slave Set: ms-7ebb5240-b565-4330-b74b-97298ab1e28d [vha-7ebb5240-b565-
4330-b74b-97298ab
1e28d]
     Masters: [ R720-2 ]
     Slaves: [ R720-1 ]
```

3. Mount the /u01 directory, and assign ownership to Oracle.

```
root@Primary # mount /dev/vgca0_vha /u01/
root@Primary # mount /dev/vgca0_vha1 on /u01/ type ext4 (rw)
root@Primary # chown -R oracle:oinstall /u01
root@Primary # chmod -R 775 /u01
```

4. Run `root.sh` (/u01/app/oracle/product/11.2.0/dbhome_1/root.sh) to complete the Oracle install.
5. Edit the /etc/oratab file to ensure that the parameter set is Y.

```
# This file is used by ORACLE utilities.  It is created by root.sh
[…]
# Multiple entries with the same $ORACLE_SID are not allowed.
orcl:/u01/app/oracle/product/11.2.0/dbhome_1:Y
```

Start and validate the Oracle database on the new primary

1. Run `lsnrctl status` to check if the listener is running.
2. If it is not running, start it with `lsnrctl start`, and check the status again.
3. Check the log files for errors.

```
[oracle@primary ~]$ lsnrctl start
```

```
LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 24-APR-2013 09:15:47

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

Starting /u01/app/oracle/product/11.2.0/dbhome_1//bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 11.2.0.1.0 - Production
System parameter file is
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/tm07/listener/alert/log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=127.0.0.1)(PORT=1521)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 11.2.0.1.0 - Production
Start Date                24-APR-2013 09:15:49
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/tm07/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=127.0.0.1)(PORT=1521)))
The listener supports no services
The command completed successfully
```

4. Start the database with dbstart.

5. If an error appears stating that the memory on the server is not supported, check the /dev/shm directory to verify that it is empty.

```
[oracle@primary ~]$ dbstart
```

```
ORACLE_HOME_LISTNER is not SET, unable to auto-start Oracle Net Listener
Usage: /u01/app/oracle/product/11.2.0/dbhome_1/bin/dbstart ORACLE_HOME
Processing Database instance "orcl": log file
/u01/app/oracle/product/11.2.0/dbh
ome_1/startup.log

[root@tm07 ~]# vi /u01/app/oracle/product/11.2.0/dbhome_1/startup.log

/u01/app/oracle/product/11.2.0/dbhome_1/bin/dbstart: Starting up database
"orcl"
Wed Apr 24 09:11:42 PDT 2013

SQL*Plus: Release 11.2.0.1.0 Production on Wed Apr 24 09:11:42 2013

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

SQL> Connected to an idle instance.
SQL> ORACLE instance started.

Total System Global Area 1085640704 bytes
Fixed Size                  2212536 bytes
Variable Size             654314824 bytes
Database Buffers          419430400 bytes
Redo Buffers                9682944 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 11g Enterprise Edition Release
11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

/u01/app/oracle/product/11.2.0/dbhome_1/bin/dbstart: Database instance "orcl"
warm started.
```

Validation of the connection is client (using ODBC)

1. Edit the tnsnames .ora, and update the IP to match the new primary for access to the database.

```
ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 172.16.72.124)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )
```

2. Use ODBC to validate the connection.
3. After database access is validated, shut down the database, and unmount /u01/.
4. If the volume is busy, `run lsof(8)`, and kill the oracle processes.

```
root@primary# dbshut
```

```
Processing Database instance "orcl": log file
/u01/app/oracle/product/11.2.0/dbhome_1/shutdown.log
[root@primary ~]# vi /u01/app/oracle/product/11.2.0/dbhome_1/shutdown.log

SQL*Plus: Release 11.2.0.1.0 Production on Wed Apr 24 09:43:53 2013

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

SQL> Database instance "orcl" shut down.
SQL>exit
root@primary ~# umount /u01
```

Stop the Pacemaker and Corosync services on the secondary server to avoid primitives starting on both servers.

5. Run service pacemaker stop and service corosync stop on the secondary.

Creating Oracle primitives (on the new primary)

Create the filesystem primitive

1. Create a Corosync primitive, p_fs_oracle, for the filesystem running on the FlashMAX II device.
2. Configure it to mount to the Oracle datadir /u01 for use by the highly available instance of Oracle.
3. Specify the vHA device node and the filesystem type in the primate command.

When completed, a commit command inside the crm shell is required for it to take effect.

```
root@Primary# crm configure
crm(live)configure# p_fs_oracle ocf:heartbeat:Filesystem params
device="/dev/vgca0_vha" directory="/u01" fstype="ext4"
crm(live)configure# commit
```

Create the virtual IP (VIP) primitive

1. Create the virtual IP primitive, p_ip_oracle, that all remote clients will use to connect to the Oracle instances.
2. Ensure that both the primary and secondary servers have the same name for the network card that will be used.

   When completed, commit it as well.

```
root@Primary# crm configure
crm(live)configure# p_ip_oracle ocf:heartbeat:IPaddr2 params ip="172.16.72.233"
cidr_netmask="24" nic="em3"
crm(live)configure# commit
```

Create the Oracle listener primitive

1. Create the listener primitive, p_oralsnr, to start the Oracle listener.
2. Ensure that the paths to listener are correctly set.
3. Start the listener before the oracle database to register the database instance.

   This will help with dependicies.

```
root@Primary# crm configure
crm(live)configure# p_oralsnr ocf:heartbeat:oralsnr params sid="orcl"
home="/u01/app/oracle/product/11.2.0/dbhome_1"
crm(live)configure# commit
```

Create the Oracle database primitive

1. Create the database primitive, p_oracle, to start the Oracle database.
2. Ensure that the paths to the database are set correctly.

   This primitive sets the Oracle home and uses the sid defined to start the Oracle instance.
3. Start the listener primitive before the database primitive.

   This will also help with dependencies.

```
root@Primary# crm configure
```

```
crm(live)configure# primitive p_oracle ocf:heartbeat:oracle params sid="orcl"
home="/u01/app/oracle/product/11.2.0/dbhome_1" ipcrm="instance" op start
timeout=120s op stop timeout=120s op monitor interval=20s timeout=120s
crm(live)configure# commit
```

Create the Pacemaker management group

Create a management group for the Pacemaker encompassing primitives and resources. As part of the failover process, these will move to the other node and start automatically by the cluster manager. `Commit` is required to cause the configuration change to take effect.

```
root@Primary# crm configure
crm(live)configure# group g_oracle p_fs_oracle p_ip_oracle p_oralsnr p_oracle
crm(live)configure# commit
```

Create the order and define dependencies

The group g_oracle must run on the primary server node. Define a collocation rule where the second parameter is the UUID of the vHA cluster prepended with `ms-`. A condition ensures that the g_oracle group is migrated with the host and that the host is promoted to the primary for the group to start. `Commit` completes the cluster setup.

```
root@Primary# crm configure
crm(live)configure# colocation c_oracle_on_vha inf: g_oracle ms-7ebb5240-b565-
4330-b74b-97298ab1e28d:Master
crm(live)configure# order o_vha_before_oracle inf: ms-7ebb5240-b565-4330-b74b-
97298ab1e28d:promote g_oracle:start
crm(live)configure# commit
```

Validate completed cluster configuration

1. Use the script `vgc-setup-vha-oracle` or manually use the `crm_mon` tool, a part of the Corosync and Pacemaker packages, to monitor cluster operation.
2. Verify that both servers in the ms-UUID set are online and active, and that the three previously defined resource groups started on the current primary.

   All cluster utilities and `crm_mon` can run on either the primary or the secondary cluster.

```
root@Primary # crm_mon
```

```
============
Last updated: Mon Mar 18 07:29:29 2013
Last change: Mon Mar 18 00:18:51 2013 via cibadmin on R720-1
Stack: openais
Current DC: R720-1 - partition with quorum
Version: 1.1.7-6.el6-148fccfd5985c5590cc601123c6c16e966b85d14
2 Nodes configured, 2 expected votes
5 Resources configured.
============
Online: [ R720-2]
OFFLINE: [ R720-1 ]
Master/Slave Set: ms-7ebb5240-b565-4330-b74b-97298ab1e28d
                      [vha-7ebb5240-b565-4330-b74b-97298ab1e28d]
     Masters: [ R720-2 ]
     Stopped: [ vha-7ebb5240-b565-4330-b74b-97298ab1e28d:1 ]
 Resource Group: g_oracle
     p_fs_oracle         (ocf::heartbeat:Filesystem):    Started R720-2
     p_ip_oracle         (ocf::heartbeat:IPaddr2):       Started R720-2
     p_oralsnr   (ocf::heartbeat:oralsnr):       Started R720-2
     p_oracle    (ocf::heartbeat:oracle):        Started R720-2
```

The master/slave set identifies the two servers and the FlashMAX Connect vHA namespace of the same UUID. The p_fs_oracle manages the filesystem mount, the p_ip_oracle manages the virtual IP migration, p_oralsnr handles the listener migration, and p_oracle handles the Oracle migration.

Validate that /dev/vgca0_vha correctly mounted to the primary server

On the primary server, verify that the Oracle database directory is mounted using the standard mount command. On the secondary server, the filesystem will not be mounted.

```
root@Primary # mount /dev/vgca0_vha on /u01 type ext4 (rw)
```

Validate that the Oracle database is accessible from a client via VIP

1. Connect to the server using the virtual IP from an external host to verify the full operation of Oracle.

This verifies that all levels of the stack, in addition to the network infrastructure, can handle the virtual IP routing.

2. Direct applications to connect to the virtual IP for service and to handle temporary disconnections during failover transition times.
3. Edit the tnsnames.ora on the client to add the VIP to the HOST field.

```
ORCL =
```

```
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 172.16.72.233)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )
```

4. Using ODBC, validate the connection.
5. Run service corosync start and service pacemaker start.
6. Run `crm_mon` to verify that both the primary and secondary are online.

```
root@Primary # crm_mon
```

```
============
Last updated: Mon Mar 18 07:29:29 2013
Last change: Mon Mar 18 00:18:51 2013 via cibadmin on R720-1
Stack: openais
Current DC: R720-1 - partition with quorum
Version: 1.1.7-6.el6-148fccfd5985c5590cc601123c6c16e966b85d14
2 Nodes configured, 2 expected votes
5 Resources configured.
============
Online: [ R720-1, R720-2]
Master/Slave Set: ms-7ebb5240-b565-4330-b74b-97298ab1e28d
                      [vha-7ebb5240-b565-4330-b74b-97298ab1e28d]
     Masters: [ R720-2 ]
     Slaves: [ R720-1 ]
 Resource Group: g_oracle
     p_fs_oracle        (ocf::heartbeat:Filesystem):    Started R720-2
     p_ip_oracle        (ocf::heartbeat:IPaddr2):       Started R720-2
     p_oralsnr  (ocf::heartbeat:oralsnr):       Started R720-2
     p_oracle   (ocf::heartbeat:oracle):        Started R720-2
```

Monitoring the Oracle cluster

Use `crm_mon` and `vgc-vha-monitor` to monitor the Oracle cluster on both the primary and secondary hosts.

Use crm_mon to check the cluster state

Use `crm_monitor` to report the state of the cluster and services therein. It can run from either the primary or the secondary server.

```
root@Primary # crm_mon
```

```
============
Last updated: Mon Mar 18 07:29:29 2013
Last change: Mon Mar 18 00:18:51 2013 via cibadmin on R720-1
Stack: openais
Current DC: R720-1 - partition with quorum
Version: 1.1.7-6.el6-148fccfd5985c5590cc601123c6c16e966b85d14
2 Nodes configured, 2 expected votes
5 Resources configured.
============
Online: [ R720-2 R720-1 ]
Master/Slave Set: ms-7ebb5240-b565-4330-b74b-97298ab1e28d
                      [vha-7ebb5240-b565-4330-b74b-97298ab1e28d]
     Masters: [ R720-2 ]
     Slaves: [ R720-1 ]
 Resource Group: g_oracle
     p_fs_oracle        (ocf::heartbeat:Filesystem):    Started R720-2
     p_ip_oracle        (ocf::heartbeat:IPaddr2):       Started R720-2
     p_oralsnr  (ocf::heartbeat:oralsnr):       Started R720-2
     p_oracle   (ocf::heartbeat:oracle):        Started R720-2
```

Use vgc-vha-monitor to check the FlashMAX Connect vHA namespace state

Run `vgc-vha-monitor` to see the performance of the Virident FlashMAX II device, and the FlashMAX Connect vHA namespace, on either the primary or the secondary server. Statistics on connection state, total bytes read and written, and resynchronization state are all provided.

```
root@(Primary,Secondary) # vgc-vha-monitor --detail /dev/vgca0_vha
```

```
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53313.V3)

 vHA Device         Local Device       Role           State
 /dev/vgca0_vha    /dev/vgca0         primary        Connected

  vHA Configuration:
    Cluster UUID               : 7ebb5240-...
    Capacity                   : 1800 GB
    Peer IP addr               : 172.16.72.120
    Peer hostname              : R720-1
  IO Statistics:
    Total Bytes Written        : 0 (0 GB)
    Total Bytes Read           : 135168 (0 GB)
    Total Read Errors          : 0
    Total Write Errors         : 0
  vHA State:
    Master-Slave               : master
    Connection Score           : 3
    Nodes in sync              : 1
  vHA Statistics:
    Total bytes mirrored       : 0 (0 GB)
    Total bytes resynced       : 0 (0 GB)
    Bytes resynced last        : 0 (0 GB)
```

Oracle HA cluster failover (with crm)

Base state of the primary server before failover

1. Use `crm_mon` to check the state of the services on the cluster.
2. Ensure that both hosts are online and there is a designated primary server.

```
root@Primary  # crm_mon
```

```
============
Last updated: Thu Mar 21 14:29:28 2013
Last change: Thu Mar 21 10:02:48 2013 via crmd on R720-2a
Stack: openais
Current DC: R720-1 - partition with quorum
Version: 1.1.7-6.el6-148fccfd5985c5590cc601123c6c16e966b85d14
2 Nodes configured, 2 expected votes
5 Resources configured.
============
Online: [ R720-1 R720-2 ]
Master/Slave Set: ms-7ebb5240-b565-4330-b74b-97298ab1e28d [vha-7ebb5240-b565-
4330-b74b-97298ab
1e28d]
     Masters: [ R720-2 ]
     Slaves: [ R720-1 ]
 Resource Group: g_oracle
     p_fs_oracle        (ocf::heartbeat:Filesystem):    Started R720-2
     p_ip_oracle        (ocf::heartbeat:IPaddr2):       Started R720-2
     p_oralsnr  (ocf::heartbeat:oralsnr):       Started R720-2
     p_oracle   (ocf::heartbeat:oracle):        Started R720-2
```

3. Use vgc-vha-monitor to check that the base state of the primary is connected.

```
root@Primary(R720-2)#  vgc-vha-monitor
```

```
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53741.V3)

  ------------------------------------------------------------------------
-
  vHA Device          Local Device        Role           State          UUID
  ------------------------------------------------------------------------
-
  /dev/vgca0_vha      /dev/vgca0          primary        Connected      7ebb...
```

Base state of the secondary before failover

Use `vgc-vha-monitor` command to check the FlashMAX Connect vHA namespace state is secondary and is connected.

```
root@Secondary(R720-1)# vgc-vha-monitor
```

```
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53741.V3)

  ------------------------------------------------------------------------------
-
  vHA Device          Local Device        Role           State          UUID
  ------------------------------------------------------------------------------
-
  /dev/vgca0_vha     /dev/vgca0          secondary      Connected      7ebb...
```

Validate that the Oracle database is accessible from the client

1. Connect to the Oracle database using the virtual IP.
2. Ensure that tnsnames.ora is configured correctly.

```
ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 172.16.72.233)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )
```

3. Use ODBC to validate connection.

Initiate the failover, reboot the primary server

Reboot the current primary server to force a failover from primary to secondary.

```
root@Primary(R720-2) # reboot
```

```
Broadcast message from root@R720-1
        (/dev/pts/1) at 8:33 ...

The system is going down for reboot NOW!
```

State of the secondary server when the primary reboots

1. After failover, use `crm_mon` and `vgc-vha-monitor` to validate that all the Oracle services have moved to the new primary server.

Since its peer is disconnected, the former primary server will show as disconnected in `vgc-vha-monitor`. The database is accessible via the same virtual IP as before.

```
root@newPrimary(R720-1) # crm_mon
```

```
============
Last updated: Thu Mar 21 14:29:28 2013
Last change: Thu Mar 21 10:02:48 2013 via crmd on R720-2a
Stack: openais
Current DC: R720-1 - partition with quorum
Version: 1.1.7-6.el6-148fccfd5985c5590cc601123c6c16e966b85d14
2 Nodes configured, 2 expected votes
5 Resources configured.
============
Online: [ R720-1 ]
OFFLINE: [ R720-2 ]
Master/Slave Set: ms-7ebb5240-b565-4330-b74b-97298ab1e28d [vha-7ebb5240-b565-
4330-b74b-97298ab
1e28d]
     Masters: [ R720-1 ]
     Stopped: [ vha-7ebb5240-b565-4330-b74b-97298ab:1 ]
 Resource Group: g_oracle
     p_fs_oracle        (ocf::heartbeat:Filesystem):    Started R720-1
     p_ip_oracle        (ocf::heartbeat:IPaddr2):       Started R720-1
     p_oralsnr  (ocf::heartbeat:oralsnr):       Started R720-1
     p_oracle   (ocf::heartbeat:oracle):        Started R720-1
root@newPrimary(R720-1) # vgc-vha-monitor
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53741.V3)

  ------------------------------------------------------------------------
-                                 ---------------------------------
  vHA Device          Local Device        Role           State          UUID
  ------------------------------------------------------------------------
-                                 ---------------------------------
  /dev/vgca0_vha      /dev/vgca0          primary        Disconnected    7ebb…
```

2. Use the same client and virtual IP to verify database accessibility.
3. Verify that tnsnames.ora has the VIP.

Restore the cluster back to HA connected state

The former primary server must be demoted manually using the vgc-vha-config utility with `--demote` option to restore the cluster back to the HA connected state.

1. Run `vgc-vha-monitor --detail` to get the connection scores for the first server and note the connection score.
2. Compare the connection score reported by `vgc-vha-monitor -detail` for both servers, and demote the server with the lower connection score.

This connection score is a monotonically increasing value kept by the FlashMAX Connect driver on every connection and promotion or demotion.

```
root@newPrimary (R720-1) # vgc-vha-monitor --detail /dev/vgca0_vha
```

```
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53741.V3)

  vHA Device          Local Device        Role           State
  /dev/vgca0_vha      /dev/vgca0          primary        Disconnected

vHA Configuration:
    vHA UUID                  : 7ebb5240-b565-4330-b74b-97298ab1e28d
    Capacity                  : 1800 GB
    Peer IP addr              : 172.16.72.120
    Peer hostname             : R720-2
  IO Statistics:
    Total Bytes Written       : 241664 (0 GB)
    Total Bytes Read          : 8902656 (0 GB)
    Total Read Errors         : 0
    Total Write Errors        : 0
  vHA State:
    Master-Slave              : master
    Connection Score          : 21
    Nodes in sync             : 1
  vHA Statistics:
    Total bytes mirrored      : 0 (0 GB)
    Total bytes resynced      : 0 (0 GB)
    Bytes resynced last       : 0 (0 GB)
```

```
root@oldPrimary (R720-2) # vgc-vha-monitor --detail vha /dev/vgca0_vha
```

```
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53741.V3)

  vHA Device          Local Device        Role           State
  /dev/vgca0_vha      /dev/vgca0          primary        Disconnected

  vHA Configuration:
    vHA UUID                    : 7ebb5240-b565-4330-b74b-97298ab1e28d
    Capacity                    : 1800 GB
    Peer IP addr                : 172.16.72.124
    Peer hostname               : R720-1
  IO Statistics:
    Total Bytes Written         : 0 (0 GB)
    Total Bytes Read            : 414720 (0 GB)
    Total Read Errors           : 0
    Total Write Errors          : 0
  vHA State:
    Master-Slave                : slave
    Connection Score            : 20
    Nodes in sync               : 0
  vHA Statistics:
    Total bytes mirrored        : 0 (0 GB)
    Total bytes resynced        : 0 (0 GB)
    Bytes resynced last         : 0 (0 GB)
```

Manual demotion of the new secondary

Use `vgc-vha-config --demote` on the current secondary, the former primary, to cause the FlashMAX Connect vHA namespace to connect and begin automatic resynchronization.

```
root@newSecondary (R720-2) # vgc-vha-config --demote --force /dev/vgca0_vha
```

```
vgc-vha-config: FlashMAX Connect Software Suite 1.0(53741.V3)

Modify role success
Done.
```

Secondary resynchronization progresses automatically

The secondary server will resynchronize with the primary server to get the most recent data. Only updated portions of the data will resynchronize. During this process, the data and services on the primary are fully accessible. Use `vgc-vha-monitor` to see the resynchronization state.

```
root@newSecondary (R720-2) # vgc-vha-monitor
```

Validate that the cluster is in the HA connected state

Synchronization depends on the amount of updated data. After synchronization completes, the vHA cluster will show as primary/secondary connected.

1. Run `vgc-vha-monitor` on both nodes to verify connection.

```
root@newSecondary (R720-2) # vgc-vha-monitor
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53741.V3)
  ------------------------------------------------------------------------
  vHA Device              Local Device      Role           State          UUID
  ------------------------------------------------------------------------
  /dev/vgca0_vha       /dev/vgca0       secondary      Connected      7ebb...

root@newPrimary (R720-1) # vgc-vha-monitor
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53741.V3)
  ------------------------------------------------------------------------
-
  vHA Device              Local Device       Role           State          UUID
  ------------------------------------------------------------------------
-
  /dev/vgca0_vha       /dev/vgca0       primary        Connected      7ebb...
```

1. Validate HA functionality with `crm_mon`.

```
root@newPrimary(R720-1) # crm_mon
============
Last updated: Thu Mar 21 14:29:28 2013
Last change: Thu Mar 21 10:02:48 2013 via crmd on R720-2a
Stack: openais
Current DC: R720-1 - partition with quorum
Version: 1.1.7-6.el6-148fccfd5985c5590cc601123c6c16e966b85d14
2 Nodes configured, 2 expected votes
5 Resources configured.
============
Online: [ R720-1 R720-2 ]
Master/Slave Set: ms-7ebb5240-b565-4330-b74b-97298ab1e28d [vha-7ebb5240-b565-
4330-b74b-97298ab
1e28d]
     Masters: [ R720-1 ]
     Slaves: [ R720-2 ]
 Resource Group: g_oracle
     p_fs_oracle         (ocf::heartbeat:Filesystem):     Started R720-1
     p_ip_oracle         (ocf::heartbeat:IPaddr2):        Started R720-1
     p_oralsnr   (ocf::heartbeat:oralsnr):        Started R720-1
     p_oracle    (ocf::heartbeat:oracle):         Started R720-1
```

Cluster node states

The cluster goes through several states during and after failover. Figure 7 shows cluster progression at a high level.

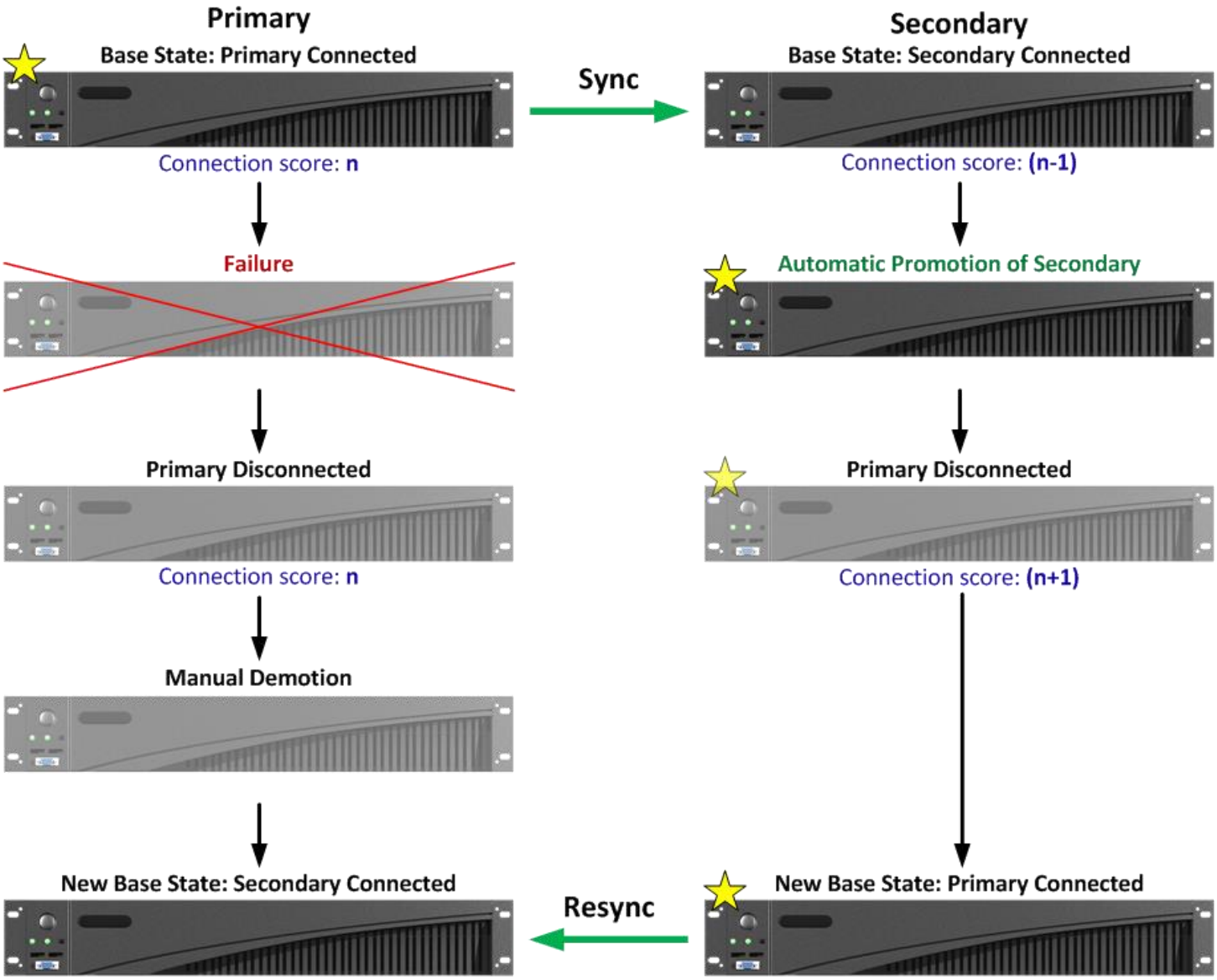

Figure 7: Cluster progression overview.

Updating the vHA cluster

1. Complete the software update on the secondary server first, initiate a failover, and repeat the update process for the remaining server, which will show as the new secondary.

   Follow the correct procedure when updating the software on a running cluster.

2. Stop the cluster manager using the service commands.

```
root@Secondary # service pacemaker stop
```

```
Signaling Pacemaker Cluster Manager to terminate:          [  OK  ]
Waiting for cluster services to unload:.......             [  OK  ]
```

3. Unload the Virident FlashMAX Connect device driver.

```
root@Secondary # service vgcd stop
```

```
Unloading FlashMAX HA kernel modules...                    [  OK  ]
Unloading kernel modules...                                [  OK  ]
```

4. Uninstall the old driver and utilities RPMs on the secondary.

```
root@Secondary # rpm -qa | grep vgc | xargs rpm -e
```

```
Unloading FlashMAX HA kernel modules...                    [  OK  ]
Unloading kernel modules...                                [  OK  ]
```

5. Install the new driver and utilities RPMs on the secondary server, using the `--nodeps` option to the `rpm` command as during the initial installation.

```
root@Secondary # rpm -ivh --nodeps *.rpm
```

```
Preparing...                ########################################### [100%])
   1:vgc-utils              ########################################### [ 33%])
   2:vgc-tools              ########################################### [ 67%])
   3:kmod-vgc-2.6.32-220.el########################################### [100%])
```

6. Start the vgcd service to load the FlashMAX Connect device drivers.

```
root@Secondary # service vgcd start
```

7. Use `vgc-vha-monitor` to verify that the secondary is back up and running.

```
Loading kernel modules...                                  [  OK  ]
Rescanning SW RAID volumes...                              [  OK  ]
Rescanning LVM volumes...                                  [  OK  ]
Enabling swap devices...                                   [  OK  ]
Rescanning mount points...                                 [  OK  ]
```

```
root@Secondary # vgc-vha-monitor
```

```
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53741.V3)
  ------------------------------------------------------------------------
  vHA Device     Local Device     Role            State           UUID
  ------------------------------------------------------------------------
  /dev/vgca0_vha1     /dev/vgca0    secondary       Connected       7ebb...
```

8. Restart the Corosync and Pacemaker services on the secondary.

```
root@Secondary # service corosync start
```

```
Starting Corosync Cluster Engine (corosync):               [  OK  ]
```

```
root@Secondary # service pacemaker start
```

```
Starting Pacemaker Cluster Manager:                        [  OK  ]
```

9. Use `crm_mon` to verify that the full cluster manager recovered, the services started, and the secondary server reconnected.
10. Verify that both servers are listed as online.

```
root@Secondary # crm_mon
```

```
============
Last updated: Mon Mar 18 07:29:29 2013
Last change: Mon Mar 18 00:18:51 2013 via cibadmin on R720-1
Stack: openais
Current DC: R720-1 - partition with quorum
Version: 1.1.7-6.el6-148fccfd5985c5590cc601123c6c16e966b85d14
2 Nodes configured, 2 expected votes
5 Resources configured.
============
Online: [ R720-2a R720-1 ]
Master/Slave Set: ms-7ebb5240-b565-4330-b74b-97298ab1e28d
                        [vha-7ebb5240-b565-4330-b74b-97298ab1e28d]
     Masters: [ R720-1 ]
     Slaves: [ R720-2a ]
 Resource Group: g_oracle
     p_fs_oracle (ocf::heartbeat:Filesystem):    Started R720-1
     p_ip_oracle (ocf::heartbeat:IPaddr2):       Started R720-1
     p_oracle    (ocf::heartbeat:oracle):         Started R720-1
```

11. Reboot the primary server to make the updated secondary server the new primary server.
12. After the original primary server reboots, follow the same upgrade procedure as the original secondary server up to the stage of installing the FlashMAX Connect device driver and restarting the vgcd service.
13. Run vgc-vha-monitor.

    Since the former primary server is not connected to the vHA namespace, its state will be primary disconnected.
14. Use `vgc-vha-config` to reconnect the server to the vHA namespace, and set it as the secondary server.
15. Use `vgc-vha-monitor` to verify that vHA namespace is connected and in either a state of resync or connected.

```
root@oldPrimary # vgc-vha-monitor
```

```
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53741.V3)
  ------------------------------------------------------------------------------
  vHA Device          Local Device        Role           State          UUID
  ------------------------------------------------------------------------------
-
  /dev/vgca0_vha      /dev/vgca0          primary        Disconnected
7ebb5...


root@oldPrimary # vgc-vha-config --demote --force vha1 /dev/vgca0
vgc-vha-config: FlashMAX Connect Software Suite 1.0(53741.V3)

Modify role success
Done.

root@oldPrimary # vgc-vha-monitor
vgc-vha-monitor: FlashMAX Connect Software Suite 1.0(53741.V3)
  ------------------------------------------------------------------------------
  vHA Device          Local Device    Role           State          UUID
  ------------------------------------------------------------------------------
  /dev/vgca0_vha      /dev/vgca0          secondary      Resync     7ebb5...
```

The former primary server will start the Corosync and Pacemaker services to finish the cluster restart.

16. Use `crm_mon` to verify that the former primary has demoted to the secondary server and that all the resources are connected and running:

```
root@oldPrimary # service corosync start
```

```
Starting Corosync Cluster Engine (corosync):               [  OK  ]
```

```
root@oldPrimary # service pacemaker start
```

```
Starting Pacemaker Cluster Manager:                        [  OK  ]
```

```
root@oldPrimary# crm_mon
```

```
============
Last updated: Mon Mar 18 07:29:29 2013
Last change: Mon Mar 18 00:18:51 2013 via cibadmin on R720-1
Stack: openais
Current DC: R720-1 - partition with quorum
Version: 1.1.7-6.el6-148fccfd5985c5590cc601123c6c16e966b85d14
2 Nodes configured, 2 expected votes
5 Resources configured.
============
Online: [ R720-2a R720-1 ]
Master/Slave Set: ms-7ebb5240-b565-4330-b74b-97298ab1e28d
                        [vha-7ebb5240-b565-4330-b74b-97298ab1e28d]
     Masters: [ R720-1 ]
     Slaves: [ R720-2 ]
  Resource Group: g_oracle
     p_fs_oracle         (ocf::heartbeat:Filesystem):     Started R720-2
     p_ip_oracle         (ocf::heartbeat:IPaddr2):        Started R720-2
     p_oralsnr   (ocf::heartbeat:oralsnr):        Started R720-2
     p_oracle    (ocf::heartbeat:oracle):         Started R720-2
```

ABOUT PRINCIPLED TECHNOLOGIES

Principled Technologies, Inc.
1007 Slater Road, Suite 300
Durham, NC, 27703
www.principledtechnologies.com

We provide industry-leading technology assessment and fact-based marketing services. We bring to every assignment extensive experience with and expertise in all aspects of technology testing and analysis, from researching new technologies, to developing new methodologies, to testing with existing and new tools.

When the assessment is complete, we know how to present the results to a broad range of target audiences. We provide our clients with the materials they need, from market-focused data to use in their own collateral to custom sales aids, such as test reports, performance assessments, and white papers. Every document reflects the results of our trusted independent analysis.

We provide customized services that focus on our clients' individual requirements. Whether the technology involves hardware, software, Web sites, or services, we offer the experience, expertise, and tools to help our clients assess how it will fare against its competition, its performance, its market readiness, and its quality and reliability.

Our founders, Mark L. Van Name and Bill Catchings, have worked together in technology assessment for over 20 years. As journalists, they published over a thousand articles on a wide array of technology subjects. They created and led the Ziff-Davis Benchmark Operation, which developed such industry-standard benchmarks as Ziff Davis Media's Winstone and WebBench. They founded and led eTesting Labs, and after the acquisition of that company by Lionbridge Technologies were the head and CTO of VeriTest.