Introduction

This document discusses the various configuration scenarios and the corresponding workflows for setting up Oracle VM for SPARC (LDoms) with multiple I/O domains configured to support deployment of Storage Foundation for High Availability (SFHA). The scenarios describe the deployment of SFHA on LDoms in a single physical host as well as on multiple physical hosts.

Intended Audience

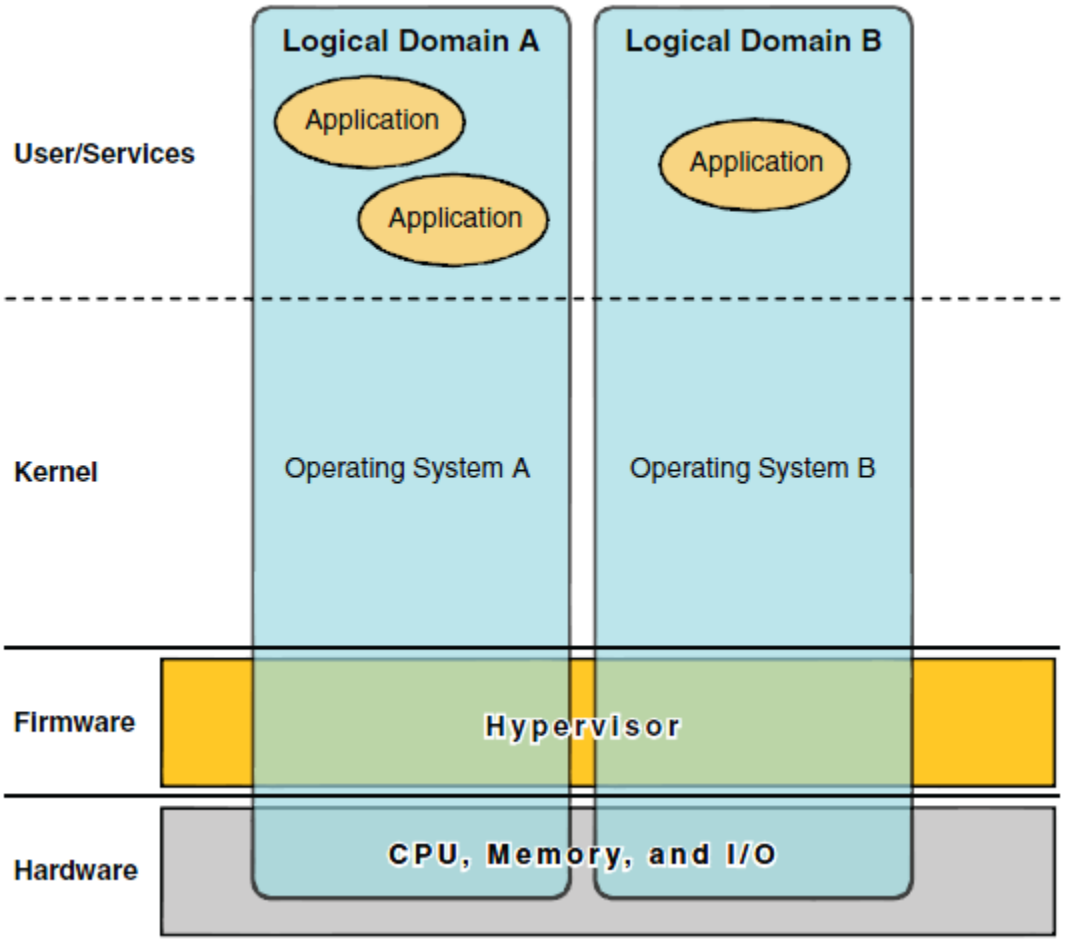
This document is intended for Symantec Systems Engineers (SE), Technical Support Engineers (TSE), and System Administrators for understanding, evaluating, or setting up virtualized environments using Oracle VM in a highly resilient architecture for deploying SFHA.

Introduction to Oracle VM for SPARC

Oracle VM Server for SPARC (previously called Sun Logical Domains) provides highly efficient, enterprise-class virtualization capabilities for Oracle's SPARC T-series servers. Oracle VM Server for SPARC leverages the built-in SPARC hypervisor to subdivide a supported platform's resources (CPUs, memory, network, and storage) by creating partitions called logical (or virtual) domains. Each logical domain can run an independent operating system. Oracle VM Server for SPARC provides the flexibility to deploy multiple Oracle Solaris operating systems simultaneously on a single platform.

Oracle VM Server for SPARC solution is supported on Oracle Solaris Cool Threads technology-based servers powered by Chip Multithreading Technology (CMT) processors. Refer to Oracle VM Server documentation for more information on the latest supported hardware.

FIGURE 1–1 Hypervisor Supporting Two Domains



Logical Domain Roles

Control domain: The Logical Domains Manager runs in this domain, which enables you to create and manage other logical domains, and to allocate virtual resources to other domains. You can have only one control domain per server. The control domain is the first domain created when you install the Oracle VM Server for SPARC software. The control domain is named as primary.

Service domain: A service domain provides virtual device services to other domains, such as a virtual switch, a virtual console concentrator, and a virtual disk server. Any domain can be configured as a service domain

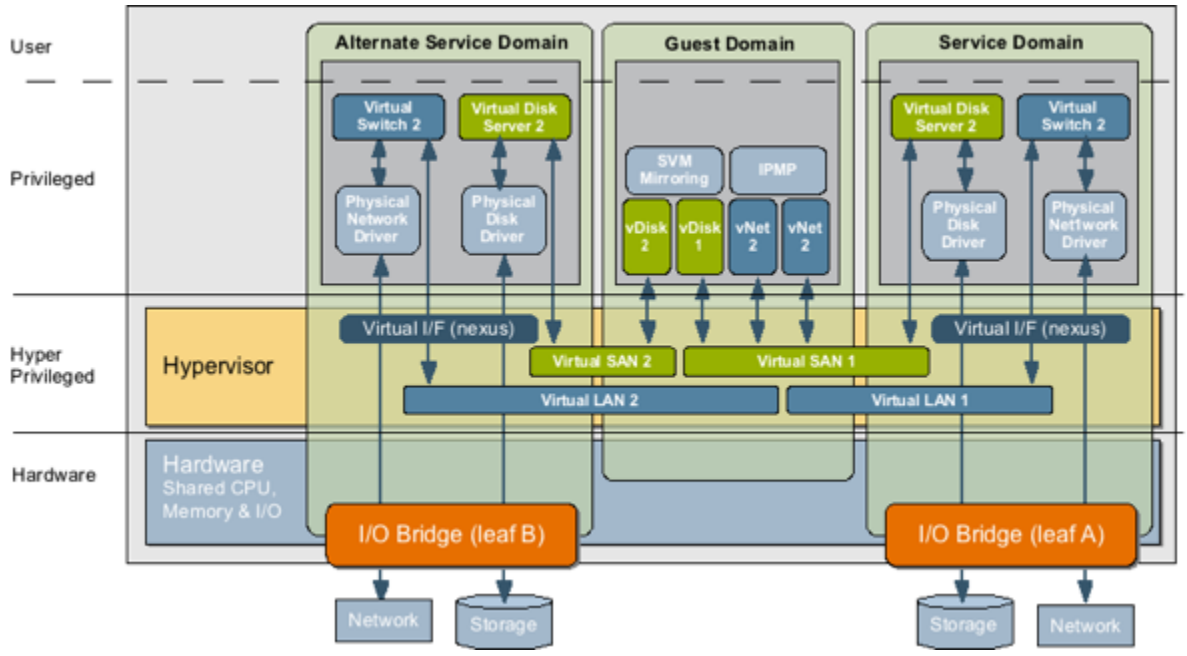
I/O domain: An I/O domain has direct access to a physical I/O device, such as a network card in a PCI EXPRESS (PCIe) controller. An I/O domain can own a PCIe root complex, or it can own a PCIe slot or on-board PCIe device by using the direct I/O (DIO) feature.

Root domain: A root domain has a PCIe root complex assigned to it. This domain owns the PCIe fabric and provides all fabric-related services, such as fabric error handling. A root domain is also an I/O domain, as it owns and has direct access to physical I/O devices.

Guest domain: A guest domain is a non-I/O domain that consumes virtual device services that are provided by one or more service domains. A guest domain does not have any physical I/O devices, but only has virtual I/O devices, such as virtual disks and virtual network interfaces.

Redundant virtual I/O services

To build a higher level of resiliency in case of I/O device or service failures, configuring additional I/O domains helps Guest domains meet these requirements. To configure additional I/O domains the server should have more than one PCI bus. Refer to the Oracle Sun Hardware documentation to see if your server meets the requirements.



In the above figure, the Guest domain uses the virtual disk from the services provided through the Primary and the Alternate I/O domain. The virtual disks are than configured in a mirrored configuration within the Guest domain to provide data availability and reliability. A virtual NIC is provided to the Guest domain through each of the virtual switch services configured in the Primary and the Alternate I/O domains. IPMP is configured in the Guest domain to provide Network availability in the Guest domain.

Benefits of using SFHA with LDoms

The LDom technology provides a very cost-effective alternative architecture for deploying SFHA. The same physical server can be used for multiple applications within various logical domains with optimal resource utilization. The underlying hardware availability is increased by using the Split-PCI bus technology that the CMT servers provide and the SFHA component completes the availability by ensuring higher uptime for applications by monitoring the device paths through all I/O domains.

Configuration scenarios for SFHA with LDoms

This section describes the server configuration, on which various scenarios have been tested, the pre-requisites for the scenarios, and the configuration scenarios for setting up high availability to the guest domains.

The following server configuration is used for the setup scenarios presented in this document:
Server: 2 SUN T5240 Servers (Server hostnames: primhost and sechost)
Processor: 2 UltraSPARC-T2+ processors
Memory: 32 GB
PCI bus : 2 PCI bus (pci@400, pci@500)
PCI Devices: 1 Quad NIC Card and 1 dual-port FC HBA connected to bus pci@400, Onboard Quad NIC and 1 dual-port FC HBA connected to bus pci@500
Firmware Version: Sun System Firmware 7.3.0
Operating System: Solaris 10 update 9
Software: LDoms Manager 2.0, SFHA 6.0
In addition to the local disks there are a few SAN LUN's that are accessible to all the I/O domains (including the primary domain) in the cluster.

Configuration Scenarios
The following configurations scenarios are tested for deployment with SFHA with Alternate I/O domain configured.
Scenario 1) SFHA on all I/O domains and guest domains configured using raw LUN devices.
Scenario 2) SFHA on all I/O domains and guest domains configured using ZFS volume.

Pre-requisites for the configuration scenarios

Logical Domain Manager installation

1. Ensure that the system firmware matches the logical domain manager that is planned for installation. Refer to the Oracle VM Server for SPARC Release Notes to find the appropriate firmware version and to the Oracle VM Server for SPARC Administration Guide for installation steps to upgrade the system firmware.

2. Download the Oracle VM Server for SPARC, version 2.0 or later from the Oracle web site.
3. Extract the archive and install the package. Refer to the Oracle VM Server for SPARC Administration Guide for installation procedures.
4. Set the PATH variable to point to the logical domain manager binaries.

Control Domain Configuration

After the Oracle VM Server for SPARC software has been installed, the system has to be configured to become a control domain. To do so, the following actions needs to be performed on each physical server that is part of the cluster.

1. Create a virtual console concentrator (vcc) service
primhost# ldm add-vcc port-range=5000-5100 primary-vcc0 primary

2. Create a virtual disk server (vds) to allow importing virtual disks into a logical domain
primhost# ldm add-vds primary-vds0 primary

3. Create a virtual switch service (vsw) to enable networking between virtual network (vnet) devices in logical domains.
primhost# ldm add-vsw net-dev=e1000g3 linkprop=phys-state primary-vsw0 primary

4. List the services created on the primary domain.
primhost# ldm ls-services primary

```
VCC
    NAME          LDOM          PORT-RANGE
    primary-vcc0  primary       5000-5100

VSW
    NAME          LDOM      MAC               NET-DEV  ID  DEVICE    LINKPROP    MTU   MODE
    primary-vsw0  primary   00:14:4f:f9:8d:3f e1000g3  0   switch@0  phys-state  1500

VDS
    NAME          LDOM          VOLUME        OPTIONS       MPGROUP       DEVICE
```

5. Configure the control domain (primary) resources
primhost# ldm set-mau 2 primary
primhost# ldm set-vcpu 16 primary
primhost# ldm set-memory 4G primary

6. Change the primary interface to be the first virtual switch interface. The following command configures the control domain to plumb and use the interface vsw0 instead of e1000g0
primhost# mv /etc/hostname.e1000g0 /etc/hostname.vsw0

7. Verify that the control domain (primary) owns more than one PCIe bus. By default the primary domain owns all buses present on the system.

primhost# ldm ls-io

```
[root@vcssx229 /]# ldm ls-io
IO          PSEUDONYM      DOMAIN
--          ---------      ------
pci@400     pci_0          primary
pci@500     pci_1          primary

PCIE            PSEUDONYM  STATUS  DOMAIN
----            ---------  ------  ------
pci@400/pci@0/pci@c  PCIE1     OCC     primary
pci@400/pci@0/pci@9  PCIE2     OCC     primary
pci@400/pci@0/pci@d  PCIE3     EMP     -
pci@400/pci@0/pci@8  MB/SASHBA  OCC     primary
pci@500/pci@0/pci@9  PCIE0     UNK     -
pci@500/pci@0/pci@d  PCIE4     OCC     - primary
pci@500/pci@0/pci@c  PCIE5     UNK     -
pci@500/pci@0/pci@8  MB/NET0    OCC     - primary
```

Note: The internal disks on the servers may be connected to a single PCIe bus and may be in use by the control domain. If a domain is booted from an internal disk, do not remove that bus from the domain. Also, ensure that you are not removing a bus with devices (such as network ports) that are used by a domain. If you remove the wrong bus, a domain might not be able to access the required devices and could become unusable.

8. Determine the device path of the control domain boot disk, which needs to be retained.

■ For UFS file systems, run the df / command to determine the device path of the boot disk.

primhost# df /

/ (/dev/dsk/c0t1d0s0): 1309384 blocks 457028 files

■ For ZFS file systems, first run the df / command to determine the pool name, and then run the zpool status command to determine the device path of the boot disk.

primhost# df /

/ (rpool/ROOT/s10s_u9wos_14a):241223335 blocks 241223335 files

primhost# zpool status rpool

primhost# zpool status rpool

```
  pool: rpool
 state: ONLINE
 scrub: none requested
config:

        NAME                    STATE     READ WRITE CKSUM
```

```
rpool                   ONLINE      0      0        0
 c0t1d0s0               ONLINE      0      0        0
```

errors: No known data errors

9. Determine the physical device to which the block device is linked.

The following example uses block device c1t0d0s0:

primhost# ls -l /dev/dsk/c1t0d0s0

lrwxrwxrwx 1 root root 49 Nov 16 2010 /dev/dsk/c1t0d0s0 ->

../../devices/pci@400/pci@0/pci@8/scsi@0/sd@0,0:a

In this example, the physical device for the primary domain's boot disk is connected to bus pci@400, which corresponds to the earlier listing of pci_0. This means that you *cannot* assign pci_0 (pci@400) to another domain.

10. Determine the network interface that is used by the system to provide network services to the guest domains. In this example, we have used e1000g3 network interface to provide network services to the guest domain.

primhost# ifconfig –a

lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1

inet 127.0.0.1 netmask ff000000

e1000g0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2

inet 10.209.74.220 netmask fffffc00 broadcast 10.209.75.255

ether 0:21:28:69:f5:f6

e1000g3: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3

inet 10.209.75.114 netmask fffffc00 broadcast 10.209.75.255

ether 0:21:28:69:f5:f9

11. Determine the physical device to which the network interface is linked.

primhost# ls -l /dev/e1000g3

lrwxrwxrwx 1 root root 46 Nov 16 2010 /dev/e1000g3 ->

../devices/pci@400/pci@0/pci@c/network@0,3:e1000g3

12. From the above outputs it is evident that the primary domain is using disk and network devices hosted on pci_0. Remove the buses that do not contain the boot disk or the network interface from the primary domain.

primhost#ldm rm-io pci_1 primary

primhost#ldm add-config initial

13. Reboot the **primary** domain so that the change takes effect.

primhost# shutdown -i6 -g0 –y

The primary domain will be rebooted with the new configuration. The primary domain also performs function of the I/O domain and the service domains.

14. The same configuration needs to be performed on the second server , in our example it is sechost. If the devices configured on the other hosts in the cluster are same then you can export the primary domain xml configuration to the other host (sechost in this configuration has same physical hardware and configuration as primhost).

primhost# ldm ls-constraints –x primary > /tmp/primary.xml
primhost# scp /tmp/primary.xml sechost:/tmp/primary.xml

15. Import the xml configuration for the primary domain on node sechost.

sechost# ldm set-domain –i /tmp/primary.xml primary
sechost# ldm ls-services primary

```
VCC
    NAME          LDOM          PORT-RANGE
    primary-vcc0  primary       5000-5100

VSW
    NAME          LDOM      MAC               NET-DEV  ID  DEVICE    LINKPROP     MTU   MODE
    primary-vsw0  primary   00:14:4f:fb:de:1c e1000g3  0   switch@0  phys-state   1500

VDS
    NAME          LDOM          VOLUME        OPTIONS       MPGROUP       DEVICE
```

16. Save the configuration to the controller and reboot the host for the new configuration to take effect.

sechost#ldm add-config initial
sechost# shutdown -i6 -g0 –y

Alternate I/O domain configuration

1. Create the alternate I/O domain configuration

primhost# ldm add-domain alternate
primhost# ldm set-mau 2 alternate
primhost# ldm add-vcpu 16 alternate
primhost# ldm add-mem 4G alternate
primhost# ldm set-var auto-boot\?=true alternate

2. Assign the PCI bus unassigned during the configuration of the primary domain to the I/O domain.

primhost# ldm add-io pci_1 alternate

3. Install the Oracle Solaris OS on the I/O domain from a ISO file

primhost# ldm add-vdsdev /software/solaris10.iso dvdiso@primary-vds0
primhost# ldm add-vdisk cdrom dvdiso@primary-vds0 alternate
primhost# ldm bind alternate
primhost# ldm start alternate

4. Connect to the console of the IO domain and begin the OS installation. Once the OS installation has been completed on the alter I/O domain, you can begin configuring the services that would be provided through the alternate I/O domain for providing high availability of I/O services to the guest logical domains.

5. Create the virtual disk and virtual switch services for the alternate I/O domain

primhost# ldm add-vds alternate-vds0 alternate

primhost# ldm add-vsw net-dev=nxge3 linkprop=phys-state alternate-vsw0 alternate

6. List the services created on the alternate domain.

primhost# ldm ls-services alternate

```
VSW
    NAME             LDOM       MAC               NET-DEV   ID   DEVICE     LINKPROP     MTU   MODE
    primary-vsw0     primary    00:14:4f:f9:8d:3f   nxge3     0    switch@0   phys-state   1500

VDS
    NAME          LDOM          VOLUME        OPTIONS        MPGROUP        DEVICE
```

7. Save the configuration to the controller.

primhost# ldm add-config splitbus

8. The above mentioned set of steps will be executed on *sechost* to configure the alternate I/O domain.

Veritas SFHA installation

Based on the configuration, you need to install the Veritas SFHA stack for providing reliable storage and high availability to the guest logical domains.

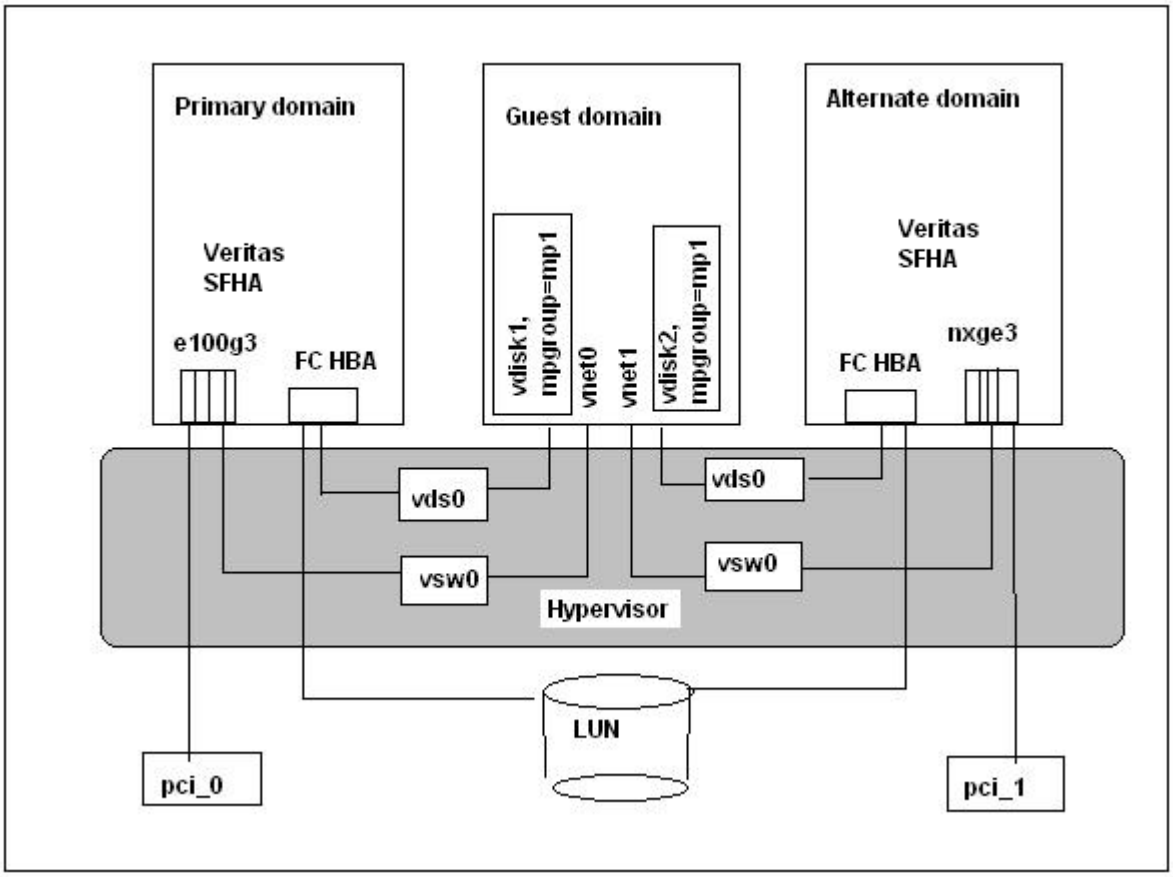
Guest domain storage configuration	**Veritas Software**
Raw SAN LUN's / ZFS volumes or image files	SFHA
VxVM volumes or image files	VCS

Install the Veritas SFHA stack on the Primary and Alternate I/O domain of each node in the cluster. Refer to Veritas Installation Guides.

Scenario 1	Virtual disks configured using raw SAN LUN's as backend devices provided to the guest domain through the virtual disk services from the control domain and the alternate I/O domain. The virtual devices are configured in an mpgroup to provide multipathing at the hypervisor level.	Virtual network device provided to the guest domain through the virtual switch services from the control domain and the alternate I/O domain. IPMP configured inside the guest domain for providing network availability.

Scenario 2	Virtual disks configured using distinct ZFS volumes as backend devices provided to the guest domain through the virtual disk services from the control domain and the alternate I/O domain. The virtual disks are then configured in a ZFS mirror for providing storage reliability and availability.	Virtual network device provided to the guest domain through the virtual switch services from the control domain and the alternate I/O domain. IPMP configured inside the guest domain for providing network availability.

Scenario 1



Setup:

- The boot disk for the guest domain will be configured using raw SAN LUN provided through the virtual disk service from the primary and alternate domain.
- The multipathing to the virtual disk provided to the guest domain will be handled by the hypervisor by configuring it as part of the same mpgroup.
- The virtual network interfaces provided through the inside the Guest domain will be configured in an IPMP group to provide network fault resiliency.
- VCS is installed in the control and alternate I/O domain. The underlying physical storage and network resources will be monitored by VCS from the primary and alternate domain.
- The AlternateIO agent provides a consolidated status of the underlying infrastructure resources availability to the LDom resource configured under VCS.

Guest domain configuration

1. Create the guest domain and configure resources for the domain.

primhost# ldm add-domain guest1
primhost# ldm add-vcpu 8 guest1
primhost# ldm add-mem 4G guest1

2. Attach the Solaris OS ISO image to the guest domain for installation.

primhost# ldm add-vdisk cdrom dvdiso@primary-vds0 guest1

3. Specify the LUN device to be exported by the virtual disk service as a virtual disk to the guest domain. In this example, the same SAN LUN device is exported from the primary and the alternate virtual disk service.

primhost# ldm add-vdsdev mpgroup=mp1 /dev/dsk/c3t50060E8000C46C50d2s2 rawboot@primary-vds0
primhost# ldm add-vdsdev mpgroup=mp1 /dev/dsk/c1t50060E8000C46C50d2s2 rawboot@alternate-vds0

4. Add the virtual disk to the guest domain. You need to assign only one virtual disk to the guest domain even though there are multiple paths that have been configured to the back end device. The hypervisor will seamlessly handle switching device paths configured in the same multipathing group when access to the storage device is not lost.

primhost# ldm add-vdisk timeout=10 vdisk1 rawboot@primary-vds0 guest1

5. Specify the virtual network device.

primhost# ldm add-vnet vnet1 primary-vsw0 guest1
primhost# ldm add-vnet vnet2 alternate-vsw0 guest1

6. Perform similar configuration on other host in the clusters **OR** if the device configurations are similar on the other hosts you need to perform step 5 explicitly on those host and export the configuration and import it to other hosts in the cluster.

7. Bind the devices to the guest logical domain and start the logical domain.

primhost# ldm bind guest1
primhost# ldm start guest1

8. Connect to the guest domain console and install the Oracle Solaris OS in the guest domain. Once the guest domain is installed, you can configure the network interfaces to be part of an IPMP group to provide network reliability and availability.

9. Configure IPMP (IP Multipathing) within the guest logical domain. Below configuration provides network multipathing between the network devices vnet0 and vnet1. The IP address 192.168.1.100 will made high available in this configuration.

guest1# cat /etc/hostname.vnet0
192.168.1.1 netmask 255.255.255.0 broadcast + group nwgrp deprecated -failover up addif 192.168.1.100 netmask 255.255.252.0 broadcast + failover up
guest1# cat /etc/hostname.vnet1
192.168.1.2 netmask 255.255.255.0 broadcast + group nwgrp deprecated -failover up

Veritas Cluster Server configuration

1. Create service group to monitor the storage device made available to the guest logical domain. In this example, *primhost-strg* is the service group that contains the resources to monitor the storage services provided to the guest logical domain through the primary and alternate I/O domain on host *primhost*.

Primhost# hagrp –add primhost-strg
Primhost# hagrp –modify primhost-strg SystemList primhost 0 primhost-alt 1
Primhost# hagrp –modify primhost-strg Parallel 1
Primhost# hares –add rawdisk1 Disk primhost-strg
Primhost# hares –modify rawdisk1 Critical 1
Primhost# hares –modify rawdisk1 Enabled 1

2. The resource configuration may have to be localized. In this example, the same SAN LUN made available through the primary and alternate I/O domain have different device names.

Primhost# hares –local rawdisk1 Partition
Primhost# hares –modify rawdisk1 Partition /dev/rdsk/c3t50060E8000C46C50d2s2 –sys primhost
Primhost# hares –modify rawdisk1 Partition /dev/rdsk/c1t50060E8000C46C50d2s2 –sys primhost-alt

3. Since the Service group contains only persistent resources (Disk), you need to configure a phantom resource to ensure that the service group status is reflected.

Primhost# hares –add phantom1 Phantom primhost-strg

4. Create service group to monitor the network device made available to the virtual switch service. In this example, *primhost-nw* is the service group used to monitor the network devices on host *primhost*.

Primhost# hagrp –add primhost-nw
Primhost# hagrp –modify primhost-nw SystemList primhost 0 primhost-alt 1
Primhost# hagrp –modify primhost-nw Parallel 1
Primhost# hares –add nic1 NIC primhost-nw
Primhost# hares –modify nic1 Critical 1
Primhost# hares –modify nic1 Enabled 1
Primhost# hares –local nic1 Device
Primhost# hares –modify nic1 Device e1000g3 –sys primhost
Primhost# hares –modify nic1 Device nxge3 –sys primhost-alt

5. Since the Service group contains only persistent resources (NIC), you need to configure a phantom resource to ensure that the service group status is reflected.

Primhost# hares –add phantom2 Phantom primhost-nw

6. You need to create similar service groups to monitor the storage and network services on the other hosts in the cluster. In this example, *sechost-strg* is the service group that contains the resources to monitor the storage services provided to the guest logical domain through the primary and alternate I/O domain on host *sechost*.

Primhost# hagrp –add sechost-strg
Primhost# hagrp –modify sechost-strg SystemList sechost 0 sechost-alt 1
Primhost# hagrp –modify sechost-strg Parallel 1
Primhost# hares –add rawdisk2 Disk sechost-strg
Primhost# hares –modify rawdisk2 Critical 1
Primhost# hares –modify rawdisk2 Enabled 1
Primhost# hares –local rawdisk2 Partition
Primhost# hares –modify rawdisk2 Partition /dev/rdsk/c3t50060E8000C46C50d2s2 –sys sechost
Primhost# hares –modify rawdisk2 Partition /dev/rdsk/c1t50060E8000C46C50d2s2 –sys sechost-alt
Primhost# hares –add phantom3 Phantom sechost-strg

7. Create service group to monitor the network device made available to the virtual switch service. In this example, *sechost-nw* is the service group used to monitor the network devices on host *sechost*.

Primhost# hagrp –add sechost-nw
Primhost# hagrp –modify sechost-nw SystemList sechost 0 sechost-alt 1
Primhost# hagrp –modify sechost-nw Parallel 1
Primhost# hares –add nic2 NIC sechost-nw
Primhost# hares –modify nic2 Critical 1
Primhost# hares –modify nic2 Enabled 1
Primhost# hares –local nic2 Device
Primhost# hares –modify nic2 Device e1000g3 –sys sechost
Primhost# hares –modify nic2 Device nxge3 –sys sechost-alt
Primhost# hares –add phantom4 Phantom sechost-nw

8. Configure the service group containing the AlternateIO resource . In this example, the underlying storage services are of type Online only and these resources will be online on all the hosts in the cluster. Hence the service group aiosg1 will be configured as a Parallel service group. The system list of this service group will contain the hostnames of the control domains of all the nodes in the cluster.

Primhost# hagrp –add aiosg1
Primhost# hagrp –modify aiosg1 SystemList primhost 0 sechost 1

Primhost# hagrp –modify aiosg1 Parallel 1
Primhost# hares –add aiores1 AlternateIO aiosg1
Primhost# hares –modify aiores1 Critical 1
Primhost# hares –modify aiores1 Enabled 1
Primhost# hares –local aiores1 StorageSG
Primhost# hares –modify aiores1 StorageSG primhost-strg 0 –sys primhost
Primhost# hares –modify aiores1 StorageSG sechost-strg 0 –sys sechost
Primhost# hares –local aiores1 NetworkSG
Primhost# hares –modify aiores1 NetworkSG primhost-nw 0 –sys primhost
Primhost# hares –modify aiores1 NetworkSG sechost-nw 0 –sys sechost

9. Configure the service group containing the LDom resource. In this example, *ldomsg1* will contain the LDom resource. The system list of this service group will contain the hostnames of the control domains of all the nodes in the cluster.

Primhost# hagrp –add ldomsg1
Primhost# hagrp –modify ldomsg1 SystemList primhost 0 sechost 1
Primhost# hares –add ldmres1 LDom ldomsg1
Primhost# hares –modify ldmres1 Critical 1
Primhost# hares –modify ldmres1 Enabled 1
Primhost# hares –modify ldmres1 LDomName guest1

10. The service group containing the LDom resource will have a online local hard dependency with the service group containing the AlternateIO resource.

Primhost# hagrp –link ldomsg1 aiosg1 online local hard

VCS configuration file

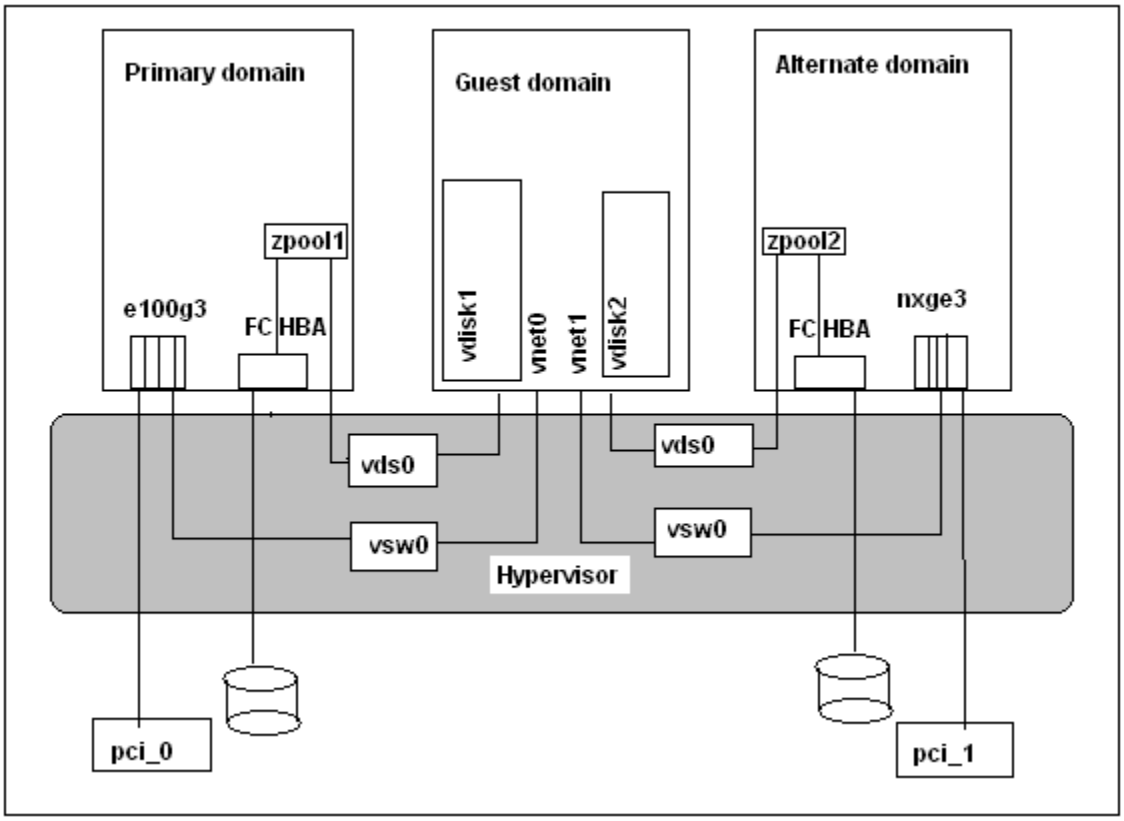
Main.cf

```
include "types.cf"

cluster aioclus (
        UserNames = { admin = XXXXXXXX }
        Administrators = { admin }
        HacliUserLevel = COMMANDROOT
        )

system primhost (
        )
```

```
        system primhost-alt (
                )

        system sechost (
                )

        system sechost-alt (
                )

        group ldomsg1 (
                SystemList = { primhost = 0, sechost = 1 }
                        )
                LDom ldmres1 (
                LDomName = guest1
                        )

                requires group aiosg1 online local hard

        group aiosg1 (
                SystemList = { primhost = 0, sechost = 1 }
                Parallel = 1
                                )
         AlternateIO aiores1 (
          StorageSG @primhost = { primhost-strg  = 0 }
         StorageSG @sechost = { sechost-strg = 0 }
         NetworkSG @primhost = { primhost-nw = 0 }
          NetworkSG @sechost = { sechost-nw = 0 }
        )

group primhost-strsg (
        SystemList = { primhost = 0, primhost-alt = 1 }
        Parallel = 1
        AutoStartList = { primhost, primhost-alt }
        )

        Disk rawdisk1 (
                Partition @primhost = "/dev/rdsk/c3t50060E8000C46C50d2s2"
                Partition @primhost-alt = "/dev/rdsk/c1t50060E8000C46C50d2s2"
                )
        Phantom phantom1 (
                )
```

```
group primhost-nw (
        SystemList = { primhost = 0, primhost-alt = 1 }
        Parallel = 1
        AutoStartList = { primhost, primhost-alt }
        )

        NIC nicres1 (
                Device @primhost = e1000g3
                Device @primhost-alt = nxge3
                )

        Phantom phantom2 (
                )

group sechost-strsg (
        SystemList = { sechost = 0, sechost-alt = 1 }
        Parallel = 1
        AutoStartList = { sechost, sechost-alt }
        )

        Disk rawdisk2 (
                Partition @sechost = "/dev/rdsk/c3t50060E8000C46C50d2s2"
                Partition @sechost-alt = "/dev/rdsk/c1t50060E8000C46C50d2s2"
                )
        Phantom phantom3 (
                )

group sechost-nw (
        SystemList = { sechost = 0, sechost-alt = 1 }
        Parallel = 1
        AutoStartList = { sechost, sechost-alt }
        )

        NIC nicres2 (
                Device @sechost = e1000g3
                Device @sechost-alt = nxge3
                )

        Phantom phantom4 (
                )
```

Scenario 2



Setup:

- The boot disk for the guest domain will be configured using distinct ZFS volume provided through the virtual disk service from the primary and alternate domain.
- The OS is configured on a ZFS root pool. The ZFS root pool will be mirrored to provide additional reliability.
- The virtual network interfaces provided to the Guest domain will be configured in an IPMP group to provide network fault resiliency.
- VCS is installed in the control and alternate I/O domain. The underlying ZFS pool and network resources will be monitored by VCS from the primary and alternate domain.
- The AlternateIO agent provides a consolidated status of the underlying infrastructure resources availability to the LDom resource configured under VCS.

Guest domain configuration

1. Create the guest domain and configure resources for the domain.

primhost# ldm add-domain guest2
primhost# ldm add-vcpu 8 guest2

primhost# ldm add-mem 4G guest2

2. Attach the Solaris OS ISO image to the guest domain for installation.

primhost# ldm add-vdisk cdrom dvdiso@primary-vds0 guest2

3. Create the ZFS volume that will be exported through the primary virtual disk service. Setting the Altroot to disable cachefile to none.

Primhost# zpool create –R / zpool1 c3t50060E8000C46C50d3
Primhost#zfs create –V 15g zpool1/guest2/disk0

4. Create the ZFS volume that will be exported through the alternate virtual disk service. Setting the Altroot to disable cachefile to none.

Primhost-alt# zpool create –R / zpool2 c1t50030D630636F50d1
Primhost-alt#zfs create –V 15g zpool2/guest2/disk1

5. Specify the ZFS filesystem device to be exported by the virtual disk service as a virtual disk to the guest domain. Here two distinct ZFS volumes having similar size are exported from the primary and the alternate virtual disk service.

Primhost# ldm add-vdsdev /dev/zvol/dsk/ zpool1/guest2/disk0 vol0@primary-vds0
Primhost# ldm add-vdsdev /dev/zvol/dsk/ zpool2/guest2/disk1 vol1@alternate-vds0

6. Assign the virtual disk to the guest logical domain.

Primhost# ldm add-vdisk timeout=10 bdisk1 vol1@primary-vds0 guest2
Primhost# ldm add-vdisk timeout=10 bdisk2 vol2@alternate-vds0 guest2

7. Specify the virtual network device.

primhost# ldm add-vnet vnet1 primary-vsw0 guest2
primhost# ldm add-vnet vnet2 alternate-vsw0 guest2

8. Perform similar configuration on other host in the clusters **OR** if the device configurations are similar on the other hosts you need to perform step 5 explicitly on those host and export the configuration and import it to other hosts in the cluster.

9. Bind the devices to the guest logical domain and start the logical domain.

primhost# ldm bind guest2
primhost# ldm start guest2

10. Install the Oracle Solaris OS in the guest domain. During the installation select ZFS as the option for the root file system

```
Choose Filesystem Type

Select the filesystem to use for your Solaris installation
```

```
[ ] UFS
[X] ZFS
```

Once the guest domain is installed, you can configure the ZFS root pool mirror and configure the network interfaces in an IPMP group to provide better availability and reliability.

11. Configuring the ZFS root pool mirror.
    a. Determine the disk on which the root pool is configured.

```
Guest2# zpool status
  pool: rpool
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        rpool       ONLINE       0     0     0
          c0d0s0  ONLINE       0     0     0
```

    b. Determine the disk device to be attached to the root pool to form a mirror.

```
Guest2# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
       0. c0d0 <SUN-DiskImage-12GB cyl 339 alt 2 hd 96 sec 768>
          /virtual-devices@100/channel-devices@200/disk@0
       1. c0d2 <SUN-DiskImage-12GB cyl 339 alt 2 hd 96 sec 768>
          /virtual-devices@100/channel-devices@200/disk@2
Specify disk (enter its number):
```

    c. Label the disk device to be added to the root pool

```
Guest2# prtvtoc /dev/rdsk/c0d0s0 | fmthard -s - /dev/rdsk/c0d2s0
```

    d. Attach the disk device to the root pool

```
Guest2# zpool attach rpool c0d0s0 c0d2s0
```

    e. Check the root pool status

```
Guest2# zpool status
  pool: rpool
 state: ONLINE
status: One or more devices is currently being resilvered.  The pool will
        continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
 scan: resilver in progress since Fri Aug 12 15:54:54 2011
```

2.16G scanned out of 14.8G at 71.3M/s, 0h3m to go
2.16G resilvered, 14.59% doneconfig:

```
NAME        STATE     READ WRITE CKSUM
rpool       ONLINE       0     0     0
  mirror-0  ONLINE       0     0     0
    c0d0s0  ONLINE       0     0     0
    c0d2s0  ONLINE       0     0     0 (resilvering)
```

errors: No known data errors

f. Install the boot block to make the disk bootable.
Guest2# installboot -F zfs /usr/platform/`uname -i`/lib/fs/zfs/bootblk /dev/rdsk/c0d2s0

12. Configure IPMP (IP Multipathing) within the guest logical domain. Below configuration provides network multipathing between the network devices vnet0 and vnet1. The IP address 192.168.1.101 will made high available in this configuration.
Guest2# cat /etc/hostname.vnet0
192.168.1.3 netmask 255.255.255.0 broadcast + group nwgrp deprecated -failover up addif 192.168.1.101 netmask 255.255.252.0 broadcast + failover up
Guest2# cat /etc/hostname.vnet1
192.168.1.4 netmask 255.255.255.0 broadcast + group nwgrp deprecated -failover up

VCS configuration

1. Create service group to monitor the storage device made available to the guest logical domain. In this example, *primhost-zfssg* is the service group that contains the resources to monitor the storage services provided to the guest logical domain through the primary and alternate I/O domain on host *primhost*.

Primhost# hagrp -add primhost-zfssg
Primhost# hagrp -modify primhost-zfssg SystemList primhost 0 primhost-alt 1
Primhost# hagrp -modify primhost-zfssg Parallel 0

2. Configure the zpool resource to monitor the Zpool volumes that are provided to the Guest logical domain. In this example, the zpool monitored in the control and alternate I/O domains are different.
Note: To ensure proper failover of the guest logical domain it is required to set the *FailMode* attribute of the Zpool to *panic*. Setting the *FailMode* attribute to *panic* causes the zpool resource to fault when the zpool looses access to the underlying storage devices. However, if the zpool configured in the control domain looses access to the underlying storage devices, it will cause the control domain to panic and resources configured under VCS control to failover to the other node in the cluster .

Refer to the Veritas BARG to know more about configuring zpool resources.

Primhost# hares -add zfs1 Zpool primhost-zfssg
Primhost# hares -modify zfs1 Critical 1
Primhost# hares -modify zfs1 FailMode panic
Primhost# hares -local zfs1 PoolName
Primhost# hares -modify zfs1 PoolName zpool2 -sys primhost-alt
Primhost# hares -modify zfs1 PoolName zpool1 -sys primhost
Primhost# hares -modify zfs1 Enabled 1

3. Create service group to monitor the network device made available to the virtual switch service. In this example, *primhost-znw* is the service group used to monitor the network devices on host *primhost*.

Primhost# hagrp –add primhost-znw
Primhost# hagrp –modify primhost-znw SystemList primhost 0 primhost-alt 1
Primhost# hagrp –modify primhost-znw Parallel 1
Primhost# hares –add nic3 NIC primhost-znw
Primhost# hares –modify nic3 Critical 1
Primhost# hares –modify nic3 Enabled 1
Primhost# hares –local nic3 Device
Primhost# hares –modify nic3 Device e1000g3 –sys primhost
Primhost# hares –modify nic3 Device nxge3 –sys primhost-alt
Primhost# hares –add phantom5 Phantom primhost-nw

4. Create service group to monitor the storage device made available to the guest logical domain on *sechost*. In this example, *sechost-zfssg* is the service group that contains the resources to monitor the storage services provided to the guest logical domain through the primary and alternate I/O domain on host *sechost*.

Primhost# hares -add zfs2 Zpool sechost-zfssg
Primhost# hares -modify zfs2 Critical 1
Primhost# hares -modify zfs2 FailMode panic
Primhost# hares -local zfs2 PoolName
Primhost# hares -modify zfs2 PoolName zpool2 -sys sechost-alt
Primhost# hares -modify zfs2 PoolName zpool1 -sys sechost
Primhost# hares -modify zfs2 Enabled 1

5. Configure the service group to monitor the network device on node *sechost*. In our example, *sechost-znw* is the service group configured to monitor the network devices providing network access to the guest logical domains on *sechost*.

Primhost# hagrp –add sechost-znw
Primhost# hagrp –modify sechost-nw SystemList sechost 0 sechost-alt 1
Primhost# hagrp –modify sechost-znw Parallel 1
Primhost# hares –add nic4 NIC sechost-znw

Primhost# hares –modify nic4 Critical 1
Primhost# hares –modify nic4 Enabled 1
Primhost# hares –local nic4 Device
Primhost# hares –modify nic4 Device e1000g3 –sys sechost
Primhost# hares –modify nic4 Device nxge3 –sys sechost-alt
Primhost# hares –add phantom6 Phantom sechost-znw

6. Configure the service group containing the AlternateIO resource . In this example, the underlying storage services (zpool) will be online only on a single node in the cluster and hence the service group *aiosg2* containing the AlternateIO resource will be configured as a failover service group. The system list of this service group will contain the hostnames of the control domains of all the nodes in the cluster.
   Note: PREONLINE triggers have to be configured for service groups of failover type containing AlternateIO resources. These triggers ensure that the service groups configured as part of the StorageSG attribute values whose key are set to 1 are bought OFFLINE or are already in an OFFLINE or OFFLINE|FAULTED state before bringing the AlternateIO resource on any node in the cluster.

Primhost# hagrp –add aiosg2
Primhost# hagrp –modify aiosg2 SystemList primhost 0 sechost 1
Primhost# hagrp –modify aiosg2 Parallel 0
Primhost# hagrp -modify aiosg2 TriggerPath bin/AlternateIO
Primhost# hagrp -modify aiosg2 TriggersEnabled PREONLINE -sys primhost
Primhost# hagrp -modify aiosg2 TriggersEnabled PREONLINE -sys sechost
Primhost# hares –add aiores2 AlternateIO aiosg2
Primhost# hares –modify aiores2 Critical 1
Primhost# hares –modify aiores2 Enabled 1
Primhost# hares –local aiores2 StorageSG
Primhost# hares –modify aiores2 StorageSG primhost-zfssg 1 –sys primhost
Primhost# hares –modify aiores2 StorageSG sechost-zfssg 1 –sys sechost
Primhost# hares –local aiores2 NetworkSG
Primhost# hares –modify aiores2 NetworkSG primhost-znw 0 –sys primhost
Primhost# hares –modify aiores2 NetworkSG sechost-znw 0 –sys sechost

7. You need to configure the preonline trigger at the service group level for StorageSG whose attribute value is set to 1, e.g. primhost-zfssg and sechost-zfssg in this scenario should not be online on both the nodes in the cluster as this may cause data corruption. In this case the resources that are managed by both the service groups are same but since the names are different across the cluster nodes, VCS cannot prevent a concurrency violation when you try to online these service groups independently. The trigger is stored in /opt/VRTSvcs/ bin/AlternateIO/StorageSG/preonline directory.

Primhost# hagrp -modify primhost-zfssg   TriggerPath   bin/AlternateIO/StorageSG
Primhost# hagrp -modify   primhost-zfssg   TriggersEnabled PREONLINE –sys primhost
Primhost# hagrp -modify   primhost-zfssg   TriggersEnabled PREONLINE –sys primhost-alt

Primhost# hagrp -modify sechost-zfssg TriggerPath bin/AlternateIO/StorageSG
Primhost# hagrp -modify sechost-zfssg TriggersEnabled PREONLINE –sys sechost
Primhost# hagrp -modify sechost-zfssg TriggersEnabled PREONLINE –sys sechost-alt

8. Configure the service group containing the LDom resource. In our example, *ldomsg2* will contain the LDom resource. The system list of this service group will contain the hostnames of the control domains of all the nodes in the cluster.

Primhost# hagrp –add ldomsg2
Primhost# hagrp –modify ldomsg2 SystemList primhost 0 sechost 1
Primhost# hares –add ldmres2 LDom ldomsg2
Primhost# hares –modify ldmres2 Critical 1
Primhost# hares –modify ldmres2 Enabled 1
Primhost# hares –modify ldmres2 LDomName guest2

9. The service group containing the LDom resource will have a online local hard dependency with the service group containing the AlternateIO resource.

Primhost# hagrp –link ldomsg2 aiosg2 online local hard

VCS Configuration file

Main.cf

```
cluster aioclus (
    UserNames = { admin = XXXXXXXX }
    Administrators = { admin }
    HacliUserLevel = COMMANDROOT
    )

system primhost (
    )

system primhost-alt (
    )

system sechost (
    )

system sechost-alt (
    )
```

```
        group ldomsg2  (
         SystemList = { primhost = 0, sechost = 1 }
                    )

         LDom ldmres2 (
         LDomName = guest2
         )

          requires group aiosg2 online local hard

        group aiosg2 (
          SystemList = { primhost = 0, sechost = 1 }
           TriggerPath = "bin/AlternateIO"
          TriggersEnabled @primhost = { PREONLINE }
          TriggersEnabled @sechost = { PREONLINE }
           )

          AlternateIO aiores2 (
         StorageSG @primhost = { primhost-zfssg = 1 }
         StorageSG @sechost = { sechost-zfssg = 1 }
         NetworkSG @primhost = { primhost-znw = 0 }
         NetworkSG @sechost = { sechost-znw = 0 }
         )

group primhost-zfssg (
      SystemList = { primhost = 0, primhost-alt = 1 }
      TriggerPath = "bin/AlternateIO/StorageSG"
      TriggersEnabled @primhost = { PREONLINE }
      TriggersEnabled @primhost-alt = { PREONLINE }
       AutoStart = 0
      Parallel = 1
      )

      Zpool zfs1 (
          Failmode = panic
           PoolName @primhost = zfsprim
           PoolName @primhost-alt = zfsmirr
           )

group primhost-znw (
      SystemList = { primhost = 0, primhost-alt = 1 }
      Parallel = 1
```

```
        AutoStartList = { primhost, primhost-alt }
        )

        NIC nicres3 (
                Device @primhost = e1000g3
                Device @primhost-alt = nxge3
                )

        Phantom phantom5 (
                )

group sechost-zfssg (
        SystemList = { sechost = 0, sechost-alt = 1 }
        TriggerPath = "bin/AlternateIO/StorageSG"
        TriggersEnabled @sechost = { PREONLINE }
        TriggersEnabled @sechost-alt = { PREONLINE }
        AutoStart = 0
        Parallel = 1
         )

        Zpool zfs2 (
             Failmode = panic
              PoolName @sechost = zfsprim
              PoolName @sechost-alt = zfsmirr
              )

group sechost-znw (
        SystemList = { sechost = 0, sechost-alt = 1 }
        Parallel = 1
        AutoStartList = { sechost, sechost-alt }
        )

        NIC nicres3 (
                Device @sechost = e1000g3
                Device @sechost-alt = nxge3
                )

        Phantom phantom6 (
                )
```