Veritas InfoScale™ 7.4.1 Virtualization Guide - Solaris

Last updated: 2019-02-01

Legal Notice

THE DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. VERITAS TECHNOLOGIES LLC SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be commercial computer software as defined in FAR 12.212 and subject to restricted rights as defined in FAR Section 52.227-19 "Commercial Computer Software - Restricted Rights" and DFARS 227.7202, et seq. "Commercial Computer Software and Commercial Computer Software Documentation," as applicable, and any successor regulations, whether delivered by Veritas as on premises or hosted services. Any use, modification, reproduction release, performance, display or disclosure of the Licensed Software and Documentation by the U.S. Government shall be solely in accordance with the terms of this Agreement.

Veritas Technologies LLC
500 E Middlefield Road
Mountain View, CA 94043

http://www.veritas.com

Technical Support

Technical Support maintains support centers globally. All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policies. For information about our support offerings and how to contact Technical Support, visit our website:

https://www.veritas.com/support

You can manage your Veritas account information at the following URL:

https://my.veritas.com

If you have questions regarding an existing support agreement, please email the support agreement administration team for your region as follows:

Worldwide (except Japan)	CustomerCare@veritas.com
Japan	CustomerCare_Japan@veritas.com

Documentation

Make sure that you have the current version of the documentation. Each document displays the date of the last update on page 2. The latest documentation is available on the Veritas website:

https://sort.veritas.com/documents

Documentation feedback

Your feedback is important to us. Suggest improvements or report errors or omissions to the documentation. Include the document title, document version, chapter title, and section title of the text on which you are reporting. Send feedback to:

infoscaledocs@veritas.com

You can also see documentation information or ask a question on the Veritas community site:

http://www.veritas.com/community/

Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a website that provides information and tools to automate and simplify certain time-consuming administrative tasks. Depending on the product, SORT helps you prepare for installations and upgrades, identify risks in your datacenters, and improve operational efficiency. To see what services and tools SORT provides for your product, see the data sheet:

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

Contents

Section 1

Overview of Veritas InfoScale Solutions used in Solaris virtualization

Chapter 1

Overview of Veritas InfoScale Virtualization Solutions

This chapter includes the following topics:

Overview of the Veritas InfoScale Products Virtualization Guide

This document provides information about Veritas InfoScale products support for Solaris virtualization technologies. It contains:

- High-level conceptual information for Veritas InfoScale products and how they function in Solaris virtual environments.
- High level implementation information for setting up Veritas InfoScale products in Solaris virtual environments.
- Use case chapters with examples of how Veritas InfoScale products can improve performance outcomes for common Solaris virtualization use cases.

See "Solaris virtualization documentation" on page 219.

About Veritas InfoScale support for Solaris virtualization technology

This section describes the Veritas InfoScale support for Solaris virtualization technology:

About SmartIO in the Solaris virtualized environments

Storage Foundation and High Availability Solutions are supported in the Solaris virtualized environments. This section describes how you can use SmartIO in the Solaris virtualized environments.

Table 1-1 shows how SmartIO can be used in the Oracle VM Server for SPARC environment.

When you install Storage Foundation and High Availability Solutions in the guest, you can use SmartIO to cache data onto an SSD or any other supported fast device. The SSD used for the cache can be either a PCIe or SAS device, or an array-based SSD.

When you install Storage Foundation and High Availability Solutions in the control domain, you can use VxVM read caching, VxFS read caching and write-back caching at the control domain level.

SmartIO caching is supported for Storage Foundation for Oracle RAC only on guest domains.

If an array-based SSD is used, live migration is supported with SmartIO caching. With direct attached devices (PCIe), live migration is not supported if SmartIO caching is enabled. If you need to perform live migration, you can use manual steps.

See “Performing live migration between LDOMs in the SmartIO environment” on page 14.

Table 1-1 Solaris: Oracle VM Server for SPARC

Configuration in guest	Configuration in control domain	Caching takes place:	VxVM read caching	VxFS read caching	VxFS writeback caching
"Split stack" VxFS	SF (VxVM/CVM/DMP)	in the control domain	Yes	No	No
"Guest-based stack" SF	Any	in the guest	Yes	Yes	Yes

Table 1-1 Solaris: Oracle VM Server for SPARC *(continued)*

Configuration in guest	Configuration in control domain	Caching takes place:	VxVM read caching	VxFS read caching	VxFS writeback caching
"Guest-based stack" SFCFS	Any	in the guest	Yes	Yes	Yes
"Guest-based stack" SFRAC	No SF stack	in the guest	Yes	Yes	No
"Layered stack" SF	DMP	in the guest	Yes	Yes	Yes
ZFS	SF, CVM,VxFS, or CFS	in the control domain	Yes	N/A	N/A

Table 1-2 shows how SmartIO can be used in the Solaris zones environment.

Table 1-2 Solaris: zones

Configuration in non-global zones	Configuration in global zone	Caching takes place:	VxVM read caching	VxFS read caching	VxFS writeback caching
SF	SF	in the global zone	Yes	Yes	Yes
SFCFS	SF	N/A	N/A	N/A	N/A

Performing live migration between LDOMs in the SmartIO environment

If an array-based SSD is used, live migration is supported with SmartIO caching. With direct attached devices (PCIe), live migration is not supported if SmartIO caching is enabled. If you need to perform live migration, you can use manual steps.

To perform live migration in the SmartIO environment

1 To prepare the LDOM for the live migration, perform the following steps:

- Offline the cache area that is created inside the LDOM.

```
Ldom1:/root# sfcache offline cachearea_name
```

- Delete the cache area.

  ```
  Ldom1:/root# sfcache delete cachearea_name
  ```

2. Remove the SSD device from the VxVM configuration so that the device can be unexported from the LDOM.

   ```
   Ldom1:/root# vxdisk rm ssd_device_name
   ```

3. Verify that the SSD device is removed from VxVM. The SSD device is not visible in the output of the following command:

   ```
   Ldom1:/root# vxdisk list
   ```

4. Unexport the device from the LDOM.

   ```
   Cdom1:/root> ldm remove-vdisk vdisk_name ldom1
   ```

5. After unexporting the local SSD device, perform the live migration of the LDOM. During the live migration, make sure that the application and mountpoints that use the SFHA objects are intact and running properly.
6. After the live migration completes, export the PCIe SSD devices that are available on the other control domain.

   ```
   Cdom1:/root> ldm add-vdsdev vxvm_device_path vds_device_name>@vds
   ```

   ```
   Cdom1:/root> ldm add-vdisk vdisk_name vds_device_name@vds ldom1
   ```

7. After exporting the local PCIe SSD devices, include the devices in the VxVM configuration that is inside the LDOM.

   ```
   Ldom1:/root> vxdisk scandisks
   ```

8. Verify that the SSD device is visible in the output of the following command:

   ```
   Ldom1:/root# vxdisk list
   ```

9. After the local PCIe device is available to the VxVM configuration, you can create the required SmartIO cache area.
10. To live migrate back the LDOM from target control domain to source control domain, follow step 1 to step 9.

Virtualization use cases addressed by Veritas InfoScale products

Virtualization use cases addressed by Veritas InfoScale products:

Table 1-3 Virtualization use cases

Use case	Details
Application failover	For LDOMs: See "Cluster Server setup to fail over an Application running inside logical domain on a failure of Application" on page 158. For Zones: See "Configuring VCS in zones" on page 35.
Application monitoring and management	For LDoms: For Zones: See "Configuring VCS in zones" on page 35.
Fast failover	For LDOMs: See "Clustering using Cluster Volume Manager" on page 140. For Zones: See "SFCFSHA mounts" on page 63.
Live migration	See "Oracle VM Server for SPARC guest domain migration in VCS environment" on page 159.
Physical to virtual migration	Physical to virtual migration in zone and LDom environments are basically provided from the operating system.
Physical to virtual migration for disaster recovery	For LDOMs: See "Configuring Oracle VM Server for SPARC guest domains for disaster recovery" on page 134. For Zones: See "Configuring Solaris non-global zones for disaster recovery" on page 76.

Table 1-3 Virtualization use cases *(continued)*

Use case	Details
Simplified management ▪ Consistent device naming ▪ Storage management and provisioning ▪ Storage management and boot disk provisioning	See “Provisioning storage for a guest domain” on page 118.
Storage availability	See “DMP as a multi-pathing solution in an Oracle VM Server for SPARC environment ” on page 108. See “Oracle VM Server for SPARC deployment models” on page 91.
Server consolidation	For LDOMs: See “About Oracle VM Server for SPARC” on page 89. For Zones: See “About Solaris Zones” on page 20.
Storage migration - array migration	See “Array migration” on page 92.
Virtual machine availability	For LDoms: See “About VCS in a Oracle VM Server for SPARC environment” on page 148. For Zones: See “About VCS support for zones” on page 20.

Section 2

Zones and Projects

Chapter 2

Storage Foundation and High Availability Solutions support for Solaris Zones

This chapter includes the following topics:

- Configuring Solaris non-global zones for disaster recovery
- Software limitations of Storage Foundation support of non-global zones

About Solaris Zones

Solaris Zones is a software partitioning technology, which provides a means of virtualizing operating system services to create an isolated environment for running applications. This isolation prevents processes that are running in one zone from monitoring or affecting processes running in other zones.

You can configure non-global zones with a shared-IP address or an exclusive-IP address. The shared-IP zone shares a network interface with global-zone and the exclusive-IP zone does not share network interface with global-zone.

See the *Oracle Solaris Administration: Oracle Solaris Zones and Resource Management* Solaris operating environment document.

Oracle provides regular updates and patches for the Oracle Solaris Zones feature. Contact Oracle for more information.

About VCS support for zones

Cluster Server (VCS) provides application management and high availability to applications running in zones.

With the Intelligent Monitoring Framework (IMF), VCS supports intelligent resource monitoring. The Zone agent is IMF-aware and uses the asynchronous monitoring framework (AMF) kernel driver for IMF notification.

For more information about the Intelligent Monitoring Framework (IMF) and intelligent resource monitoring, refer to the *Cluster Server Administrator's Guide*.

For more information about how to perform intelligent resource monitoring for the Zone agent, see the *Cluster Server Bundled Agents Reference Guide*.

Overview of how VCS works with zones

You can use VCS to perform the following:

- Start, stop, monitor, and fail over a non-global zone.
- Start, stop, monitor, and fail over an application that runs in a zone.

Topic	Description
How VCS models containers	VCS provides zone aware resource types to monitor applications inside the zone. For the zone aware resources, C based entry points run from global zone and fetch information from non-global zone when ever required. The script based entry points run with in non-global zone. If any resource faults, based on the configuration, VCS fails over either the service group with the zone or the service group with the application running in the zone to another node. You can configure VCS to use Veritas Product Authentication Service to run in a secure environment. Communication from non-global zones to global zones is secure in this environment.
Installing and configuring zones in VCS environments	Install and configure the zone. Create the service group with the standard application resource types (application, storage, networking) and the Zone resource. VCS manages the zone as a resource. You then configure the service group's ContainerInfo attribute. You can use the `hazonesetup` and `hazoneverify` utilities to create and verify the configuration.
Configuring the ContainerInfo attribute	The service group attribute ContainerInfo specifies information about the zone. When you have configured and enabled the ContainerInfo attribute, you have enabled the zone-aware resources in that service group to work in the zone environment. VCS defines the zone information at the level of the service group so that you do not have to define it for each resource. You may specify a per-system value for the ContainerInfo attribute. VCS has zone aware resource types which can be configured to monitor resources inside the non-global zone or on the global zone. For example, if a network interface is shared with the zone, the associated NIC resource should run in the global zone. If a resource is to be monitored inside non-global zone, you can define service group attribute ContainerInfo.
ResContainerInfo	If you want to configure more than one zone resource in a service group, you can set ResContainerInfo attribute at resource level for all the zone aware agents. In this case you must not set ContainerInfo attribute at service group level.

About the ContainerInfo service group attribute

The ContainerInfo attribute has the Name key, Type key, and Enabled key. The Name key defines the name of the container. The Type key lets you select the type of container that you plan to use. The Enabled key enables the Zone-aware resources within the service group. The ContainerInfo attribute specifies that the resources that belong to that Service Group will run inside the local zone, if they are Zone-aware.

Note: Zone-aware resources are resources capable of running inside a local zone.

Assign the following values to the ContainerInfo attribute:

- Name
  The name of the container.
- Type
  The type of container. You can set this to Zone.
- Enabled
  Specify the value as 0, if you want to disable the container. Specify the value as 1, if you want to enable the container. Specify the value as 2, to enable physical to virtual and virtual to physical failovers. When the value is 2, the Zone resource mimics a non-existent entity.

You can set a per-system value for this attribute.

About the ContainerOpts resource type attribute

The ContainerOpts resource attribute is pre-set for Zone-aware resource types. It determines the following:

- Whether the zone-aware resource can run in the zone.
- Whether the container information that is defined in the service group's ContainerInfo attribute is passed to the resource.

These values are only effective when you configure the ContainerInfo service group attribute.

Attribute's keys follow:

The ContainerOpts resource type attribute's definitions for Zone-aware types contain the following values:

- RunInContainer (RIC)
  When the value of the RunInContainer key is 1, the agent function (entry point) for that resource runs inside of the local container.
  When the value of the RunInContainer key is 0, the agent function (entry point) for that resource runs outside the local container (in the global environment).
  A limitation for the RunInContainer value is that only script agent functions (entry points) can run inside a container.

- PassCInfo (PCI)

  When the value of the PassCInfo key is 1, the agent function (entry point) receives the container information that is defined in the service group's ContainerInfo attribute. An example use of this value is to pass the name of the container to the agent.

About the ResContainerInfo resource type attribute

Set ResContainerInfo attribute for each of the zone aware resources in a service group. The ResContainerInfo attribute has the Name key, Type key, and Enabled key. The Name key defines the name of the container.

Assign the following values to this attribute:

- Name: The name of the container.
- Type: The type of container. You can set this to Zone.
- Enabled: Specify the value as 1 to enable the container. Specify the value as 0 to disable the container. Specify the value as 2 to enable failovers from physical computers to virtual machines and from virtual machines to physical computers.

When ResContainerInfo attribute is set at any resource level, you need not set ContainerInfo attribute at service group level. If you want to configure multiple zone resources in a service group, you can use ResContainerInfo attribute.

Zone-aware resources

Table 2-1 1ists the ContainerOpts attributes default values for resource types. Zone-aware resources have predefined values for the ContainerOpts attribute.

Note: Veritas recommends that you do not modify the value of the ContainerOpts attribute, with the exception of the Mount agent.

See "About the Mount agent" on page 24.

See "About networking agents" on page 34.

Table 2-1 ContainerOpts attribute default values for applications and resource types

Resource type	RunInContainer	PassCInfo
Apache	1	0
Application	1	0

Table 2-1 ContainerOpts attribute default values for applications and resource types *(continued)*

Resource type	RunInContainer	PassCInfo
ASMInst	1	0
ASMDG	1	0
Db2udb	1	0
NIC	0	1
IP	0	1
IPMultiNIC	0	1
IPMultiNICB	0	1
Process	1	0
Zone	0	1
Oracle	1	0
Netlsnr	1	0
Sybase	1	0
SybaseBk	1	0
ProcessOnOnly	1	0
Project	0	1

About the Mount agent

You may need to modify the ContainerOpts values for the Mount resource in certain situations.

In certain situations where the block device is not exported to zone, you can make the file system available inside local zone. Mount the block device on the directory that has a path that includes the zone root from global zone, for example:

```
BlockDevice = /dev/vx/dsk/dg/vol1
MountPoint = /zones/zone-test/root/mntpt
```

Where */zones/zone-test* is the zone root of the local zone.

Mount agent supports the following configuration for mount points

1 Direct mount of file system with mount point as full path seen from global zone. Typical mount resource configuration for this type of mount is shown below:

```
group mntgrp (
                  SystemList = { Sys1 = 0, Sys1 = 1 }
       )

       Mount mnt-direct (
               MountPoint = "/zones/zone-test/root/mnt"
               BlockDevice = "/dev/vx/dsk/dg/vol"
               FSType = vxfs
               FsckOpt = "-y"
               )
```

2 Loop-back file system mount inside non-global zone for file system mounted in global zone. Typical mount resource configuration for this type of mount is shown below:

```
group loopbacksg (
        SystemList = { sysA = 0, sysB = 1 }
        ContainerInfo@sysA = { Name = zone1, Type = Zone, Enabled = 1 }
        ContainerInfo@sysB = { Name = zone1, Type = Zone, Enabled = 1 }
        )

        Mount zone_mnt (
                MountPoint = "/export/home/zone1/root/lofs_mnt"
                BlockDevice = "/mnt1/m1"
                FSType = lofs
        )

        Zone z1 (
                       )
        Mount global_mnt (
                MountPoint = "/mnt1"
                BlockDevice = "/dev/vx/dsk/tdg/tvol1"
                FSType = vxfs
                FsckOpt = "-y"
                )
        zone_mnt requires z1
        zone_mnt requires global_mnt
```

3. Direct mount of NFS based file system inside non-global zone. Typical mount resource configuration for this type of mount is shown below:

```
group mntgrp (
       SystemList = { Sys1 = 0, Sys1 = 1 }
       ContainerInfo = { Name = zone-test, Type = Zone, Enabled = 1 }
       )

       Mount mntnfs (
               MountPoint = "/mnt"
               BlockDevice = "system:/shared-dir"
               FSType = nfs
               FsckOpt = "-n"
               ContainerOpts = { RunInContainer = 1, PassCInfo = 0 }
               )
```

4. Support for direct mount of VxFS file system inside non-global zone. VCS Mount agent supports the direct mount of VxFS file system inside non-global zone. Typical mount resource configuration for this type of mount is shown below:

```
group mntgrp  (
       SystemList = { sys1 = 0, sys2 = 1 }
       ContainerInfo = { Name = zone-test, Type = Zone, Enabled = 1 }
       Administrators = { z_zoneres_sys1, z_zoneres_sys2 }
       )

       Mount mnt-zone-direct (
               BlockDevice = "/dev/vx/dsk/data_dg/data_vol"
               MountPoint = "/mnt1"
               FSType = vxfs
               FsckOpt = "-y"
               ContainerOpts = { RunInContainer = 1, PassCInfo = 0 }
               )
```

Sample configurations for Physical-to-Virtual (P2V) scenarios

In the following sample configuration, the local zone (zone1) runs only on system 1 (sys1) and it does not exist in system 2 (sys2). The mount resource comes online inside the zone. On system 1, the `/export/home/zone1/root/mnt` mount point exists inside the zone. On system 1, the `/export/home` mount point is mounted on

/export/home/zone1/root/mnt inside the zone. On system 2, the /export/home mount point is mounted on /mnt in the global zone.

```
group mountgrp (
         SystemList = { sys1 = 0, sys2 = 1 }
         ContainerInfo @sys1 = { Name = zone1, Type = Zone, Enabled = 1 }
         ContainerInfo @sys2 = { Name = zone1, Type = Zone, Enabled = 2 }
         Administrators = { z_mountres_sys1, z_zoneres_sys1 }
         )


         Mount loopback (
                 MountPoint @sys1 = "/export/home/zone1/root/mnt"
                 MountPoint @sys2 = "/mnt"
                 BlockDevice = "/export/home"
                 FSType = lofs
                 FsckOpt = "-n"
                 )

         Zone zoneres (
                 )

         loopback requires zoneres
```

In this sample configuration, there are two mount resources. The first mount resource (mountres) mounts the actual shared volume /dev/vx/dsk/vdg/nfs_vol on /lockinfo. Next, /lockinfo is loop-backed on /export/home/zone1/root/mnt inside the zone on system 1. On system 2, the /dev/vx/dsk/vdg/nfs_vol volume is mounted on /lockinfo and then /lockinfo is loop-backed and mounted on /mnt in the global zone. In the following sample, localization has been done for the mount resource:

```
group mountgrp (
         SystemList = { sys1 = 0, sys2 = 1 }
         ContainerInfo @sys1 = { Name = zone1, Type = Zone, Enabled = 1 }
         ContainerInfo @sys2 = { Name = zone1, Type = Zone, Enabled = 2 }
         Administrators = { z_mountres_sys1, z_zoneres_sys1 }
         )

         DiskGroup dgres (
                 DiskGroup = vdg
                 )

         Mount loopback (
                 MountPoint @sys1 = "/export/home/zone1/root/mnt"
                 MountPoint @sys2 = "/mnt"
                 BlockDevice = "/lockinfo"
                 FSType = lofs
                 FsckOpt = "-n"
                 )

         Mount mountres (
                 MountPoint = "/lockinfo"
                 BlockDevice = "/dev/vx/dsk/vdg/nfs_vol"
                 FSType = vxfs
                 FsckOpt = "-y"
                 )

         Zone zoneres (
                 )

         loopback requires mountres
         loopback requires zoneres
         mountres requires dgres
```

Bringing a Mount resource online in the zone

The Mount resource is brought online in the global zone by default (RunInContainer = 0). If you want to bring a mount resource online inside the non-global zone, perform the following:

- Export the block device to the zone through zone configuration. Ensure that the raw volume is used appropriately to prevent the possibility of data corruption.
- Modify the ContainerInfo attribute for the service group and set values for the Name, Type, and Enabled keys.

  ```
  # hagrp -modify service_group ContainerInfo Name zone_name \
  Type Zone Enabled 1
  ```

- Override the ContainerOpts attribute at the resource level.
- Set the value of the RunInContainer key to 1, for example:

  ```
  # hares -override Mountres ContainerOpts
  # hares -modify Mountres ContainerOpts \
  RunInContainer 1 PassCInfo 0
  ```

For information on overriding resource type static attributes, see the *Cluster Server Administrator's Guide*.

Setting the attribute values for a Mount resource for NFS mounts

For NFS mounts, you must mount in the non-global zone.

- Modify the ContainerInfo attribute for the service group and set values for Name, Type and Enabled keys.
- Override the ContainerOpts attribute at the resource level.
- Set the value of the RunInContainer key to 1.

  Set the RIC value to 1. When you set RIC=1, specify the value of the MountPoint attribute relative to the zone root, for example:

  ```
  BlockDevice = abc:/fs1
  MountPoint = /mnt1
  ```

  The file system is mounted on `/zone_root/mnt1`.

Configuring a direct mount of VxFS file system in a non-global zone with VCS

Typical steps to configure a direct mount inside a non-global zone.

To configure a direct mount inside a non-global zone

1. Create a VxVM disk group and volume:

   - Create a VxVM disk group from a device:

     ```
     global# vxdg init data_dg c0t0d1
     ```

   - Create a volume from a disk group:

     ```
     global# vxassist -g data_dg make data_vol 5G
     ```

     For more information, see the *Storage Foundation Administrator's Guide*.

2. Create a zone:

   - Create a root directory for a zone local-zone and change its permission to 700:

     ```
     global# mkdir -p /zones/local-zone
     global# chmod 700 /zones/local-zone
     ```

   - On Solaris 11, configure a zone local-zone:

```
global# zonecfg -z local-zone

local-zone: No such zone configured
Use `create' to begin configuring a new zone.

zonecfg:local-zone> create
zonecfg:local-zone> set zonepath=/zones/local-zone
zonecfg:local-zone> set ip-type=shared
zonecfg:local-zone> add net
zonecfg:local-zone:net> set physical=eri0
zonecfg:local-zone:net> set address=192.168.5.59
zonecfg:local-zone:net> end
zonecfg:local-zone > verify
zonecfg:local-zone > commit
zonecfg:local-zone > exit
```

The zone is in configured state.

- Install the zone:

  ```
  global# zoneadm -z local-zone install
  ```

- Login to the zone console to setup the zone from terminal 1:

  ```
  global# zlogin -C local-zone
  ```

- Boot the zone from another terminal:

  ```
  global# zoneadm -z local-zone boot
  ```

- Follow the steps on terminal 1 on the zone console to setup the zone. See the *Oracle* documentation for more information about creating a zone.

3. Add VxVM volumes to the zone configuration:

   - Check the zone status and halt the zone, if it is running:

     ```
     global# zoneadm list -cv
     ID NAME       STATUS   PATH           BRAND    IP
     0 global      running  /              native    shared
     2 local-zone  running  /zones/myzone  native    shared

     global# zoneadm -z myzone halt
     ```

   - Add the VxVM devices to the zone's configuration:

```
global# zonecfg -z local-zone
zonecfg:local-zone:fs> add device
zonecfg:local-zone:fs> set match=/dev/vxportal
zonecfg:local-zone:fs> end
zonecfg:local-zone:fs> add device
zonecfg:local-zone:fs> set match=/dev/fdd
zonecfg:local-zone:fs> end
zonecfg:local-zone:fs> add device
zonecfg:local-zone:fs> set match=/dev/vx/rdsk/data_dg/data_vol
zonecfg:local-zone:fs> end
zonecfg:local-zone:fs> add device
zonecfg:local-zone:fs> set match=/dev/vx/dsk/data_dg/data_vol
zonecfg:local-zone:fs> end
zonecfg:local-zone:fs> add fs
zonecfg:local-zone:fs> set dir=/etc/vx/licenses/lic
zonecfg:local-zone:fs> set special=/etc/vx/licenses/lic
zonecfg:local-zone:fs> set type=lofs
zonecfg:local-zone:fs> end
zonecfg:local-zone> verify
zonecfg:local-zone> commit
zonecfg:local-zone> exit
```

- On Solaris 11, you must set `fs-allowed` to `vxfs` and `odm` in the zone's configuration:

  ```
  global# zonecfg -z myzone
  zonecfg:myzone> set fs-allowed=vxfs,odm
  zonecfg:myzone> commit
  zonecfg:myzone> exit
  ```

  See "Adding a direct mount to a zone's configuration" on page 61.
  See "Mounting VxFS as VxFS inside a non-global zone" on page 60.

- Boot the zone:

  ```
  global# zoneadm -z myzone boot
  ```

4 Create a VxFS file system on the volume inside a non-global zone:

- Login to the local-zone:

  ```
  global# zlogin myzone
  ```

- Create a VxFS file system on the block device:

```
bash-3.00# mkfs -F vxfs /dev/vx/dsk/data_dg/data_vol
```

5 Create a mount point inside the zone:

- Login to the local-zone:

  ```
  global# zlogin myzone
  ```

- Create a mount point inside the non-global zone:

  ```
  bash-3.00# mkdir -p /mydata
  ```

- Mount the VxFS file system on the mount point:

  ```
  bash-3.00# mount -F vxfs /dev/vx/dsk/data_dg/data_vol /mydata
  ```

6 Configure the zone service group:

- On the first node, create the service group with password-less communication with global zone:

  ```
  global# hazonesetup -g zone_grp -r zone_res -z myzone \
  -p password -s sysA,sysB
  ```

- Switch the service group from the first node to the second node and run the `hazonesetup` command to setup password-less communication from the next node.
- Repeat step 6 for all the nodes in the cluster where the zone can go online.

See "Configuring a zone resource in a failover service group with the hazonesetup utility" on page 45.

7 Create a mount, disk group, and volume resources into the service group:

- Add a disk group resource to the service group:

  ```
  global# hares -add dg_res DiskGroup zone_grp
  global# hares -modify dg_res DiskGroup data_dg
  global# hares -modify dg_res Enabled 1
  ```

- Add a volume resource to the service group:

  ```
  global# hares -add vol_res Volume zone_grp
  global# hares -modify vol_res Volume data_vol
  global# hares -modify vol_res DiskGroup data_dg
  global# hares -modify vol_res Enabled 1
  ```

- Add a Mount resource to the service group:

  ```
  global# hares -add mnt_res Mount zone_grp
  global# hares -modify mnt_res BlockDevice \
  /dev/vx/dsk/data_dg/data_vol
  global# hares -modify mnt_res MountPoint /mydata
  global# hares -modify mnt_res FSType vxfs
  global# hares -modify mnt_res FsckOpt %-y
  global# hares -modify mnt_res Enabled 1
  ```

- Create a resource dependency between the resources in the service group:

  ```
  global# hares -link zone_res vol_res
  global# hares -link vol_res dg_res
  global# hares -link mnt_res zone_res
  ```

8. Set the ContainerOpts attribute for the Mount resource for VxFS direct mount:

   - Override the ContainerOpts attribute at the resource level for mnt_res:

     ```
     global# hares -override mnt_res ContainerOpts
     ```

   - Set the value of the RunInContainer key to 1:

     ```
     global# hares -modify mnt_res ContainerOpts RunInContainer \
     1 PassCInfo 0
     ```

For information on overriding resource type static attributes: see the *Cluster Server Administrator's Guide*.

9 Here is a sample configuration for the VxFS direct mount service groups in the `main.cf` file:

```
group zone_grp (
       SystemList = {sysA = 0, sysB = 1 }
       ContainerInfo = { Name = local-zone, Type = Zone, Enabled = 1 }
       Administrators = { z_zoneres_sysA, z_zoneres_sysB }
        )

       Mount mnt_res (
                BlockDevice = "/dev/vx/dsk/data_dg/data_vol"
                MountPoint = "/mydata"
                FSType = vxfs
                FsckOpt = "-y"
                ContainerOpts = { RunInContainer = 1, PassCInfo = 0 }
                 )

       DiskGroup dg_res (
                DiskGroup = data_dg
        )

       Volume vol_res (
                Volume = data_vol
                DiskGroup = data_dg
        )

       Zone zone_res (
        )

       zone_res requires vol_res
       vol_res requires dg_res
       mnt_res requires zone_res
```

About networking agents

Enable the attribute ExclusiveIPZone for resources of type IP and NIC when these resources are configured to manage the IP and the NIC inside an exclusive-IP zone. This attribute is disabled by default. The IP agent and the NIC agent assumes the native zone (shared-IP) by default.

Cluster Server (VCS) brings resources online in the global zone by default.

If you want to bring these resources online inside the exclusive-IP zone, perform the following tasks:

- Make sure that the resource is in a service group that has valid ContainerInfo attribute value configured.
- Set the value of the ExclusiveIPZone attribute to 1.

Note: The exclusive-IP zone supports the IP and NIC networking agents. For more information about these agents, see the *Cluster Server Bundled Agents Reference Guide*.

About the Zone agent

The Zone agent monitors zones, brings them online, and takes them offline. For more information about the agent, see the *Cluster Server Bundled Agents Reference Guide*.

Use the `hazonesetup` utility to create user account with group administrative privileges. The DeleteVCSZoneUser attribute of zone resource controls removing the user account when the zone resource is taken offline. For more information, see the *Cluster Server Bundled Agents Reference Guide*.

About configuring failovers among physical and virtual servers

You can configure VCS to fail over from a physical system to a virtual system and vice versa. A physical to virtual failover gives an N + N architecture in an N + 1 environment. For example, several physical servers with applications can fail over to containers on another physical server.

See “Configuring for physical to virtual and virtual to physical failovers—a typical setup” on page 57.

Configuring VCS in zones

Configuring VCS in zones involves the following tasks:

First	Review the prerequisites. See “Prerequisites for configuring VCS in zones” on page 36.
Second	Decide on the location of the zone root, which is either on local storage or shared storage. See “Deciding on the zone root location” on page 37.

Third	Install the application in the zone. See "About installing applications in a zone" on page 41.
Fourth	Create the application service group and configure its resources. See "Configuring the service group for the application" on page 42.

Prerequisites for configuring VCS in zones

Review the following prerequisites for configuring VCS in zones:

- For Oracle Solaris 11, VCS supports only ZFS for zone root.

Method for file system access inside non-global zone

File system mounts must meet one of the following two conditions:

- **Use a loopback file system with zone configuration.** All mounts that the application uses must be part of the zone configuration and must be configured in the service group. For example, you can create a zone, z-ora, and define the file system containing the application's data to have the mount point as `/oradata`. When you create the zone, you can define a path in the global zone. An example is `/export/home/oradata`, which the mount directory in the non-global zone maps to. The MountPoint attribute of the Mount resource for the application is set to `/export/home/oradata`. Confirm that `/export/home/oradata` maps to `/oradata` with the `zonecfg -z zone_name info` command. You can also look into the zone configuration `/etc/zones/zone_name.xml` file. The Zone resource depends on the Mount resource.

- **Mount the file system inside zone's root-path.** All file system mount points that the application uses that run in a zone must be set relative to the zone's root. For example, if the Oracle application uses `/oradata`, and you create the zone with the zonepath as `/z_ora`, then the mount must be `/z_ora/root/oradata`. The MountPoint attribute of the Mount resource must be set to this path. The Mount resource depends on the Zone resource.

Using custom agents in zones

If you use custom agents, review the following information for their use in zones:

- If you use custom agents to monitor the applications that run in zones, you can use C++ or script-based entry points.

- If you want the custom agent to monitor an application in the zone, for the custom agent type, set the following values for the ContainerOpts attribute: RunInContainer = 1 and the PassCInfo = 0.

- If you don't want the custom agent to monitor an application in the zone, for the custom agent type, set the following values for the ContainerOpts attribute: RunInContainer = 0 and the PassCInfo = 0.
- Two main use cases exist where you might want to use a RunInContainer = 0 and PassCInfo = 1, descriptions of these follow.
  - The first is the Zone agent's use of these values. The Zone agent's entry points cannot run inside of the non-global zone but the agent itself manages the zone. RunInContainer requires a value of 0 because the agent must run in the global zone. PassCInfo has a value of 1 because the Zone agent requires the name of the container from the ContainerInfo service group attribute.
  - The second case is how the IP agent uses RunInContainer and PassCInfo. The IP agent's entry points must run outside of the non-global zone because a shared-IP zone may cause the networking stack to not completely run in the non-global zone. You cannot perform an `ifconfig` command and then plumb the IP from inside of a non-global zone. When you run the `ifconfig` command in the global zone with the zone option - it plumbs the IP and makes it available to the zone that you specify. The need for the container's name comes from the use of this command, even though it cannot run in the container. This is applicable to all networking agents.
    If the non-global zone is of type exclusive IP zone, the agent runs the script based entrypoints inside the non-global zone if the zone is in running state. The agents runs the script based entrypoints in global zone if the zone is not in running state.
    For NIC assigned to exclusive IP Zone and IP to be plumbed inside exclusive IP zone you can set ExclusiveIPZone attribute available to NIC and IP resources.

Deciding on the zone root location

Each zone has its own section of the file system hierarchy in the zone root directory. Processes that run in the zone can access files only within the zone root.

You can set the zone root in the following two ways:

- Zone root on local storage
  In this configuration, you must configure and install a zone on each node in the cluster.
  To configure a zone under VCS control using the hazonesetup utility when the zone root is on local storage
- Zone root on shared storage

In this configuration, configure and install a zone in shared storage from one system and duplicate the configuration on each node in the cluster.

Setting the zone root on shared storage means you need to install the non-global zone on shared storage from one system only. The zone root can fail over to the other systems. To do this, the system software, including the patches, must be identical on each system during the existence of the zone.

To configure a zone under VCS control using the hazonesetup utility when the zone root is on shared storage

Creating a zone with root on local disk

Create a zone root on the local disk on each node in the cluster. The file system for application data is on a shared device and is either the loopback type or the direct mount type. For a direct mount file system, run the mount command from the global zone with the mount point specified as the complete path that starts with the zone root. For a loopback file system, add it into the zone's configuration before you boot the zone.

To create a zone root on local disks on each node in the cluster

1. Configure the zone with the `zonecfg` command.

   ```
   zonecfg -z newzone
   zonecfg:newzone> create
   ```

2. Set the zonepath parameter to specify a location for the zone root.

   ```
   zonecfg:newzone> set zonepath=/export/home/newzone
   ```

3. For exclusive IP zone, set the ip-tye to exclusive:

   ```
   zonecfg:newzone> set ip-type=exclusive
   ```

4. Add network interface to zone configuration. This is required for non-global zone to communicate with had running in global zone.

   ```
   zonecfg:newzone> add net
   zonecfg:newzone:net> set physical=bge1
   zonecfg:newzone:net> set address=192.168.1.10
   zonecfg:newzone:net> end
   ```

5. Make sure global zone can be pinged from non-global zone with global zone hostname. You may need to add global zone hostname entry to `/etc/hosts` file inside non-global zone or enable DNS access from inside the non-global zone.

6 If your application data resides on a loopback mount file system, create the loopback file system in the zone.

7 Exit the zonecfg configuration.

```
zonecfg> exit
```

8 Create the zone root directory.

```
mkdir zonepath
```

9 Set permissions for the zone root directory.

```
chmod 700 zonepath
```

10 Install the non-global zone.

```
zoneadm -z newzone install
```

11 Repeat step 1 to step 10 on each system in the service group's SystemList.

12 If the application data is on a loopback file system, mount the file system containing the application's data on shared storage.

13 Boot the zone.

```
zoneadm -z newzone boot
```

14 If the application data is on a direct mount file system, mount the file system from the global zone with the complete path that starts with the zone root.

Creating a zone with root on shared storage

Create a zone with root which points to the shared disk's location on each node in the cluster. The file system for application data is on a shared device and is either the loopback type or the direct mount type. For a direct mount file system, run the mount command from the global zone with the mount point specified as the complete path that starts with the zone root. For a loopback file system, add it into the zone's configuration before you boot the zone.

To create a zone root on shared disks on each node in the cluster

1. Create a file system on shared storage for the zone root. The file system that is to contain the zone root may be in the same disk group as the file system that contains the application data.

2. Configure the zone with the `zonecfg` command.

   ```
   zonecfg -z newzone
   zonecfg:newzone> create
   ```

3. Set the zonepath parameter to specify a location for the zone root.

   ```
   zonecfg:newzone> set zonepath=/export/home/newzone
   ```

4. Add network interface to zone configuration. This is required for non-global zone to communicate with had running in global zone.

   ```
   zonecfg:newzone> add net
   zonecfg:newzone:net> set physical=bge1
   zonecfg:newzone:net> set address=192.168.1.10
   zonecfg:newzone:net> end
   ```

5. Make sure global zone can be pinged from non-global zone with global zone hostname. You may need to add global zone hostname entry to `/etc/hosts` file inside non-global zone or enable DNS access from inside the non-global zone.

6. If your application data resides on a loopback mount file system, create the loopback file system in the zone.

7. Exit the zonecfg configuration.

   ```
   zonecfg> exit
   ```

8. Create the zone root directory.

   ```
   mkdir zonepath
   ```

9. Set permissions for the zone root directory.

   ```
   chmod 700 zonepath
   ```

10. Repeat step 2 to step 9 on each system in the service group’s SystemList.

11. Mount the file system that contains the shared storage on one of the systems that share the storage to the directory specified in zonepath.

12 Run the following command to install the zone on the system where the zone path is mounted.

```
zoneadm -z newzone install
```

13 If the application data is on a loopback file system, mount the file system containing the application's data on shared storage.

14 Boot the zone.

```
zoneadm -z newzone boot
```

15 If the application data is on a direct mount file system, mount the file system from the global zone with the complete path that starts with the zone root.

Performing the initial internal zone configuration

When a zone is booted for the first time after installation, the zone is in an unconfigured state. The zone does not have an internal configuration for naming services. Its locale and time zone have not been set, and various other configuration tasks have not been performed. You need to perform the initial internal zone configuration after zone installation.

You can perform the internal zone configuration in the following ways:

- sysidcfg tool
- Zone console login

For more details refer to Oracle documentation about "Performing the Initial Internal Zone Configuration" section in the *Oracle Solaris Administration: Oracle Solaris Zones and Resource Management* guide.

About installing applications in a zone

Perform the following tasks to install the application in a zone:

- If you have created zones locally on each node in the cluster, install the application identically in all zones on all nodes. If you are installing an application that is supported by a Veritas High Availability agent, see the installation and configuration guide for the agent.
- Install the agent packages on the global zone and the currently existing zones. Installs the agents in future zones when they are installed.
  - On Solaris 11, the package needs to be installed separately inside each non-global zone.

- You must define all the mount points that the application uses that are configured in the zone in the service group's configuration.

Configuring the service group for the application

You need to configure the application service group and the required resource dependencies. The following diagrams illustrate different examples of resource dependencies. In one case the zone root is set up on local storage. In the other, zone root is set up on shared storage.

Resource dependency diagrams: zone root on local disks

The following resource dependency diagrams show zone configurations on local disks configured for loopback and direct mounted file systems.

If the zone root is on a cluster file system, ensure that service group containing zone resource is a fail-over type service group.

Figure 2-1 depicts the dependency diagram when the zone root is set up on local storage with the loopback file system for the application. You can replace the Mount resource with the CFSMount resource and the DiskGroup resource with the CVMVolDg resource in the following diagram. In this configuration, decide if you want the service group to be a parallel service group. If so, you may need to localize certain attributes for resources in the service group. For example, you have to change the IP resource's Address attribute for each node.

Figure 2-1 Zone root on local disks with loopback file system

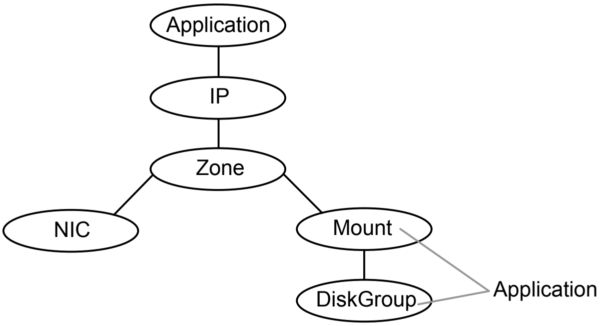
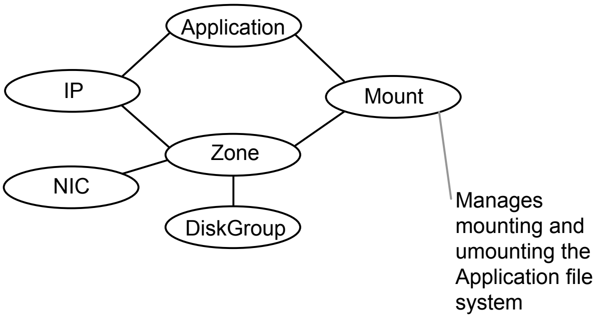


Figure 2-2 depicts the dependency diagram when the zone root is set up on local storage with a direct mount file system for the application. You can replace the Mount resource with the CFSMount resource and the DiskGroup resource with the CVMVolDg resource in the following diagram. In this configuration, decide if you want the service group to be a parallel service group. If so, you may need to localize certain attributes for resources in the service group. For example, you have to change the IP resource's Address attribute for each node.

Figure 2-2 Zone root on local disks with direct mount file system



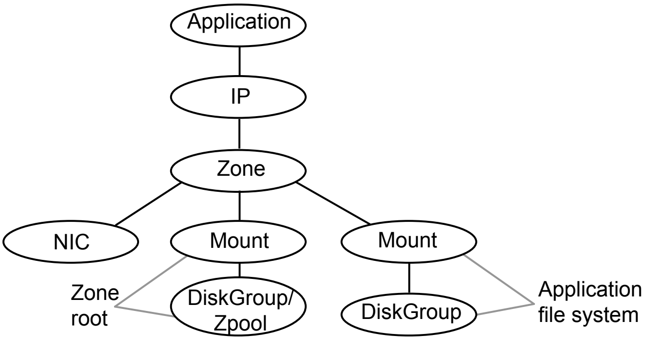
In a typical use case where application runs in parallel inside zones, you can have following configuration on Solaris. You can configure parallel service group containing zone resource whose zone roots are on a VxVM volume or Zpool and not shared between the nodes. The storage for the application is set up on shared storage between the nodes and managed using CFSMount and CVMVolDG resources. You may need to localize certain attributes for resources in the service group. For example you may have to change the IP resource’s Address attribute, the DiskGroup resource’s DiskGroup attribute or the Zpool resource’s PoolName attribute.

Resource dependency diagrams: zone root on shared disks

The following resource dependency diagrams show zone configurations on shared disks configured for loopback and direct mounted file systems.

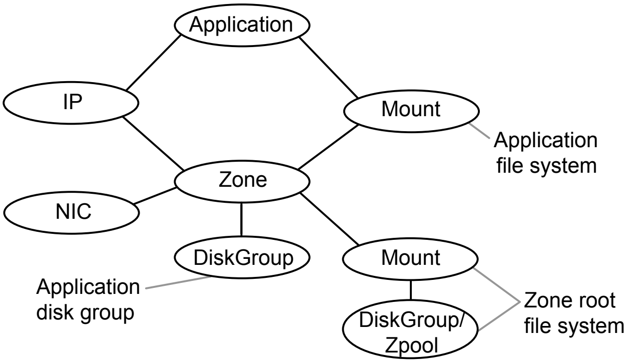
Figure 2-3 depicts the dependency diagram when a zone root is set up on shared storage with the loopback file system. You can replace the Mount resource with the CFSMount resource and the DiskGroup resource with the CVMVolDg resource in the following diagram for application. In this configuration, decide if you want the service group to be a parallel service group. If so, you may need to localize certain attributes for resources in the service group. For example, you have to change the IP resource's Address attribute for each node.

Figure 2-3 Zone root on shared storage with loopback file system



[Figure 2-4](#) depicts the dependency diagram when a zone root is set up on shared storage with the direct mount file system for the application. You can replace the Mount resource with the CFSMount resource and the DiskGroup resource with the CVMVolDg resource in the following diagram for application. In this configuration, decide if you want the service group to be a parallel service group. If so, you may need to localize certain attributes for resources in the service group. For example, you have to change the IP resource's Address attribute for each node.

Figure 2-4 Zone root on shared storage a direct mounted file system



Use the following principles when you create the service group:

- Set the MountPoint attribute of the Mount resource to the mount path.
- If the application requires an IP address, configure the IP resource in the service group.

Configuring a zone resource in a failover service group with the hazonesetup utility

The `hazonesetup` utility helps you configure a zone under VCS. This section covers typical scenarios based on where the zone root is located.

Two typical setups for zone configuration in a failover scenario follow:

- Zone root on local storage
  To configure a zone under VCS control using the hazonesetup utility when the zone root is on local storage
- Zone root on shared storage
  To configure a zone under VCS control using the hazonesetup utility when the zone root is on shared storage

Consider an example in a two-node cluster (sysA and sysB). Zone local-zone is configured on both the nodes.

To configure a zone under VCS control using the hazonesetup utility when the zone root is on local storage

1. Boot the non-global zone on first node outside VCS.

   ```
   sysA# zoneadm -z local-zone boot
   ```

2. To use the `hazonesetup` utility, ensure you have a IP configured for the non-global zone and hostname of the global zone is resolvable from non-global zone.

   ```
   # zlogin local-zone
   # ping sysA
   ```

3. Run the `hazonesetup` utility with correct arguments on the first node. This adds failover zone service group and zone resource in VCS configuration.

   ```
   sysA# hazonesetup -g zone_grp -r zone_res -z local-zone\
   -p password -a -s sysA,sysB
   ```

   Note: If you want to use a particular user for password-less communication use `-u` option of the `hazonesetup` utility. If `-u` option is not specified a default user is used for password-less communication.

4. Switch the zone service group to next node in the cluster.

   ```
   sysA# hagrp -switch zone_grp -to sysB
   ```

5. Run the `hazonesetup` utility with correct arguments on the node. The `hazonesetup` utlity detects that the zone service group and zone resource are already present in VCS configuration and update the configuration accordingly for password-less communication.

   ```
   sysB# hazonesetup -g zone_grp -r zone_res -z local-zone\
   -p password -a -s sysA,sysB
   ```

6. Repeat step 4 and step 5 for all the remaining nodes in the cluster.

To configure a zone under VCS control using the hazonesetup utility when the zone root is on shared storage

1. Configure a failover service group with required storage resources (DiskGroup, Volume, Mount, etc.) to mount the zone root on the node. Set the required dependency between storage resources (DiskGroup->Volume->Mount). Make sure you configure all the required attributes of all the storage resources in order to bring them online on cluster node.

   ```
   sysA# hagrp -add zone_grp
   sysA# hagrp -modify zone_grp SystemList sysA 0 sysB 1
   sysA# hares -add zone_dg DiskGroup zone_grp
   sysA# hares -add zone_vol Volume zone_grp
   sysA# hares -add zone_mnt Mount zone_grp
   sysA# hares -link zone_mnt zone_vol
   sysA# hares -link zone_vol zone_dg
   sysA# hares -modify zone_dg DiskGroup zone_dg
   sysA# hares -modify zone_dg Enabled 1
   sysA# hares -modify zone_vol Volume volume_name
   sysA# hares -modify zone_vol DiskGroup zone_dg
   sysA# hares -modify zone_vol Enabled 1
   sysA# hares -modify zone_mnt MountPoint /zone_mnt
   sysA# hares -modify zone_mnt BlockDevice /dev/vx/dsk/zone_dg/volume_name
   sysA# hares -modify zone_mnt FSType vxfs
   sysA# hares -modify zone_mnt MountOpt rw
   sysA# hares -modify zone_mnt FsckOpt %-y
   sysA# hares -modify zone_mnt Enabled 1
   ```

   When the zone root is on a ZFS file system, use the following commands:

   ```
   sysA# hagrp -add zone_grp
   sysA# hagrp -modify zone_grp SystemList sysA 0 sysB 1
   sysA# hares -add zone_zpool Zpool zone_grp
   sysA# hares -modify zone_zpool AltRootPath /zone_root_mnt
   sysA# hares -modify zone_zpool PoolName zone1_pool
   sysA# hares -modify zone_zpool Enabled 1
   ```

2. Bring the service group online on first node. This mounts the zone root on first node.

   ```
   sysA# hagrp -online zone_grp -sys sysA
   ```

3 Boot the local zone on first node outside VCS.

```
sysA# zoneadm -z local-zone boot
```

4 To use the `hazonesetup` utility, ensure you have a IP configured for the non-global zone and hostname of the global zone is resolvable from non-global zone.

```
# zlogin local-zone
# ping sysA
```

5 Run the `hazonesetup` utility with correct arguments on the first node. Use the service group configured in step 1. This adds the zone resource to VCS configuration.

```
sysB# hazonesetup -g zone_grp -r zone_res -z local-zone \
-p password -a -s sysA,sysB
```

Note: If you want to use a particular user for password-less communication use `-u` option of the `hazonesetup` utility. If `-u` option is not specified a default user is used for password-less communication.

6 Set the proper dependency between the Zone resource and other storage resources. The Zone resource should depend on storage resource (Mount or Zpool ->Zone).

```
sysA# hares -link zone_res zone_mnt
```

When the zone root is on a ZFS file system, use following command:

```
sysA# hares -link zone_res zone_zpool
```

7 Switch the service group to next node in the cluster.

```
sysA# hagrp -switch zone_grp -to sysB
```

8 Run the `hazonesetup` utility with correct arguments on the node. The `hazonesetup` utility detects that the service group and the zone resource are already present in VCS configuration and update the configuration accordingly for password-less communication.

```
sysB# hazonesetup -g zone_grp -r zone_res -z local-zone\
-p password -a -s sysA,sysB
```

9 Repeat step 7 and step 8 for all the remaining nodes in the cluster

Configuring zone resource in a parallel service group with the hazonesetup utility

The `hazonesetup` utility helps you configure a zone under VCS. This section covers typical scenarios based on the location of the zone root.

In the case of a zone resource in parallel service group, the zone root can be on local or shared storage that the node owns.

Consider an example in a two-node cluster (sysA and sysB). Zone local-zone1 is configured on sysA and local-zone2 is configured on sysB.

To configure a zone under VCS control using the hazonesetup utility when the zone root is on local storage

1 Boot the local zone on all the nodes outside VCS.

```
sysA# zoneadm -z local-zone1 boot
sysB# zoneadm -z local-zone2 boot
```

2 To use the `hazonesetup` utility, ensure you have a IP configured for the non-global zone and hostname of the global zone is resolvable from non-global zone.

3 Run the hazonesetup utility with correct arguments on all the nodes successively.

```
sysA# hazonesetup -g zone_grp -r zone_res -z local-zone1\
-p password -a -l -s sysA,sysB
sysB# hazonesetup -g zone_grp -r zone_res -z local-zone2\
-p password -a -l -s sysA,sysB
```

Note: If you want to use a particular user for password-less communication use -u option of the hazonesetup command. If -u option is not specified a default user is used for password-less communication.

4 Running the hazonesetup utility on first node adds parallel zone service group and zone resource in VCS configuration. Running the hazonesetup utility on other nodes detect that the zone service group and zone resource are already present in VCS configuration and update the configuration accordingly for password-less communication.

Note: Run the hazonesetup utility on all the nodes in the cluster that have a zone running on that node. This is required as hazonesetup runs the halogin command inside the local zone that enables password-less communication between local zone and global zone.

You can use the same user for multiple zones across systems. Specify the same user name using the -u option while running the hazonesetup utility for different zones on different systems. When you do not specify a user name while running the hazonesetup utility, the utility creates a user with the default user name z_resname_hostname for a non-secure cluster and z_resname_clustername for a secure cluster.

To configure a zone under VCS control using the hazonesetup utility when the zone root is on shared storage that the node owns

1. Configure a parallel service group with required storage resources (DiskGroup, Volume, Mount, etc.) to mount the zone root on the nodes. Set the required dependency between storage resources (DiskGroup->Volume->Mount). Make sure that you configure all the required attributes of all the storage resources to bring them online on cluster node. You may have to localize certain attributes of storage resources to online them in parallel on all the nodes of the cluster. If you have a parallel Service Group and you use a Diskgroup resource, the attributes for this resource must be localized, otherwise you may end up importing the same diskgroup on 2 nodes at the same time on a non-CVM environment.

   ```
   sysA# hagrp -add zone_grp
   sysA# hagrp -modify zone_grp Parallel 1
   sysA# hagrp -modify zone_grp SystemList sysA 0 sysB 1
   sysA# hares -add zone_dg DiskGroup zone_grp
   sysA# hares -add zone_vol Volume zone_grp
   sysA# hares -add zone_mnt Mount zone_grp
   sysA# hares -link zone_mnt zone_vol
   sysA# hares -link zone_vol zone_dg
   ```

   See the *Cluster Server Bundled Agents Reference Guide* for more details on configuring storage resources.

2. Bring the service group online on all the nodes. This command mounts the zone root on all the nodes.

   ```
   sysA# hagrp -online zone_grp -any
   ```

3. Boot the local zone on all the nodes outside VCS.

   ```
   sysA# zoneadm -z local-zone1 boot
   sysB# zoneadm -z local-zone2 boot
   ```

4 Run the hazonesetup utility with correct arguments on all the nodes successively.

```
sysA# hazonesetup -g zone_grp -r zone_res -z \
local-zone1 -p password -a -l -s sysA,sysB
sysB# hazonesetup -g zone_grp -r zone_res -z \
local-zone2 -p password -a -l -s sysA,sysB
```

Running the hazonesetup utility on first node adds parallel zone service group and zone resource in VCS configuration. Running the hazonesetup utility on other nodes detect that the zone service group and zone resource are already present in VCS configuration and update the configuration accordingly for password-less communication.

Note: If you want to use a particular user for password-less communication use -u option of the hazonesetup command. If -u option is not specified a default user is used for password-less communication.

5 Set the proper dependency between the Zone resource and other storage resources. The Zone resource should depend on storage resource (Mount->Zone).

```
sysA# hares -link zone_res zone_mnt
```

Note: Run the hazonesetup utility on all the nodes in the cluster that have a zone running on that node. This is required as the hazonesetup utility runs the halogin command inside the local zone that enables password-less communication between local zone and global zone.

You can use the same user for multiple zones across systems. Specify the same user name using the -u option while running the hazonesetup utility for different zones on different systems. When you do not specify a user name while running the hazonesetup utility, the utility creates a user with the default user name sz_resname_hostname for a non-secure cluster and z_resname_clustername for a secure cluster.

Configuring multiple zone resources using same VCS user for password less communication

The `hazonesetup` utility helps you configure multiple zones under VCS, which are using same VCS user for password less communication between non-global zone and global zone.

Consider an example in a two-node cluster (sysA and sysB). Zones local-zone1 and local-zone2 are configured on both the nodes.

To configure zones under VCS control in failover mode using the hazonesetup utility when the zone root is on local storage

1 Boot the local zones on first node outside VCS.

```
sysA# zoneadm -z local-zone1 boot
sysA# zoneadm -z local-zone2 boot
```

2 Run the `hazonesetup` utility with the correct arguments on the first node. This adds failover zone service group, zone resource in VCS configuration, configures same VCS user (zone_user) to be used for password less communication between non-global zone, and global zone.

```
sysA# hazonesetup -g zone1_grp -r zone1_res -z local-zone1\
-u zone_user -p password -a -s sysA,sysB
sysA# hazonesetup -g zone2_grp -r zone2_res -z local-zone2\
-u zone_user -p password -a -s sysA,sysB
```

3 Switch the zone service group to next node in the cluster.

```
sysA# hagrp -switch zone1_grp -to sysB
sysA# hagrp -switch zone2_grp -to sysB
```

4 Run the `hazonesetup` utility with correct arguments on the node. The `hazonesetup` utility detects that the zone service group, zone resource are already present in VCS configuration, and update the configuration accordingly for password-less communication.

```
sysB# hazonesetup -g zone1_grp -r zone1_res -z local-zone1\
-u zone_user -p password -a -s sysA,sysB
sysB# hazonesetup -g zone2_grp -r zone2_res -z local-zone2\
-u zone_user -p password -a -s sysA,sysB
```

5 Repeat step 3 and step 4 for all the remaining nodes in the cluster.

Modifying the service group configuration

Perform the following procedure to modify a service group's configuration.

To modify the configuration to manage a zone

1 Run the `hazonesetup` script to set up the zone configuration.

```
# hazonesetup [-t] -g sg_name -r res_name -z zone_name\
[-u user_name] -p password [-a] [-l] -s systems
```

Where the values are:

-t	Updates the password for the VCS zone user.
-g *sg_name*	Name of the zone service group to be created in VCS configuration.
-r *res_name*	Name of the zone resource to be created in VCS configuration.
-z *zone_name*	Name of the zone that is configured on the system.
-u *user_name*	Name of the VCS user used for password less communication between the local zone and the global zone. If no username is specified the default username is used.
-p *password*	Password for the VCS user used for password less communication. If Veritas Authentication Service is enabled, the password should be at least six characters long.
-a	Populate AutoStartList for the group.
-l	Configure a parallel service group. If you do not specify the -l option, a failover service group is created by default.
-s *systems*	A comma separated list of systems where the zone service group need to be configured, for example: sys1,sys2,sys3.

If the service group does not exist, the script creates a service group.

The script adds a resource of type Zone to the service group. The script logs in to the zone and runs the `halogin` command. It also creates a user account with group administrative privileges to enable password less communication between global zone and local zone for VCS.

2 Modify the resource dependencies to reflect your zone configuration. See the resource dependency diagrams for more information.

See "Configuring the service group for the application" on page 42.

3 Save the service group configuration and bring the service group online.

Verifying the zone configuration

Run the `hazoneverify` command to verify the zone configuration.

The command verifies the following requirements:

- The systems hosting the service group have the required operating system to run zones.
- The service group does not have more than one resource of type Zone.
- The dependencies of the Zone resource are correct.

To verify the zone configuration

1. If you use custom agents make sure the resource type is added to the APP_TYPES or SYS_TYPES environment variable.

   See "Using custom agents in zones" on page 36.

2. Run the `hazoneverify` command to verify the zone configuration.

   ```
   # hazoneverify servicegroup_name
   ```

Synchronizing the zone configuration across cluster nodes

Use the `hazoneconfigsync` command to synchronize the zone configuration across cluster nodes. The `hazoneconfigsync` command propagates the changes that are made to zone configuration on one node to other cluster nodes.

To synchronize non-global zone's configuration on all cluster nodes

◆ Run the `hazoneconfigsync` command.

```
# hazoneconfigsync<comma separated zone names>
```

See `hazoneconfigsync(1M)` manual page for more details.

Performing maintenance tasks

Perform the following maintenance tasks as necessary:

- Make sure that the zone configuration files are consistent on all the nodes at all times. The `/etc/zones/`*`zone_name`*`.xml` file is located on only the global zones.
- When you add a patch or upgrade the operating system on one node, make sure to upgrade the software on all nodes.
- Make sure that the application configuration is identical on all nodes. If you update the application configuration on one node, apply the same updates to all nodes.

To update password of VCS user used for password less communication

1. Run the `hazonesetup` utility with correct arguments on the node where Zone resource is online.

   ```
   sysA# hazonesetup -t -g zone_grp -r zone_res -z local-zone\
   -u zoneuser -p new_password -a -s sysA,sysB
   ```

2. Switch the zone service group to next node in the cluster.

   ```
   sysA# hagrp -switch zone_grp -to sysB
   ```

3. Run the `hazonesetup` utility with correct arguments on the node.

   ```
   sysB# hazonesetup -t -g zone_grp -r zone_res -z local-zone\
   -u zoneuser -p new_password -a -s sysA,sysB
   ```

4. Repeat step 2 through step 3 for all the remaining nodes in the cluster.

Troubleshooting zones

Use following information to troubleshoot Cluster Server (VCS) and zones:

- VCS HA commands do not work.

  Recommended actions:

  - Verify the VCS packages are installed.
  - Run the `halogin` command from the zone.
    For more information on the `halogin` command, refer to the *Cluster Server Administrator's* Guide.
  - Verify your VCS credentials. Make sure the password is not changed.
  - Verify the VxSS certificate is not expired.

- Resource does not come online in the zone.

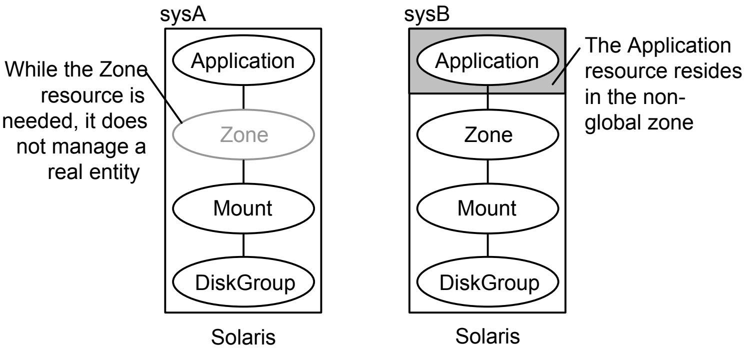
  Recommended actions:

  - Verify VCS and the agent packages are installed correctly.
  - Verify the application is installed in the zone.
  - Verify the configuration definition of the resource.

Configuring for physical to virtual and virtual to physical failovers—a typical setup

In this configuration, you have two physical nodes. One node runs Solaris without zones configured (sysA) and another node runs Solaris with zones configured (sysB).

Figure 2-5 An application service group that can fail over into a zone and back


sysA
sysB
While the Zone resource is needed, it does not manage a real entity
Application
Zone
Mount
DiskGroup
Application
Zone
Mount
DiskGroup
The Application resource resides in the non-global zone
Solaris
Solaris

In the main.cf configuration file, define the container name, type of container, and whether it is enabled or not in the service group definition.

```
ContainerInfo@sysA = {Name = Z1, Type = Zone, Enabled = 2}
ContainerInfo@sysB = {Name = Z1, Type = Zone, Enabled = 1}
```

On sysA, set the value of Enabled to 2 to ignore zones so that the application runs on the physical system. When the service group fails over to sysB, the application runs inside the zone after the failover because Enabled is set to 1 on sysB. The application can likewise fail over to sysA from sysB.

When ContainerInfo::Enabled is set to 2, the Zone agent reports resource state based on state of the corresponding group.

IMF monitoring must be disabled on the node where ContainerInfo::Enable is set to 2 (sysA in this example).

- To disable IMF monitoring, set the Mode key of IMF attribute to 0:

```
# hares -override zone_res IMF
# hares -local zone_res IMF
# hares -modify  zone_res IMF Mode 0 MonitorFreq 5 \
RegisterRetryLimit 3 -sys sysA
```

Adding VxFS file systems to a non-global zone

There are three ways in which you can provide VxFS file system to a non-global zone.

- Mounting VxFS as lofs into a non-global zone
- Mounting VxFS directly into a non-global zone from global zone
- Mounting VxFS as VxFS inside a non-global zone

Mounting VxFS as lofs into a non-global zone

VxFS file systems that were previously created in the global zone can be made available in the non-global zone using a loopback file system mount. This functionality is especially useful when the sole purpose of making the file system available in the non-global zone is to share access of this file system with one or more non-global zones. For example, if a configuration file is available in a particular file system and this configuration file is required by the non-global zone, then the file system can be shared with the non-global zone using a loopback file system mount.

The following commands share access of file system `/mnt1` as a loopback file system mount with an existing non-global zone `newzone`:

```
# zonecfg -z newzone
zonecfg:newzone> add fs
zonecfg:newzone:fs> set dir=/mnt1
zonecfg:newzone:fs> set special=/mnt1
zonecfg:newzone:fs> set type=lofs
zonecfg:newzone:fs> end
zonecfg:newzone> verify
zonecfg:newzone> commit
zonecfg:newzone> exit
```

The value of `dir` is a directory in the non-global zone. The value of `special` is the mount point for the file system inside the local zone.

This mount may be referred later as VxFS loopback mount.

Caution: Sharing file systems with non-global zones through a loopback file system mount makes the file system available for simultaneous access from all the non-global zones. This method should be used only when you want shared read-only access to the file system.

Mounting VxFS directly into a non-global zone from global zone

To direct mount a VxFS file system in a non-global zone, the directory to mount must be in the non-global zone and the mount must take place from the global zone. The following procedure mounts the directory `dirmnt` in the non-global zone `newzone` with a mount path of `/zonedir/newzone/root/dirmnt`.

Note: VxFS entries in the global zone `/etc/vfstab` file for non-global zone direct mounts are not supported, as the non-global zone may not yet be booted at the time of `/etc/vfstab` execution.

Once a file system has been delegated to a non-global zone through a direct mount, the mount point will be visible in the global zone through the `mount` command, but not through the `df` command.

To direct mount a VxFS file system in a non-global zone

1. Log in to the zone and make the mount point:

   ```
   global# zlogin newzone
   newzone# mkdir dirmnt
   newzone# exit
   ```

2. Mount the file system from the global zone:

   - Non-cluster file system:

     ```
     global# mount -F vxfs /dev/vx/dsk/dg/vol1 /zonedir/zone1\
     /root/dirmnt
     ```

   - Cluster file system:

     ```
     global# mount -F vxfs -o cluster /dev/vx/dsk/dg/vol1 \
     /zonedir/zone1/root/dirmnt
     ```

3. Log in to the non-global zone and ensure that the file system is mounted:

   ```
   global# zlogin newzone
   newzone# df | grep dirmnt
   /dirmnt (/dirmnt):142911566 blocks 17863944 files
   ```

Mounting VxFS as VxFS inside a non-global zone

You can create a VxFS file system inside non-global zones.

To create the VxFS file system inside non-global zones

1 Check the zone status and halt the zone:

```
global# zoneadm list -cv
ID NAME             STATUS     PATH                BRAND    IP
0 global           running    /                   solaris  shared
1 myzone           running    /zone/myzone        solaris  shared
global# zoneadm -z myzone halt
```

2 Add devices to the zone's configuration:

```
global# zonecfg -z myzone
zonecfg:myzone> add device
zonecfg:myzone:device> set match=/dev/vxportal
zonecfg:myzone:device> end
zonecfg:myzone> add device
zonecfg:myzone:device> set match=/dev/fdd
zonecfg:myzone:device> end
zonecfg:myzone> add device
zonecfg:myzone:device> set match=/dev/vx/rdsk/dg_name/vol_name
zonecfg:myzone:device> end
zonecfg:myzone> add device
zonecfg:myzone:device> set match=/dev/vx/dsk/dg_name/vol_name
zonecfg:myzone:device> end
zonecfg:myzone> add fs
zonecfg:myzone:fs> set dir=/etc/vx/licenses/lic
zonecfg:myzone:fs> set special=/etc/vx/licenses/lic
zonecfg:myzone:fs> set type=lofs
zonecfg:myzone:fs> end
zonecfg:myzone> verify
zonecfg:myzone> commit
zonecfg:myzone> exit
```

3. On Solaris 11, you must set fs-allowed=vxfs,odm to the zone's configuration:

   ```
   global# zonecfg -z myzone
   zonecfg:myzone> set fs-allowed=vxfs,odm
   zonecfg:myzone> commit
   zonecfg:myzone> exit
   ```

   If you want to use ufs, nfs and zfs inside the zone, set fs-allowed=vxfs,odm,nfs,ufs,zfs.

4. Boot the zone:

   ```
   global# zoneadm -z myzone boot
   ```

5. Login to the non-global zone and create the file system inside the non-global zone:

   ```
   global# zlogin myzone
   myzone# mkfs -F vxfs /dev/vx/rdsk/dg_name/vol_name
   ```

6. Create a mount point inside the non-global zone and mount it:

   ```
   myzone# mkdir /mnt1
   myzone# mount -F vxfs /dev/vx/dsk/dg_name/vol_name /mnt1
   ```

   Mounting a VxFS file system as a cluster file system from the non-global zone is not supported.

Adding a direct mount to a zone's configuration

A non-global zone can also be configured to have a VxFS file system direct mount automatically when the zone boots using `zonecfg`. The `fsck` command is run, before the file system is mounted. If the `fsck` command fails, the zone fails to boot.

To add a direct mount to a zone's configuration

1 Check the status and halt the zone:

```
global# zoneadm list -cv
ID NAME      STATUS     PATH                BRAND    IP
0 global     running    /                   solaris  shared
1 myzone     running    /zone/myzone        solaris  shared
global# zoneadm -z myzone halt
```

2 Add devices to the zone's configuration:

```
global# zonecfg -z myzone
zonecfg:myzone> add device
zonecfg:myzone:device> set match=/dev/vxportal
zonecfg:myzone:device> end
zonecfg:myzone> add device
zonecfg:myzone:device> set match=/dev/fdd
zonecfg:myzone:device> end
zonecfg:myzone> add fs
zonecfg:myzone:fs> set dir=/dirmnt
zonecfg:myzone:fs> set special=/dev/vx/dsk/dg_name/vol_name
zonecfg:myzone:fs> set raw=/dev/vx/rdsk/dg_name/vol_name
zonecfg:myzone:fs> set type=vxfs
zonecfg:myzone:fs> end
zonecfg:myzone> add fs
zonecfg:myzone:fs> set dir=/etc/vx/licenses/lic
zonecfg:myzone:fs> set special=/etc/vx/licenses/lic
zonecfg:myzone:fs> set type=lofs
zonecfg:myzone:fs> end
zonecfg:myzone> verify
zonecfg:myzone> commit
zonecfg:myzone> exit
```

3 On Solaris 11, you must set fs-allowed=vxfs,odm to the zone's configuration:

```
global# zonecfg -z myzone
zonecfg:myzone> set fs-allowed=vxfs,odm
zonecfg:myzone> commit
zonecfg:myzone> exit
```

If you want to use ufs, nfs and zfs inside the zone, set fs-allowed=vxfs,odm,nfs,ufs,zfs.

4 Boot the zone:

```
global# zoneadm -z myzone boot
```

5 Ensure that the file system is mounted:

```
myzone# df | grep dirmnt
/dirmnt (/dirmnt):142911566 blocks 17863944 files
```

Benefits of a VxFS mount in a non-global zone over VxFS mount from global zone into the non-global zone

Direct mount of the VxFS file system in a non-global zone have several benefits over direct mount from global zone into the non-global zone:

- With VxFS direct mount inside a non-global zone, the zone administrator can take advantage of VxFS file system.
- Limits the visibility and access to the VxFS file system for other non-global zone.
- The zone administrator can mount and unmount the file system directly inside the zone.

SFCFSHA mounts

Storage Foundation Cluster File System (SFCFSHA) provides support for the same file system to be made available from multiple nodes that have been grouped together as a cluster. VxFS supports the sharing or delegation of cluster-mounted file systems in the non-global zone.

Note: A non-global zone root can not be created on VxFS or CFS for Solaris 11. The only supported file system for non-global zone root on Solaris 11 is zfs.

See “Mounting VxFS as lofs into a non-global zone” on page 58.

The requirements to support SFCFSHA in non-global zones are similar to those for SFCFSHA support in global zones. Some key points are as follows:

- Both lofs and direct mount are supported; Veritas recommends direct mount.
- The device must be visible and shared on all nodes.

- The zone configuration must be the same on all nodes. The zone name can be different.

Mounting a VxFS file system as a cluster file system from the non-global zone is not supported.

Support for SFCFSHA in a non-global zone is available in Veritas File System 5.0 Maintenance Pack 1 and later.

To direct mount a VxFS file system as a cluster file system in a non-global zone

1 Log in to the zone and make the mount point:

```
global# zlogin newzone
newzone# mkdir dirmnt
newzone# exit
```

2 Mount the file system from the global zone.

Cluster File System:

```
global# mount -F vxfs -o cluster /dev/vx/dsk/dg/vol1 \
/zonedir/zone1/root/dirmnt
```

Note: It is not possible to make SFCFSHA cluster between different non-global zones of the same node.

Note: Cluster file system should not be used as a part of non-global zone configuration. It should be used through VCS as a part of `main.cf` configuration or it should be mounted manually from global zone. If using LOFS to mount the cluster file system within the non global zone, then do not use any cluster file system related options in the zone configuration since the cluster file system will already be mounted in the global zone.

Concurrent I/O access in non-global zones

Concurrent I/O allows multiple processes to read from or write to the same file without blocking other `read`(2) or `write`(2) calls. POSIX semantics requires `read` and `write` calls to be serialized on a file with other `read` and `write` calls. Concurrent I/O is generally used by applications that require high performance for accessing data and do not perform overlapping writes to the same file.

Storage Foundation supports concurrent I/O for applications running in the non-global zones as well. This implies that a process in a non-global zone can access the file concurrently with other processes in the global or non-global zone. The application or running threads are responsible for coordinating the write activities to the same file when using Concurrent I/O.

An application must perform the following activities to enable the concurrent I/O advisory on a file:

```
fd=open(filename, oflag)
ioctl(fd, VX_SETCACHE, VX_CONCURRENT)
write(fd, buff, numofbytes)
```

Veritas Extension for Oracle Disk Manager

The Veritas Extension for Oracle Disk Manager (ODM) is specifically designed for Oracle 10gR2 or later to enhance file management and disk I/O throughput. The features of ODM are best suited for databases that reside in a Veritas File System (VxFS). ODM allows Oracle 10gR2 or later users to improve database throughput for I/O intensive workloads with special I/O optimization.

The Veritas extension for ODM is supported in non-global zones. To run Oracle 11g Release 2 on a non-global zone and use ODM, the Oracle software version must be 11.2.0.3.

Care must be taken when installing and removing packages when working with the VRTSodm package, for more information refer to the following:

- See “Package installation in non-global zones” on page 79.
- See “Package removal with non-global zone configurations ” on page 79.

Enabling ODM file access from non-global zones with VxFS

This section describes how to the enable ODM file access from non-global zones with VxFS:

- On Solaris 11:
  - If there is no existing zone.
    See “On Solaris 11: To enable ODM file access from non-global zones with VxFS, if there is no existing zone” on page 66.
  - If there is existing zone.
    See “On Solaris 11: To enable ODM file access from non-global zones with VxFS, if there is existing zone” on page 67.

On Solaris 11: To enable ODM file access from non-global zones with VxFS, if there is no existing zone

1. Install SF in the global zone.

   See the *Storage Foundation Configuration and Upgrade Guide*.

2. Set publisher for installation after going in the `pkgs` folder of the same installer:

   ```
   global# pkg set-publisher -P -g VRTSpkgs.p5p Veritas
   ```

3. Create a zone with the following configuration:

   ```
   zonecfg:myzone> create
   create: Using system default template 'SYSdefault'
   zonecfg:myzone> set zonepath=/export/home/myzone
   zonecfg:myzone> set fs-allowed=default,vxfs,odm
   zonecfg:myzone> add fs
   zonecfg:myzone:fs> set dir=/etc/vx/licenses/lic
   zonecfg:myzone:fs> set special=/etc/vx/licenses/lic
   zonecfg:myzone:fs> set type=lofs
   zonecfg:myzone:fs> end
   zonecfg:myzone:fs> remove anet linkname=net0
   zonecfg:myzone> add device
   zonecfg:myzone:device> set match=/dev/vxportal
   zonecfg:myzone:device> end
   zonecfg:myzone> add device
   zonecfg:myzone:device> set match=/dev/fdd
   zonecfg:myzone:device> end
   zonecfg:myzone> add device
   zonecfg:myzone:device> set match=/dev/odm
   zonecfg:myzone:device> end
   zonecfg:myzone> verify
   zonecfg:myzone> commit
   ```

4. Install the zone:

   ```
   global# zoneadm -z myzone install
   ```

5. Boot the zone.

   ```
   global# zoneadm -z myzone boot
   ```

6 Configure the zone:

```
global# zlogin -C myzone
```

7 Install VRTSvxfs, VRTSodm, and VRTSvlic in the zone:

```
myzone# pkg install --accept VRTSvxfs VRTSodm VRTSvlic
```

8 Enable the `vxodm` service inside the zone:

```
myzone# svcadm enable vxodm
```

9 Execute the `mount -p | grep odm` in the local zone and confirm the output looks similar to the following:

```
/dev/odm - /dev/odm odm - no nodevices,smartsync,zone=myzone,sharezone=5
```

10 Unset the publisher after coming inside the global zone:

```
global# pkg unset-publisher Veritas
```

On Solaris 11: To enable ODM file access from non-global zones with VxFS, if there is existing zone

1 Check whether, SF is installed in the global zone or not. If not, Install SF in the global zone.

See the *Storage Foundation Configuration and Upgrade Guide*.

2 2. Set publisher for installation after going in the `pkgs` folder of the same installer with which SF was installed in the global zone.

```
global# pkg set-publisher -P -g VRTSpkgs.p5p Veritas
```

3 Check whether the zone is in running or in installed state. If it is running then shut it down:

```
global# zoneadm -z myzone shutdown
```

4 Set `fs-allowed` to be `default,vxfs,odm`:

```
zonecfg -z myzone
zonecfg:myzone> set fs-allowed=default,vxfs,odm
zonecfg:myzone> verify
zonecfg:myzone> commit
```

5 Add license directory as `fs` in zone configuration file:

```
zonecfg -z myzone
zonecfg:myzone> add fs
zonecfg:myzone:fs> set dir=/etc/vx/licenses/lic
zonecfg:myzone:fs> set special=/etc/vx/licenses/lic
zonecfg:myzone:fs> set type=lofs
zonecfg:myzone:fs> end
zonecfg:myzone> verify
zonecfg:myzone> commit
```

6 Add three devices `vxportal`, `fdd`, and `odm` in the zone configuration file:

```
zonecfg -z myzone
zonecfg:myzone> add device
zonecfg:myzone:device> set match=/dev/vxportal
zonecfg:myzone:device> end
zonecfg:myzone> add device
zonecfg:myzone:device> set match=/dev/fdd
zonecfg:myzone:device> end
zonecfg:myzone> add device
zonecfg:myzone:device> set match=/dev/odm
zonecfg:myzone:device> end
zonecfg:myzone> verify
zonecfg:myzone> commit
```

7 Boot the zone:

```
global# zoneadm -z myzone boot
```

8 Install VRTSvxfs, VRTSodm, and VRTSvlic in the zone:

```
myzone# pkg install --accept VRTSvxfs VRTSodm VRTSvlic
```

9 Enable the `vxodm` service inside the zone:

```
myzone# svcadm enable vxodm
```

10 Execute the `mount -p | grep odm` in the local zone and confirm the output looking similar to the following:

```
/dev/odm - /dev/odm odm - no nodevices,smartsync,zone=myzone,sharezone=5
```

11 11. Unset the publisher after coming inside the global zone:

```
global# pkg unset-publisher Veritas
```

Exporting VxVM volumes to a non-global zone

A volume device node can be exported for use in non-global zone using the `zonecfg` command. The following procedure makes a volume `vol1` available in the non-global zone `myzone`.

Caution: Exporting raw volumes to non-global zones has implicit security risks. It is possible for the zone administrator to create malformed file systems that could later panic the system when a mount is attempted. Directly writing to raw volumes, exported to non-global zones, and using utilities such as `dd` can lead to data corruption in certain scenarios.

To export VxVM volumes to a non-global zone

1 Create a volume `vol1` in the global zone:

```
global# ls -l /dev/vx/rdsk/rootdg/vol1
crw-------   1 root     root     301, 102000 Jun  3
12:54 /dev/vx/rdsk/rootdg/vol1crw-------   1 root  sys   301, 10200
0 Jun  3 12:54 /devices/pseudo/vxio@0:rootdg,vol1,102000,raw
```

2 Add the volume device `vol1` to the non-global zone `myzone`:

```
global# zonecfg -z myzone
zonecfg:myzone> add device
zonecfg:myzone:device> set match=/dev/vx/rdsk/mydg/vol1
zonecfg:myzone:device> end
zonecfg:myzone> commit
```

3. Ensure that the devices will be seen in the non-global zone:

   ```
   global# zoneadm -z myzone halt
   global# zoneadm -z myzone boot
   ```

4. Verify that `/myzone/dev/vx` contains the raw volume node and that the non-global zone can perform I/O to the raw volume node.

   The exported device can now be used for performing I/O or for creating file systems.

VxVM devices in Oracle Solaris global zones

On the Oracle Solaris operating environment, there are two physical nodes corresponding to each volume node entry, `/devices` and `/dev`, respectively, with the same major and minor number. The physical nodes appear as follows:

```
/devices raw volume node : /devices/pseudo/vxio@0:
dgname,volname,minor_number,raw
/devices block volume node : /devices/pseudo/vxio@0:
dgname,volname,minor_number,blk
/dev raw volume node : /dev/vx/rdsk/dgname/volumename
/dev block volume node : /dev/vx/dsk/dgname/volumename
```

The following example provides sample values in `/devices`:

```
ls -l /devices/pseudo/vxio*vol1*
brw-------   1 root     sys      302, 66000 Mar 25
17:21 /devices/pseudo/vxio@0:mydg,vol1,66000,blk
crw-------   1 root     sys      302, 66000 Mar 25
17:21 /devices/pseudo/vxio@0:mydg,vol1,66000,raw
```

The following example provides sample values in `/dev`:

```
ls -l /dev/vx/*dsk/mydg/vol1
brw-------   1 root     root     302, 66000 Mar 25 17:21 /dev/vx/dsk/mydg/vol1
crw-------   1 root     root     302, 66000 Mar 25 17:21 /dev/vx/rdsk/mydg/vol1
```

Removing a VxVM volume from a non-global zone

The following procedure removes a VxVM volume from a non-global zone.

To remove a VxVM volume from a non-global zone

1. You must halt the local zone before modifying the configuration.
2. Remove the volume device `vol1` from the non-global zone `myzone`:

   ```
   global# zonecfg -z myzone
   zonecfg:myzone> remove device match=/dev/vx/rdsk/rootdg/vol1
   zonecfg:myzone> end
   zonecfg:myzone> commit
   ```

About SF Oracle RAC support for Oracle RAC in a zone environment

This release supports the installation and configuration of two non-global zones in each global zone. The SF Oracle RAC cluster must comprise non-global zones from different global zones.

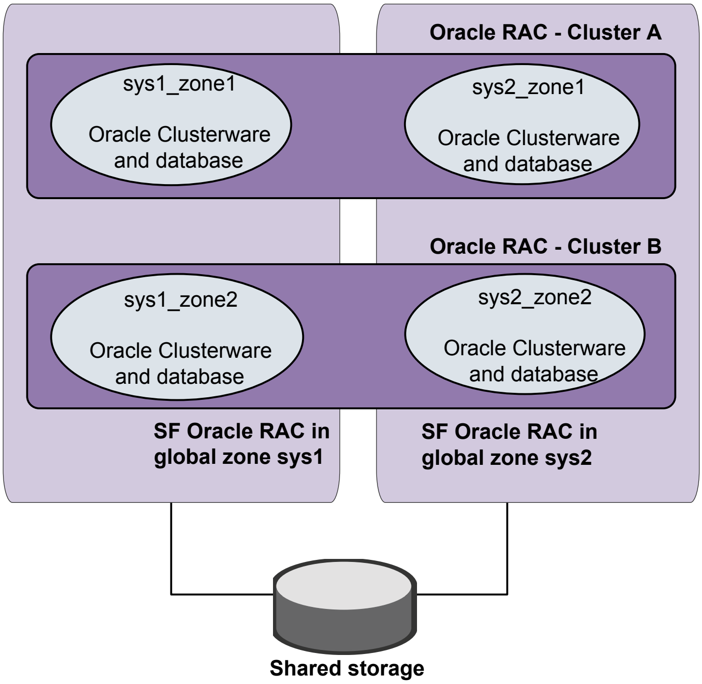
Note: SF Oracle RAC does not support a cluster formed of non-global zones from the same global zone.

SF Oracle RAC and the necessary agents run in the global zone. Oracle RAC runs in the non-global zone. You must configure non-global zones with an exclusive-IP zone. The exclusive-IP zone does not share the network interface with global-zone.

Using SF Oracle RAC, you can start, stop, and monitor a non-global zone and provide high availability to Oracle RAC instances inside the non-global zone.

Figure 2-6 illustrates the SF Oracle RAC configuration in a zone environment.

Figure 2-6 SF Oracle RAC with Oracle RAC in a zone environment


Oracle RAC - Cluster A
sys1_zone1
Oracle Clusterware and database
sys2_zone1
Oracle Clusterware and database
Oracle RAC - Cluster B
sys1_zone2
Oracle Clusterware and database
sys2_zone2
Oracle Clusterware and database
SF Oracle RAC in global zone sys1
SF Oracle RAC in global zone sys2
Shared storage

Supported configuration

The configuration supported by SF Oracle RAC for a zone environment is as follows:

Architecture	Solaris SPARC systems
Oracle RAC version	11.2.0.3
Operating system version	Refer to *Veritas InfoScale Release Notes* for the supported OS versions.
Zone IP address type	Exclusive IP zone **Note:** For exclusive IP zone, you need a minimum of three network interfaces for each non-global zone, one as public link and two as private links.

Note: All private interfaces inside a non-global zone must be configured under LLT as private interfaces. If you plan to have only one non-global zone cluster across global zones, it is recommended that the private interfaces configured for a non-global zone be exactly the same in name and total number as those which have been used for LLT configuration in the global zone. However, if you configure a subset of LLT interfaces as private interfaces in non-global zones, Oracle Clusterware will take cluster reconfiguration decisions in the event of network partition.

Known issues with supporting SF Oracle RAC in a zone environment

This section describes the known issues in supporting SF Oracle RAC in a zone environment.

CFS mount agent does not support mounting VxVM devices inside non-global zones

The CFS mount agent does not support mounting VxVM devices inside non-global zones because CVM binaries (`/etc/vx/bin/vxclustadm`) will not work or be present inside non-global zone.

Workaround: Perform a direct mount of storage under the non-global zones root directory (`/zoneroot/root/mnt_point`). This makes it available to non-global zones.

Issue with VCS agents

If the host name of the non-global zone is different from the name of the non-global zone, you may observe unexpected behavior with the VCS agents configured for the non-global zone.

Workaround: Ensure that the host name of the non-global zone is the same as the name of the non-global zone.

Stopping non-global zones configured with direct-mount file systems from outside VCS causes the corresponding zone resource to fault or go offline

Stopping non-global zones, which are configured with direct-mount file systems, from outside VCS causes the corresponding zone resource to fault or to go offline. The status of the zone shows `down` and the corresponding zone resource faults or goes offline. As a result, VCS cannot bring the zone resource online.

Workaround:

1. Log into the global zone as the root user.
2. Unmount the mount points that were in use by the zone and are still mounted:

   ```
   # umount -o mntunlock=VCS /mount_point
   ```

3. Stop the zone:

   ```
   # zoneadm -z zone_name halt
   ```

   This changes the status of the non-global zone to `installed` or `configured`.

Error message displayed for PrivNIC resource if zone is not running

If the PrivNIC resource is configured for a zone in a non-global zone environment and the respective zone is not running, the following error message is displayed in the VCS engine log file `/var/VRTSvcs/log/engine_*.log`:

```
VCS ERROR V-16-20035-0 (sys1)
PrivNIC:ora_priv:monitor:Zone [zone1] not running.
```

Warning messages displayed when VCS restarts

When you restart VCS, the following warning message is displayed before the multi-user services inside a zone are started:

```
VCS WARNING V-16-10001-14056 (sys1)
Zone:vcszoneres:monitor:Zone is running without specified
milestone [multi-user-server] online - returning offline.
```

You may safely ignore the message.

The installer log of non-global zone contains warning messages

The installer log of non-global zone contains warning messages related to the VRTS packages.

Workaround:

Before installing the new non-global zone, you must set the parent directory of zonepath to be 755. The parent directory of the zone can be derived from the complete zonepath by running the `dirname` command on the complete zonepath including the zonename.

Issue with CFS mounts

Attempts to mount the CFS mounts on a global zone, after they are unmounted using the `hares` or `umount` command, fail with the following error if the CFS mounts on the global zone are mounted on a non-global zone as an lofs file system:

```
VCS WARNING V-16-20011-5508 (sys1)
CFSMount:ocrvote_mnt:online:Mount Error :
UX:vxfs mount: ERROR: V-3-21264:
/dev/vx/dsk/ocrvotedg/ocrvotevol is already mounted,
/ocrvote is busy,allowable number
of mount points exceeded
```

Workaround:

Perform the following steps to resolve the issue:

1. Log into the global zone as the root user.
2. View the CFS and lofs mounts that are unmounted on the global zone:

   ```
   # cat /etc/mnttab |grep mount_point
   ```

   For example:

   ```
   # cat /etc/mnttab |grep ocrvote/ocrvote \
   /zonevol/sys1_zone1/root/ocrvote lofs \
   dev=53859d8 12971587943
   ```

3. Unmount the CFS and lofs mounts:

   ```
   # umount /zonevol/sys1_zone1/root/mount_point
   ```

4. Check if there are any active CFS and lofs mounts:

   ```
   # cat /etc/mnttab |grep mount_point
   ```

5. Mount the CFS and lofs mounts in one of the following ways on the global zone.

   Using hares command:

   ```
   # hares -online res_name -sys sys_name
   ```

   Manually:

   ```
   # mount -F vxfs -o cluster /dev/vx/dsk/\
   dg_name/vol_name /mount_point
   ```

6. Verify if the CFS mounts are mounted successfully:

```
# cat /etc/mnttab |grep mount_point
```

For example:

```
# cat /etc/mnttab |grep ocrvote/dev/vx/dsk/ocrvotedg/ocrvotevol \
/ocrvote vxfsrw,suid,delaylog,largefiles,qio,cluster,\
ioerror=mdisable,crw,dev=53859d8 1297159501
```

Configuring Solaris non-global zones for disaster recovery

Solaris Zones can be configured for disaster recovery by replicating the zone root using replication methods like Hitachi TrueCopy, EMC SRDF, Veritas Volume Replicator, and so on. The network configuration for the Zone in the primary site may not be effective in the secondary site if the two sites are in different IP subnets. Hence, you need to make these additional configuration changes to the Zone resource.

To configure the non-global zone for disaster recovery, configure VCS on both the sites in the global zones with GCO option.

Refer to the *Cluster Server Administrator’s Guide* for more information about global clusters, their configuration, and their use.

To set up the non-global zone for disaster recovery

1. On the primary site, create the non-global Zone and configure its network parameters.
   - Create the non-global zone on the primary site using the `zonecfg` command.
   - Add the network adapters to the non-global zone’s configuration if the zone is configured as an exclusive IP zone. Assign an IP address to the network adapter along with the Netmask and Gateway.
   - After the zone boots, set up the other network-related information such as the HostName, DNS Servers, DNS Domain, and DNS Search Path in the appropriate files (/etc/hostname, /etc/resolve.conf).
2. On the primary site, shut down the zone.
3. Use replication-specific commands to failover the replication to the secondary site from primary site.
4. Repeat step 1 on the secondary site.

5 Perform step 6, step 7, step 8, and step 9 on the primary cluster and secondary clusters.

6 Create a VCS service group with a VCS Zone resource for the non-global zone.

Configure the DROpts association attribute on the Zone resource with the following keys and site-specific values for each: HostName, DNSServers, DNSSearchPath, and DNSDomain. If the non-global zone is an exclusive IP zone in this site, configure the following keys in the DROpts association attribute on the Zone resource: Device (network adapter name), IPAddress, Netmask, and Gateway.

7 Add the appropriate Mount resources and DiskGroup resources for the File System and DiskGroup on which the non-global zone's zoneroot resides. Add a resource dependency from the Zone resource to the Mount resource and another dependency from the Mount resource to the Diskgroup resource.

8 Add one of the following VCS replication resources to the service group for managing the replication.

- A hardware replication agent
  Examples of these agents include SRDF for EMC SRDF, HTC for Hitachi TrueCopy, MirrorView for EMC MirrorView, etc. Refer to the appropriate VCS replication agent guide for configuring the replication resource.

- The VVRPrimary agent

For VVR-based replication, add the RVGPrimary resource to the service group.

Refer to the following manuals for more information:

- For information about configuring VVR-related resources, see the *Veritas InfoScale™ Replication Administrator's Guide*.

- For information about the VVR-related agents, see the *Cluster Server Bundled Agents Reference Guide*.

9 Add a dependency from the DiskGroup resource to the replication resource.

Figure 2-7 Sample resource dependency diagram for hardware replication based non-global zones

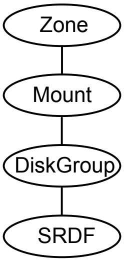
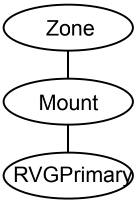


Figure 2-8 Sample resource dependency diagram for VVR replication-based non-global zones



When the resource is online in a site, the replication resource makes sure of the following:

- The underlying replicated devices are in primary mode, and the underlying storage and eventually the zone root goes into read-write mode.
- The remote devices are in secondary mode.

Thus, when the Zone resource goes online the resource modifies the appropriate files inside the non-global zone to apply the disaster recovery-related parameters to the non-global zone.

Software limitations of Storage Foundation support of non-global zones

This section describes the software limitations of Storage Foundation support of non-global zones in this release.

Administration commands are not supported in non-global zone

All administrative tasks, such as resizing a volume, adding a volume to a volume set, and file system reorganization, are supported only in the global zone. Consequently, administrative commands, such as `fsadm`, `fsvoladm`, and administrative ioctls are not supported in the non-global zone by both VxFS and VxVM.

VxFS file system is not supported as the root of a non-global zone

For Solaris 11, the root of a non-global zone cannot currently be on a VxFS file system.

QIO and CQIO are not supported

Quick I/O and Cached Quick I/O are not supported by VxFS in non-global zones.

Package installation in non-global zones

On Solaris 11, the packages are not propagated to the non-global zones automatically whenever the package is installed in the global zone. Refer to Install Guide of the product for instructions to install packages inside the non-global zone.

Package removal with non-global zone configurations

If non-global zones are part of the system configuration and the VRTSodm package is installed, ensure that `/dev/odm` is unmounted in each non-global zone prior to VRTSodm package removal or product uninstallation. This ensures there are no non-global zone odm module references that might prevent the global zone from unloading the odm module.

You can unmount `/dev/odm` in the non-global zone with the following commands:

```
global# zlogin myzone
myzone# svcadm disable vxodm
```

The `/etc/vx/licenses/lic` file which is lofs mounted inside the local zones needs to be removed from the zone configuration prior to uninstallation of VRTSvxfs and VRTSvlic packages in the non-global zone. This is specific to Solaris 11.

Remove the zone configuration

1 If running Solaris 11, enter the following commands:

```
zoneadm -z myzone shutdown
zonecfg -z myzone
zonecfg:myzone> remove fs
dir=/etc/vx/licenses/lic
zonecfg:myzone> verify
zonecfg:myzone> commit
zoneadm -z myzone boot
```

2 Uninstall the VRTSvxfs and VRTSvlic packages.

During package removal from global zone for VRTSvxfs. Remove all lofs mounted vxfs mount path and vxfs mounted mount path (automatic mounting inside the non-global zone by setting `type=vxfs` in the zone configuration file) as it interferes with the uninstallation of VRTSvxfs package in the global zone. These removal should be done after halting the zone and by modifying the zone configuration files.

Remove all lofs mounted vxfs mount path and vxfs mounted mount path

◆ Enter the following commands:

```
zoneadm -z myzone halt
zonecfg -z myzone
zonecfg:myzone> remove fs dir=/dirmnt
zonecfg:myzone> verify
zonecfg:myzone> commit
zonecfg:myzone> end
zoneadm -z myzone boot
```

Root volume cannot be added to non-global zones

The root volume cannot be added to non-global zones.

Some Veritas Volume Manager operations can cause volume device names to go out of sync

If a volume is exported to a non-global zone, some Veritas Volume Manager operations can cause the global and non-global volume device names to go out of sync, which can create data corruption. This is because the Solaris operating environment zone support is not aware of the `devfsadm`(1M) command, and thus the zone configuration is not updated with any changes to the `/dev` or `/devices` namespaces.

The following operations can cause device names to go out of sync:

- Removing a volume
- Importing a disk group
- Deporting a disk group
- Renaming a disk group or volume
- Reminoring a disk group
- Restarting `vxconfigd` or resetting the kernel

To prevent device names from to going out of sync, if a volume is exported to a non-global zone and an operation that can cause device names to go out of sync occurs on that volume, remove the volume from the zone configuration using the `zonecfg` command and reboot the zone using the `zoneadm` command.

See the `zonecfg`(1M) and `zoneadm`(1M) manual pages.

Note: This issue applies to any Solaris device for which the `/dev` or `/devices` device node is changed and has been configured in the non-global zone before the change.

Chapter 3

Storage Foundation and High Availability Solutions support for Solaris Projects

This chapter includes the following topics:

About Solaris Projects

The Solaris operating system provides the projects facility to identify workloads. The project serves as an administrative tag that you use to group related work in a useful manner. You can create one project for a sales application and another project for a marketing application. By placing all processes related to the sales application in the sales project and the processes for the marketing application in the marketing project, you can separate and control the workloads in a way that makes sense to the business.

A user that is a member of more than one project can run processes in multiple projects at the same time. This multiple project approach makes it possible for users to participate in several workloads simultaneously. All processes that a process starts inherits the project of the parent process. As a result, switching to a new project in a startup script runs all child processes in the new project.

For more information, refer to the Solaris operating environment document *System Administration Guide: Solaris Containers--Resource Management and Solaris Zones* .

About VCS support for Solaris projects

VCS provides application management and high availability to applications that run in Solaris projects.

Overview of how VCS works with Solaris projects

You can use VCS to perform the following:

- Start, stop, monitor, and fail over a Solaris project.
- Start, stop, monitor, and fail over an application that runs inside a Solaris project.

How VCS models containers

VCS and necessary agents run in the global zone. For the applications that run in a Solaris project, the agents can run online entry point inside the project. If any resource faults, VCS fails over the service group.

Installing and configuring projects in a VCS environment

Install and configure the project. Create the service group with the standard application resource types (application, storage, networking) and the Project resource. VCS manages the project as a resource. You then configure the service group's ContainerInfo attribute.

Configuring the ContainerInfo attribute

The service group attribute ContainerInfo specifies information about the Solaris project. When you have configured and enabled the ContainerInfo attribute, you have enabled the project-aware resources in that service group to work in the project environment. VCS defines the project information at the level of the service group so that you do not have to define it for each resource. You need to specify a per-system value for the ContainerInfo attribute.

About the ContainerInfo service group attribute

The ContainerInfo attribute has the Name key, Type key, and Enabled key. The Name key defines the name of the container. The Type key lets you select the type of container that you plan to use. The Enabled key enables the Project-aware resources within the service group. The ContainerInfo attribute specifies if you can use the service group with the container.

Assign the following values to the ContainerInfo attribute:

- Name
  The name of the container.
- Type
  The type of container. You can set this to Project.
- Enabled
  Specify the value as 0, if you want to disable the container. Specify the value as 1, if you want to enable the container. Specify the value as 2, to enable physical to virtual and virtual to physical failovers. When the value is 2, the Project resource mimics a non-existent entity.

You can set a per-system value for this attribute.

About the ContainerOpts resource type attribute

The ContainerOpts resource attribute is pre-set for project-aware resource types. It determines the following:

- Whether the project-aware resource can run in the project.
- Whether the container information that is defined in the service group's ContainerInfo attribute is passed to the resource.

These values are only effective when you configure the ContainerInfo service group attribute.

attribute's keys follow:

The ContainerOpts resource type attribute's definitions for project-aware types contain the following values:

- RunInContainer
  When the value of the RunInContianer key is 1, only online agent function (entry point) for that resource runs inside of the project.
  When the value of the RunInContainer key is 0, the agent function (entry point) for that resource runs outside the local container (in the global environment).
  A limitation for the RunInContainer value is that only script agent functions (entry points) can run inside a container.
- PassCInfo
  When the value of the PassCInfo key is 1, the agent function receives the container information that is defined in the service group's ContainerInfo attribute. An example use of this value is to pass the name of the container to the agent.

Project-aware resources

At present Process, Application and Oracle resources are project aware. If a service group, which is configured for Solaris Project contains resources other than Process, Application, or Oracle, Veritas recommends you set RunInContainer to 0.

About the Project agent

The Project agent monitors Solaris Project, brings them online, and takes them offline.

For more information about the agent, see the *Cluster Server Bundled Agents Reference Guide*.

Configuring VCS in Solaris projects

Configuring VCS in projects involves the following tasks:

First	Review the prerequisites. See "Prerequisites for configuring VCS in projects" on page 85.
Second	Decide on the location of the project root, which is either on local storage or shared storage.
Third	Install the application in the project.
Fourth	Create the application service group and configure its resources.

Prerequisites for configuring VCS in projects

Review the following prerequisites for configuring Cluster Server (VCS) in projects: VCS support only Process, Application and Oracle agent.

Using custom agents in projects

If you use custom agents, review the following information for their use in projects:

- If you use custom agents to monitor the applications that run in project, make sure that the agents use script-based entry points. VCS does not support running C++ entry points inside a project.
- If you want the custom agent to monitor an application in the project, for the custom agent type, set the following values for the ContainerOpts attribute: RunInContainer = 1 and the PassCInfo = 0.

- If you don't want the custom agent to monitor an application in the project, for the custom agent type, set the following values for the ContainerOpts attribute: RunInContainer = 0 and the PassCInfo= 0.

For an example, refer to the *Cluster Server Bundled Agents Reference Guide*.

Section 3

Oracle VM Server for SPARC

Chapter 4

Storage Foundation and High Availability Solutions support for Oracle VM Server for SPARC

This chapter includes the following topics:

About Oracle VM Server for SPARC

Oracle VM Server for SPARC is a virtualization technology that enables the creation of independent virtual machine environments on the same physical system. Oracle VM Server for SPARC provides a virtualized computing environment abstracted from all physical devices, which allows you to consolidate and centrally manage your workloads on a system. The logical domains can be specified roles such as a control domain, service domain, I/O domain , and guest domain. Each domain is a full virtual machine where the operating systems can be started, stopped, and rebooted independently.

Note: VCS treats LDOM analogous to a Physical machine.

Terminology for Oracle VM Server for SPARC

The following terminology is used in configuring the Storage Foundation software in Oracle VM Server for SPARC.

Table 4-1 Lists the terminology for Oracle VM Server for SPARC

Term	Definition
Logical domain	Logical domain or virtual machine with its own operating system, resources, and identity within the same physical host.

Table 4-1 Lists the terminology for Oracle VM Server for SPARC *(continued)*

Term	Definition
Hypervisor	A firmware layer that provides a set of hardware-specific support functions to the operating systems running inside logical domains through a stable interface, known as the sun4v architecture. The hypervisor is interposed between the operating system and the hardware layer.
Logical Domains Manager	Software that communicates with the Hypervisor and logical domains to sequence changes, such as the addition and removal of resources or creation of a logical domain. The Logical Domains Manager provides an administrative interface and keeps track of the mapping between the physical and virtual devices in a system.
Control domain	The primary domain which provides a configuration platform to the system for the setup and teardown of logical domains. Executes Logical Domains Manager software to govern logical domain creation and assignment of physical resources.
I/O domain	Controls direct, physical access to input/output devices, such as PCI Express cards, storage units, and network devices. The default I/O domain is the control domain.
Guest domain	Utilizes virtual devices offered by control and I/O domains and operates under the management of the control domain.
Virtual devices	Physical system hardware, including CPU, memory, and I/O devices that are abstracted by the Hypervisor and presented to logical domains within the platform.
Logical Domains Channel (LDC)	A logical domain channel is a point-to-point, full-duplex link created by the Hypervisor. LDCs provide a data path between virtual devices and guest domains and establish virtual networks between logical domains.

Table 4-1 Lists the terminology for Oracle VM Server for SPARC *(continued)*

Term	Definition
Virtual Disk Client	A Solaris kernel module in the guest domain which controls the virtual disks visible to that guest, providing standard device interfaces to applications.
Virtual Disk Server	A Solaris kernel module in the control domain which is responsible for exporting various backend devices as virtual disks to guest domains.

Oracle VM Server for SPARC deployment models

Oracle VM Server for SPARC is a virtualization technology on the Solaris SPARC platform that enables the creation of independent virtual machine environments on the same physical system. This lets you consolidate and centrally manage your workloads on one system.

Storage Foundation supports logical domains in the following three deployments models:

- Split Storage Foundation stack
- Guest-based Storage Foundation stack
  Storage Foundation Cluster File System High Availability (SFCFSHA) is supported only in the guest-based Storage Foundation stack.
- Layered Storage Foundation stack

See “About Cluster Server configuration models in an Oracle VM Server for SPARC environment” on page 152.

Split Storage Foundation stack

The support for this model was introduced in 5.0 MP1 release and this model continues to be supported in this release.

See “Split Storage Foundation stack model” on page 97.

See “Cluster Server setup to fail over a logical domain on a failure of logical domain” on page 153.

Guest-based Storage Foundation stack

The support for this model was introduced in 5.0 MP3 release.

See “Guest-based Storage Foundation stack model” on page 100.

See “Features” on page 93.

See “Known issues” on page 138.

Note: The SFCFSHA stack can be installed across multiple I/O domains within or across physical servers.

See “Cluster Server limitations” on page 151.

Layered Storage Foundation stack

Only I/O fencing support is added with this configuration.

See “Layered Storage Foundation stack model” on page 107.

Benefits of deploying Storage Foundation High Availability solutions in Oracle VM server for SPARC

There are several benefits to a Oracle VM Server for SPARC environment.

Standardization of tools

Independent of how an operating system is hosted, consistent storage management tools save an administrator time and reduce the complexity of the environment.

Storage Foundation in the control domain provides the same command set, storage namespace, and environment as in a non-virtual environment.

Array migration

Data migration for Storage Foundation can be executed in a central location, migrating all storage from an array utilized by Storage Foundation managed hosts.

This powerful, centralized data migration functionality is available with Storage Foundation Manager 1.1 and later.

Moving storage between physical and virtual environments

Storage Foundation can make painful migrations of data from physical to virtual environments easier and safer to execute.

With Storage Foundation, there is no need to copy any data from source to destination, but rather the administrator reassigns the same storage or a copy of the storage for a test migration, to the virtual environment.

Boot Image Management

Using Storage Foundation in this environment the user can utilize features such as instant snapshots to contain boot images and manage them from a central location in the control domain.

Features

This section describes some of the features in Oracle VM Server for SPARC using the products in the Storage Foundation and High Availability Solutions.

Storage Foundation features

The following features apply for Storage Foundation.

The vxloadm utility enables access to a file system contained in a VxVM volume from the Control Domain

The `vxloadm` utility lets you access a file system contained inside a VxVM volume from outside the guest domain, that is from the Control Domain. This is done by mapping all the partitions contained within that volume using the `vxlo` driver. The partitions can then be mounted if they contain valid file systems.

To use this vxloadm utility

1 Check if the driver is loaded in memory:

```
# modinfo| grep vxlo
226 7b3ec000   3870 306   1  vxlo (Veritas Loopback Driver 0.1)
```

2 Run the `vxloadm` utility:

```
# /etc/vx/bin/vxloadm
```

3. You can now use the utility.

   See "Examples of using the vxloadm utility" on page 94.

4. Veritas recommends once you are done using the `vxloadm` utility to unload the `vxlo` driver:

   ```
   # rem_drv vxlo
   # modinfo| grep vxlo
   226 7b3ec000   3870 306   1  vxlo (Veritas Loopback Driver 0.1)
   # modunload -i 226
   ```

   where *226* is the module ID from the `modinfo | grep vxlo` command.

Examples of using the vxloadm utility

Use the `vxloadm addall` command to create device(s) mapping the various partition(s) contained in a VxVM volume. For example:

```
# /etc/vx/bin/vxloadm addall vol1 /dev/vx/dsk/testdg/vol1
```

This creates a device node entry for every slice or partition contained within the volume in the `/dev/vxlo/dsk/` and `/dev/vxlo/rdsk/` directories.

```
# ls -l /dev/vxlo/dsk/
lrwxrwxrwx 1 root root 46 Sep 25 14:04 vol1s0
-> ../../../devices/pseudo/vxlo@0:vol1s0,1,blk
lrwxrwxrwx 1 root root 46 Sep 25 14:04 vol1s3
-> ../../../devices/pseudo/vxlo@0:vol1s3,2,blk
```

```
# ls -l /dev/vxlo/rdsk/
lrwxrwxrwx 1 root root 46 Sep 25 14:04 vol1s0
-> ../../../devices/pseudo/vxlo@0:vol1s0,1,raw
lrwxrwxrwx 1 root root 46 Sep 25 14:04 vol1s3
-> ../../../devices/pseudo/vxlo@0:vol1s3,2,raw
```

Use the `vxloadm get` command to display the list of all currently mapped partition(s) created using the `vxloadm` utility. For example:

```
# /etc/vx/bin/vxloadm get
VxVM  INFO V-5-1-0      NAME       FILENAME
MOUNT   OFFSET    C/H/S
VxVM  INFO V-5-1-15260  vol1s0  /dev/vx/dsk/testdg/vol1
6180       6787/1/618
VxVM  INFO V-5-1-15260  vol1s3  /dev/vx/dsk/testdg/vol1
4326000 50902/1/618
```

Use the appropriate file system commands to access the file system(s). For example:

```
# fstyp /dev/vxlo/rdsk/vol1s0
ufs
# mount -F ufs /dev/vxlo/dsk/vol1s0 /mnt
```

Use the `vxloadm delete` to remove the partition mappings of a volume. For example:

```
# /etc/vx/bin/vxloadm delete vol1s0
# /etc/vx/bin/vxloadm delete vol1s3
```

Note: This `vxloadm` utility should only be used on volumes that are currently not in use or held open by a guest domain.

The vxformat utility automatically relabels the virtual disk backed by a VxVM volume in the guest domain

The `vxformat` utility provides the user the ability to automatically relabel a virtual disk backed by a VxVM volume. This utility is meant to be executed from inside the guest domain only.

The `vxformat` utility is particularly useful when a VxVM volume with existing partitions is grown in size and you need to access the new size from the guest domain.

Requirements for relabeling to succeed

- The relabel succeeds only- if it can find a new cylinder size that is aligned with the start and size of each of the existing partitions.
  In case the `vxformat` command cannot find such cylinder size, it displays the following descriptive message and then exits:

  ```
  Cannot re-label device /dev/rdsk/c0t1d2s2 since we failed to
  find new cylinder size that's aligned with all the existing partitions
  ```

- The relabel succeeds only - if the available blocks is greater than the last sector of each and every non-s2 partition.
  Otherwise, the`vxformat` command displays the following message and then exits:

  ```
  Cannot re-label device /dev/rdsk/c0d2s2 since the last sector of a
  non-s2 partition is greater than the available blocks
  ```

Example of using the vxformat utility

Use the `vxformat` command to relabel the virtual disk. For example:

```
# /etc/vx/bin/vxformat c0d1s2
rawpath: /dev/rdsk/c0d1s2
Old disk capacity: 2097000 blocks
New disk capacity: 4194000 blocks
Device /dev/rdsk/c0d1s2 has been successfully re-labeled.
Please use prtvtoc(1) to obtain the latest partition table information
```

If the underlying device size has not changed, the `vxformat` command displays the following message without changing the label. For example:

```
# /etc/vx/bin/vxformat c0d1s2
Old disk capacity: 2343678 blocks
New disk capacity: 2343678 blocks
size of device /dev/rdsk/c0d2s2 is unchanged
```

Oracle VM Server for SPARC features

SFHA solutions support the following features for Oracle VM server for SPARC.

Guest domain migration

The guest domain migration feature is supported for cold, warm, and live migrations by Storage Foundation with all the below deployment models:

- Split Storage Foundation stack
- Guest-based Storage Foundation stack
- Layered Storage Foundation stack

Virtual I/O dynamic reconfiguration

The virtual I/O dynamic reconfiguration feature is supported with all the below deployment models:

- Split Storage Foundation stack
- Guest-based Storage Foundation stack
- Layered Storage Foundation stack

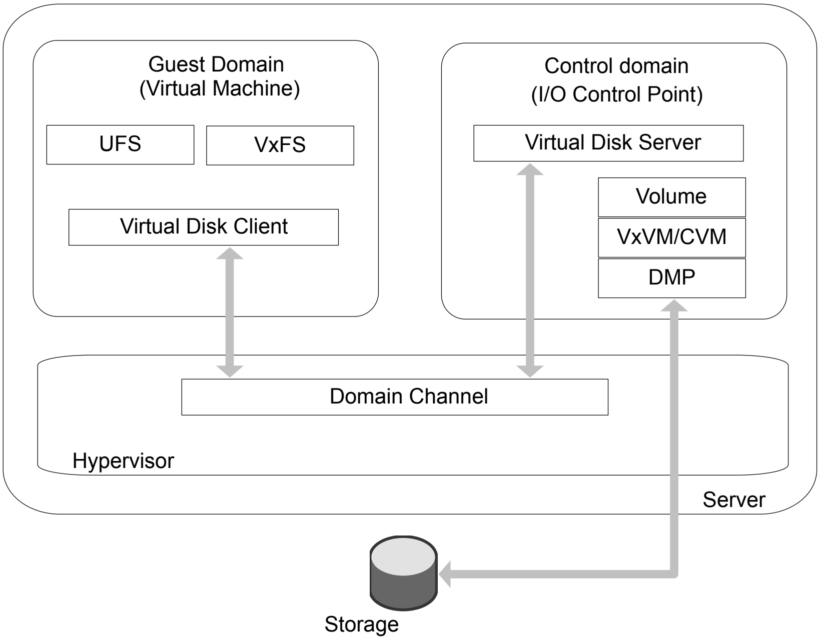
Note: For resizing a volume exported as a single slice: The new size should be visible dynamically in the guest immediately.

For resizing a volume exported as a full disk: Even though the new size is visible dynamically in the guest, the new space allocated in the volume cannot be utilized unless the label in the vdisk has been adjusted to reflect the new sectors. This adjustment of the label needs to be done carefully.

Split Storage Foundation stack model

Figure 4-1 illustrates the split Storage Foundation stack model with Oracle VM Server for SPARC logical domains.

Figure 4-1 Split Storage Foundation stack model with logical domains



How Storage Foundation and High Availability Solutions works in the Oracle VM Server for SPARC

Storage Foundation and High Availability Solutions supports Oracle VM Server for SPARC logical domains in both single-node, multiple-node, and multiple-node high availability configurations.

Figure 4-1 illustrates the recommended placement of Storage Foundation stack component products in this model.

Following indicates the recommended placement of Storage Foundation stack component products:

- For a single node configuration, Veritas Volume Manager (VxVM) including DMP is placed in the control domain, and Veritas File System (VxFS) is placed in the guest domain.
- For clustered nodes, Cluster Volume Manager (CVM) is placed in the control domain, and VxFS is placed in the guest domain.
  See "Clustering using Cluster Volume Manager" on page 140.
  See "Installing Storage Foundation on multiple nodes in a Logical Domain" on page 141.
  See "Cluster Volume Manager in the control domain for providing high availability" on page 143.
- For clustered nodes in a highly available environment, install Cluster Server (VCS) in the control domain.
  See "About VCS in a Oracle VM Server for SPARC environment" on page 148.
  See "About Cluster Server configuration models in an Oracle VM Server for SPARC environment" on page 152.
  See "Configuring VCS to fail over an application on a failure" on page 159.
- VxFS drivers in the guest domain cannot currently interact with the VxVM drivers in the control domain. This renders some features, which require direct VxVM-VxFS coordination, unusable in such a configuration.
  See " Storage Foundation features restrictions" on page 98.

Note: VxFS can also be placed in the control domain, but there will be no coordination between the two VxFS instances in the guest and the control domain.

Storage Foundation features restrictions

The following Storage Foundation software features are restricted in the split Storage Foundation stack model:

- Smartmove and Thin Reclamation — These features require co-ordination between VxVM and VxFS and hence are not supported in this model.
- VxVM volume snapshots — Due to the inability of VxFS in the guest domain to coordinate with VxVM in the control domain, taking a data consistent snapshot

of a VxVM volume containing a VxFS file system requires shutting down the application and unmounting the file system before taking the snapshot.

- Resizing VxVM volumes and any type of file system on top of the volume with `vxresize` — Resizing any type of file system on the guest whose underlying device is backed by a VxVM volume in the control domain, requires resizing the VxVM volume and the file system in the guest individually.

  If you are growing a VxFS file system in the guest whose underlying device is backed by a VxVM volume requires you to first grow the volume in the control domain using the `vxassist` command, and then the file system in the guest domain using the `fsadm` command.

  Shrinking a VxFS file system, on the other hand, requires you to first shrink the file system in the guest domain using the `fsadm` command, and then the volume in the control domain using the `vxassist` command. Using the `vxassist` command requires you to use the `-f` option of the command, as in the following example.

  ```
  # vxassist -g [diskgroup] -f shrinkto volume length
  ```

  Caution: Do not shrink the underlying volume beyond the size of the VxFS file system in the guest as this can lead to data loss.

- Exporting a volume set to a guest domain is not supported.
- Veritas Volume Replicator is not supported in the Split Storage Foundation stack model.
- Multi-volume DST
- File-level Smartsync
- The following VxFS tunables are not set to their default values based on the underlying volume layout, due to VxFS being in the guest domain and VxVM being installed in the control domain:
  - read_pref_io
  - write_pref_io
  - read_nstream
  - write_nstream

  If desired, you can set the values of these tunables based on the underlying volume layout in the `/etc/vx/tunefstab` file.

  See the *Storage Foundation Administrator's Guide* for more information about tuning I/O.

- Storage Foundation Cluster File System High Availability is not supported in this deployment model.

Guest-based Storage Foundation stack model

Figure 4-2 Guest-based Storage Foundation stack model

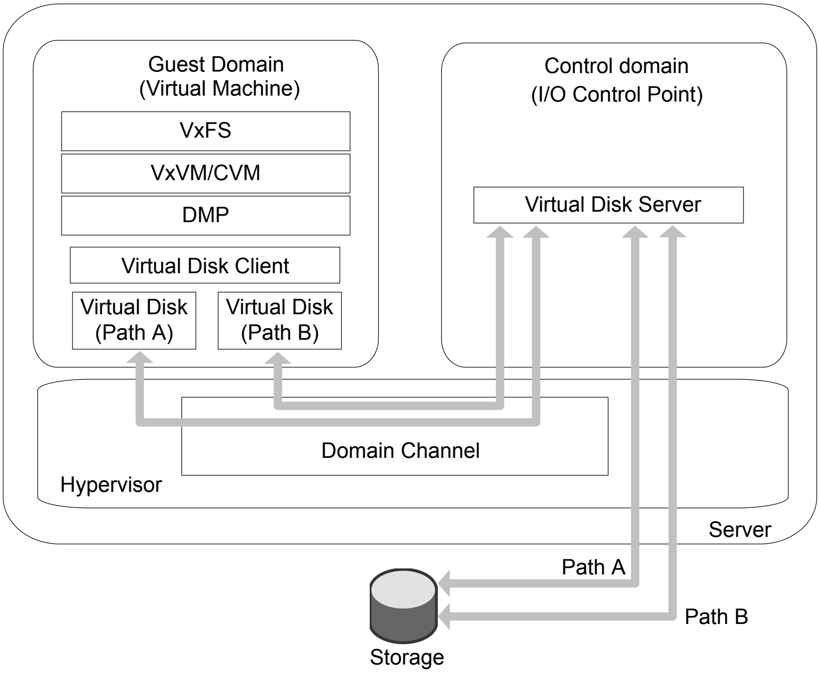


Figure 4-2 illustrates the guest-based Storage Foundation stack model with guest logical domains.

How Storage Foundation and High Availability Solutions works in the guest domains

The entire Storage Foundation stack is co-located within the guest in this deployment model.

Veritas recommends that you export all paths to a disk which is being exported to a guest and let DMP do the multi-pathing of the disk in the guest domain.

Note: It is also possible to have the VxVM stack be fully functional (guest-based model limitations still apply) in the guest with DMP only seeing a single path to each disk and DMP in the Control or I/O domains managing the multiple paths to these disks. This implies all features including Thin Reclamation.

See "Enabling DMP in the control and alternate I/O domains" on page 108.

Note: Only full SCSI disks can be used under Veritas Volume Manager (VxVM) and DMP in this model. Non-SCSI devices (volume, file, slice, etc) are not supported.

Storage Foundation and High Availability Solutions and Storage Foundation Cluster File System High Availability supports running in the guest domains in both single-node, multiple-node, and multiple-node high availability configurations.

- For a single node configuration, VxVM (including DMP) and VxFS are co-located in the guest domain.
- For clustered nodes, CVM can also be used inside the guest domain. As with regular physical nodes, forming a CVM cluster of logical domain guests requires shared storage visibility across the guests that are part of the cluster.

  See the *Storage Foundation Cluster File System High Availability Administrator's Guide* for CVM information.

  See the *Storage Foundation Configuration and Upgrade Guide* for installation and configuration information.
- For clustered nodes in a highly available environment, install Cluster Server (VCS) in the guest domains.

  See the *Cluster Server* documentation for more information.
- Veritas Volume Replicator (VVR) is supported in the guest-based Storage Foundation stack model in the following configurations:
  - A guest domain on one host acting as the VVR primary, and another guest on another host acting as the VVR secondary.
  - Two guest domains on the same physical host, but you must export separate LUNs or disks to the data volumes and Storage Replicator Logs of the two guest domains.

In this model, the boot disk of the guest can be a VxVM volume. For more details on this support:

See "Provisioning Veritas Volume Manager volumes as boot disks for guest domains" on page 120.

About SFCFSHA in an Oracle VM Server for SPARC environment

Storage Foundation Cluster File System (SFCFSHA) allows clustered servers to mount and use a file system simultaneously as if all applications using the file system were running on the same server for a Oracle VM Server for SPARC.

Supported configurations with SFCFSHA and multiple I/O Domains

Figure 4-3 SFCFSHA cluster across two guest domains

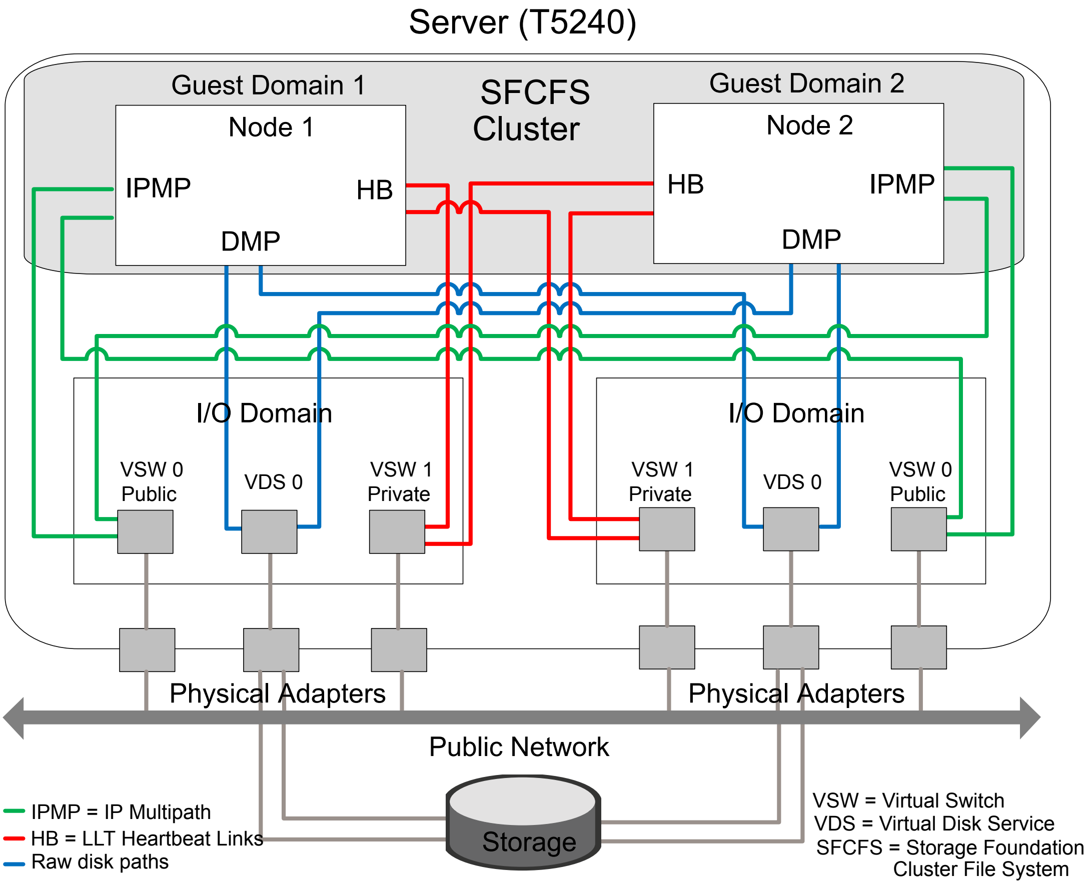


Figure 4-3 illustrates that each guest domain gets network and disk storage redundancy from the two I/O domains.

Figure 4-4 SFCFSHA cluster across two guest domains

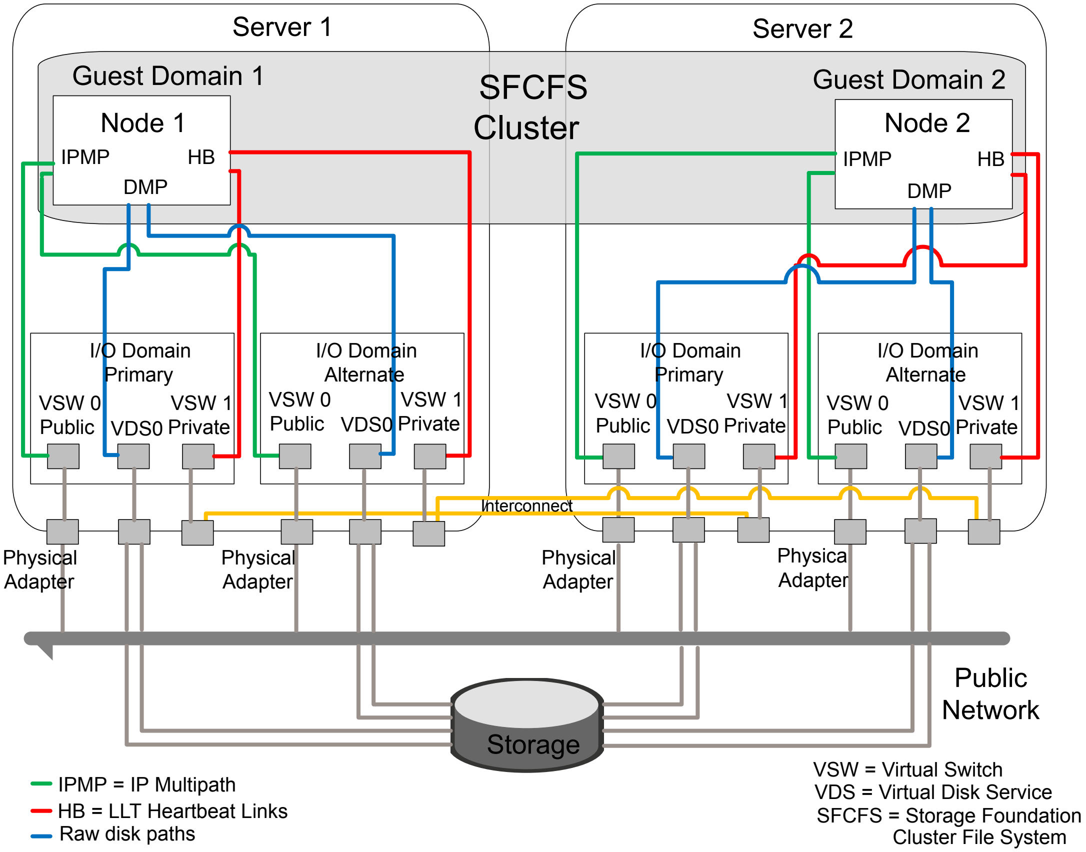


Figure 4-4 illustrates that each guest domain gets network and disk storage redundancy from the two I/O domains on that physical server. The guest cluster spans across two physical servers.

Figure 4-5 SFCFSHA cluster across four guest domains

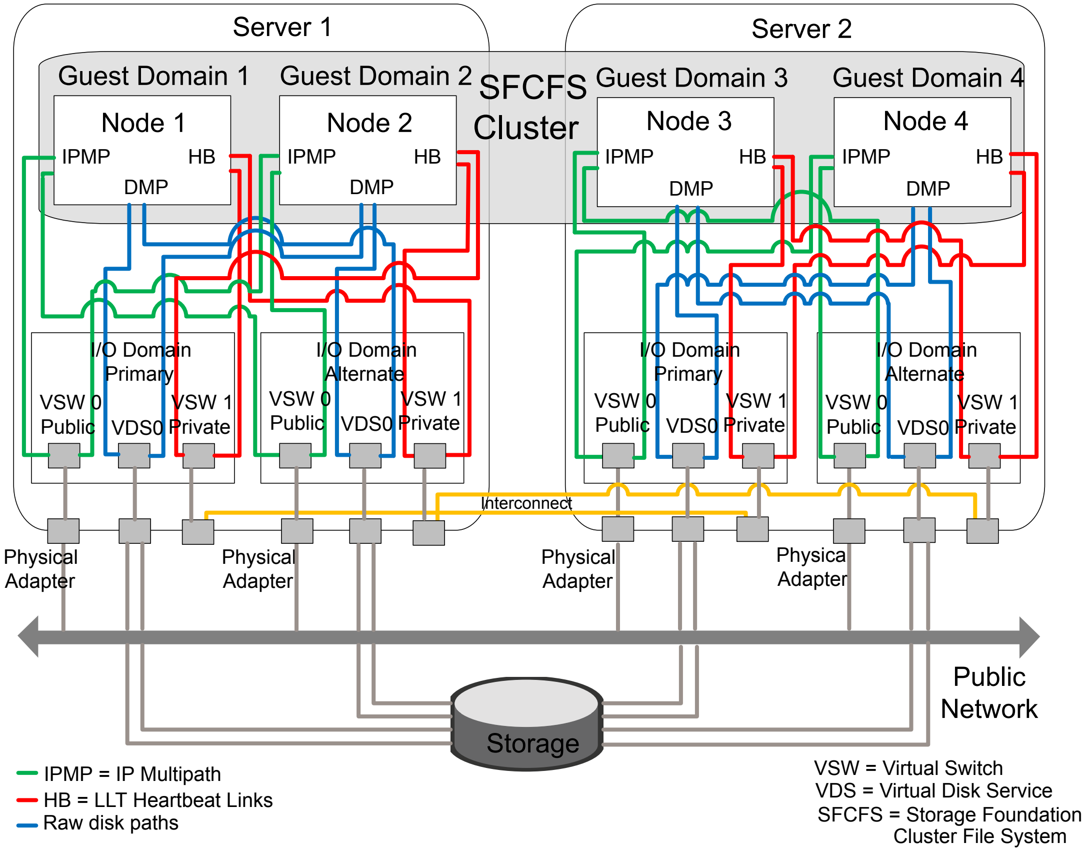


Figure 4-5 illustrates that each guest domain gets network and disk storage redundancy from two I/O domains on that physical server. The guest cluster spans across two physical servers.

Figure 4-6 SFCFSHA cluster across four guest domains

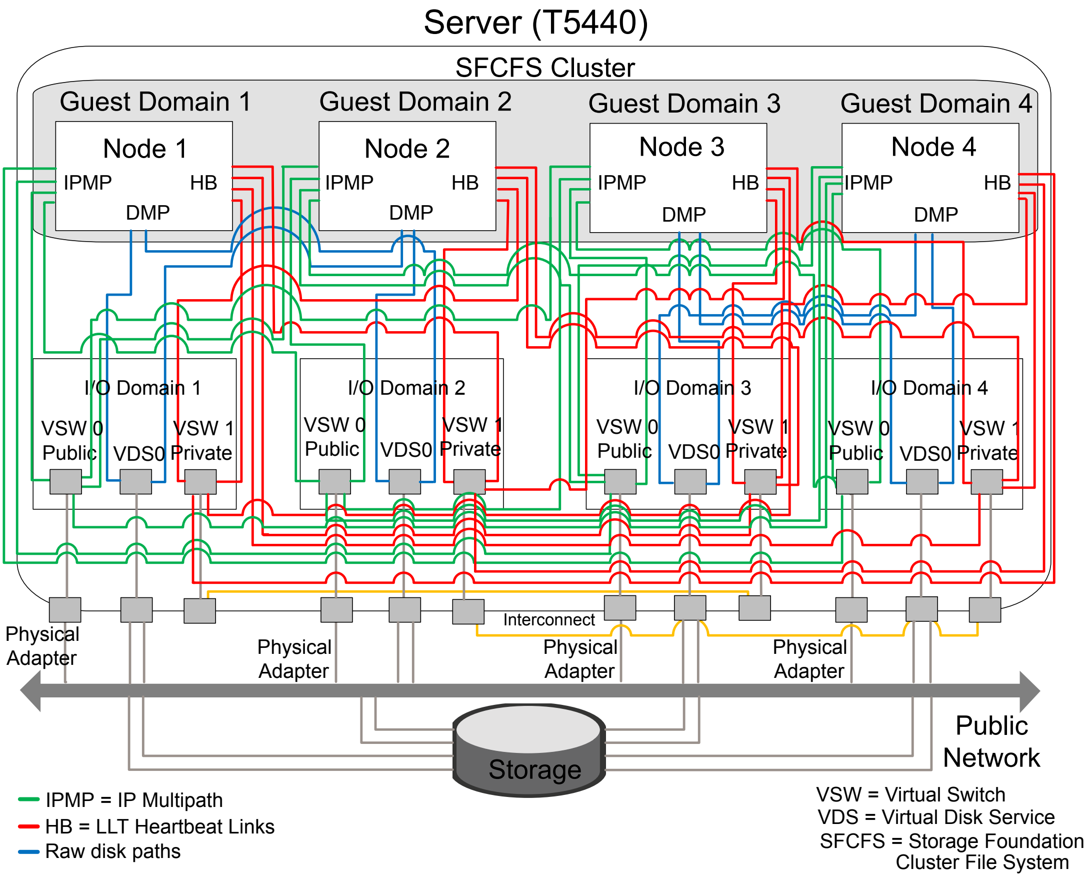


Figure 4-6 illustrates each guest gets its disk storage redundancy from two out of the four I/O domains. Each guest gets its network redundancy from all the four I/O domains.

SFCFSHA limitations

Depending on the configuration model, the following limitations apply to using Storage Foundation Cluster File System High Availability (SFCFSHA) in an Oracle VM Server for SPARC environment.

Limitations for SFCFSHA configuration in the guest domain:

- There is no support for replicating a shared disk group using VVR, when one or more guest domains share the disk group.

- If you want to configure disk based I/O fencing in guest domain, then export physical devices to more than one guest domain on the same physical node using different HBAs. Otherwise, I/O fencing fences off the device whenever one of the guest domain dies. This situation causes the other guest domains also to lose access to the device.

Note: Veritas recommends you to disable I/O fencing if you exported the same physical device to multiple guest domains. If you use fencing on LDOMs, only 1 node of the cluster LDOM can be on each physical machine. Use of CPS servers if you need more that one node LDOM of the same cluster on the same physical machine.

Storage Foundation features restrictions

The following Storage Foundation software features are restricted in the Oracle VM Server for SPARC guest domain environment.

Veritas Volume Replicator bunker replication

Veritas Volume Replicator (VVR) currently does not support configuring a guest domain as a bunker node in the bunker replication mode. This restriction may be removed in a later release.

Mirroring across controllers using vxassist the mirror=ctlr option

Currently, all virtual disks in the guest are under the same virtual controller c0. When `vxassist` tries to look for a second controller to place the mirror on, it fails and therefore the command fails.

All disks fall under the c0 controller, even if they are exported using different physical paths in the backend and coming from different HBAs.

DMP SCSI bypass

The virtual disk client (VDC) driver controls all virtual disks in the guest domain and not the SCSI driver.

Therefore, it is not possible to construct a SCSI packet and send the packet down through DMP. Setting the tunable dmp_fast_recovery to on has no effect.

Event Source Daemon (vxesd) fabric monitoring capabilities

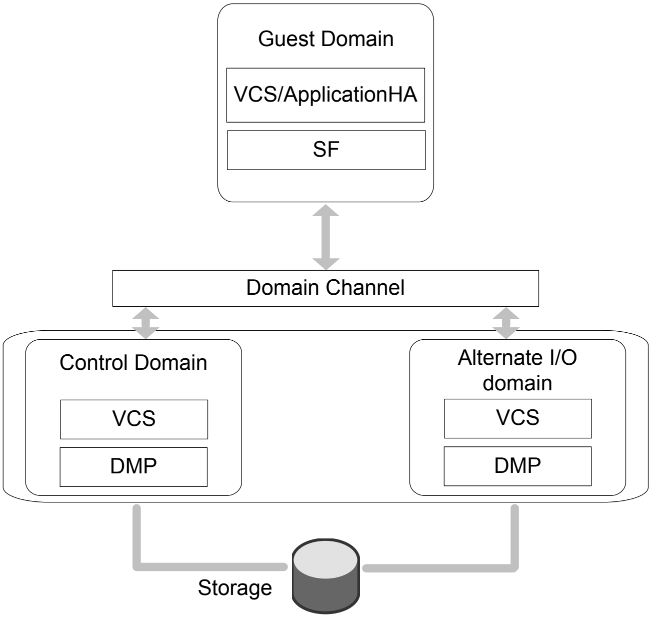
One of the features of the `vxesd` daemon is to register with the HBA API to listen to fabric events. Even though the HBA API is loaded in the guest, since there is currently no concept of a directly connected HBA in the guest, this API is not of any use. Therefore, this capability of `vxesd` is not available.

Physical WWN for a path

It is not possible to create a Sub Path Failover Group (SFG) without Physical WWN. Physical World Wide IDs cannot be obtained from inside the guest because they are fetched by DMP using the HBA API which is currently not functional inside the guest.

Layered Storage Foundation stack model

Figure 4-7 illustrates the layered Storage Foundation stack model with Oracle VM server for SPARC logical domains.



DMP as a multi-pathing solution in an Oracle VM Server for SPARC environment

Using Dynamic Multi-Pathing (DMP), you can choose to manage multiple paths to a system's storage in the following domains:

- control domain
- I/O domains
- guest domains

For configurations that have multiple I/O domains, Veritas recommends that you have DMP manage the paths inside of the I/O domains.

Note: Not having a multi-pathing solution can cause accidental data corruption when the same device is accessed.

Using DMP in the control domain lets you use all of the Storage Foundation (SF) features from inside of the guest. You must enable DMP in the control and alternate I/O domains.

See “Enabling DMP in the control and alternate I/O domains” on page 108.

You can enable DMP path failover in the guest domain.

See “Enabling DMP path failover in the guest domain” on page 110.

You can enable I/O fencing in the guest domain.

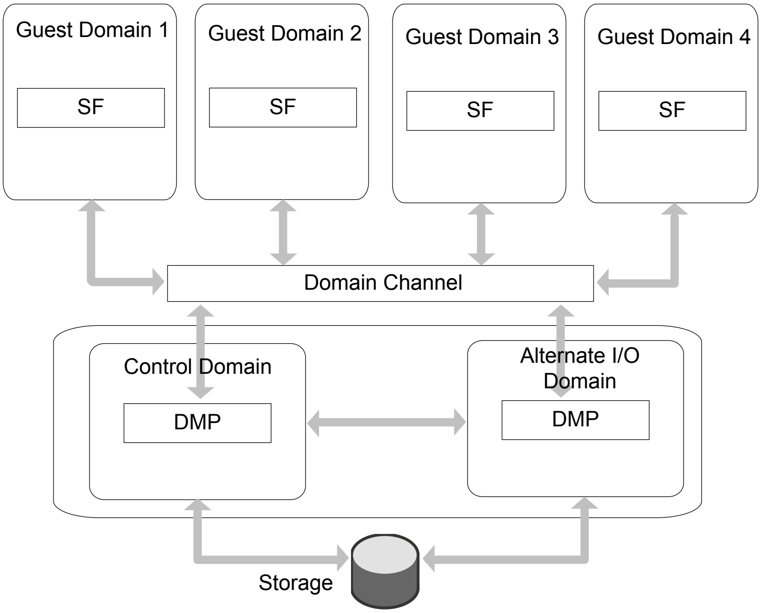
See “Enabling I/O fencing in the guest domain” on page 112.

Enabling DMP in the control and alternate I/O domains

This section describes how to enable Dynamic Multi-Pathing (DMP) in the control and alternate I/O domains.

Figure 4-8 illustrates the architecture of DMP.

Figure 4-8 DMP model



To enable DMP in the control and alternate I/O domains

1. Install VRTSvxvm and VRTSaslapm packages on both the control and alternate I/O domains.

   For information on manual installation of DMP:

   See the *Storage Foundation Configuration and Upgrade Guide*.

   Note: Info related to DMP installation and configuration has been merged to the *Storage Foundation Configuration and Upgrade Guide*.

2. Create the Virtual Disk Service (VDS) devices on DMP metanodes that can be provisioned to the guest domains.

   For example:

   ```
   # ldm add-vdsdev /dev/vx/dmp/xyz vol0015-001-p1@primary-vds0
   # ldm add-vdsdev /dev/vx/dmp/xyz vol0015-001-p2@alternate-vds0
   ```

3. While provisioning, export the DMP metanodes from both the control and alternate I/O domain:

   ```
   # ldm add-vdsdev /dev/vx/dmp/xyz vol0015-001-p1@primary-vds0
   # ldm add-vdsdev /dev/vx/dmp/xyz vol0015-001-p2@alternate-vds0
   # ldm add-vdisk timeout=30 vdsk0015-001-p1 \
   vol0015-001-p1@primary-vds0 hsxd0015
   # ldm add-vdisk timeout=30 vdsk0015-001-p2 \
   vol0015-001-p2@alternate-vds0 hsxd0015
   ```

   This allows DMP in the guest domain to see two access paths, one through the control domain and the other through the alternate domain to the storage.

   DMP in the guest domain can take care of the control and alternate I/O domain failures.

Enabling DMP path failover in the guest domain

In Oracle VM Server configurations the Virtual Disk Client (VDC) driver timeout is set to zero by default which signifies infinity. This can cause the failed I/O not to return to the guest domain, if either the control or alternate I/O domain crashes unexpectedly. As a result the guest domain cannot get back the failed I/Os and cannot route them through the alternate domain. If this issue occurs or to avoid this issue, you must set the VDC driver timeout.

There are two ways to set the VDC driver timeout:

Modify globally all of the LUNs that are exported to the current guest domain. This requires a reboot to all the guest domains.	See "To change the VDC driver timeout globally" on page 111.
Manually export every LUN directly to the guest domain and set the timeout parameter to 30 seconds. No reboot is required.	See "To change the VDC driver timeout for each LUN" on page 111.

To change the VDC driver timeout globally

1. On each guest domain, edit the `/etc/system` file and add the following line to set the VDC driver timeout to 30 seconds:

   ```
   set vdc:vdc_timeout=30
   ```

2. Reboot the guest domains.

To change the VDC driver timeout for each LUN

1. Create the primary domain using four internal disks and get all required SAN LUNs for the guest domains allocated to the primary domain.

2. Remove half of the system's I/O from the primary domain:

   ```
   # ldm remove-io pci_X primary_domain_name
   ```

   where *`pci_x`* is the name of the PCI bus for your system.

   where *`prinary_domain_name`* is the name of the primary domain.

   For example:

   ```
   # ldm remove-io pci_@400 primary
   ```

3 Create the alternate I/O domain on the other four internal disks and add the I/O that was removed from the primary domain:

```
# ldm add-io pci_X primary_domain_name
```

where *pci_x* is the name of the PCI bus for your system.

where *primary_domain_name* is the name of the primary domain.

For example:

```
# ldm add-io pci_@400 primary
```

4 On the primary domain, create the guest domains. In the sample the enclosure-based name of one of the LUNs is `xyz` and the guest domain is `hsxd0015`:

```
# ldm add-vdsdev /dev/vx/dmp/xyz vol0015-001-p1@primary-vds0
# ldm add-vdsdev /dev/vx/dmp/xyz vol0015-001-p2@alternate-vds0
# ldm add-vdisk timeout=30 vdsk0015-001-p1 \
vol0015-001-p1@primary-vds0 hsxd0015
# ldm add-vdisk timeout=30 vdsk0015-001-p2 \
vol0015-001-p2@alternate-vds0 hsxd0015
```

The same set of four commands for each SAN LUN that gets placed in a guest domain. Use three SAN LUNs for SAN boot in the guest domain and the rest for application data. Each LUN in the guest domain has one path backup through the primary domain and one backup through the alternate domain. That means each LUN only uses one LDC in each domain. Also, since you are using DMP you still only use 1 LDC in each domain, even if the LUN has more than two paths from the array.

Enabling I/O fencing in the guest domain

Starting with 6.0.5 release, LDOM guest can use SCSI-3 PR fencing with virtual devices backed by DMP devices in I/O domains.

Note: Primary and Alternate IO domains should have Storage Foundation 6.1.1 or higher.

See the *Cluster Server Administrator's Guide* for information about configuring I/O fencing.

How Storage Foundation and High Availability Solutions works in the Oracle VM Server for SPARC

Storage Foundation and High Availability Solutions supports Oracle VM Server for SPARC logical domains in both single-node, multiple-node, and multiple-node high availability configurations.

Following indicates the possible placement of Storage Foundation stack component products:

- For a single node configuration inside the guest domain, either single node Cluster Server (VCS) or ApplicationHA can be placed inside guest domain. Cluster Server (VCS) can be placed inside the control domain for guest domain management. If guest domain is provided with storage I/O services from multiple I/O domains, then Dynamic Multi-Pathing (DMP) can be placed in the IO domains as wells as in the guest domains

  See "About configuring VCS for Oracle VM Server for SPARC with multiple I/O domains" on page 171.

- For clustered nodes, VCS can be placed inside the control domain for guest domain management and inside the guest domains for application availability.

Storage Foundation features restrictions

When VCS is installed inside the control domains to manage the guest domains and also VCS is installed inside those guest domains for application management, configuring IO fencing for both the clusters is not supported.

System requirements

See the *Veritas InfoScale Release Notes* for the system requirements.

Hardware requirements

Visit the Oracle Web site for information about the supported hardware for Oracle VM Server for SPARC

Product release notes

See *Veritas InfoScale Release Notes* before installing Veritas InfoScale products.

Product licensing

Customers running Storage Foundation or Storage Foundation Cluster File System High Availability in an Oracle VM Server for SPARC environment are entitled to use an unlimited number of logical domains on each licensed server or CPU.

Installing Storage Foundation in a Oracle VM Server for SPARC environment

This section describes how to install Storage Foundation in several Oracle VM Server for SPARC environments.

To install the Split Storage Foundation stack model environment, you must complete the following sections in order:

- See "Installing and configuring Oracle VM Server for SPARC and domains" on page 114.
- See "Installing Storage Foundation in the control domain or guest" on page 115.
- See "Installing Veritas File System in the guest domain" on page 115.
- See "Verifying the configuration" on page 116.

To install the Guest based Storage Foundation stack model environment, which includes Storage Foundation Cluster File System High Availability, you must complete the following sections in order:

- See "Installing and configuring Oracle VM Server for SPARC and domains" on page 114.
- See "Installing Storage Foundation in the control domain or guest" on page 115.
- See "Verifying the configuration" on page 116.

To install and configure Cluster Server in a Oracle VM Server for SPARC environment, see the following sections:

- See "Configuring VCS to fail over an application on a failure" on page 159.

Installing and configuring Oracle VM Server for SPARC and domains

Refer to the Oracle documentation for instructions about installing and configuring the Oracle VM Server for SPARC software and configuring the control and guest domains.

Installing Storage Foundation in the control domain or guest

This section describes how to install Storage Foundation in the control domain or guest domain.

Installing the split Storage Foundation stack model

If you are installing the split Storage Foundation stack model, the entire stack must be in the control domain and VxFS must be in the guest domain.

Use the procedures in the Storage Foundation Configuration and Upgrade Guide to install Storage Foundation in the control domain.

To install the split Storage Foundation stack model

◆ Install the product.

See the *Storage Foundation Configuration and Upgrade Guide for Solaris*.

See the *Veritas InfoScale Release Notes for Solaris*.

Installing the guest-based Storage Foundation stack model

If you are installing the guest-based Storage Foundation stack model environment, the entire stack must be in the guest domain.

Note: The Veritas Volume Manager (VxVM) must to be installed in the control domain, if you allow the use of volumes as vdisks for the boot device on the guest hosts.

Use the procedures in the SF or SFCFSHA Configuration and upgrade Guide, as appropriate, to install SF or SFCFSHA in a guest domain.

To install the guest-based Storage Foundation stack model

◆ Install the product.

See the *Storage Foundation Configuration and Upgrade Guide for Solaris* for SF.

See the *Veritas InfoScale Release Notes for Solaris* for SF.

See the *Storage Foundation Cluster File System High Availability Configuration and Upgrade Guide for Solaris* for SFCFSHA.

See the *Veritas InfoScale Release Notes for Solaris* for SFCFSHA.

Installing Veritas File System in the guest domain

This section describes how to install Veritas File System (VxFS) in the guest domain.

To install VxFS in the guest domain

◆
- On Solaris 11:
  If the guest OS is Oracle Solaris 11, refer to the Installation Guide for the product to install VxFS inside the guest domain.

Verifying the configuration

Verify the configuration of Oracle VM server for SPARC in the control domain and the guest domain. Refer to the Oracle documentation for details.

Verify the installation in both the control domain and the guest domain. Refer to the following guides for more information:

- See the *Veritas InfoScale Release Notes for Solaris.*
- See the *Storage Foundation Cluster File System High Availability Configuration and Upgrade Guide for Solaris*.

Exporting a Veritas volume to a guest domain from the control domain

Use the following procedure to migrate a VxVM disk group from a non-logical domain environment to a Oracle VM Server for SPARC environment.

Note: This section applies to only the Split Storage Foundation model.

In the following example control domain is named `primary` and the guest domain is named `ldom1` The prompts in each step show in which domain to run the command.

To create virtual disks on top of the Veritas Volume Manager data volumes using the ldm command

1. The VxVM diskgroup on the target host is imported in the control domain, after which volumes are visible from inside the control domain.

   See the *Storage Foundation Administrator's Guide* to move disk groups between systems.

2. In the control domain (primary), configure a service exporting the VxVM volume containing a VxFS or UFS filesystem as a slice using the `options=slice` option:

   ```
   primary# ldm add-vdiskserverdevice options=slice \
   /dev/vx/dsk/dg-name/volume_name \
   volume_name volume_name@primary-vds0
   ```

   Caution: A volume by default shows up as a full disk in the guest. The Virtual Disk Client driver writes a VTOC on block 0 of the virtual disk, which will end up as a WRITE on block 0 of the VxVM volume. This can potentially cause data corruption, because block 0 of the VxVM volume contains user data. Using `options=slice` exports a volume as a slice to the guest and does not cause any writes to block 0, therefore preserving user data.

3. Add the exported disk to a guest domain:

   ```
   primary# ldm add-vdisk vdisk1 volume_name
   volume_name@primary-vds0 ldom1
   ```

4. Start the guest domain, and ensure that the new virtual disk is visible.

   ```
   primary# ldm bind ldom1
   ```

   ```
   primary# ldm start ldom1
   ```

5 If the new virtual disk device node entires do not show up in the`/dev/[r]dsk` directories, then run the `devfsadm` command in the guest domain:

```
ldom1# devfsadm -C
```

In this example, the new disk appears as /dev/[r]dsk/c0d1s0.

```
ldom1# ls -l /dev/dsk/c0d1s0
```

```
lrwxrwxrwx 1 root root 62 Sep 11 13:30 /dev/dsk/c0d1s0 ->
../../devices/virtual-devices@100/channel-devices@200/disk@1:a
```

6 Mount the file system on the disk to access the application data:

```
ldom1# mount -F vxfs /dev/dsk/c0d1s0 /mnt
```

```
ldom1# mount -F ufs /dev/dsk/c0d1s0 /mnt
```

Provisioning storage for a guest domain

Use the following procedure to provision storage for a guest domain. You can provision both boot disks and data disks.

Note: This section applies to the Split Storage Foundation stack model only.

For the guest-based Storage Foundation model:

See “How Storage Foundation and High Availability Solutions works in the guest domains” on page 100.

Provisioning Veritas Volume Manager volumes as data disks for guest domains

The following procedure uses VxVM volumes as data disks (virtual disks) for guest domains.

VxFS can be used as the file system on top of the data disks.

The example control domain is named `primary` and the guest domain is named `ldom1`. The prompts in each step show in which domain to run the command.

To provision Veritas Volume Manager volumes as data disks

1. Create a VxVM disk group (`mydatadg` in this example) with some disks allocated to it:

   ```
   primary# vxdg init mydatadg TagmaStore-USP0_29 TagmaStore-USP0_30
   ```

2. Create a VxVM volume of the desired layout (in this example, creating a simple volume):

   ```
   primary# vxassist -g mydatadg make datavol1 500m
   ```

3. Configure a service exporting the volume datavol1 as a virtual disk:

   ```
   primary# ldm add-vdiskserverdevice /dev/vx/dsk/mydatadg/datavol1 \
   datadisk1@primary-vds0
   ```

4. Add the exported disk to a guest domain.

   ```
   primary# ldm add-vdisk vdisk1 datadisk1@primary-vds0 ldom1
   ```

5. Start the guest domain, and ensure that the new virtual disk is visible:

   ```
   primary# ldm bind ldom1
   ```

   ```
   primary# ldm start ldom1
   ```

6. If the new virtual disk device node entires do not show up in the`/dev/[r]dsk` directories, then run the `devfsadm` command in the guest domain:

   ```
   ldom1# devfsadm -C
   ```

7. Label the disk using the `format` command to create a valid label before trying to access it.

   See the `format`(1M) manual page.

8. Create the file system where c0d1s2 is the disk.

   ```
   ldom1# mkfs -F vxfs /dev/rdsk/c0d1s2
   ```

9 Mount the file system.

```
ldom1# mount -F vxfs /dev/dsk/c0d1s2 /mnt
```

10 Verify that the file system has been created:

```
ldom11# df -hl -F vxfs
```

```
Filesystem size used avail capacity Mounted on
/dev/dsk/c0d1s2 500M 2.2M 467M 1% /mnt
```

Provisioning Veritas Volume Manager volumes as boot disks for guest domains

The following procedure provisions boot disks for a guest domain.

Note: This deployment model applies to Guest-based Storage Foundation stack model.

A VxVM volume appears as a full disk by default and can be used as a boot disk for a guest domain.

The following process gives the outline of how a VxVM volume can be used as a boot disk.

The example control domain and is named `primary` the guest domain is named `ldom1`. The prompts in each step show in which domain to run the command.

To provision Veritas Volume Manager volumes as boot disks for guest domains

1 On the control domain, create a VxVM volume of a size that is recommended for Solaris 11 installation. In this example, a 7GB volume is created:

```
primary# vxassist -g boot_dg make bootdisk-vol 7g
```

2 Configure a service by exporting the `/dev/vx/dsk/boot_dg/bootdisk1-vol` volume as a virtual disk:

```
primary# ldm add-vdiskserverdevice \
/dev/vx/dsk/boot_dg/bootdisk1-vol bootdisk1-vol@primary-vds0
```

3 Add the exported disk to ldom1:

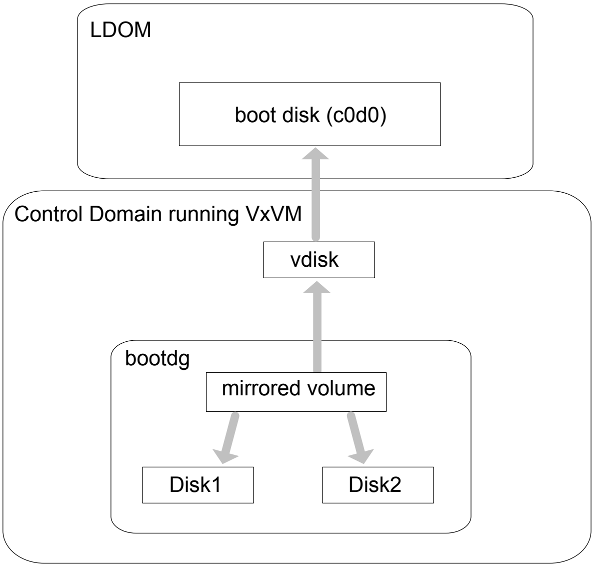
```
primary# ldm add-vdisk vdisk1 bootdisk1-vol@primary-vds0 \
ldom1
```

4 Follow Oracle's recommended steps to install and boot a guest domain, and use the virtual disk vdisk1 as the boot disk during the network, CD, or ISO image install.

Note: It is not supported to encapsulate such a boot disk inside the guest using VxVM or any other 3rd party Volume Management software.

Using VxVM mirrored volumes as boot devices for Ldoms

Figure 4-9 VxVM mirrored volumes as boot devices for Ldoms



For providing high availability and redundancy of the guest boot disks, Veritas recommends that you use mirrored volumes as the backend storage for the boot disks.

The following are some advantages of using this configuration:

- You need to export only a single "vdisk" to the guest LDOM, using only a single LDC channel. This saves on the overall LDC channels being utilized in the control domain.

- Boot disks are managed in a single central location, possibly even in one disk group in the control domain.
- You can easily take snapshots of the boot image using VxVM snapshot feature in the control domain.
- With VxVM doing mirroring in the control domain and a single device being exported to the LDOM, even if the primary boot disk goes bad, the volume still remains enabled in the control domain and is accessible using the same device in the LDOM.
- There is no need to carry out the extra step of encapsulating the boot disk inside the guest for mirroring as it is already mirrored in the Control Domain.

Provisioning boot disks for a guest domain

This section describes how to provision boot disks for a guest domain.

A VxVM volume appears as a full disk by default and can be used as a boot disk for a guest domain.

The following process gives the outline of how a VxVM volume can be used as a boot disk.

The example control domain is named *primary* and the guest domain is named *ldom1*. The prompts in each step show in which domain to run the command.

To provision boot disks for guest domains

1. On the control domain, create a VxVM volume of required size for hosting the guest boot image. In this example, a 7GB volume is created:

   ```
   primary# vxassist -g boot_dg make bootdisk1-vol 7g
   ```

   Veritas recommends that you have a mirrored volume to store the boot image:

   ```
   primary# vxassist -g boot_dg make bootdisk1-vol 7g \
   layout=mirror
   ```

   See the `vxassist`(1M) manual page for more options.

2. Configure a service by exporting the `/dev/vx/dsk/boot_dg/bootdisk1-vol` volume as a virtual disk:

   ```
   primary# ldm add-vdiskserverdevice \
   /dev/vx/dsk/boot_dg/bootdisk1-vol bootdisk1-vol@primary-vds0
   ```

3. Add the exported disk to *ldom1*:

   ```
   primary# ldm add-vdisk vdisk1 bootdisk1-vol@primary-vds0 ldom1
   ```

4. Follow Oracle's recommended steps to install and boot a guest domain, and use the virtual disk `vdisk1` as the boot disk during the network, CD, or ISO image install.

Note: It is not supported to encapsulate such a boot disk inside the guest using VxVM or any other 3rd party Volume Management software. Full SCSI disks can be encapsulated, while vdisks that come from volumes cannot.

Using VxVM snapshot as a backup copy of the boot image during the upgrade

You have the ability to preserve a backup copy of the guest boot image using the Veritas Volume Manager (VxVM) snapshot feature.

Veritas recommends the following configuration:

- VxVM 7.4.1 in the control domain.
- A mirrored VxVM volume per guest LDom boot image.
- For ease of management, you may want to group all the LDOM boot image volumes into a separate disk group.

To upgrade the guest boot image

1. Shut down the guest after synchronizing the operating system file system:

   ```
   # sync
   # init 0
   ```

2. Stop and unbind the guest:

   ```
   # ldm stop guest
   # ldm unbind guest
   ```

3. (Optional) Taking a snapshot of a VxVM volume requires allocating a DCO object that is done by executing the `vxsnap prepare` command. Veritas recommends that you mirror the DCO for redundancy. If you choose to do so, add two disks of reasonable size (such as 2gb) to the disk group containing the boot volumes:

   ```
   # vxdg -g disk_group adddisk [ disk3 disk4 ]
   ```

4. Perform one of the following:

   - If you performed step 3, prepare the separate disks in the disk group for DCO for the snapshot operation:

     ```
     # vxsnap -g disk_group prepare boot_volume [alloc=disk3 disk4]
     ```

   - If you skipped step 3, prepare the boot volume for snapshot operation:

     ```
     # vxsnap -g disk_group prepare boot_volume
     ```

5. Ensure that the mirror plexes of the volume are completely synchronized:

   ```
   # vxtask list
   ```

   The output of the `vxtask list` command shows if any currently synchronize operation is in progress. If there is any such task in progress then you have to wait until it completes.

   ```
   # vxsnap -g disk_group print
   ```

   This should show that the dirty percentage is 0% and the valid percentage is 100% for both the original and the snapshot volumes. If not, wait until the original and the snapshot volumes are synchronized.

6. Take the snapshot of the boot volume and specify the name of the plex that you want to use for backup.

   ```
   # vxsnap -g disk_group make \
   source=boot_volume/new=backup_vol/plex=backup_plex
   ```

   where *`backup_plex`* is the plex that you want to use for backup.

   This operation creates a snapshot volume using the *`backup_plex`*. This snapshot volume can be used to revert to the boot image to the point-in-time it was taken.

7 Ensure that the new snapshot volume is completely synchronized:

```
# vxtask list
```

The output of the `vxtask list` command shows if any currently synchronize operation is in progress. If there is any such task in progress then you have to wait until it completes.

```
# vxsnap -g disk_group print
```

This should show that the dirty percentage is 0% and the valid percentage is 100% for both the original and the snapshot volume. If not, wait until the original and the snapshot volumes are synchronized.

8 Bind and restart the guest and boot the guest:

```
# ldm bind guest
# ldm start guest
```

The guest is now booting from the primary plex.

9 Perform the upgrade of the intended guest.

10 Once the upgrade is successful reattach the snapshot volume to the original boot volume. This operation causes the backup plex to get re-attached back to the boot volume as a mirror, making the volume redundant again with two mirrored plexes.

```
# vxsnap -g disk_group reattach backup_vol source=boot_volume
```

Restore the original environment

In the event of an upgrade failure, you have the option to revert to the original boot environment prior to the upgrade using the following procedure:

To restore the original environment

1. Stop and unbind the guest:

   ```
   # ldm stop guest
   # ldm unbind guest
   ```

2. Restore the boot volume from the backup snapshot volume:

   ```
   # vxsnap -g disk_group restore boot_volume source=backup_vol
   ```

   This operation restores the boot image to the point-in-time that the snapshot was taken prior to the upgrade.

   See "Using VxVM snapshot as a backup copy of the boot image during the upgrade" on page 123.

3. Ensure that the boot volume is completely restored

   ```
   # vxtask list
   ```

   The output of the `vxtask list` command shows if any currently synchronize operation is in progress. If there is any such task in progress then you have to wait until it completes.

   ```
   # vxsnap -g disk_group print
   ```

   This should show that the dirty percentage is 0% and the valid percentage is 100% for both the original and the snapshot volume. If not, wait until the original and the snapshot volumes are synchronized.

4. Bind and restart the guest:

   ```
   # ldm bind guest
   # ldm start guest
   ```

5. Verify that the boot environment is restored correctly.

6. Reattach the snapshot volume to the source volume as a plex. This operation causes the backup plex to get re-attached back to the boot volume as a mirror, making the volume redundant again with two mirrored plexes:

   ```
   # vxsnap -g disk_group reattach backup_vol source=boot_volume
   ```

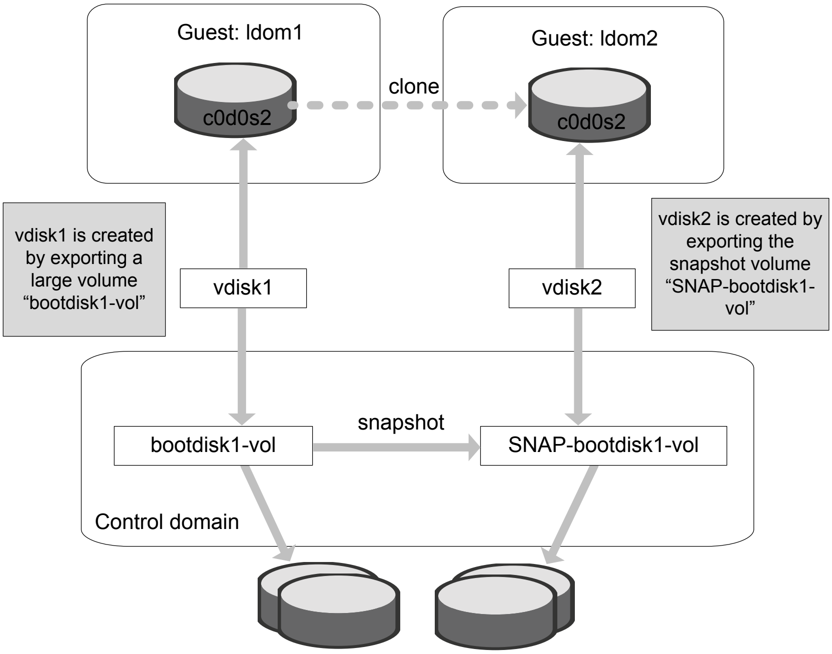
Using Veritas Volume Manager snapshots for cloning logical domain boot disks

The following highlights the steps to clone the boot disk from an existing logical domain using VxVM snapshots, and makes use of the third-mirror breakoff snapshots.

See "Provisioning Veritas Volume Manager volumes as boot disks for guest domains" on page 120.

Figure 4-10 illustrates an example of using Veritas Volume Manager snapshots for cloning Logical Domain boot disks.

Figure 4-10 Example of using Veritas Volume Manager snapshots for cloning Logical Domain boot disks


Guest: ldom1
c0d0s2
clone
Guest: ldom2
c0d0s2
vdisk1 is created by exporting a large volume "bootdisk1-vol"
vdisk1
vdisk2
vdisk2 is created by exporting the snapshot volume "SNAP-bootdisk1-vol"
bootdisk1-vol
snapshot
SNAP-bootdisk1-vol
Control domain

Before this procedure, ldom1 has its boot disk contained in a large volume, `/dev/vx/dsk/boot_dg/bootdisk1-vol`.

This procedure involves the following steps:

- Cloning the logical domain configuration to form a new logical domain configuration.

This step is a Solaris logical domain procedure, and can be achieved using the following commands:

```
# ldm list-constraints -x
```

```
# ldm add-domain -i
```

Refer to the Oracle documentation for more information about cloning the logical domain configuration to form a new logical domain configuration.
See the *Oracle VMServer for SPARC Administration Guide*.

- After cloning the configuration, clone the boot disk and provision it to the new logical domain.
  To create a new logical domain with a different configuration than that of ldom1, skip this step of cloning the configuration and create the desired logical domain configuration separately.

To clone the boot disk using Veritas Volume Manager snapshots

1. Create a snapshot of the source volume bootdisk1-vol. To create the snapshot, you can either take some of the existing ACTIVE plexes in the volume, or you can use the following command to add new snapshot mirrors to the volume:

   ```
   primary# vxsnap [-b] [-g diskgroup] addmir volume \
   [nmirror=N] [alloc=storage_attributes]
   ```

   By default, the `vxsnap addmir` command adds one snapshot mirror to a volume unless you use the `nmirror` attribute to specify a different number of mirrors. The mirrors remain in the SNAPATT state until they are fully synchronized. The `-b` option can be used to perform the synchronization in the background. Once synchronized, the mirrors are placed in the SNAPDONE state.

   For example, the following command adds two mirrors to the volume, bootdisk1-vol, on disks mydg10 and mydg11:

   ```
   primary# vxsnap -g boot_dg addmir bootdisk1-vol \
   nmirror=2 alloc=mydg10,mydg11
   ```

   If you specify the `-b` option to the `vxsnap addmir` command, you can use the `vxsnap snapwait` command to wait for synchronization of the snapshot plexes to complete, as shown in the following example:

   ```
   primary# vxsnap -g boot_dg snapwait bootdisk1-vol nmirror=2
   ```

2 To create a third-mirror break-off snapshot, use the following form of the `vxsnap make` command.

Caution: Shut down the guest domain before executing the `vxsnap` command to take the snapshot.

```
primary# vxsnap [-g diskgroup] make \
source=volume[/newvol=snapvol] \
{/plex=plex1[,plex2,...]|/nmirror=number]}
```

Either of the following attributes may be specified to create the new snapshot volume, snapvol, by breaking off one or more existing plexes in the original volume:

plex	Specifies the plexes in the existing volume that are to be broken off. This attribute can only be used with plexes that are in the ACTIVE state.
nmirror	Specifies how many plexes are to be broken off. This attribute can only be used with plexes that are in the SNAPDONE state. Such plexes could have been added to the volume by using the `vxsnap addmir` command.

Snapshots that are created from one or more ACTIVE or SNAPDONE plexes in the volume are already synchronized by definition.

For backup purposes, a snapshot volume with one plex should be sufficient.

For example,

```
primary# vxsnap -g boot_dg make \
source=bootdisk1-vol/newvol=SNAP-bootdisk1-vol/nmirror=1
```

Here bootdisk1-vol makes source; SNAP-bootdisk1-vol is the new volume and 1 is the nmirror value.

The block device for the snapshot volume will be `/dev/vx/dsk/boot_dg/SNAP-bootdisk1-vol.`

3 Configure a service by exporting the`/dev/vx/dsk/boot_dg/SNAP-bootdisk1-vol` file as a virtual disk.

```
primary# ldm add-vdiskserverdevice \
/dev/vx/dsk/boot_dg/SNAP-bootdisk1-vol vdisk2@primary-vds0
```

4 Add the exported disk to ldom1 first.

```
primary# ldm add-vdisk vdisk2 \
SNAP-bootdisk1-vol@primary-vds0 ldom1
```

```
primary# ldm bind ldom1
primary# ldm start ldom1
```

5 Start ldom1 and boot ldom1 from its primary boot disk vdisk1.

```
primary# ldm bind ldom1
primary# ldm start ldom1
```

6 If the new virtual disk device node entries do not show up in the`/dev/[r]dsk` directories, then run the `devfsadm` command in the guest domain:

```
ldom1# devfsadm -C
```

where vdisk2 is the c0d2s# device.

```
ldom1# ls /dev/dsk/c0d2s*
/dev/dsk/c0d2s0 /dev/dsk/c0d2s2 /dev/dsk/c0d2s4 /dev/dsk/c0d2s6
/dev/dsk/c0d2s1 /dev/dsk/c0d2s3 /dev/dsk/c0d2s5 /dev/dsk/c0d2s7
```

7 Mount the root file system of c0d2s0 and modify the `/etc/vfstab` entries such that all c#d#s# entries are changed to c0d0s#. You must do this because ldom2 is a new logical domain and the first disk in the operating system device tree is always named as c0d0s#.

8 Stop and unbind ldom1 from its primary boot disk vdisk1.

```
primary# ldm stop ldom1
primary# ldm unbind ldom1
```

9 After you change the `vfstab` file, unmount the file system and unbind vdisk2 from ldom1:

```
primary# ldm remove-vdisk vdisk2 ldom1
```

10 Bind vdisk2 to ldom2 and then start and boot ldom2.

```
primary# ldm add-vdisk vdisk2 vdisk2@primary-vds0 ldom2
primary# ldm bind ldom2
primary# ldm start ldom2
```

After booting ldom2, appears as ldom1 on the console because the other host-specific parameters like hostname and IP address are still that of ldom1.

```
ldom1 console login:
```

11 To change the parameters bring ldom2 to single-user mode and run the `sys-unconfig` command.

12 Reboot ldom2.

During the reboot, the operating system prompts you to configure the host-specific parameters such as hostname and IP address, which you must enter corresponding to ldom2.

13 After you have specified all these parameters, ldom2 boots successfully.

Support of live migration for Solaris LDOMs with fencing configured in DMP mode

Consider the case where guest domains form a cluster and I/O fencing is configured in the cluster. If one of the guest domains which is part of the cluster need to be migrated from one physical system to another physical system, then following considerations need to be made.

Prerequisites

- Veritas recommends that the LDOMs configured as part of the cluster are on different Solaris hosts.
- When a guest domain is migrated from one physical host to another,
  - if the guest domains are under VCS management from the control domain and data disks and co-ordinator disks are assigned from above mentioned Veritas qualified arrays, you can make use of ReregPGR attribute in LDOM agent.
    - it is recommended to run the `vxdmpadm pgrrereg` command manually immediately inside the guest domain after migration.
    - If the ReregPGR is enabled and if you use the `hagrp -migrate` VCS command to migrate, VCS will automatically re-registers the key. Refer

to LDOM agent section in VCS bundled agents reference guide for more details.

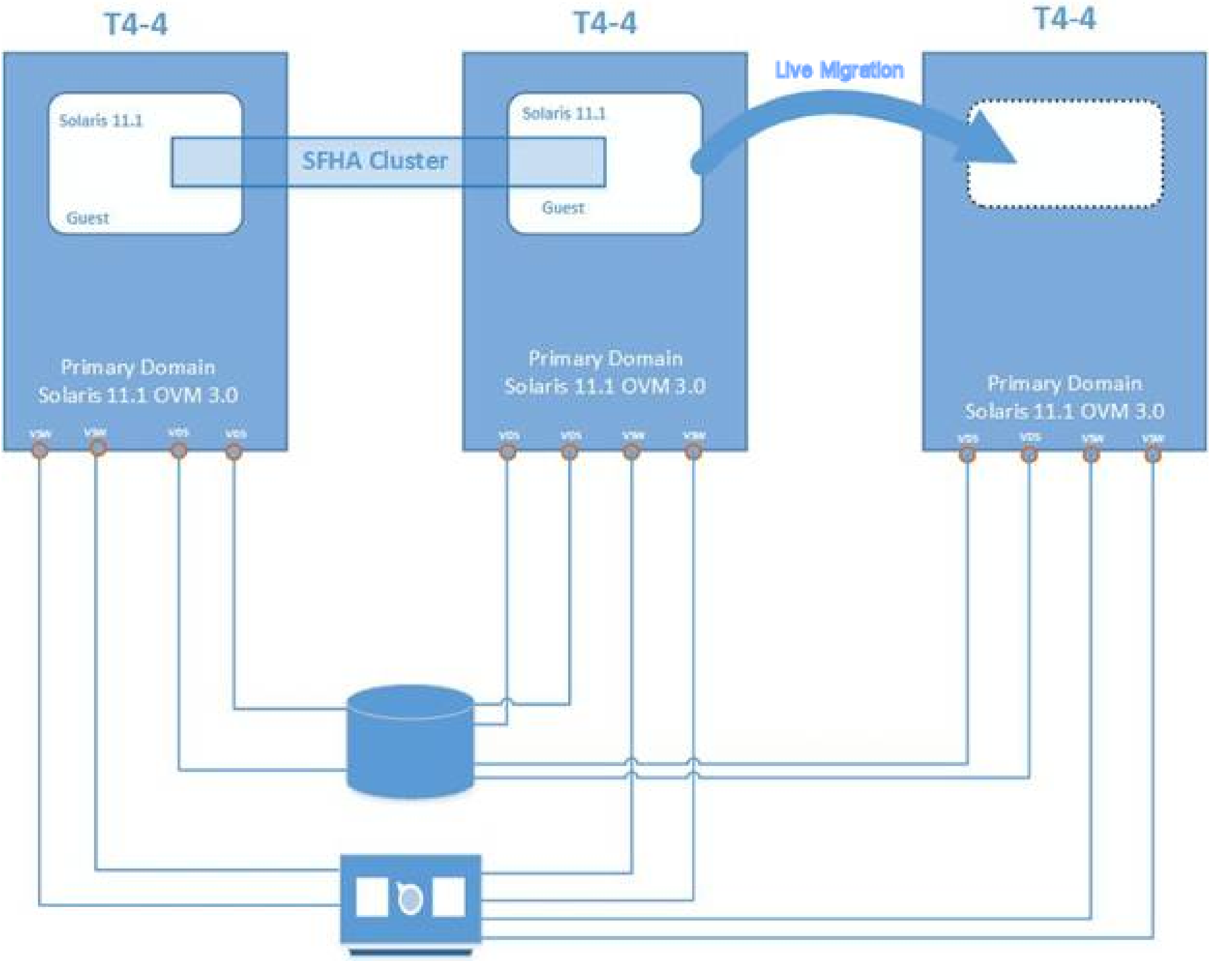
The `vxdmpadm pgrrereg` command is tested and recommended with following arrays only:

- Fujitsu ETERNUS DX440 S2
- Fujitsu ETERNUS DX80 S2
- EMC symmetrix VMAX
- EMC symmetrix DMX
- Hitachi VSP

For example in a typical setup the configuration for this environment:

1. OS inside control and I/O domains: Solaris 11.1.
2. OVM version: 3.0
3. OS inside Guest LDOMs: Solaris 11.1.
4. SFHA configured only inside LDOMs

Figure 4-11 Example of a typical setup configuration



Configuring Oracle VM Server for SPARC guest domains for disaster recovery

The Oracle VMs can be configured for disaster recovery by replicating the boot disk using replication methods like Hitachi TrueCopy, EMC SRDF, Volume Replicator, and so on. The network configuration for the Oracle VM in the primary site may not be effective in the secondary site if the two sites are in different IP subnets. You will need to make these additional configuration changes to the LDom resource.

Note: This deployment model applies to Split Storage Foundation stack model.

To configure the guest domains for disaster recovery, you need to configure VCS on both the sites in the Control Domains with GCO option.

Refer to the *Cluster Server Administrator's Guide* for more information about global clusters.

To set up the guest domain for disaster recovery

1. On the primary site, create the guest domain using `ldm` commands and configure the network-related parameters.
2. On the primary site after you boot the guest domain, copy and install the package VRTSvcsnr from the VCS installation media in the guest domain. This package installs the vcs-network-reconfig service in the guest domain. This service makes sure that the site-specific network parameters are applied when the guest domain boots.
3. On the primary site, shut down the guest domain.
4. Use replication specific commands to failover the replication to the secondary site from the primary site.
5. Repeat step 1 on the secondary site.
6. Perform step7, step 8, step 9, and step 10 on both the primary cluster and the secondary clusters.
7. Create a VCS service group and add a VCS LDom resource for the guest domain.

   Configure the following disaster recovery-related attributes on the LDom resource with site-specific values for each: IPAddress, Netmask, Gateway, DNS (DNS Server).

   Set the value of the ConfigureNetwork attribute to 1 to make the changes effective. The LDom agent does not apply the disaster recovery-related attributes to the guest domain if the value of the ConfigureNetwork attribute is 0.
8. Add the appropriate Mount and DiskGroup resources in the service group for the file system and the disk group on which the boot image of the guest domain resides.

   Add a resource dependency from the LDom resource to the Mount resource and another dependency from the Mount resource to the Diskgroup resource.
9. Add the appropriate VCS replication resource in the service group. Examples of hardware replication agents are SRDF for EMC SRDF, HTC for Hitachi TrueCopy, MirrorView for EMC MirrorView, etc.

Refer to the appropriate VCS Replication agent guide for configuring the replication resource.

For VVR-based replication, add the appropriate RVGPrimary resource to the service group.

Refer to the following manuals for more information:

- For information about configuring VVR-related resources, see the *Storage Foundation and High Availability Solutions Replication Administrator's Guide*.
- For information about the VVR-related agents, see the *Cluster Server Bundled Agents Reference Guide*.

10 Add a dependency from the DiskGroup resource to the replication resource.

Figure 4-12 Sample resource dependency diagram for hardware replication based guest domains

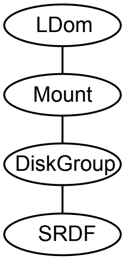
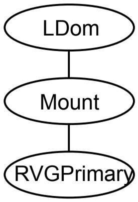


Figure 4-13 Sample resource dependency diagram for VVR replication-based guest domains



The replication resource makes sure that when the resource is online in a site, the underlying replicated devices are in primary mode and the remote devices are in secondary mode. Thus, when the LDom resource goes online, the underlying storage will always be in read-write mode. When the LDom resource goes online, it sets the DR related parameters in the EEPROM parameter network-boot-arguments for the guest domain before starting the guest domain. When the guest domain boots, the vcs-network-reconfig service starts inside the guest domain. This service reads the EEPROM parameter and applies the disaster recovery related parameters by modifying the appropriate files inside the guest domain.

Software limitations

The following section describes some of the limitations of the Oracle VM server for SPARC software and how those software limitations affect the functionality of the Storage Foundation products.

When an I/O domain fails, the vxdisk scandisks or vxdctl enable command take a long time to complete (2791127)

When an I/O domain fails, the vxdisk scandisks or vxdctl enable from the Oracle VM Server for SPARC guest take a long time to complete. `vdc_ioctls` like `DKIOCGGEOM` and `DKIOCINFO` also take more time to return. These issues seem to be due to retry operations performed at the Solaris operating system layer.

Reducing the vdc_timeout value to lower value might help to bring down time. Dynamic multi-pathing (DMP) code is optimized to avoid making such `vdc_ioctl` calls in an Oracle VM Server for SPARC guest environment as much possible. This change considerably reduces delays.

A complete resolution to this issue may require changes at the Solaris operating system level.

Resizing a Veritas Volume Manager volume (exported as a slice or full disk) does not dynamically reflect the new size of the volume in the guest

On resizing a VxVM volume exported to a guest, the virtual disk still shows the old size of the volume. The virtual disk drivers do not update the size of the backend volume after the volume is resized.

Oracle has an RFE for this issue (CR 6699271 Dynamic virtual disk size management).

Workaround: The guest must be stopped and rebound for the new size to be reflected.

This Oracle bug is fixed in Oracle patch 139562-02.

Known issues

The following section describes some of the known issues of the Oracle VM Server for SPARC software and how those known issues affect the functionality of the Storage Foundation products.

Guest-based known issues

The following are new known issues in this release of Storage Foundation and High Availability Solutions Support for Oracle VM Server for SPARC.

Encapsulating a non-scsi disk may fail (1745409)

Trying to encapsulate a non-scsi disk which is a slice of a disk or a disk exported as a slice may fail with the following error:

```
VxVM vxslicer ERROR V-5-1-599 Disk layout does not support swap shrinking
VxVM vxslicer ERROR V-5-1-5964 Unsupported disk layout.
Encapsulation requires at least 0 sectors of unused space either at the
beginning or end of the disk drive.
```

This is because while installing the OS on such a disk, it is required to specify the entire size of the backend device as the size of slice "s0", leaving no free space on the disk.

Boot disk encapsulation requires free space at the end or the beginning of the disk for it to proceed ahead.

Guest domain node shows only one PGR key instead of two after rejecting the other node in the cluster

For configuration information concerning the guest domain node shows only 1 PGR key instead of 2 after rejecting the other node in the cluster:

See Figure 4-4 on page 103.

This was observed while performing a series of reboots of the primary and alternate I/O domains on both the physical hosts housing the two guests. At some point one key is reported missing on the coordinator disk.

This issue is under investigation. The `vxfen` driver can still function as long as there is 1 PGR key. This is a low severity issue as it will not cause any immediate interruption. Veritas will update this issue when the root cause is found for the missing key.

Disk paths intermittently go offline while performing I/O on a mirrored volume

This was observed while testing the SFCFSHA stack inside a 4-node guest cluster where each node gets its network and virtual disk resources from multiple I/O domains within the same host.

See “Supported configurations with SFCFSHA and multiple I/O Domains” on page 102.

While performing I/O on a mirrored volume inside a guest, it was observed that a vdisk would go offline intermittently even when at least one I/O domain which provided a path to that disk was still up and running.

Split Storage Foundation stack known issues

The following are new known issues in this release of Storage Foundation and High Availability Solutions Support for Oracle VM Server for SPARC.

Caching of data writes on the backend volume in the service domain

This was observed by a customer during their evaluation of Oracle VM Server for SPARC with Storage Foundation. This issue occurs because data written to the virtual disk might get cached into the service domain before it is effectively written to the virtual disk backend. This can cause potential data loss if the service domain crashes while the data is still cached.

Oracle (Sun) bug id is: 6684721 (file backed virtual I/O should be synchronous)

This Oracle (Sun) bug is fixed in Oracle (Sun) patch 139562-02 that has been obsoleted by 138888-07.

A volume can become inaccessible from the guest in the event of control domain reboot

All access to such a volume hangs if the primary domain is rebooted. This is due to the vdisk corresponding to the volume does not come back online after the control domain reboots.

This issue has been identified and fixed under Oracle (Sun) bug id: 6795836 (vd_setup_vd() should handle errors from vd_identify_dev() better)

This Oracle (Sun) bug is fixed in Oracle (Sun) patch 141777-01.

Chapter 5

Cluster Server support for using CVM with multiple nodes in a Oracle VM Server for SPARC environment

This chapter includes the following topics:

Clustering using Cluster Volume Manager

The Veritas Volume Manager cluster functionality (CVM) makes logical volumes and raw device applications accessible throughout a cluster.

In the split Storage Foundation model, CVM is placed in the control domain and VxFS is placed in the guest domain. In this model, CVM provides high availability and shared storage visibility at the control domain level across multiple physical nodes in the cluster.

See "Cluster Volume Manager in the control domain for providing high availability" on page 143.

In the guest-based Storage Foundation stack model, CVM is placed in the guest domain, providing high availability and shared storage visibility at the guest domain level across multiple guest domains that act as the nodes in the cluster.

Installing Storage Foundation on multiple nodes in a Logical Domain

To install Storage Foundation on multiple nodes in a Solaris Logical Domains environment, you must complete the following operations, which are the same as on a single node:

- See "Installing and configuring Oracle VM Server for SPARC and domains" on page 114.
- See "Installing Storage Foundation in the control domain or guest" on page 115.
- See "Installing Veritas File System in the guest domain" on page 115.
- See "Verifying the configuration" on page 116.

Reconfiguring the clustering agents for Cluster Volume Manager

This section applies to only the Split Storage Foundation model.

For a Storage Foundation CVM, the following additional configuration steps are necessary:

- See "Removing the vxfsckd resource" on page 141.
- See "Creating CVMVolDg in a group" on page 142.

Removing the vxfsckd resource

After configuring Storage Foundation and CVM, remove the vxfsckd resource.

To remove the vxfsckd resource

1. Make the configuration writeable:

```
# haconf -makerw
```

2. Delete the resource:

```
# hares -delete vxfsckd
```

3. Make the configuration read-only:

```
# haconf -dump -makero
```

4 Stop the resources:

```
# hastop -all
```

5 Restart the resources.

```
# hastart
```

Run the `hastart` command on all nodes in the cluster.

Creating CVMVolDg in a group

The following procedure creates CVMVolDg in a given group.

To create CVMVolDg

1 Make the configuration writeable:

```
# haconf -makerw
```

2 Add the CVMVolDg resource:

```
# hares -add name_of_resource CVMVolDg name_of_group
```

3 Add a diskgroup name to the resource:

```
# hares -modify name_of_resource CVMDiskGroup diskgroup_name
```

4 Make the attribute local to the system:

```
# hares -local name_of_resource CVMActivation
```

5 Add the attribute to the resource.

```
# hares -modify name_of_resource CVMActivation \
activation_value -sys nodename
```

Repeated this step on each of the nodes.

6 If you want to monitor volumes, enter the following command:

```
# hares -modify name_of_resource CVMVolume \
name_of_volume
```

In a database environment, Veritas recommends you monitor volumes.

7 Modify the resource so that a failure of this resource does not bring down the entire group:

```
# hares -modify name_of_resource Critical 0
```

8 Enable the resources:

```
# hares -modify name_of_resource Enabled 1
```

9 Make the configuration read-only:

```
# haconf -dump -makero
```

10 Verify the configuration:

```
# hacf -verify
```

This should put the resource in the `main.cf` file.

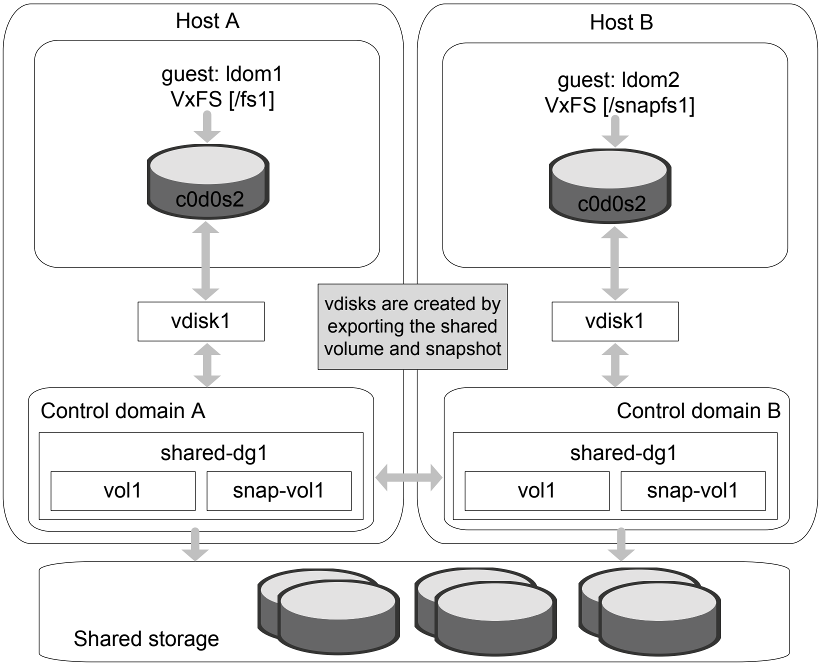
Cluster Volume Manager in the control domain for providing high availability

The main advantage of clusters is protection against hardware failure. Should the primary node fail or otherwise become unavailable, applications can continue to run by transferring their execution to standby nodes in the cluster.

CVM can be deployed in the control domains of multiple physical hosts running Oracle VM Server for SPARC, providing high availability of the control domain.

Figure 5-1 illustrates a CVM configuration.

Figure 5-1 CVM configuration in an Oracle VM Server for SPARC environment



If a control domain encounters a hardware or software failure causing the domain to shut down, all applications running in the guest domains on that host are also affected. These applications can be failed over and restarted inside guests running on another active node of the cluster.

Caution: As such applications running in the guests may resume or time out based on the individual application settings. The user must decide if the application must be restarted on another guest on the failed-over control domain. There is a potential data corruption scenario if the underlying shared volumes get accessed from both of the guests simultaneously.

Shared volumes and their snapshots can be used as a backing store for guest domains.

Note: The ability to take online snapshots is currently inhibited because the file system in the guest cannot coordinate with the VxVM drivers in the control domain.

Make sure that the volume whose snapshot is being taken is closed before the snapshot is taken.

The following example procedure shows how snapshots of shared volumes are administered in such an environment. In the example, datavol1 is a shared volume being used by guest domain ldom1 and c0d1s2 is the front end for this volume visible from ldom1.

To take a snapshot of datavol1

1. Unmount any VxFS file systems that exist on c0d1s0.
2. Stop and unbind ldom1:

   ```
   primary# ldm stop ldom1
   primary# ldm unbind ldom1
   ```

   This ensures that all the file system metadata is flushed down to the backend volume, datavol1.
3. Create a snapshot of datavol1.

   See the *Storage Foundation Administrator's Guide* for information on creating and managing third-mirror break-off snapshots.
4. Once the snapshot operation is complete, rebind and restart ldom1.

   ```
   primary# ldm bind ldom1
   primary# ldm start ldom1
   ```

5. Once ldom1 boots, remount the VxFS file system on c0d1s0.

   Note: If CVM is configured inside the guest domain and the guest domain is planned for migration, perform this step:

   Set the value of the LLT peerinact parameter to sufficiently high value on all nodes in the cluster. You set the value to very high value so that while the logical domain is in migration, the system is not thrown out of the cluster by the other members in the cluster.

   If the CVM stack is unconfigured, the applications can stop.

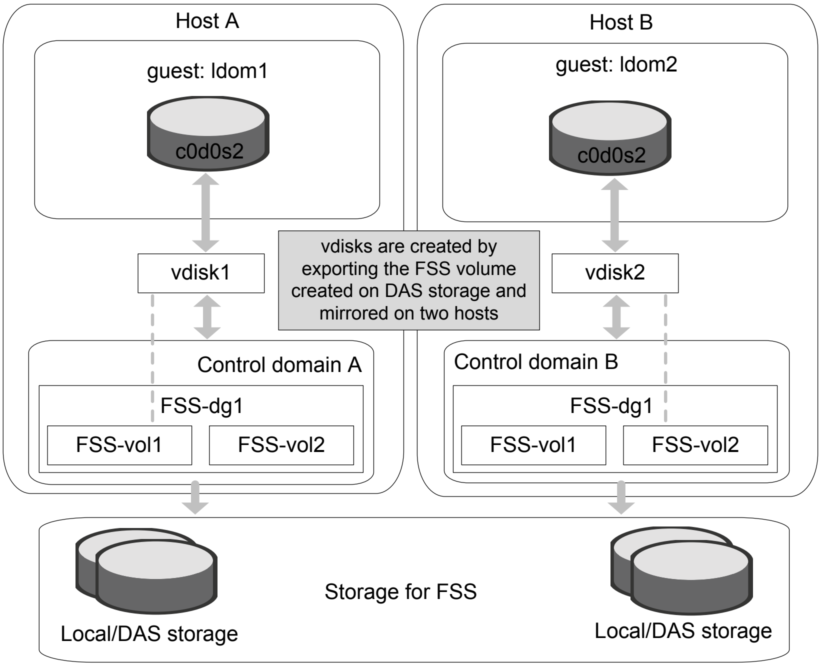
   See the *Cluster Server Administrator’s Guide* for LLT tunable parameter configuration instructions.

Provisioning storage to guests with Flexible Storage Sharing volumes of control domain

Flexible Storage Sharing (FSS) volumes created with shared nothing storage of the control domains of multiple physical hosts can be used in provisioning storage to the guest domains. FSS volumes are mirrored across the control domains cluster nodes. The applications running in the guest domains on one physical host can be failed over to the guests running on another physical host that has active node of the control domain cluster. Live logical domain migration is also feasible with FSS volumes in control domains even though the target host during migration does not have access to the entire storage of the FSS volume.

Figure 5-2 illustrates storage for an FSS volume provisioned to guest domains.

Figure 5-2 FSS storage provisioned to guest domains



Note: When performing Live LDOM migration, make sure that the Support Repository Updates (SRUs) in both the source and the target physical hosts are compatible. See the Oracle documentation for more information on LDOM migration.

Veritas recommends the use of high-speed GAB interconnects when a physical host is running an LDOM (guest domain) with an FSS volume as the root or data disk and local storage is not available for the FSS volume on the physical host.

Chapter 6

VCS: Configuring Oracle VM Server for SPARC for high availability

This chapter includes the following topics:

About VCS in a Oracle VM Server for SPARC environment

Cluster Server (VCS) provides high availability (HA) for Oracle VM Server for SPARC. You can configure VCS to monitor the Logical Domain, services to the Logical Domain, and the applications that run in logical domain, or to monitor only the applications that run in the logical domain.

See "About Cluster Server configuration models in an Oracle VM Server for SPARC environment" on page 152.

Table 6-1 lists the failure scenarios and the VCS failover options based on which you can plan your VCS configuration in an Oracle VM Server for SPARC environment.

Table 6-1 VCS failover options for logical domain failure

Failure scenario	VCS failover	Typical VCS configuration
Logical domains, their storage, or switches fail	VCS fails over the logical domain from one node to another node	VCS is installed in the control domain of each node. See "Cluster Server setup to fail over a logical domain on a failure of logical domain" on page 153.
Applications that run in logical domains fail Or logical domain where the application is running fails	VCS fails over the application from one logical domain to another.	VCS is installed in the guest domain of each node. See "Cluster Server setup to fail over an Application running inside logical domain on a failure of Application" on page 158.

Benefits of using VCS in Oracle VM server for SPARC environments to manage logical domains

The following is a list of benefits of using Cluster Server (VCS) in Oracle VM server for SPAC environments to manage logical domains:

- VCS responds gracefully to user initiated native domain migrations (live/warm/cold).
- VCS provides command line interface `hagrp -migrate` to live migrate the domains in place of offline and online between cluster nodes.
- VCS can create a logical domain from a configuration file during online, failover, and switchover.
- VCS provides high availability of logical domain in a multiple I/O domain environment.
- VCS can ensure logical domain availability even during maintenance (reboot/shutdown) of control domain in multiple I/O domain environments.

- VCS supports dynamic re-configuration of Memory and CPU assigned to active logical domains.
- VCS has commands to synchronize logical domain configuration files across cluster nodes.
- VCS supports layered clustering of (between) control domain and (between) guest domains simultaneously. The control domain cluster provides availability to logical domains (infrastructure) and the guest domains cluster provides the availability to applications.

Dynamic reconfiguration of memory and CPU of a guest domain

VCS supports dynamic reconfiguration of memory and CPU assigned to a guest domain. Modify the values of the Memory and NumCPU attributes of an LDom resource to dynamically reconfigure the memory and CPU of a guest domain.

AdaptiveHA for logical domains

Cluster Server (VCS) supports AdaptiveHA (AHA) for service groups containing resources to monitor logical domains (LDoms). AdaptiveHA enables VCS to dynamically select the biggest available CPU and Memory system to fail over the logical domain. The AdaptiveHA can be enabled by setting the attribute FailOverPolicy to BiggestAvailable for the service group containing LDom resource. AdaptiveHA periodically measures the CPU (vCPU) and Memory required by the logical domain. Also AHA measures the free CPU (vCPU) and memory available with the physical server. VCS uses these values while taking decision to failover logical domain based on the FailOverPolicy attribute settings.

See the *Cluster Server Administrator's Guide* for more details on AdaptiveHA.

See the *Cluster Server Bundled Agents Reference Guide* for more details on LDom agent.

Cluster Server requirements

For installation requirements:

See “System requirements” on page 113.

For the configuration model where VCS is installed on the control domain:

- VCS requires shared storage that is visible across all the nodes in the cluster.
- Configure Logical Domain on each cluster node.
- The logical domain’s boot device and application data must reside on shared storage.

For the configuration model where VCS is installed on the guest domain:

- VCS requires the application data to reside on the shared storage.
- Each node can have more than one logical domain.
- Each logical domain can have its own boot device.

Cluster Server limitations

Depending on the configuration model, the following limitations apply to VCS in an Oracle VM Server for SPARC environment.

Limitations for VCS configuration in the control domain:

- Each logical domain configured under VCS must have at least two VCPUs. With one VCPU, the control domain always registers 100% CPU utilization for the logical domain. This is an Oracle VM Server for SPARC software issue.

Limitation for VCS configuration in the guest domain when disk based fencing is configured:

- Do not export a physical device to more than one guest domains on the same physical node. For example: If you configure I/O fencing in guest domain, and if one of the guest domains dies, then I/O fencing fences off the other guest domains as well.
- For co-ordinator disks and data disks, disks must be given from different HBAs if more than one guest domains forming the cluster are residing on the same physical host.
- The disks exported for co-ordinator diks and data disks must be full disks and SCSI compliant.

Cluster Server known issues

The following section describes known issues with VCS in a Oracle VM Server for SPARC environment.

Shutting down the control domain may cause the guest domain to crash (1631762)

Set up	Two Oracle SPARC Enterprise T5240 server physical boxes, each with a control domain and a guest domain. The guest domains in each of the physical boxes form a two node cluster. The nodes are named node 0 and node 1 in the following text.

Symptom	Gracefully shutting down the control domain of node 0 causes the guest domain of node 0 to crash.
Analysis	Even though the guest domain can continue to function when the control domain is shut down, the heartbeats between node 0 and node 1 are lost as the control domain shuts down. As a result, the cluster forms two separate sub-clusters without the sub-clusters being able to see each others' heartbeats. I/O fencing resolves the split brain situation and determines that only one sub-cluster will continue to function while the other sub-cluster should panic. Therefore, the panic of node 0 is expected behavior.
Resolution:	None; this is expected behavior. However, Veritas recommends keeping the control domain highly available for the proper function of the SFCFSHA and SF for Oracle RAC stack in the guest domains. If you have set up a virtual private LLT heartbeats between the two guests (node 0 and node1), the guest will not crash.

About Cluster Server configuration models in an Oracle VM Server for SPARC environment

When you configure Cluster Server (VCS) in an Oracle VM Server for SPARC environment, you need some specific information about the logical domain, network, and the storage devices that the logical domain requires to run.

You need to know the following information about your logical domain:

- The logical domain’s name
- The names of the primary network interfaces for each node
- The virtual switch that the logical domain uses
- The name and type of storage that the logical domain uses

VCS configuration depends on whether you want VCS to fail over the logical domain or the application on a failure:

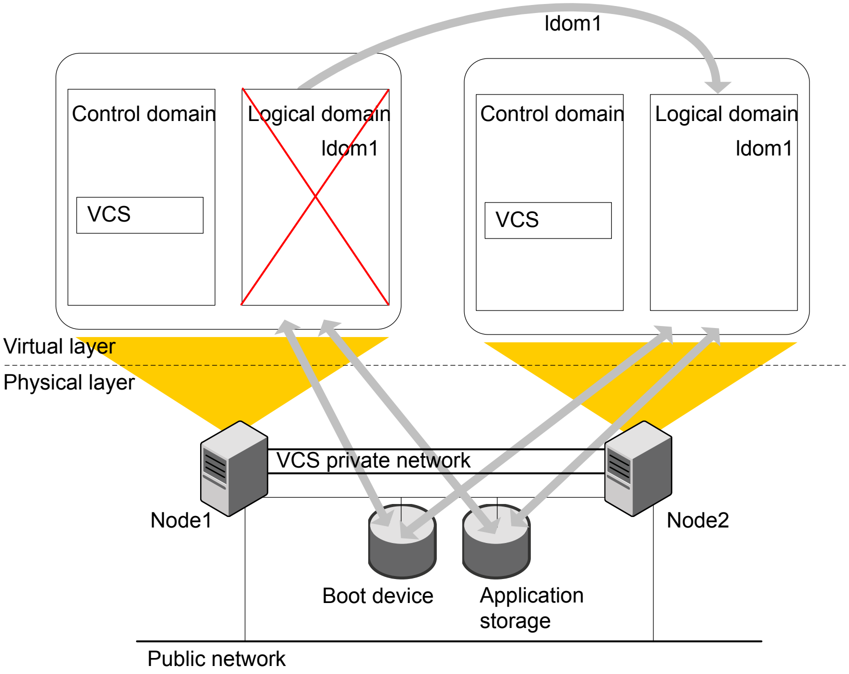
- See “Cluster Server setup to fail over a logical domain on a failure of logical domain” on page 153.
- See “Cluster Server setup to fail over an Application running inside logical domain on a failure of Application” on page 158.

See “About VCS in a Oracle VM Server for SPARC environment” on page 148.

Cluster Server setup to fail over a logical domain on a failure of logical domain

Figure 6-1 illustrates a typical setup where Cluster Server (VCS) installed in the control domain provides high availability to the logical domain and it’s resources.

Figure 6-1 Typical setup for logical domain high availability with VCS installed in control domains

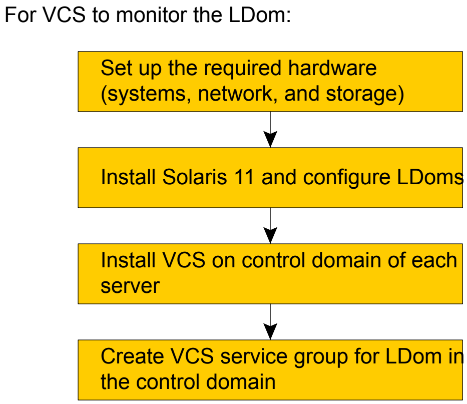


A typical two-node VCS configuration for logical domain high availability has the following software and hardware infrastructure:

- Oracle VM Server for SPARC software is installed on each system Node1 and Node2.
- Shared storage is attached to each system.

- A logical domain (ldom1) exists on both the nodes with a shared boot device.
- Each LDom has an operating system installed.
- VCS is installed in the control domains of each node.

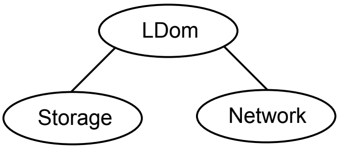
Figure 6-2 Workflow to configure VCS to fail over a logical domain on a failure


For VCS to monitor the LDom:
Set up the required hardware (systems, network, and storage)
Install Solaris 11 and configure LDoms
Install VCS on control domain of each server
Create VCS service group for LDom in the control domain

Configuration scenarios

Figure 6-3 shows the basic dependencies for an logical domain resource.

Figure 6-3 A logical domain resource depends on storage and network resources



Network configuration

Use the NIC agent to monitor the primary network interface, whether it is virtual or physical. Use the interface that appears using the `ifconfig` command.

Recommended network device to monitor.

Figure 6-4 is an example of an logical domain service group. The logical domain resource requires both network (NIC) and storage (Volume and DiskGroup) resources.

See the *Cluster Server Bundled Agents Reference Guide* for more information about the NIC agent.

Storage configurations

Depending on your storage configuration, use a combination of the Volume, DiskGroup, and Mount agents or Disk agent to monitor storage for logical domains.

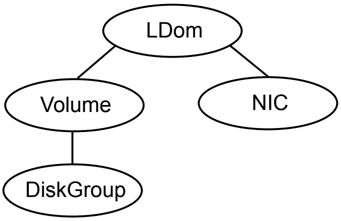
Note: Veritas recommends volumes or flat files in volumes that are managed by VxVM for LDom storage for configurations where VCS is in control domain.

Veritas Volume Manager exposed volumes

Veritas Volume Manager (VxVM) exposed volumes is the recommended storage solution for Cluster Server (VCS) in a control domain configuration. Use the Volume and DiskGroup agents to monitor a VxVM volume. VCS with VxVM provides superior protection for your highly available applications.

Figure 6-4 shows an logical domain resource that depends on a Volume and DiskGroup resource.

Figure 6-4 The logical domain resource can depend on resources such as NIC, Volume, and DiskGroup depending on the environment



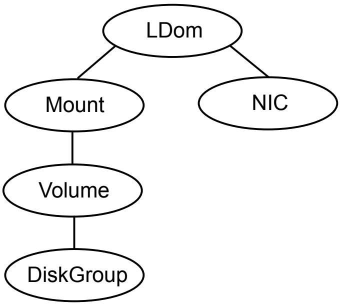
For more information about the Volume and DiskGroup agents, refer to the *Cluster Server Bundled Agents Reference Guide*.

Image files

Use the Mount, Volume, and DiskGroup agents to monitor an image file.

Figure 6-5 shows how the LDom resource depends on the resources with image file.

Figure 6-5 The LDom resource in conjunction with different storage resources



See the *Cluster Server Bundled Agents Reference Guide* for more information about the Mount agent.

Configuring logical domain

You must perform the following steps to configure logical domain.

To configure logical domain

1. Make sure that the network and storage setup meet the VCS requirements.

   See " Cluster Server requirements" on page 150.

2. Make sure that you have installed the required Solaris operating system in the logical domain.

3. Create logical domain (ldom1) on each system with an identical configuration and boot device.

Installing Cluster Server inside the control domain

You must install Cluster Server (VCS) in the control domain of each system.

To install VCS inside the control domain

◆ Install and configure VCS in the control domain of each system.

The process for installing VCS in the control domain is very similar to the regular installation of VCS. However, you must specify the host name of the control domain for the name of the host where you want to install VCS.

See the *Cluster Server Configuration and Upgrade Guide* for VCS installation and configuration instructions.

Creating the VCS service groups for logical domains

You can also create and manage service groups using the Veritas InfoScale Operations Manager, or through the command line.

See the *Cluster Server Administrator's Guide* for complete information about using and managing service groups.

Verifying a logical domain service group failover

Verify the configuration in different situations.

Using hagrp -switch command

Switch the logical domain service group to another node in the cluster to make sure the service group fails over. If all the resources are properly configured, the service group shuts down on the first node and comes up on the second node.

Other verification scenarios

In all of these verification scenarios, you are stopping or moving an logical domain, or stopping a resource for that logical domain. VCS should detect the failure, or the movement, and either failover the effected logical domain or take no action.

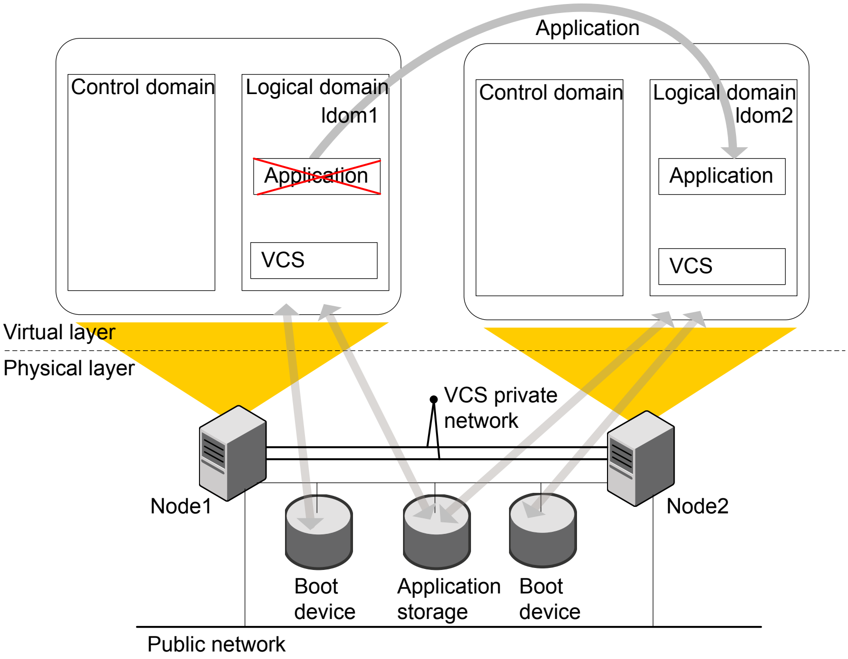
The following list presents some quick testing scenarios;

- From outside of VCS control, stop the logical domain. VCS should fail the logical domain over to the other node.
- Boot the logical domain through VCS by entering a `hagrp -online` command. move the logical domain to another node by shutting it down through VCS on the node where the logical domain is running. Boot the logical domain outside of VCS control on the other node- the service group comes online on that node.

Cluster Server setup to fail over an Application running inside logical domain on a failure of Application

Figure 6-6 illustrates a typical Cluster Server (VCS) setup to provide high availability for applications that run in guest domains.

Figure 6-6 Typical setup for applications high availability with Cluster Server installed in guest domains



A typical two-node VCS configuration that fails over the applications to keep the applications that run in logical domains highly available has the following infrastructure:

- Oracle VM Server for SPARC software is installed on each system Node1 and Node2.

- Shared storage is attached to each system.
- Logical domains are created on both the nodes that may have local boot devices.
- Each LDom has an operating system installed.
- VCS is installed in the guest domains of each node.

Workflow:

- Set up guest domain and install Solaris OS
- Install VCS on the guest domain
- Create Application service group in the guest domain

Configuring VCS to fail over an application on a failure

You must install and configure Cluster Server (VCS) in the guest domains of each system to enable VCS to manage applications in the guest domains.

To configure VCS to manage applications in the guest domains

1. Install and configure VCS in the guest domains of each system.

   See the *Cluster Server Configuration and Upgrade Guide* for installation and configuration instructions.

2. Create two virtual NICs using private virtual switches for private interconnects.

   You can configure virtual switches with no physical network interfaces if you want the failover across logical domains in the same control domain.

3. Configure VCS service group for the application that you want to monitor.

   - Configure Mount and Disk resources to monitor the storage.
   - Configure NIC resources to monitor the network.
   - Configure application resources using the application-specific agent.

   See the *Cluster Server Administrator's Guide* for more details on configuring applications and resources in VCS.

   See the *Cluster Server Bundled Agents Reference Guide* for details on the storage and networking bundled agents.

Oracle VM Server for SPARC guest domain migration in VCS environment

VCS supports cold, warm, and live migration, also known as domain migration, for Oracle VM Server for SPARC guest domains.

Domain migration enables you to migrate a guest domain from one host system to another host system. The system on which the migration is initiated is the source system. The system to which the domain is migrated is the target system.

While a migration operation is in progress, the domain that you want to migrate transfers from the source system to the migrated domain on the target system.

The domain migration until Oracle VM Server for SPARC 2.0 release is a warm migration. A warm migration is where the domain that you want to migrate enters a suspended state before the migration.

The Oracle VM Server for SPARC 2.1 introduces live migration, which provides performance improvements that enable an active domain to migrate while it continues to run.

In addition to live migration, you can migrate bound or inactive domains. This migration is a cold migration.

You can use domain migration to perform tasks such as the following:

- Balancing the load between systems
- Performing hardware maintenance while a guest domain continues to run

Overview of a warm migration

The Logical Domains Manager on the source system accepts the request to migrate a domain and establishes a secure network connection with the Logical Domains Manager running on the target system. Once this connection has been established, the migration occurs.

The migration operation occurs in the following phases:

Phase 1 After connecting with the Logical Domains Manager running in the target host, information about the source system and domain are transferred to the target host. The Logical Domains Manager on the target host uses this information to perform a series of checks to determine whether a migration is possible. The checks differ depending on the state of the source domain. For example, if the source domain is active the Logical Domains Manger performs a different set of checks than if the domain is bound or inactive.

Phase 2 When all checks in Phase 1 have passed, the source and target systems prepare for the migration. The Logical Domains Manager on the source suspends and the source domain. On the target system, the Logical Domains Manager creates a domain to receive the source domain.

Phase 3 For an active domain, the next phase is to transfer all the runtime state information for the domain to the target. The Logical Domains Manager retrieves this information from the hypervisor. On the target, the Logical Domains Manager installs the state information in the hypervisor.

Phase 4 Handoff—after all state information is transferred, the handoff occurs when the target domain resumes execution (if the source was active). The Logical Domain Manager on the source destroys the source domain. From this point on, the target domain is the sole version of the domain running.

Overview of a live migration

The Logical Domains Manager on the source system accepts the request to migrate a domain and establishes a secure network connection with the Logical Domains Manager that runs on the target system. The migration occurs after this connection has been established.

The migration operation occurs in the following phases:

Phase 1 After the source system connects with the Logical Domains Manager that runs in the target system, the Logical Domains Manager transfers information about the source system and the domain to be migrated to the target system. The Logical Domains Manager uses this information to perform a series of checks to determine whether a migration is possible. The Logical Domains Manager performs state-sensitive checks on the domain that you plan to migrate. The checks it performs is different for an active domain than for bound or inactive ones.

Phase 2 When all checks in Phase 1 have passed, the source and target systems prepare for the migration. On the target system, the Logical Domains Manager creates a domain to receive the domain. If the domain that you plant to migrate is inactive or bound, the migration operation proceeds to Phase 5.

Phase 3 If the domain that you want to migrate is active, its run-time state information is transferred to the target system. The domain continues to run, and the Logical Domains Manager simultaneously tracks the modifications that the operating system makes to this domain. The Logical Domains Manager on the source retrieves this information on the source from the source hypervisor and sends the information to the Logical Domains Manager on the target. The Logical Domains Manager on the target installs this information in the hypervisor for the target.

Phase 4 The Logical Domains Manager suspends the domain that you want to migrate. At this time, all of the remaining modified state information is re-copied to the target system. In this way, there should be little or no perceivable interruption to the domain. The amount of interruption depends on the workload.

Phase 5	A handoff occurs from the Logical Domains Manager on the source system to the Logical Domains Manager on the target system. The handoff occurs when the migrated domain resumes execution (if the domain to be migrated was active), and the domain on the source system is destroyed. From this point forward, the migrated domain is the sole version of the domain running.

With Oracle VM Server for SPARC 2.1, the default domain migration attempted is Live Migration. If the installed version of Oracle VM Server for SPARC is 2.0, the domain migration defaults to warm migration. For more details on supported configurations, see Migrating Domains in the *Oracle® VM Server for SPARC Administration Guide*.

Cluster Server (VCS) provides the following support for migration of Oracle VM Server for SPARC guest domains:

See “VCS initiated migration of service groups that have LDom resource configured to monitor Oracle VM for SPARC guest domains” on page 163.

See “User initiated migration of Oracle VM guest domains managed by VCS” on page 165.

For migration of guest domains, ensure each virtual disk back end that is used in the guest domain to be migrated is defined on the target machine. The virtual disk back end that is defined must have the same volume and service names as on the source machine. Similarly each virtual network device in the domain to be migrated must have a corresponding virtual network switch on the target machine. Each virtual network switch must have the same name as the virtual network switch to which the device is attached on the source machine. For complete list of migration requirements for a guest domain, refer to Administration Guide for appropriate Oracle VM for SPARC version that you are using.

Note: If CVM is configured inside the logical domain which is planned for migration, perform this step:

Set the value of the LLT peerinact parameter to sufficiently high value on all nodes in the cluster. You set the value to very high value so that while the logical domain is in migration, the system is not thrown out of the cluster by the other members in the cluster.

If the CVM stack is unconfigured, the applications can stop.

See the *Cluster Server Administrator’s Guide* for LLT tunable parameter configuration instructions.

Note: If the control domain exports the FSS volumes to a guest domain, live migration can be performed even if the storage is not physically connected to the hosts of the source system and the target system for migration.

See “Provisioning storage to guests with Flexible Storage Sharing volumes of control domain” on page 146.

VCS initiated migration of service groups that have LDom resource configured to monitor Oracle VM for SPARC guest domains

This section describes how to initiate migration of service groups by Cluster Server (VCS).

VCS contains the “migrate” operation for initiating service group migration. The LDom agent contains the “migrate” entry point to initiate migration of guest domains.

For details of supported service group configurations with VCS initiated migration refer to the *Cluster Server Administrator’s Guide* on Solaris.

To initiate migration of service groups by VCS

1. The guest domain must meet domain migration requirements for memory, CPU, firmware version, OVM version, etc.

   For details on domain migration requirements, refer to *Oracle VM server for SPARC* documentation.

2. Migrate a service group, enter:

   ```
   # hagrp -migrate service_group -to sys_name
   ```

3 The UserName and Password attributes of the LDom agent are mandatory attributes for supporting VCS initiated migration of service groups. A sample configuration for migration of a service group:

```
group sg1 (
        SystemList = { sys1 = 0, sys2 = 1 }
        )

        LDom ldmres1 (
                LDomName = ldm1
                CfgFile = "/etc/ldm1.xml"
                NumCPU = 16
                Memory = 8G
                Password = aabbccddee
                UserName = user1
                )
```

Refer to *Oracle* documentation for steps to authorize non-root users to issue commands to the logical domain manager.

4 The guest domain migration command uses physical system name, if VCS node name is different from physical system name, you must configure PhysicalServer attribute (system level) of VCS with the physical system name on each VCS node. On each VCS node, enter:

```
# haconf -makerw
# hasys -modify node_name PhysicalServer node_name
```

Oracle VM server for SPARC does not support migration of more than one guest domain on a system at a time either as source of migration or as target for migration. Attempting VCS initiated migration with more than one service group between two systems at a time is not supported.

Configuring more than one LDom resource in a service group is also not supported for VCS initiated migration.

Troubleshooting migration-related issues

Service group remains in ONLINE|STOPPING|MIGRATING in source node and OFFLINE|STARTING|MIGRATING in target node when migration of a service group is attempted from node A to node B and migration of another service group is attempted from node B to node A simultaneously

With older versions of OVM for SPARC software, when migration of two guests are attempted between two systems in opposite direction, it is observed that both the migration attempts fails and the `ldm` command hangs. This causes LDom resources to go into ADMIN WAIT state and also the monitor function times out

To resolve ldm command hang issue:

1 restart the ldmd service:

```
# svcadm restart svc:/ldoms/ldmd:default
```

2 To clear service group state:

```
# hagrp -flush -force service_group -sys system
```

User initiated migration of Oracle VM guest domains managed by VCS

VCS supports the following three varieties of Oracle VM Server for SPARC domain migration:

- Guest migration from one VCS node to other VCS node in the cluster, for example:

  ```
  sys1# ldm migrate ldom1 sys2
  ```

- Guest migration from a VCS node to a non-VCS node, for example:

  ```
  sys1# ldm migrate ldom1 sys3
  ```

- Renaming the logical domain during the migration to the target system, for example:

  ```
  sys1# ldm migrate ldom1 sys2:ldom2
  ```

Prerequisites before you perform domain migration

Perform the following prerequisites before you perform a domain migration:

- Verify that the value of IntentionalOffline attribute for LDom type is 1. Note that 1 is the default value.
- Make sure that the LDom resource for the LDom that you plan to migrate is in an ONLINE or OFFLINE steady state.
- To rename the logical domain when you migrate it, make sure that the LDomName attribute for the LDom resource in VCS is localized with the target LDom name for the target node. When you rename it, VCS can continue to monitor the LDom after migration.

- Make sure that CfgFile attribute for LDom is configured before migration.
- Make sure that RemoveLDomConfigForMigration attribute is set before migration. If this attribute is set, the LDom Agent removes the LDom configuration from the system on which an offline or clean is called. This arrangement helps in the scenario when a LDom is failed-over to a target node and is migrated back to the source node. In the presence of LDom configuration on the source node, migration would not be possible. Refer to the LDom Agent attribute description in the *Bundled Agents Reference Guide* for Solaris for more information.

Supported deployment models for Oracle VM Server for SPARC domain migration with VCS

The following are the supported deployment models for Oracle VM Server for SPARC domain migration with VCS:

- See "Migrating Oracle VM guest when VCS is installed in the control domain that manages the guest domain" on page 166.
- See "Migrating Oracle VM guest when VCS is installed in the control domain and single-node VCS is installed inside the guest domain to monitor applications inside the guest domain" on page 167.
- See "Migrating Oracle VM guest when VCS cluster is installed in the guest domains to manage applications for Oracle VM Server for SPARC version 2.1 and above" on page 169.
- See "Migrating Oracle VM guest when VCS cluster is installed in the guest domains to manage applications for Oracle VM Server for SPARC version 2.0" on page 169.

Migrating Oracle VM guest when VCS is installed in the control domain that manages the guest domain

Use the following information to migrate when you have VCS installed in the control domain that manages the guest domain.

To perform a migration of an LDom when VCS is installed in the control domain

◆ Use the `ldm` command for migration.

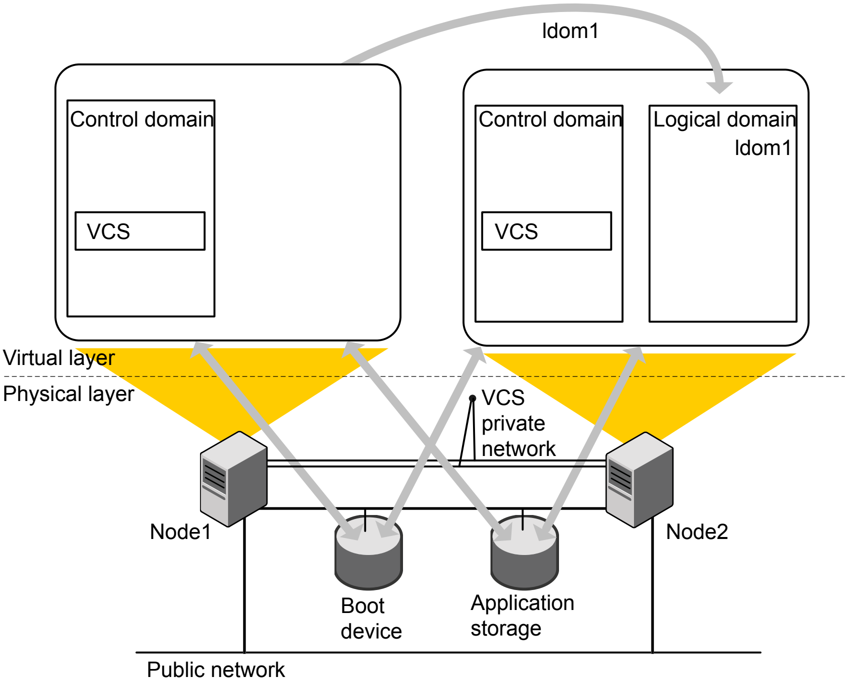
```
ldm migrate [-f] [-n] [-p password_file] source_ldom \
[user@target_host[:target_ldom]
```

For example:

```
Sys1# ldm migrate ldom1 Sys2
```

Figure 6-7 illustrates a logical domain migration when VCS is clustered between control domains.

Figure 6-7 Logical domain migration when VCS is clustered between control domains



Migrating Oracle VM guest when VCS is installed in the control domain and single-node VCS is installed inside the guest domain to monitor applications inside the guest domain

Use the following information to migrate when you have:

- VCS installed in the control domain
- The VCS in control domain manages the application in the guest domain, and
- Single-node VCS installed in the guest domain monitors the application in guest domain.

To perform a migration of the LDom when VCS is installed in the control domain that manages the applications in the guest domains

- Use the `ldm` command for migration.

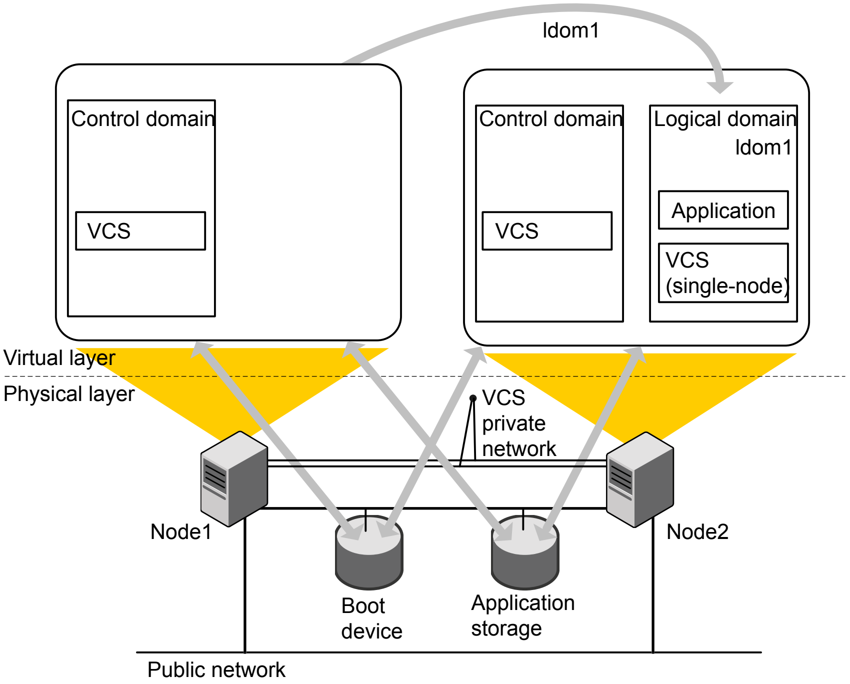
  ```
  ldm migrate [-f] [-n] [-p password_file] source_ldom \
  [user@target_host[:target_ldom]
  ```

  For example:

  ```
  Sys1# ldm migrate ldom1 Sys2
  ```

Figure 6-7 illustrates a logical domain migration when VCS is clustered between control domains and single-node VCS in the guest domain monitors applications.

Figure 6-8 Logical domain migration when VCS is clustered between control domains and single-node VCS in the guest domain monitors applications



Migrating Oracle VM guest when VCS cluster is installed in the guest domains to manage applications for Oracle VM Server for SPARC version 2.1 and above

Perform one of the following procedures when you want a migration of guest domain when VCS cluster is configured between guest domains.

To perform a migration of the LDom when VCS is installed in the guest domain that manages the applications in the guest domains for Oracle VM Server for SPARC version 2.1 and above

◆ Use the `ldm` command for migration.

```
ldm migrate [-f] [-n] [-p password_file] source_ldom \
[user@target_host[:target_ldom]
```

For example:

```
Sys1# ldm migrate ldom1 Sys2
```

Note: If CVM is configured inside the logical domain, perform this step:

Set the value of the LLT peerinact parameter to sufficiently high value on all nodes in the cluster. You set the value to very high value so that while the logical domain is in migration, the system is not thrown out of the cluster by the other members in the cluster.

If the CVM stack is unconfigured, the applications can stop.

See the *Cluster Server Administrator's Guide* for LLT tunable parameter configuration instructions.

Migrating Oracle VM guest when VCS cluster is installed in the guest domains to manage applications for Oracle VM Server for SPARC version 2.0

The domain migration is a warm migration.

Note: You do not have to start and stop LLT and GAB. In a warm migration, LLT and GAB restart themselves gracefully.

To perform a domain migration for an LDom when VCS is installed in guest domains

1. Stop VCS engine. Use the `hastop -local -force` command on the system that has the logical domain that you plan to migrate. Perform this step so that GAB does not have to kill the Cluster Server (VCS) engine process when the migration is complete. GAB wants all clients to reconfigure and restart when the configuration is not in sync with other members in the cluster.

2. If CVM is configured inside the logical domain, perform this step. Set the value of the LLT peerinact parameter to sufficiently high value on all nodes in the cluster. You set the value to very high value so that while the logical domain is in migration, the system is not thrown out of the cluster by the other members in the cluster. If the CVM stack is unconfigured, the applications can stop.

   See the *Cluster Server Administrator's Guide* for LLT tunable parameter configuration instructions.

3. If fencing is configured in single instance mode inside the logical domain, perform this step. Unconfigure and unload the vxfen module in the logical domain. Perform this step so that GAB does not panic the node when the logical domain migration is complete.

4. Migrate the logical domain from the control domain using the ldm interface. Wait for migration to complete.

   ```
   ldm migrate [-f] [-n] [-p password_file] source_ldom \
   [user@target_host[:target_ldom]
   ```

   For example:

   ```
   Sys1# ldm migrate ldom1 Sys2
   ```

5. Perform this step if you performed step 3. Load and configure vxfen module in the logical domain. See the Cluster Server Administrator's Guide for information about I/O fencing and its administration.

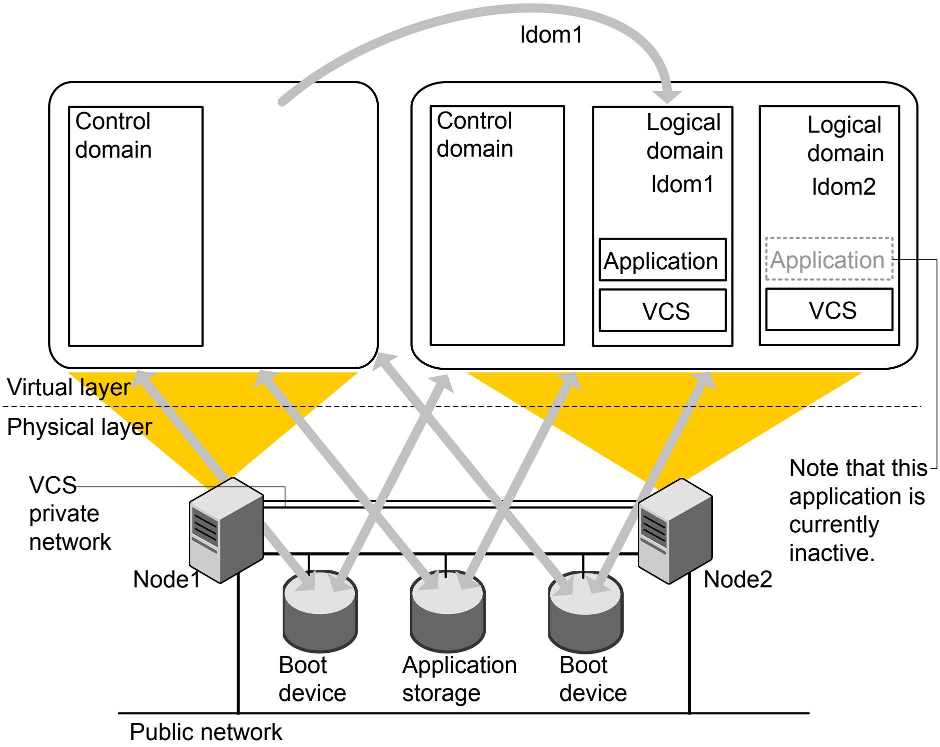
6. Perform this step if you performed step 2. Reset the value of the LLT peerinact parameter to its original value on all nodes in the cluster.

   See the *Cluster Server Administrator's Guide* for LLT tunable parameter configuration instructions.

7. Use the `hastart` command to start VCS engine inside the logical domain.

Figure 6-9 illustrates a logical domain migration when VCS is clustered between control domains and single-node VCS in the guest domain monitors applications.

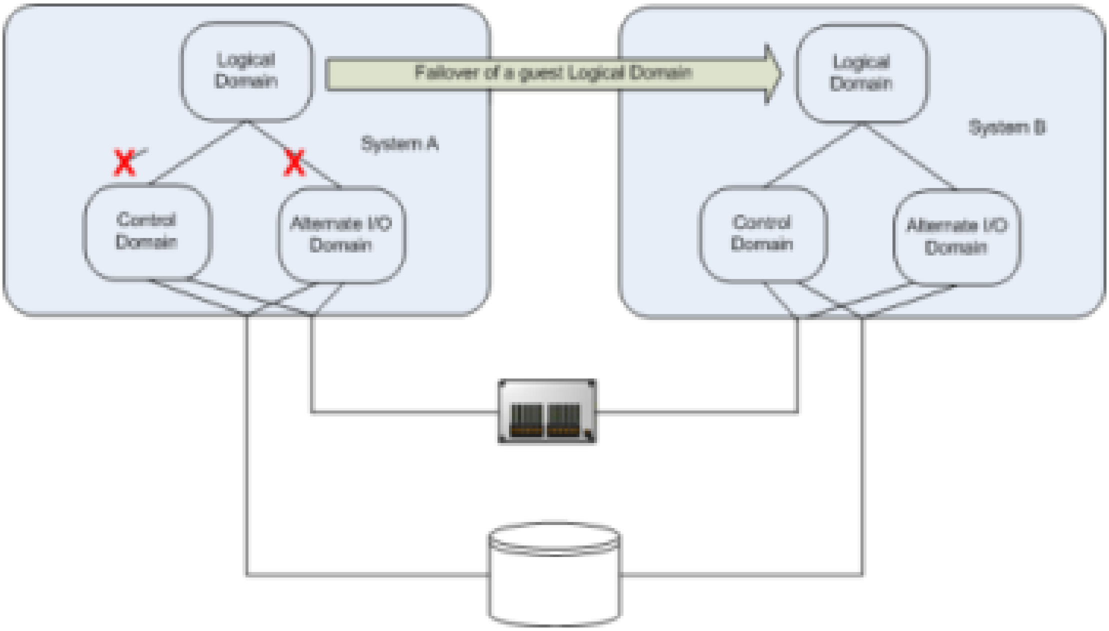
Figure 6-9 The logical domain migration when VCS is clustered between guest domains



About configuring VCS for Oracle VM Server for SPARC with multiple I/O domains

With Oracle VM Server for SPARC virtualization technology, you can create multiple I/O domains (control domain and alternate I/O domain) to provide redundant storage and network services to a guest Logical Domain. A typical cluster-setup configured to use multiple I/O domains has two physical systems. On each physical system, the control domain and alternate I/O domain provide I/O services from back-end storage and network devices to a Logical Domain.

Figure 6-10 with guest Logical Domain on System A and System B using storage and network services from the control domain and alternate I/O domain



If there is a failure of storage or network services from one of the domains, the guest Logical Domain continues to function on the same physical system because it gets I/O services from the other I/O domain. However, when there is failure of services from both the I/O domains on a physical system, the Logical Domain on the physical system fails.

Configure Cluster Server (VCS) on multiple I/O domains to manage a Logical Domain. VCS fails over the Logical Domain from one system to a Logical Domain on another system when services from both the domains fail.

Note: Failover happens when I/O services from all I/O domains fail or the control domain goes down.

About Alternate I/O domain

Alternate I/O domain, is an Oracle technology available on Oracle VM server for SPARC, which provides highly available storage and network services to guest domains on a physical system.

Setting up the Alternate I/O domain

While setting up a system to support Logical Domain, the control domain (primary domain) owns all the I/O devices on the physical system. To create alternate I/O domain, you need to relinquish ownership of one of the PCI Express bus from control domain and assign it to a Logical Domain.

For more information on creating alternate I/O domain, refer to the Oracle Solaris documentation.

Configuring VCS to manage a Logical Domain with multiple I/O domains

Proceed to configure Cluster Server (VCS). See "Configuring VCS to manage a Logical Domain using services from multiple I/O domains" on page 173.

Configuring VCS to manage a Logical Domain using services from multiple I/O domains

VCS provides high availability to Logical Domains using I/O services from multiple I/O domains. When I/O services from the control domain and alternate I/O domain fail, VCS fails over the LDom from one system to another system. LDom continues to be functional on the same system and does not need a fail over if one of the I/O domain continues to provide service.

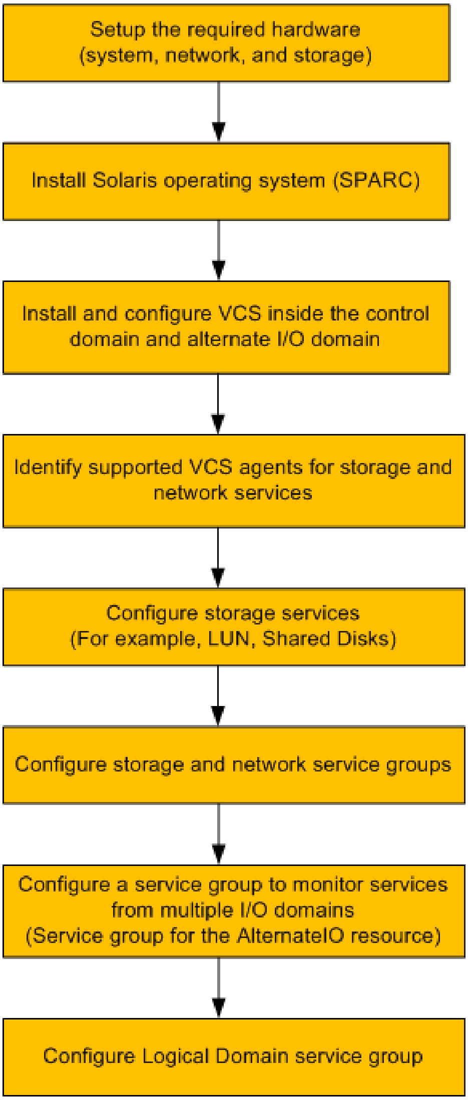
VCS uses service groups and resources to manage the storage and network services that are provided to a Logical Domain. These service groups are monitored by the AlternateIO resource. The AlternateIO service group provides the information about the state of the storage and network services to the LDom agent. VCS fails over the Logical Domain when services from both the I/O domains fail.

Perform the following tasks to configure VCS to manage a Logical Domain:

- Identify supported storage and network services
- Determine the number of nodes to form VCS cluster
- Install and configure VCS inside the control domain and alternate I/O domain
- Configuring storage services
- Configure storage service groups
- Configure network service groups
- Configure a service group to monitor services from multiple I/O domains
- Configure the AlternateIO resource

- Configure the service group for a Logical Domain

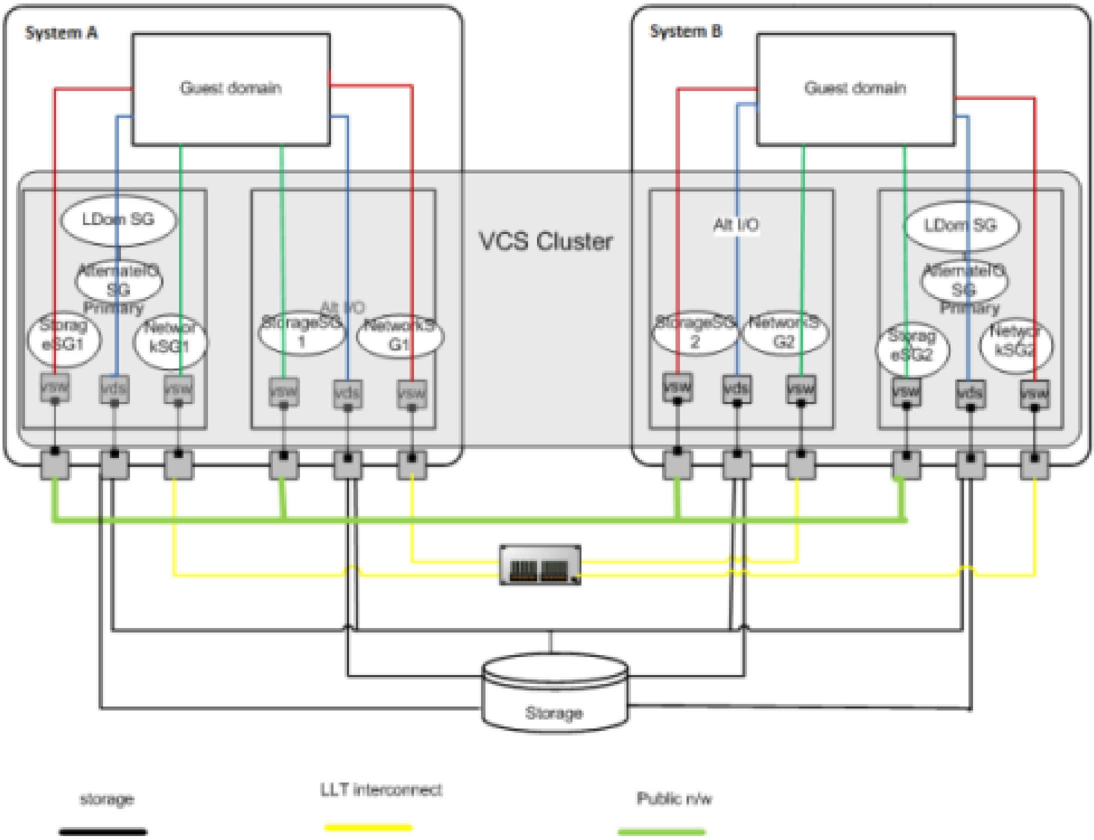
Figure 6-11 Workflow to configure VCS on a physical system to manage a Logical Domain


Setup the required hardware (system, network, and storage)
Install Solaris operating system (SPARC)
Install and configure VCS inside the control domain and alternate I/O domain
Identify supported VCS agents for storage and network services
Configure storage services (For example, LUN, Shared Disks)
Configure storage and network service groups
Configure a service group to monitor services from multiple I/O domains (Service group for the AlternateIO resource)
Configure Logical Domain service group

A typical setup for a Logical Domain with multiple I/O services

Guest logical domain using I/O services from multiple I/O domains.

Figure 6-12 shows VCS configuration to monitor a logical domain using I/O services from multiple I/O domains



System A, System B - T5440 servers

LDom SG - Logical Domain service group

AlternateIO SG - AlternateIO service group

Storage SG - Storage service group

Network SG - Network service group

Identify supported storage and network services

The following back-end storage and network services can be exported to Logical Domains.

I/O services	Back-end device	VCS agents to be used
Storage	LUN, shared disk	Disk
	Flat file	Mount
	zpool	Zpool
	Veritas CVM volume	CVMVolDG
Network	NIC	NIC

Determine the number of nodes to form VCS cluster

The total number of nodes that form a VCS cluster depends on the number of physical systems times the the control domain and alternate I/O domain on each physical system.

For example, if you have two physical systems, each having a control domain and an alternate I/O domain, you need to configure VCS as a four node cluster.

Install and configure VCS inside the control domain and alternate I/O domain

Install and configure VCS inside the control domain and alternate I/O domain

For more details, refer to the *Cluster Server Configuration and Upgrade Guide*.

Configuring storage services

Depending upon the storage service, the configuration procedure differs.

- LUN, Shared Disk, or CVM volume
  See "Configuring virtual disk with LUN, Shared Disk, or CVM Volume" on page 177.
- ZFS volume
  See "Configuring storage services when back-end device is a ZFS volume" on page 179.
- Flat file
  For more information, refer to the LDom Administrator's guide.
- Zpool
  For more information, refer to the LDom Administrator's guide.

Configuring virtual disk with LUN, Shared Disk, or CVM Volume

Use the following procedure to configure virtual disk with LUN, Shared Disk, or CVM Volume:

To configure virtual disk

1. Add the physical disk back-end path of the disk service to the primary domain.

   ```
   # ldm add-vdsdev backend_path1 volume@primary-vds0
   ```

   where *backend_path1* is the path to the virtual disk back-end from the primary domain.

2. Add the physical disk back-end path of the disk service to the alternate I/O domain for disk added in step 1.

   ```
   # ldm add-vdsdev backend_path2 volume@alternate-vds0
   ```

   where *backend_path2* is the path to the virtual disk back-end from the alternate I/O domain.

3. Export the virtual disk to the guest domain.

   ```
   # ldm add-vdisk disk_name volume@primary-vds0 ldom_name
   ```

   ```
   # ldm add-vdisk disk_name volume@alternate-vds0 ldom_name
   ```

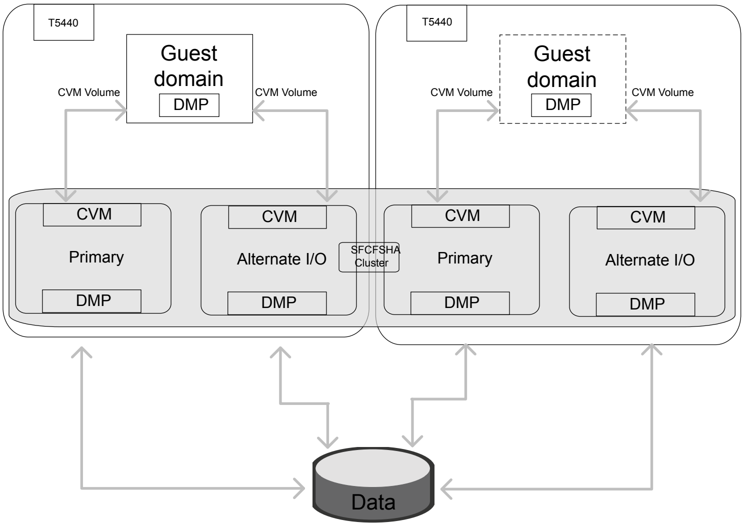
   where *disk_name* is the name of the virtual storage device.

   where *ldom_name* is the name of the Logical Domain.

   Note: Do not set the value of the **Options** attribute to exclusive; **excl**. If set to exclusive, Logical Domain cannot use the multipathing functionality.

   For more details on configuring virtual disk multipathing, refer to the *Oracle VM server for SPARC Administration Guide*.

Figure 6-13 shows a sample diagram for CVM Volume

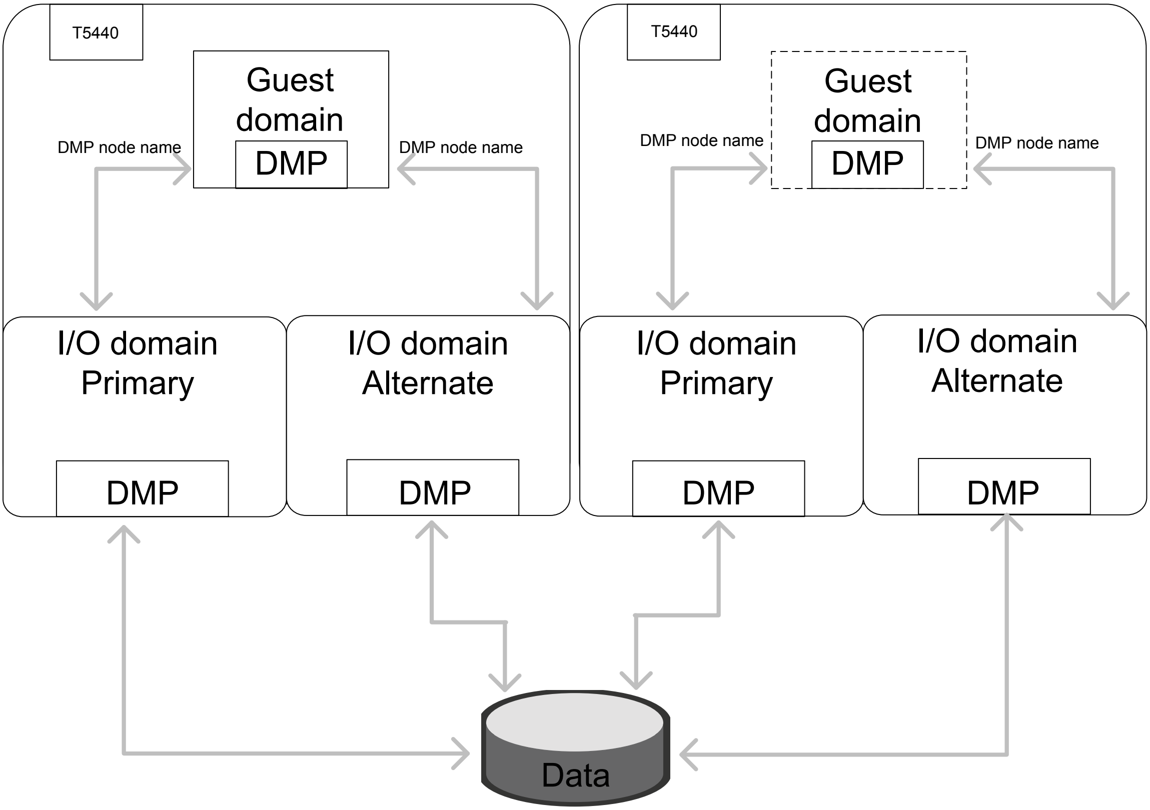


Virtual disk configurations with DMP

If the disk exported to guest domain has more than one I/O path from each I/O domain, install Dynamic Multi-Pathing (DMP) in each I/O domain. Use the DMP node name as back-end device to configure the virtual disk exported to the guest domain.

You must use DMP at guest domain for path management.

Figure 6-14 shows how layered DMP works together



Configuring storage services when back-end device is a ZFS volume

If you export ZFS volume as a back-end storage to Logical Domain, you need to

1. Export ZFS volume created in the control domain.
2. Export ZFS volume created in the alternate I/O domain.

   Note: Ensure the ZFS volume size created in both domains matches.

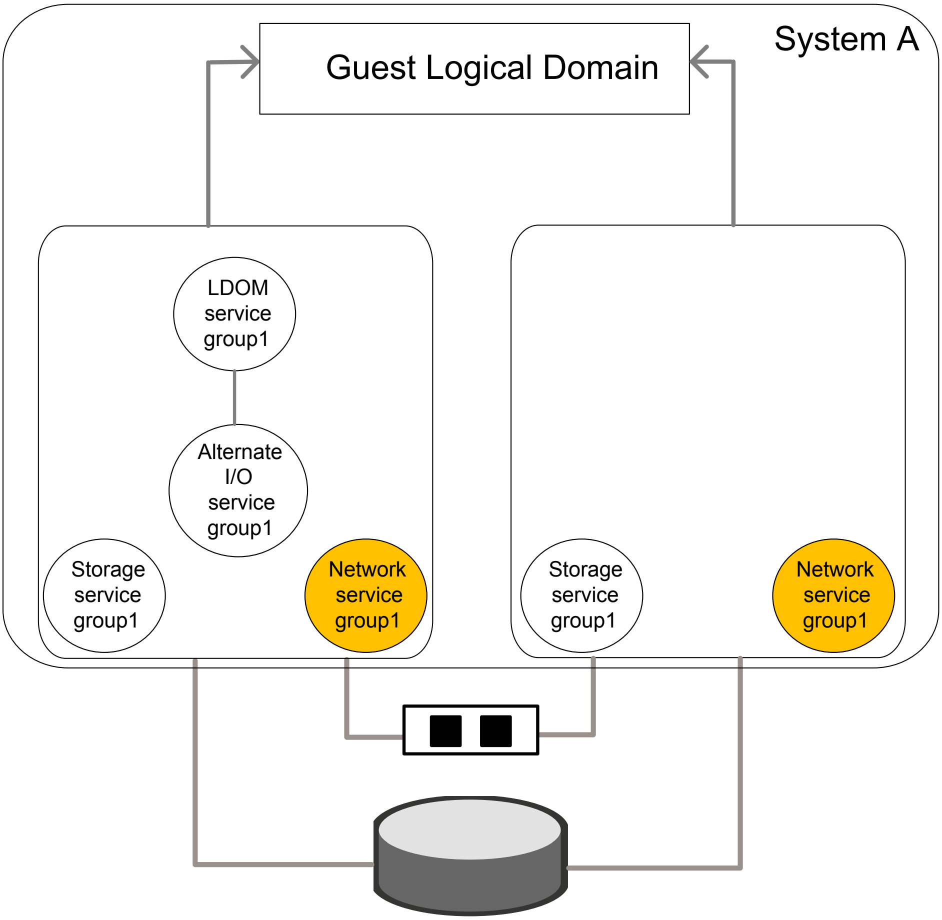
3. Create ZFS root pool mirror inside the Logical Domain from the volumes exported from control domain and alternate I/O domain.

Configure storage service groups

VCS agents manage storage services that are made available to a guest Logical Domain. Depending on the back-end storage device, use the appropriate VCS agent. For more information on supported VCS agents, see Identify supported storage and network services.

Note: You must configure a storage service group on each physical system in the cluster.

Figure 6-15 shows storage service groups in the control domain and alternate I/O domain on a physical system



Configuration parameter	Description

Localize resource attribute value	You may need to localize VCS resource attributes depending on the type of back-end storage device.
	For example, for Disk resource, if the back-end storage paths from the control and alternate I/O domains are different, you need to localize the partition attribute.
	`Disk disk1`
	` (`
	`   Partition @primary = \`
	`   "/dev/rdsk/c3t50060E8000C46C50d2s2"`
	`   Partition @alternate = \`
	`   "/dev/rdsk/c1t50060E8000C46C50d2s2"`
	` )`

Service group type	Service groups that manage storage services in the control domain and alternate I/O domain must be configured as a parallel service group.
Configure the SystemList attribute	Modify the SystemList attribute of the service group to add hostnames of the control domain and alternate I/O domain configured on the physical system.
Configure Phantom resource	If all the resources are of the type Disk, configure a Phantom resource. The Disk resource is of the type OnOnly and does not contribute to determine the state of the service group. The Phantom resource enables VCS to determine the state of parallel service groups that do not include OnOff resources. For more information on the Phantom agent, refer to the Cluster Server Bundled Agents Reference Guide.

An example of storage service group configuration from `main.cf` configuration (for a setup that has two physical systems)

Control domain host names – primary1, primary2

Alternate domain host names – alternate1, alternate2

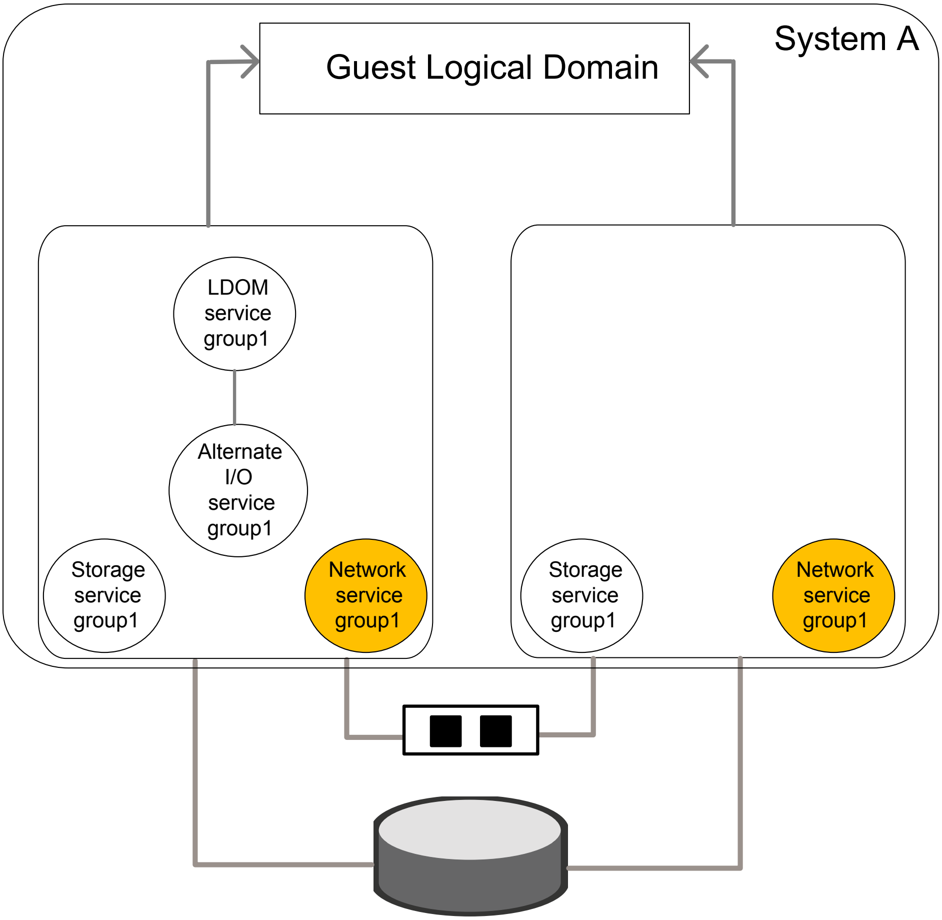
```
group primary1-strsg (
        SystemList = { primary1 = 0, alternate1 = 1 }
        AutoStartList = { primary1, alternate1 }
        Parallel = 1
        )

        Disk disk1
        (
        Partition @primary1 = "/dev/rdsk/c3t50060E8000C46C50d2s2"
        Partition @alternate1 = "/dev/rdsk/c1t50060E8000C46C50d2s2"
        )

        Phantom ph1 (
                )
group primary2-strsg (
        SystemList = { primary2 = 0, alternate2 = 1 }
        AutoStartList = { primary2, alternate2 }
        Parallel = 1
        )

        Disk disk2
        (
        Partition @primary2 = "/dev/rdsk/c3t50060E8000C46C50d2s2"
        Partition @alternate2 = "/dev/rdsk/c1t50060E8000C46C50d2s2"
        )

        Phantom ph2 (
                )
```

Configure network service groups

VCS agents manage network resources that are made available to a guest Logical Domain. Depending on the back-end storage device, use appropriate VCS agent. For more information, see Identify supported storage and network services.

Note: You must configure a network service group on each physical system in the cluster.

Figure 6-16 shows network service groups in the control domain and alternate I/O domain



Perform the configuration steps for the network service group on each physical system.

Configuration parameter	Description
Localize network resource attribute	You may need to localize VCS resources depending on the back-end network device. For example, for disk agent, if the network device exported from control and alternate I/O domain are different, you need to localize the Device attribute. `NIC primary1-network ( Device @primary = nxge3 Device @alternate = nxge4 )`
Service group type	Service groups that manage network services in the control domain and alternate I/O domain must be configured as a parallel service group.
Configure the SystemList attribute	Modify the SystemList attribute in the service group to add host names of the control domain and alternate I/O domain configured on the physical system.
Configure Phantom resource	If all the resources are of the type NIC, configure a Phantom resource. The NIC resource is of the type OnOnly and does not contribute to determine the state of the service group. The Phantom resource enables VCS to determine the state of parallel service groups that do not include OnOff resources. For more information on the Phantom agent, refer to the Cluster Server Bundled Agents Reference Guide.

An example of network service group configuration from `main.cf` (for a setup that has two physical systems)

Control domain host names – primary1, primary2

Alternate domain host names – alternate1, alternate2

```
group primary1-nwsg (
        SystemList = { primary1 = 0, alternate1 = 1 }
        AutoStartList = { primary1, alternate1 }
        Parallel = 1
        )

        NIC nicres1 (
                Device @primary1 = nxge3
                Device @alternate1 = nxge1
                )

        Phantom ph3 (
                )
group primary2-nwsg (
        SystemList = { primary2= 0, alternate2 = 1 }
        AutoStartList = { primary2, alternate2 }
        Parallel = 1
        )

        NIC  nicres2(
                Device @primary2= nxge3
                Device @alternate2 = nxge1
                )

        Phantom ph4 (
                )
```

Configure a service group to monitor services from multiple I/O domains

Configure a service group for the AlternateIO resource to monitor storage and network services that are exported from back-end devices to a Logical Domain.

Configuration notes for the service group:

- Configure the service group as a parallel or a failover service group. See, Type of service group configuration for the AlternateIO resource.
- If multiple storage services are exported to a Logical Domain, you can configure separate service groups to monitor each of the storage services. For example, you can configure separate service groups to monitor LUN and ZFS volume storage services.
- The SystemList attribute of the service group must contain only host names of the control domains present on each physical system in the cluster.
- Localize the StorageSG attribute of the service group with the list of the storage service groups configured for each node.
- Enable preonline trigger for a fail over type service group

  ```
  # hagrp -modify aiosg TriggerPath bin/AlternateIO
  ```

  where aiosg is the name of the service group

```
# hagrp -modify aiosg TriggersEnabled PREONLINE
```

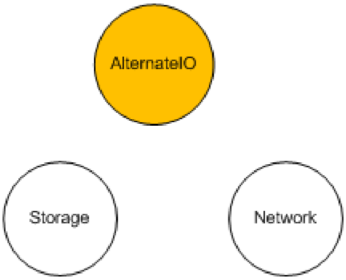
Type of service group configuration for the AlternateIO resource

Service group type	Condition
Parallel	If storage services are simultaneously accessible to all nodes in the cluster, the service group of the AlternateIO resource must be configured as a parallel service group. For example, shared LUNs, shared Disks, CVM Volume.
Fail over	If storage services must be accessible only on one physical system (control domain and alternate I/O domain) in the cluster, configure the service group of the AlternateIO resource as a failover service group. For example, zpool.

Configure the AlternateIO resource

The AlternateIO resource monitors storage and network services that are exported to a guest Logical Domain. The AlternateIO resource is not dependent on storage or network resources. However, its state depends upon the state of storage or network service groups.

Figure 6-17 shows that the AlternateIO resource does not have any dependency on storage or network resources.



Configuration parameter | **Description**

StorageSG attribute

This attribute is a key value pair. A storage service group is the key and the value of the key can either be 0 or 1.

Set the value of the key to 1 to bring the service group online when the AlternateIo resource comes online and to take the service group offline when the AlternateIo resource goes offline.

Localize the StorageSG attribute of the AlternateIO resource with the list of storage service groups that are configured on each node.

```
AlternateIO altiores1
 (
 StorageSG @primary1 = { primary1-strsg1 = 1 }
 StorageSG @primary2 = { primary2-strsg1 = 1 }
  )
```

NetworkSG attribute

This attribute is a key value pair. A network service group is the key and the value of the key can either be 0 or 1.

Set the value of the key to 1 to bring the service group online when the AlternateIo resource comes online and to take the service group offline when the AlternateIo resource goes offline.

Localize the NetworkSG attribute of the AlternateIO resource with the list of network service groups that are configured on each node.

```
AlternateIO altiores1
 (
 NetworkSG @primary1 = { primary1-nwsg = 0 }
 NetworkSG @primary2 = { primary2-nwsg = 0 }
 )
```

Preonline trigger

For any of the service groups configured in the StorageSG or NetworkSG attributes, if you set the value to 1, configure the preonline trigger at the service group level.

Configuring the preonline trigger ensures that service groups listed in the StorageSG attributes are offline on all systems except onto the system where failover or a manual switch over is initiated.

For information on enabling the preonline trigger, see Configure service group to monitor services from multiple I/O domains.

Sample service group configuration for the AlternateIO resource

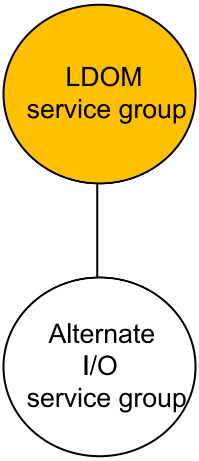
Assumption – Storage and network service groups are of the type parallel.

```
group aiosg (
        SystemList = { primary1 = 0, primary2 = 1 }
        AutoStartList = { primary1, primary2 }
        Parallel = 1
         )

        AlternateIO aiores1 (
         StorageSG @primary1 = { primary1-strsg = 0 }
         StorageSG @primary2 = { primary2-strsg = 0 }
         NetworkSG @primary1 = { primary1-nwsg = 0 }
         NetworkSG @primary2 = { primary2-nwsg = 0 }
          )
```

Configure the service group for a Logical Domain

VCS uses the LDOM agent to manage a guest logical domain. The Logical Domain resource has an online local hard dependency on the AlternateIO resource.

Figure 6-18 shows the dependency of LDom service group on the AlternateIO service group



Configuration notes:

- Configure the service group as a fail over type service group.

- The SystemList attribute in the LDOM service group must contain only host names of the control domains from each physical system in the cluster.
- The LDOM service group must have online local hard dependency with the AlternateIO service group.

If the guest domain needs to be made available even when the primary domain is rebooted or shut down for planned maintenance.

To make the guest domain available

1. Set the LDOM resource attribute DomainFailurePolicy to { primary=ignore, alternate1=stop } for all the LDOM resources in the cluster which are critical and needs to be available during primary/control domain maintenance. This setting ensure that guest domain will not be brought down when primary/control domain is taken down for planned maintenance.

   ```
   # hares -modify DomainFailurePolicy ldmres primary ignore \
   alternate1 stop
   ```

2. Set the LDOM service group attribute SysDownPolicy to AutoDisableNoOffline This setting ensures that VCS will not fail-over the service group even when the primary/control domain where the service group is online is taken down.

   ```
   # hagrp -modify ldmsg SysDownPolicy AutoDisableNoOffline
   ```

3. The service group will be auto-disabled in the cluster when the control domain is taken down for maintenance. Once the control domain is brought online again, clear the auto disabled system by executing the following command:

   ```
   # hagrp -autoenable ldmsg -sys primary1
   ```

4. Once the maintenance for the control domain is completed, set the DomainFailurePolicy attribute to it’s original values (default: {primary = stop}). Also reset the service group attribute SysDownPolicy:

   ```
   # hares -modify ldmres DomainFailurePolicy primary stop
   # hagrp -modify ldmsg SysDownPolicy -delete AutoDisableNoOffline
   ```

Sample configuration for LDom service group

The LDom service group must have online local hard dependency with the AlternateIO service group.

```
group ldmsg (
        SystemList = { primary1 = 0, primary2 = 1 }
```

```
AutoStartList = { primary1, primary2 }
SysDownPolicy = { AutoDisableNoOffline }
)

LDom ldmres (
        LDomName = ldg1
        DomainFailurePolicy={ primary=ignore, alternate1="stop"}
        )
```

Failover scenarios

Scenario	Control domain	Alternate I/O domain	VCS behavior
State of each storage service group	Online	Online	No fail over
	Offline/FAULT	Online	No fail over
	Online	Offline/FAULT	No fail over
	Offline/FAULT	Offline/FAULT	Fail over
State of each network service group	Online	Online	No fail over
	Offline/FAULT	Online	No fail over
	Online	Offline/FAULT	No fail over
	Offline/FAULT	Offline/FAULT	Fail over
Domain state	Up	Up	No fail over
	Up	down	No fail over
	Down	Up	Fail over *
	Down	down	Fail over **

***** VCS behavior would be “No fail over” with service group in auto-disabled state if the LDom resource attribute DomainFailurePolicy for the control domain is set to “ignore” and the LDom service group attribute SysDownPolicy is set to “AutoDisableNoOffline”.

****** VCS behavior would be “No fail over” with service group in auto-disabled state if the LDom resource attribute DomainFailurePolicy for the control and other I/O domain is set to “ignore” and the LDom service group attribute SysDownPolicy is set to “AutoDisableNoOffline”

Recommendations while configuring VCS and Oracle VM Server for SPARC with multiple I/O domains

- Online and offline operations for service groups in the StorageSG attribute

  To manually bring online or take offline service groups that are configured in the StorageSG attribute do not use the AlternateIO resource or its service group

  Instead, use service groups configured in the StorageSG attribute.

- Freeze the service group for the AlternateIO resource

  Freeze the AlternateIO service group before you bring online or take offline service groups configured in the StorageSG attribute of the AlternateIO resource. If you do not freeze the service group, the behavior of the Logical Domain is unknown as it is dependent on the AlternateIO service group.

- Configuring preonline trigger for storage service groups

  You must configure preonline trigger in the following scenario:

  When the service groups configured in the StorageSG attribute of the AlternateIO resource are of fail over type, and if you accidentally bring storage service groups online on another physical system in the cluster.

  It is possible to bring the storage service groups online on another physical system because resources configured to monitor back-end storage services are present in different service groups on each physical system. Thus, VCS cannot prevent resources coming online on multiple systems. This may cause data corruption.

  Note: Perform this procedure for storage service groups on each node.

  To configure preonline trigger for each service group listed in the StorageSG attribute

  - Run the following commands:

    ```
    # hagrp -modify stg-sg TriggerPath bin/AlternateIO/StorageSG
    # hagrp -modify stg-sg TriggersEnabled PREONLINE
    ```

    where *stg-sg* is the name of the storage service group

- Set connection time out period for virtual disks

  When a disk device is not available, I/O services from the guest domain to the virtual disks are blocked.

  Veritas recommends to set a connection time out period for each virtual disk so that applications times out after the set period instead of waiting indefinitely. Run the following command:

```
# ldm add-vdisk timeout=seconds disk_name \
volume_name@service_name ldom
```

- Fail over of LDom service group when all the I/O domains are down. When the attribute SysDownPolicy is set to AutoDisableNoOffline for a service group, the service group state would be changed to OFFLINE|AutoDisabled when the system on which the service group online goes down. Before auto-enabling the service group and online the service group on any other nodes, it is mandatory to ensure that guest domain is stopped on the system (control domain) that is down. This is particularly important when failure-policy of the master-domains is set to ignore.

Consider the following scenario: The DomainFailurePolicy of the LDom resource is set to {primary=”stop”} by default.

If the guest domain need to be made available even when primary domain is rebooted or shut down for maintenance.

- The DomainFailurePolicy attribute would be changed to {primary=ignore, alternate1=stop} or {primary=ignore, alternate1=ignore}.

  Guest domain will not be stopped even when primary domain is rebooted or shutdown
- The SysDownPolicy attribute would be set to AutoDisableNoOffline for planned maintenance. VCS will not fail-over the service group when the node is down instead the group would be put in to auto-disabled state.

The guest domain can continue to function normally with the I/O services available through alternate I/O domain when the control domain is taken down for maintenance.

When the control domain is under maintenance, and if the alternate I/O domain fails due one of the following:

- DomainFailurePolicy attribute is set to {primary=ignore, alternate1=stop} and only the I/O services from alternate I/O domain are unavailable (i/o domain is active, but n/w or storage loss).
- DomainFailurePolicy attribute is set to {primary=ignore, alternate1=ignore} and if the alternate I/O domain is down (domain is in-active).

In this situation guest domain will not be functioning normally and it is not possible to bring down the guest domain as there is no way to access the guest domain. In such scenarios, you must perform the following steps to online the LDom service group on any other available nodes.

To online the LDom service group

1. If the primary domain can be brought up, then bring up the primary domain and stop the guest domain:

   ```
   # ldm stop ldom_name
   ```

   If this is not possible, power off the physical system from the console so that the guest domain stops.

2. Auto enable the service group:

   ```
   # hagrp -autoenable group -sys system
   ```

3. Online the LDom service group:

   ```
   # hagrp -online group -any
   ```

Sample VCS configuration for AlternateIO resource configured as a fail over type

```
include "types.cf"
cluster altio-cluster (
 UserNames = { admin = XXXXXXXXXXX }
 Administrators = { admin }
 HacliUserLevel = COMMANDROOT
 )

system primary1 (
 )

system alternate1 (
 )

system primary2 (
 )

system alternate2 (
 )

group aiosg (
 SystemList = { primary1 = 0, primary2 = 1 }
 AutoStartList = { primary1 }
 TriggerPath = "bin/AlternateIO"
```

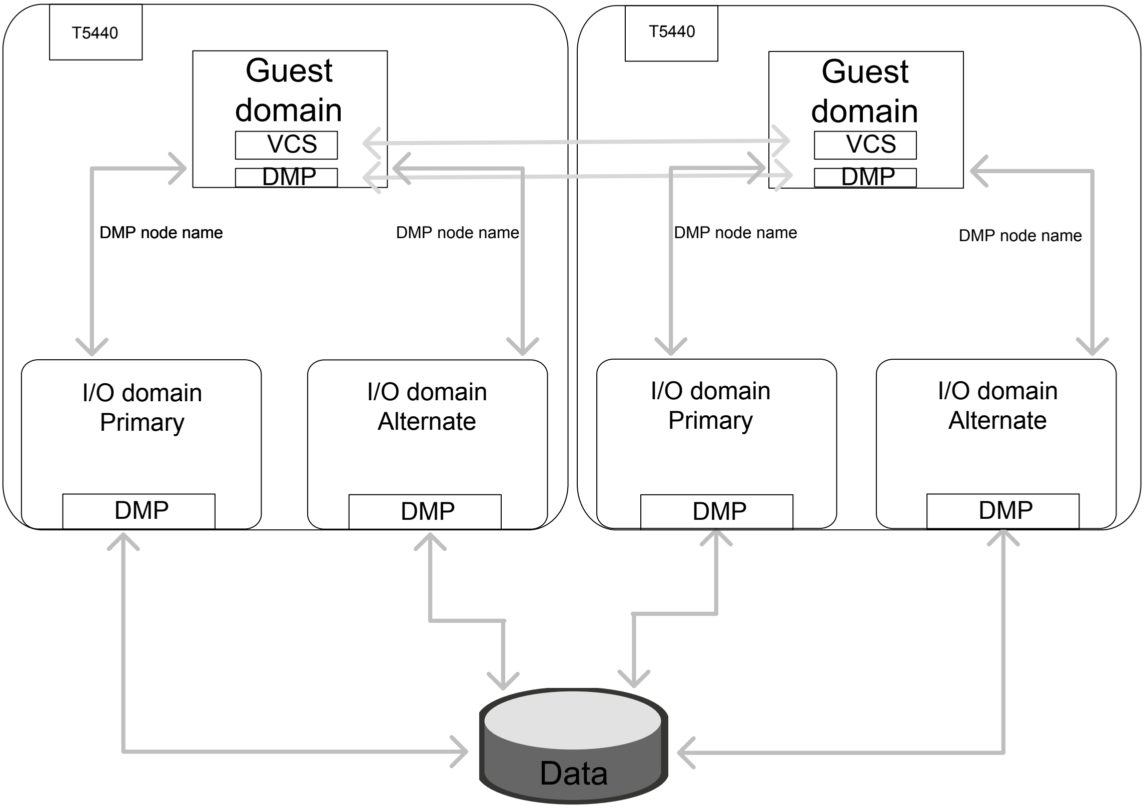
```
 TriggersEnabled @primary1 = { PREONLINE }
 TriggersEnabled @primary2 = { PREONLINE }
 )

 AlternateIO altiores (
  StorageSG @primary1 = { primary1-strsg = 1 }
  StorageSG @primary2 = { primary2-strsg  = 1 }
  NetworkSG @primary1 = { primary1-nwsg = 0 }
  NetworkSG @primary2 = { primary2-nwsg  = 0 }
  )



 // resource dependency tree
 //
 // group aiosg
 // {
 // AlternateIO altiores
 // }


group ldomsg (
 SystemList = { primary1 = 0, primary2 = 1 }
 AutoStartList = { primary1 }
 SysDownPolicy = { AutoDisableNoOffline }
 )

 LDom ldmguest (
  LDomName = ldg1
  )

 requires group aiosg online local hard


 // resource dependency tree
 //
 // group ldomsg
 // {
 // LDom ldg1
 // }


group primary1-strsg (
  SystemList = { primary1 = 0, alternate1 = 1  }
        AutoStart = 0
        Parallel = 1
        TriggerPath = "bin/AlternateIO/StorageSG"
        TriggersEnabled @primary1 = { PREONLINE }
        TriggersEnabled @alternate1 = { PREONLINE }
        AutoStartList = { primary1, alternate1 }
        )

        Zpool zpres1 (
                PoolName @primary1= zfsprim
                PoolName @alternate1 = zfsmirr
                ForceOpt = 0
                )



        // resource dependency tree
        //
        //      group primary1-strsg
        //      {
        //      Zpool zpres1
        //      }



group primary1-nwsg (
 SystemList = { primary1 = 0, alternate1 = 1 }
 Parallel = 1
 )

 Phantom ph1 (
  )

 NIC nicres1 (
                Device @primary1 = nxge3
                Device @alternate1 = nxge4
                )



 // resource dependency tree
 //
 // group primary1-nwsg
 // {
 // Phantom ph1
 // Proxy nicres1
 // }


group primary2-strsg (
  SystemList = { primary2 = 0, alternate2 = 1  }
        Parallel = 1
        TriggerPath = "bin/AlternateIO/StorageSG"
        TriggersEnabled @primary2 = { PREONLINE }
        TriggersEnabled @alternate2 = { PREONLINE }
        )

        Zpool zpres2 (
                PoolName @ primary2  = zfsprim
                PoolName @ alternate2 = zfsmirr
                ForceOpt = 0
                )



   // resource dependency tree
   //
   //      group primary2-strsg
   //      {
   //      Zpool zpres2
   //      }


group primary2-nwsg (
 SystemList = { primary2 = 0, alternate2 = 1 }
 Parallel = 1
 )

 Phantom ph2 (
  )

 NIC nicres2 (
                Device @primary2 = nxge3
                Device @alternate2 = nxge4
                )
```

```
// resource dependency tree
//
// group primary2-nwsg
// {
// Phantom ph2
// Proxy nicres2
// }
```

Configuring VCS on logical domains to manage applications using services from multiple I/O domains

This section describes how to configure Cluster Server (VCS) on logical domains to manage applications using services from multiple I/O domains.

Figure 6-19 Typical two node VCS cluster setup is configured on logical domains where each guest domain receives I/O services from multiple I/O domains



The typical two node VCS cluster at guest domain fails over the applications between guest domains to keep the applications highly available.

Configuring VCS on guest domains that uses services from multiple I/O domains

1. Set up the required hardware, system, storage, and network.
2. Install the operating system and the Oracle VM Server for SPARC on the physical system.

   Refer to the *Oracle* documentation for details.
3. Set up the alternate I/O domain.

   Refer to the *Oracle VM Server for SPARC Administration Guide* for details.

4. Set up the guest domain and provide the I/O services from all the I/O domains to the guest domain.

   Refer to the *Oracle VM Server for SPARC Administration Guide* for details.

   During the setup of the guest domain, you can choose one of the following:

   - Local disk or SAN disk as boot disk to guest domain
   - SAN disk as data disk for application

   Use SAN disk for boot disk and data disk if you plan to migrate the guest domain.

   - If the storage device contains multiple paths from each I/O domain, you have the following options:
     - Install Dynamic Multi-Pathing (DMP) on the guest domain
       Export all the paths to storage device from each I/O domain. DMP at the guest domain takes care of path management between I/O domains.
     - Install Dynamic Multi-Pathing (DMP) on I/O domains and the guest domain
       Export DMP node name from each I/O domain to the guest domain. DMP at the guest domain takes care of path management between I/O domains.

5. Install VCS on the guest domain.

   See the *Cluster Server Configuration and Upgrade Guide* for details.

6. Configure the application service group.

   See the *Cluster Server Administrator’s Guide* and *Cluster Server Bundled Agents Reference Guide* for details.

Chapter 7

SF Oracle RAC support for Oracle VM Server for SPARC environments

This chapter includes the following topics:

About deploying SF Oracle RAC in Oracle VM Server for SPARC environments

Oracle VM Server for SPARC (earlier known as Logical Domains) from Oracle is a technology that allocates resources such as processors, memory, disks or network devices to logical containers and manages them as logical domains within the physical host. The resulting logical domain has its own operating system and manages resources independently in its realm.

A logical domain can be set up to function in any of the following roles:

Control domain	The Control domain is the physical host that has access to all physical resources on the system. The Oracle VM Server software, which is used for managing the guest and I/O domains, is installed on this domain. The Control domain can also serve as an I/O domain and provide services to other guest domains.
I/O domain	The I/O domain has direct access to I/O devices. There can be a maximum of two I/O domains in a logical domain setup. The domain is also called as the service domain when it provides I/O services to other guest domains.
Guest domain	The Guest domain uses the services delivered by the service domain to access physical devices. The Guest domain exists as an independent entity with virtual resources and own copy of operating system.

Oracle VM Server for SPARC provides a cost-effective alternative architecture for deploying SF Oracle RAC. The same physical server can be used for multiple applications within various logical domains with optimal resource utilization. Oracle VM Server for SPARC is hardware-dependent and works with the latest SUN CoolThreads servers.

For detailed information, see the Oracle documentation.

Sample configuration scenarios

The following server configuration is used for the sample scenarios presented in this document:

Server	Oracle Sun SPARC Enterprise T-3 Server
Processor	24 T3 Based Processor (4Cores * 6 Threads per core)
Memory	8 GB
PCI devices	2 NIC Cards +1 HBA per BUS, all on board hard drives belong to one PCI bus
Operating system	Solaris 11 Update 1
Oracle VM Server for SPARC version	Oracle VM Server for SPARC 3.0

Database version	Oracle RAC 11g Release 2 For the latest information on supported Oracle database versions, see: https://sort.veritas.com/scl
SF Oracle RAC version	SF Oracle RAC 7.4.1
Installation and setup instructions	For instructions, see the *Storage Foundation for Oracle RAC Configuration and Upgrade Guide (7.4.1)*.

The sample deployment scenarios are as follows:

- SF Oracle RAC with Oracle RAC database on I/O domains of two hosts
  See "SF Oracle RAC with Oracle RAC database on I/O domains of two hosts" on page 203.
- SF Oracle RAC with Oracle RAC database on Guest domains of two hosts
  See "SF Oracle RAC with Oracle RAC database on guest domains of two hosts" on page 205.
- SF Oracle RAC with Oracle RAC database on Guest domains of a single host
  See "SF Oracle RAC with Oracle RAC database on guest domains of single host" on page 207.
- SF Oracle RAC with Oracle RAC database on Guest domain and I/O domain of a single host
  See "SF Oracle RAC with Oracle RAC database on I/O domain and guest domain of single host" on page 210.

Preparing to deploy SF Oracle RAC in logical domain environments

Complete the following tasks before you deploy SF Oracle RAC in Oracle VM Server for SPARC environments:

1. Install Oracle VM Server software.
2. Configure the physical host as a primary domain.

   The primary domain will serve as the control/IO/service domain by default.
3. Create new logical domain configurations.

SF Oracle RAC with Oracle RAC database on I/O domains of two hosts

This section describes the tasks required to set up I/O domains on two hosts for deploying SF Oracle RAC.

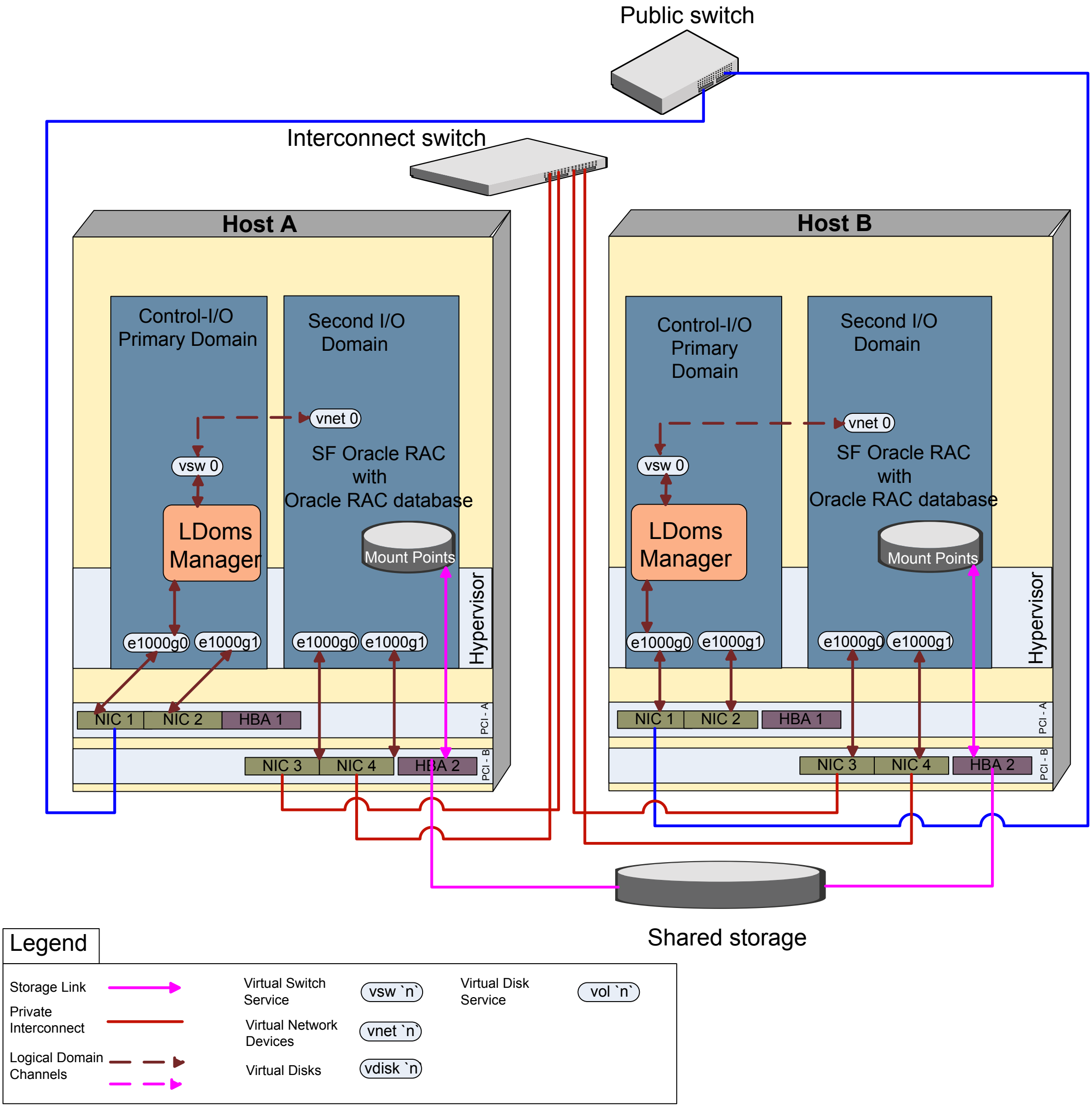
The benefits of this approach are as follows:

- The computing resources on the hosts are available to other logical domains.
- Direct access to storage ensures better database performance.

The disadvantage of this approach is that if the I/O load is extensive, the I/O latency on other logical domains will be high.

Figure 7-1 illustrates the scenario.

Figure 7-1 SF Oracle RAC with Oracle RAC database on I/O domains of two hosts



To set up I/O domains on two hosts for deploying SF Oracle RAC

1. Complete the preparatory steps for setting up a logical domain environment.

   See "Preparing to deploy SF Oracle RAC in logical domain environments" on page 202.

2. Create the secondary I/O domain.

3. Provision a PCI bus from the primary domain to the secondary I/O domain.

4. Create virtual network service in the primary domain.

   The virtual interface connected to the service will be used by the secondary I/O domain as its public interface. The interfaces that remain on the secondary I/O domain will be used for LLT heartbeat.

5. Create virtual disk service on the primary domain.

   The virtual disk connected to the service will be used by the secondary I/O domain as its root disk.

6. Install Solaris operating system on the secondary I/O domain using native operating system installation method.

   After OS installation, all devices present on the provisioned PCI bus are visible on the secondary I/O domain.

7. Repeat steps 2 to 6 on the second host.

8. Install and configure SF Oracle RAC.

9. Set up Oracle RAC database.

SF Oracle RAC with Oracle RAC database on guest domains of two hosts

This section describes the tasks required to set up guest domains on two hosts for deploying SF Oracle RAC.

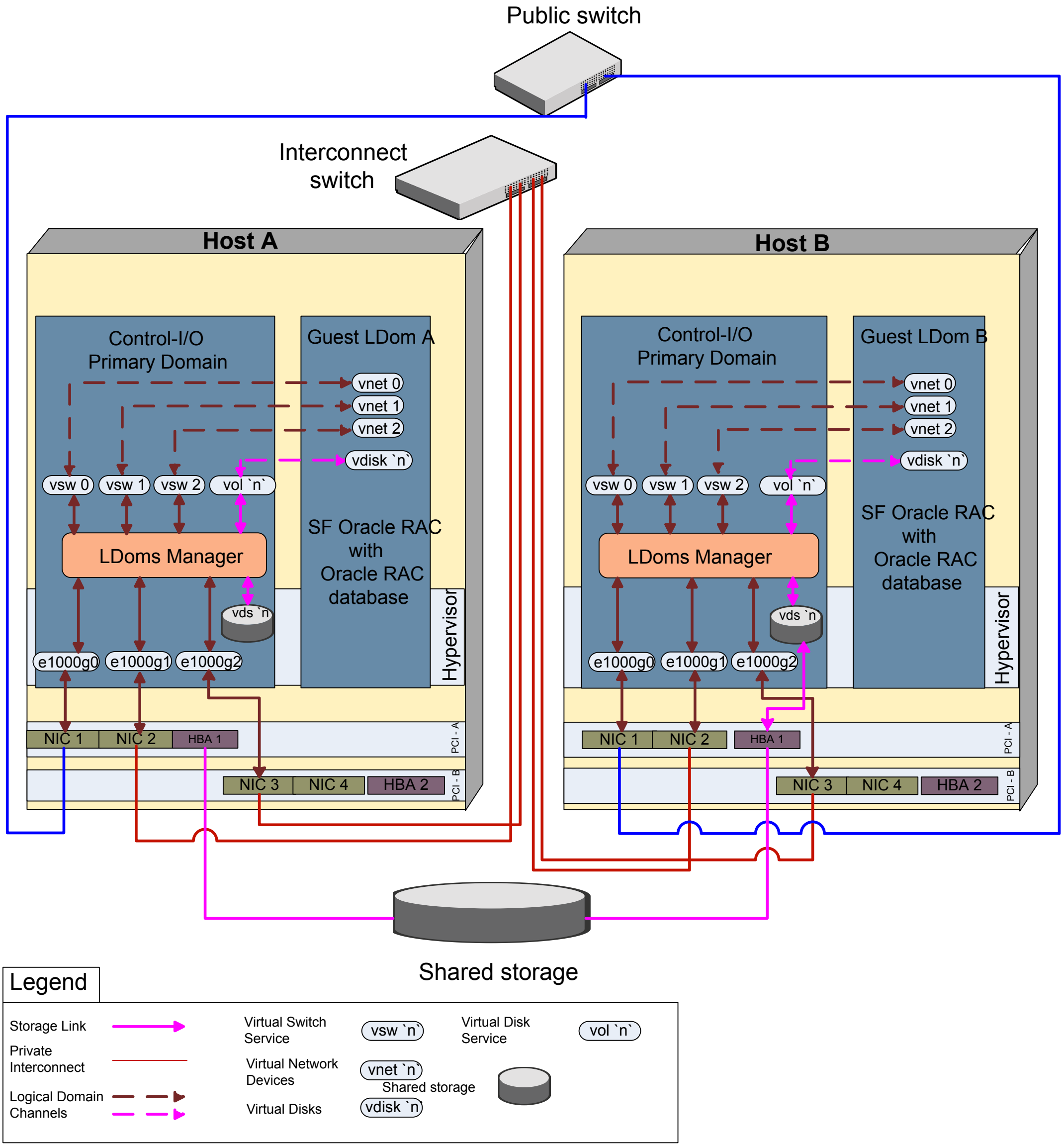
The benefits of this approach are as follows:

- The configuration provides a completely virtualized domain.
- The hardware resources can be effectively utilized with other logical domains.

The disadvantage of this approach is that since there is no direct disk access from the virtual domain to the physical disk, there could be a minor lag on disk access times.

Figure 7-2 illustrates the scenario.

Figure 7-2 SF Oracle RAC with Oracle RAC database on guest domains of two hosts



To set up guest domains on two hosts for deploying SF Oracle RAC

1 Complete the preparatory steps for setting up a logical domain environment.

   See "Preparing to deploy SF Oracle RAC in logical domain environments" on page 202.

2 Create virtual disk and network services on the primary domain.

   The virtual services are bound to the virtual devices created for the guest logical domain. A guest logical domain accesses the physical hardware through the virtual devices connected to the virtual services. Assign individual services for each disk from storage. This helps to maintain sequence of the disks in the guest domain.

3 Create the guest domain.

4 Create the virtual devices on the guest domain and bind it to the virtual services created on the primary domain.

5 Install Solaris operating system on the guest domain using native operating system installation method.

6 Repeat steps 2 to 5 on the second host.

7 Install and configure SF Oracle RAC.

8 Set up Oracle RAC database.

SF Oracle RAC with Oracle RAC database on guest domains of single host

This section describes the tasks required to set up guest domains on a single host for deploying SF Oracle RAC.

Note: This setup is recommended for use as a four-node cluster by using an additional physical host with the same configuration.

The benefits of this approach are as follows:

- Reduction in the number of physical servers used makes it a very cost-effective setup.
- The setup is easy to create and maintain. It is also flexible and portable.
- Many guest logical domains from multiple systems can be joined together to form a bigger cluster.

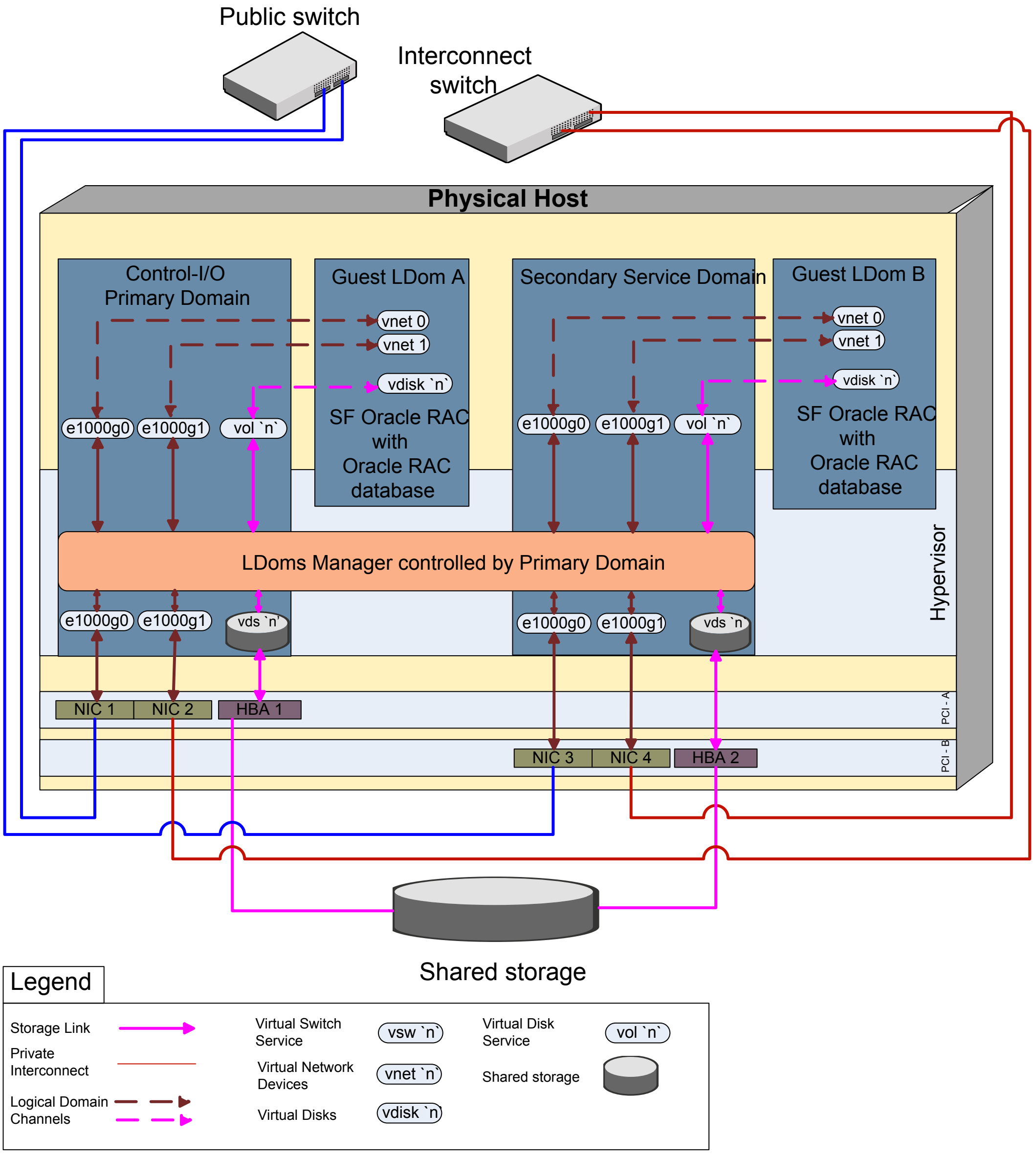
- If the primary domain reboots, only the guest logical domain attached to it is affected. The guest logical domain attached to the secondary service domain continues to be available. Please note that shutting down the primary domain halts all domains.

The disadvantages of this approach are as follows:

- Hardware failures act as a single point of failure, bringing down all associated domains.
- Almost all physical resources are consumed in the creation of this setup on a T2000 Server.

Figure 7-3 illustrates the scenario.

Figure 7-3 SF Oracle RAC with Oracle RAC database on guest domains of single host



To set up guest domains on a single host for deploying SF Oracle RAC

1 Complete the preparatory steps for setting up a logical domain environment.

See "Preparing to deploy SF Oracle RAC in logical domain environments" on page 202.

2 Create a split PCI configuration on a T5k, T4, or T2000 Server.

Each bus has 2NICs and 1HBA.

3 Create one primary domain and one secondary domain.

The primary domain must be configured to be used as a control and I/O Domain (primary service domain). The secondary domain must be initially configured as an I/O domain. Next, configure the secondary I/O domain to be used as a secondary service domain. The primary domain is used to create and assign the services for this secondary service domain.

Both the primary and secondary Service domains see the storage through its respective paths.

4 Create the disk and network services on the primary service domain and assign it to guest logical domain A.

5 Create and assign a different set of disk and network services for the secondary service domain using the primary service domain. This ensures that services for guest logical domain B are provided by the secondary service domain.

6 Install Solaris operating system on the secondary service domain and the guest logical domains using native operating system installation method.

7 Install and configure SF Oracle RAC.

Note: Set the public link to be used as a low priority heartbeat link due to unavailability of NICs.

8 Set up Oracle RAC database.

SF Oracle RAC with Oracle RAC database on I/O domain and guest domain of single host

This section describes the tasks required to set up guest domains on single host for deploying SF Oracle RAC.

Note: This setup is recommended for use as a four-node cluster by using an additional physical host with the same configuration.

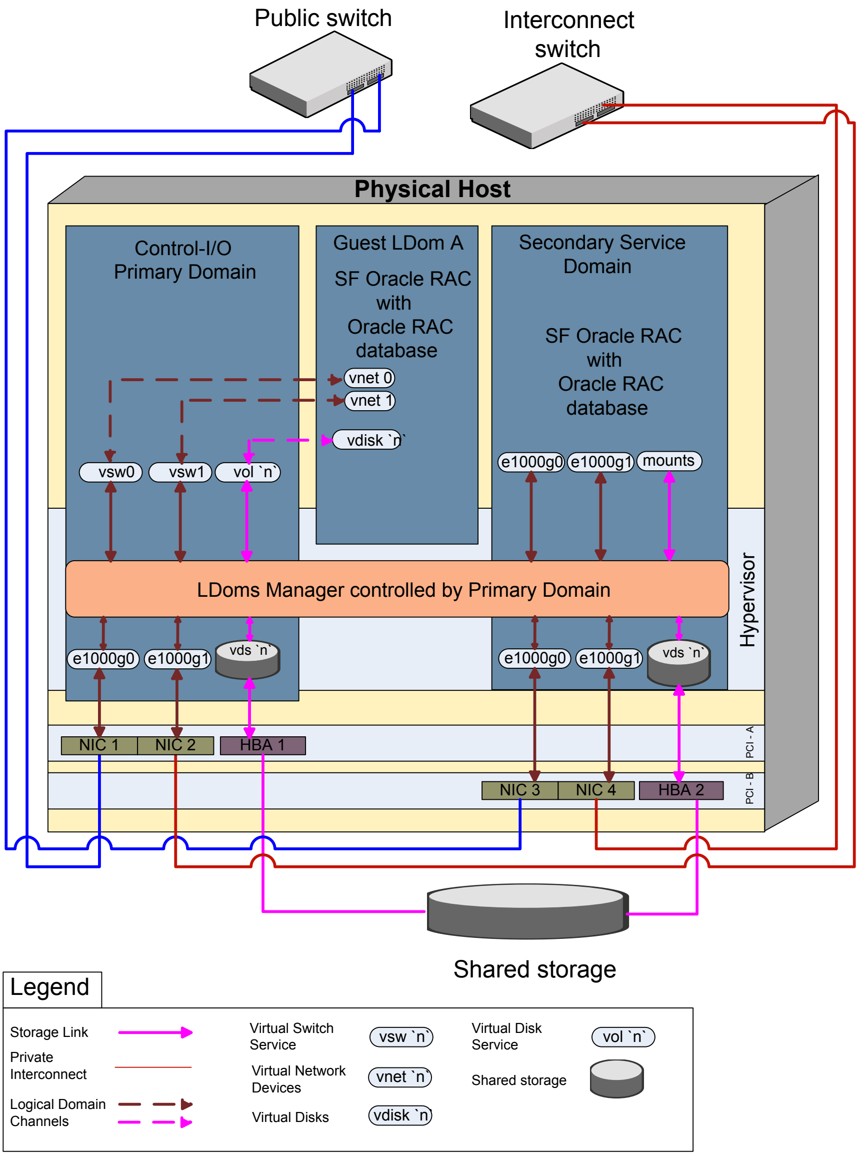
The benefits of this approach are as follows:

- Guest logical domains can be added to the host at any time unlike in Scenario 3 where all resources are utilized leaving no room for additional logical domains.
- This setup results in better disk performance as compared to Scenario 3.
- If the primary domain reboots, only the guest logical domain attached to it is affected. The guest logical domain attached to the secondary service domain continues to be available. Please note that shutting down the primary domain halts all domains.

The disadvantage of this approach is that hardware failures act as a single point of failure, bringing down all associated domains.

Figure 7-4 illustrates the scenario.

Figure 7-4 SF Oracle RAC with Oracle RAC database on I/O domain and guest domain of single host



To set up guest domains of single host for deploying SF Oracle RAC

1. Complete the preparatory steps for setting up a logical domain environment.

   See "Preparing to deploy SF Oracle RAC in logical domain environments" on page 202.

2. Create a split PCI configuration on a T2000 Server.

   Each bus has 2NICs and 1HBA.

3. Create the secondary I/O domain.

4. Provision a PCI bus from the primary domain to the secondary I/O domain.

5. Create virtual disk service on the primary domain. The virtual disk connected to the service is used by the secondary I/O domain as its root disk.

6. Create the disk and network services on the primary service domain and assign it to the guest domain.

7. Install Solaris operating system on the secondary service domain and the guest domain using native operating system installation method.

8. Install and configure SF Oracle RAC.

   Perform the following steps after configuring SF Oracle RAC:

   - Set the public link to be used as a low priority heartbeat link due to unavailability of NICs.
   - Run the command "oifcfg" after installation of Oracle Clusterware.
     This is required to resolve the issue of different interface names on the guest logical domain and the secondary I/O logical domains.
   - Configure MultiPrivNic manually.

9. Set up Oracle RAC database.

Chapter 8

Support for live migration in FSS environments

This chapter includes the following topics:

About live migration in Flexible Storage Sharing (FSS) environments

You can migrate Oracle VM Server for SPARC systems from one control domain to another in FSS environments. This support is currently tested for SF Oracle RAC and SFCFSHA environments.

The migration requires that all devices accessible to the Oracle VM Server on the original control domain are also accessible on the new control domain. However, the DAS disks attached to the control domain in FSS environments are not accessible on the new control domain. Therefore, the migration requires removal of local DAS devices from the VxVM configuration so as to unexport them from the Oracle VM Servers. All running applications and mount points continue to be active by using the storage on the remote node during the migration.

All devices exported from the source control domain must exist on the target control domain where you plan to migrate the Oracle VM Server.

It is recommended that the firmware and the architecture of the source and target control domains be the same.

Performing live migration of Oracle VM Server for SPARC systems in Flexible Storage Sharing (FSS) environments

Perform the steps in the following procedure to perform live migration of Oracle VM Server for SPARC systems in FSS environments.

To perform live migration of Oracle VM Server for SPARC systems in FSS environments

1. In the Oracle VM Server for SPARC cluster, verify that each FSS volume has a mirror across the cluster nodes of the target control domain.

   ```
   # vxprint
   ```

2. Remove the DAS devices from the VxVM configuration.

   ```
   # vxdmpadm exclude dmpnodename=vm_diskname
   # vxdisk list
   ```

   After excluding the DAS devices from DMP, they will be visible as “remote disk” in the `vxdisk list` output.

3. Verify that the applications and mount points in the migrating systems are running properly using remote plex.

   ```
   # cfsmntadm display
   ```

4. Unexport the devices from the Oracle VM Server.

   ```
   # ldm remove-vdisk vdisk_name
         oracle_vm_name
   ```

5. Migrate the Oracle VM Server systems.

   ```
   # ldm migrate oracle_vm_name
         controldomain_name
   ```

6. Verify that the applications and mount points in the migrating systems are running properly using remote plex.

   ```
   # cfsmntadm display
   ```

If you plan to move the migrated systems back to the original control domain, ensure that you export the local DAS devices to the Oracle VM Server again after migrating it back, and add them to the VxVM configuration for local access.

```
# ldm add-vdisk vdisk_name vdsdev@vds oracle_vm_name
# vxdmpadm include dmpnodename=vm_diskname
```

After adding the local DAS devices to the VxVM configuration, the local plex resyncs automatically without any manual intervention.

You can verify the state of the local plex and local the DAS devices using the following commands:

```
# vxprint
# vxdisk list -o alldgs
```

Section 4

Reference

Where to find more information

This appendix includes the following topics:

- Veritas InfoScale documentation
- Solaris virtualization documentation
- Service and support
- About Veritas Services and Operations Readiness Tools (SORT)

Veritas InfoScale documentation

The latest documentation is available on the Veritas Services and Operations Readiness Tools (SORT) website in the Adobe Portable Document Format (PDF).

See the release notes for information on documentation changes in this release.

Make sure that you are using the current version of documentation. The document version appears on page 2 of each guide. The publication date appears on the title page of each document. The documents are updated periodically for errors or corrections.

https://sort.veritas.com/documents

You need to specify the product and the platform and apply other filters for finding the appropriate document.

Solaris virtualization documentation

For Oracle VM Server for SPARC (formerly Solaris Logical Domains), Zone, Projects, and Zone installation and configuration information, refer to the Oracle site: www.oracle.com.

Oracle provides regular updates and patches for Oracle VM Server for SPARC, Zones, and Zone features. Contact Oracle for details.

Service and support

To access the self-service knowledge base, go to the following URL:

https://www.veritas.com/support/en_US.html

About Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) is a Web site that automates and simplifies some of the most time-consuming administrative tasks. SORT helps you manage your datacenter more efficiently and get the most out of your Veritas products.

SORT can help you do the following:

Prepare for your next installation or upgrade

- List product installation and upgrade requirements, including operating system versions, memory, disk space, and architecture.
- Analyze systems to determine if they are ready to install or upgrade Veritas products.
- Download the latest patches, documentation, and high availability agents from a central repository.
- Access up-to-date compatibility lists for hardware, software, databases, and operating systems.

Manage risks

- Get automatic email notifications about changes to patches, array-specific modules (ASLs/APMs/DDIs/DDLs), and high availability agents from a central repository.
- Identify and mitigate system and environmental risks.
- Display descriptions and solutions for hundreds of Veritas error codes.

Improve efficiency	■ Find and download patches based on product version and platform. ■ List installed Veritas products and license keys. ■ Tune and optimize your environment.

Note: Certain features of SORT are not available for all products. Access to SORT is available at no extra cost.

To access SORT, go to:

https://sort.veritas.com