

Confidence in a connected world.

Architecting for a Resilient, Cost Effective Logical Domain Infrastructure

A Guide to Maximizing the
Investment in an LDom Farm using
Veritas Storage Foundation

*Doug Matthews, Senior Principal Technical Product Manager
Storage and Availability Management Group*

*Misha Chawla, Principal Software Engineer
Storage and Availability Management Group*

Architecting for a Resilient , Cost Effective Logical Domain Infrastructure

A Guide to Maximizing the Investment in an LDom Farm using Veritas Storage Foundation

Contents

Introduction	4
The challenge of architecting a cost efficient LDom implementation while maintaining resiliency and reducing overhead	4
High-level Architecture Options.....	4
The Use Cases	6
Software Used in this Document.....	7
Getting Started	7
Building the Control Domain.....	8
Boot Image Management.....	12
Creating the Boot_DG.....	13
Setting up the gold image	13
Configuring a “dummy domain”	14
Rapid Provisioning using Boot Image Manager (BIM)	15
Storage Savings from Space Optimized SnapShots	19
Pooling storage with BIM + CVM across multiple nodes	20
Beyond the Boot Images	22
Considerations for Getting DATA Storage to Provisioned Guests (Choices choices and more choices)	22
Improving Application Resiliency with Veritas Cluster Server.....	26
Pooling Storage among Multiple Physical and Virtual Nodes and Advantages for Failover	28
Customer Use Case Examples	32
The Boot Image Management Use Case:	32

The Production High Availability Use Case:.....	33
Consolidated list of Recommended Best Practices.....	35
Appendix	36
Where to go for more info:	36

Introduction

A system administrator can create a Solaris Logical Domains (LDoms) server farm that allows for rapid instance provisioning, significantly less storage consumption, greater instance resiliency and true pooled storage across all instances regardless of their physical location by using Veritas Storage Foundation High Availability (SFHA). This guide demonstrates a set of reference architectures that can be customized to fit most implementations. An assumption is made that the reader understands Solaris Logical Domains and their architecture. It is also assumed the reader possesses a basic understanding of how to configure and manage Solaris Logical Domains technology using Solaris management tools. There is also an expectation that the reader is familiar with basic Veritas Storage Foundation HA product architecture and management utilities. For more information on items not covered please reference the *Where to Get More Information* section.

The challenge of architecting a cost efficient LDom implementation while maintaining resiliency and reducing overhead

Logical Domains offer the promise of driving up server physical resource utilization through the use of logical resource partitioning and hosting multiple Operating System virtual instances on a single physical node. This architecture comes with some inherent risks from the “*all your eggs in a single basket*” approach that represents a threat to virtual instance resiliency in the event of hardware failures. The potential benefits however include:

- Dramatic cost savings in the sharing of the physical node hardware resources
- Significant reduction of storage space consumed by multiple boot images
- Reduction of software licensing cost with Host and CPU based license models
- Greatly improved time to market of new installations

Using appropriate implementation techniques, these benefits can be accomplished while achieving higher application availability than a traditional standalone server.

High-level Architecture Options

Veritas Storage Foundation can be configured to run in the Control Domain as well as individual Guest Domains. Additionally sub-sets of the entire solution can be utilized at different layers to achieve specific needs. This results in numerous potential permutations of architectures. To simplify matters this paper will establish some reference architectures to explore. With that said there are significantly more options that are offered in detail in the *Veritas Storage Foundation Application Support Note for Solaris Logical Domains* (link provided in More Info section). Each of the following reference configuration options will be presented and discussed in detail in this guide including the advantages and disadvantages of each.

Architecting for a Resilient, Cost Effective Logical Domain Infrastructure

As the reference architectures are presented every configuration discussed refers only to data storage. It is assumed that Boot Image Management in all of the following reference configurations will be performed using the Veritas Boot Image Management (BIM) utility (discussed in detail in later sections) which implies that at a minimum Veritas Volume Manager (VxVM) or Cluster Volume Manager (CVM) will be run in the Control Domain even if only for boot image management. While this is not a requirement, it is generally considered a best practice and will be discussed in the Boot Image Management sections of this document. For a basic understanding of some of the configuration options available they are presented here as an introduction, and will be expanded upon in later sections.

Data storage access reference architectures:

- **Light Weight Guest Architecture**
 - Control Domain – Dynamic Multipathing (DMP), VxVM or CVM to present volumes to Guest Domains
 - Guest Domain VxFS in Guest Domain for file system
- **Full Stack in Guest**
 - Control Domain – Devices directly passed from Control Domain to Guest Domain with no Veritas software in Data Device I/O path
 - VxVM / CVM will still manage and present volumes used as the boot devices from the Control Domain
 - Guest Domain – Entire Veritas Storage Foundation stack
- **Centralized DMP**
 - Control Domain – DMP for device presentation to Guest
 - VxVM / CVM will still manage and present volumes used as the boot devices from the Control Domain
 - Guest Domain – Entire Veritas Storage Foundation stack

These architectures are further extended by the addition of high availability software for domain management and application monitoring. This is accomplished with configuration of the Veritas Cluster Server (VCS) software and the usage of the Cluster Management Console. VCS can be configured to extend any of the above reference architectures and will be discussed in detail in later sections, but again as an introduction the following is presented:

Architectures to achieve a manageable LDom farm with the highest availability

- **VCS In the Control Domain**
 - Manages Guest Domains as a single unit of control
 - Monitors status of resources required to start, run, and stop Guest Domains
 - Provide automatic restart or move individual Guest Domains in response to failures

- Provides Start / Stop / Monitor of individual Guest Domains from a common console across the entire farm using Veritas Cluster Management Console
- Controls placement of individual LDom's configured to run on multiple physical nodes in active-active or active-passive configurations
- Can be configured to monitor or control applications running in Guest Domains through the combination of the remote group agent and the VCS in the Guest Domain architecture
- **VCS in the Guest Domain architecture**
 - Manages applications running in the Guest Domains as a single unit of control
 - Monitors status of resources required to start, stop, and monitor Guest Domain based applications and reacts to changes in resource status
 - Provides automatic restart, move or migrate of individual applications between Guest Domains
 - Provides management of applications as service groups from a common console across appropriate Guest Domains in the farm using Veritas Cluster Management Console
 - Controls placement of individual applications configured to run on multiple LDom's in active-active or active-passive configurations
 - Can be configured to provide application monitoring or respond to control from VCS in the Control Domain via the use of the remote group agent

The Use Cases

The use case examples provided at the end of this document are based on observed usage trends in the Symantec customer base and are provided as examples only. While these cases are representative of usage trends, they are not meant to imply the implementation specifics of any individual customer.

Use Case #1

The Boot Image Management use case was designed to reduce cost and improve time to market. The customer faced a need to purchase additional hardware including server and storage resources to support new training environments. Given that classes are short term and environments need to be rebuilt often and rapidly the goal was to implement a virtualized solution to reduce the overall hardware spend without sacrificing performance or agility. This use case is relevant for test and development environments especially where servers are rebuilt often over a period of time.

Use Case #2

The Production High Availability use case was designed to achieve maximum uptime for production applications while minimizing the management impact associated with boot images, LDom migrations, and device multipathing. The solution was also designed with an architecture that supported separate root account control where the admin of the Control Domain was not necessarily the admin of the LDom Operating System(s).

Software Used in this Document

The software listed in the table is not the only versions supported by Symantec. Please reference the latest “*Application Support Note for Solaris Logical Domains*” (Link in the More Information section) for information on what versions are compatible.

Table 1-1 Sun Software

Software	Purpose	Download Location
Solaris 10 Update 7	Control Domain and Guest Domain Operating System	http://www.sun.com/software/solaris/get.jsp
Logical Domains 1.2 software	Provides the interface to manage the LDom implementation	http://www.sun.com/servers/coolthreads/ldoms/get.jsp

Table 1-2 Symantec Software

Software	Purpose	Download Location
Storage Foundation Cluster File System HA v 5.0 MP3 RP2 or later	Used for Volume Management, File System and Clustering	https://vias.symantec.com/labs/vpcs/vpcs/patchinfo/2520
Boot Image Management Utility	Provide Rapid LDom Deployment with limited storage impact based on a gold image	http://www.symantec.com/connect/downloads/boot-image-management-utility-ldom-logical-domains-environments-beta

Getting Started

The reference environment configured for the majority of examples in this guide includes two physical nodes in a clustered pair with shared storage visible from both physical nodes. The environment also includes (3) coordinator LUN's configured for SCSI3-PGR reservations that will be used to configure I/O Fencing as well as (2) network interfaces that will be used for intra-node cluster communication traffic.

To prepare the environment for Logical Domains the Administrator will need to perform a basic installation of the Solaris Operating 10 System Update 7 (or later) on (2) Sun SPARC Enterprise Servers based on UltraSPARC T2 or T2 Plus processors. This OS instance will ultimately be used as the Control Domain. The Admin will then install the LDom v1.2 software. For the configurations used in this guide we chose to use the “Standard Solaris Configuration” when prompted for a security profile and to configure the LDom's manually.

Building the Control Domain

In this implementation the Veritas Storage Foundation will be run within the Control Domain to present storage to the virtual guests for Boot Image Management. In some configurations Storage Foundations will also be used in the Control Domain for data device presentation to the Guest Domains. This configuration will warrant sufficient resources assigned to the Control Domain to host this workload. It is suggested that a minimum of 4 virtual CPU's and 4GB of memory be assigned to the Control Domain.

After the initial configuration suggested in the "Getting Started" section, a single Control Domain per physical node will own all of the system resources including CPU, Memory, MAU, and PCI Buses. The next steps will be to define the boundaries of the Control Domain (primary) and set up virtual servers that will deliver services to the guest domains. Many of these steps could also be performed using the LDom's configuration assistant CLI – ldmconfig, however the manual method will be shown for completeness (for more information refer to Sun information resources about ldmconfig)

Display the current configuration state:

```
# ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-c--	SP	16	16128M	0.2%	14d 17h 25m

In the example above the domain has 16 Virtual CPU's and 16GB of memory. To free resources for use in creating guest domains the administrator will change the configuration of the primary to contain 4 vCPU, 4g of memory, and no MAU resources.

An example command set to accomplish this:

```
# ldm set-vcpu 4 primary
# ldm set-mem 4g primary
# ldm set-mau 0 primary
```

The administrator will also need to configure a virtual disk server, and virtual network switch and create a virtual console device. It is then advisable that the configuration be added and saved to the service processor to be used on the next reboot.

Example:

```
# ldm add-vdiskserver vds0 primary
# ldm add-vswitch net-dev=e1000g0 vsw0 primary
# ldm add-vconscon port-range=5000-5100 vcc0 primary
# ldm add-spconfig initial-config
```

Architecting for a Resilient, Cost Effective Logical Domain Infrastructure

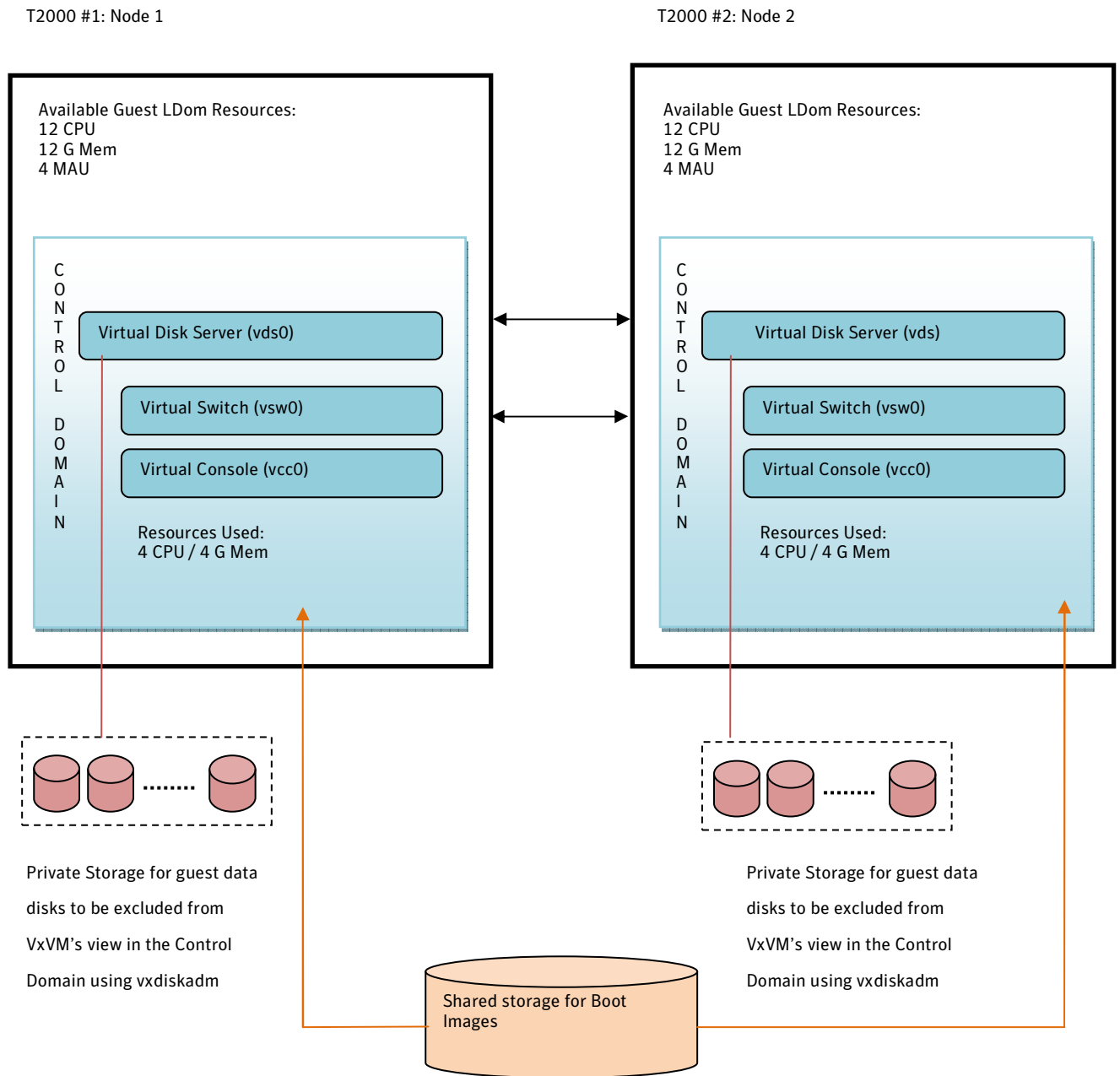


Figure 1 Initial Control Domain Configuration

Installing Veritas Software

Review the Storage Foundation Installation Guide for complete details of the installation process review. A simplified example is presented here for completeness

Upload the Storage Foundation Cluster File System HA product install tarball to a directory located in the Control Domain or mount it from a file share. Uncompress and extract the product installation binaries. Prior to running the installer the administrator will want to have the following information readily available:

- Storage Foundation Cluster File System HA Product License Key
- Systems that you will be installing
- Unique Cluster Name
- Unique Cluster ID
- Nic's that will be used for Heartbeat
- Devices that will be used in I/O Fencing Configuration

Locate and start the Storage Foundation Installer:

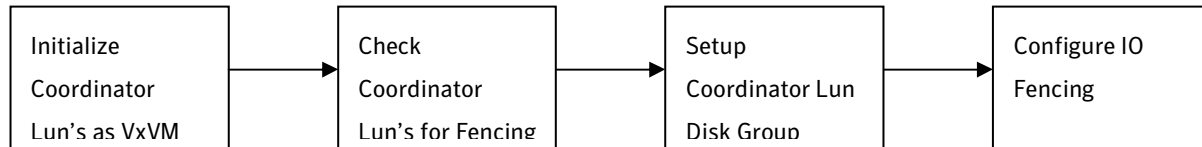
```
# ./installer
```

Using the information previously gathered answer the questions asked by the installer which will include (suggested response):

- Selecting a product to install (choose CFS)
- Entering the systems to install (User Provided)
- Providing a license key (user provided)
- Packages to install (1 for all)
- Install I/O Fencing (Answer NO – this will be setup later using DMP device names)
- Enter Cluster configuration information (Cluster Name, Cluster ID, HB Nics)
- Use Veritas Security Services (No)
- Setup SNMP / SMTP (No)
- Use Enclosure Based Names (Yes)
- Configure a default disk group (Yes – boot_dg or the name of the Disk Group that will hold Boot Devices)
- Configure SFM (Yes if available)

After the installation is complete download, uncompress and install the latest Rolling Patch (RP2 on 5.0 MP3 as of this publication). The rolling patches contain enhancements and development fixes that are crucial to successful LDom implementations and should not be considered optional in an LDom install.

Finally configure I/O Fencing to utilize DMP devices. This is beyond the scope of this document but the high-level configuration process looks like:



For more information please refer to the Veritas Cluster Server Installation Guide.

Boot Image Management

With the ever-growing application workload needs of data centers comes the requirement to dynamically create virtual environments. Thus the ability to provision and customize Logical Domains on-the-fly is required. Every Logical Domain created needs to be provisioned with CPU, memory, network and I/O resources. The primary I/O resource needed is a boot image. A boot image is an operating environment that consists of a complete filesystem with a bootable Operating System and a custom or generic software stack.

With the number of LDoms per physical server easily scaling to double digits, it becomes increasingly important to have an automatic, space-optimizing provisioning mechanism. The Boot Image Manager utility aims at providing such a solution.

This solution allows you to manage and instantly deploy LDoms based on XML templates and snapshot-based boot images (snapshots may be full or space optimized).

Benefits of the Boot Image Manager:

- Eliminates the installation, configuration and maintenance costs associated with installing the Operating System and complex stacks of software. For example a 4-node SF CFS cluster configuration can be created within minutes using a customized xml template file creating LDom nodes with identical resources. The boot images for the various nodes will be copies of the Master Image or Gold Image which contains the required software stack.
- Infrastructure cost savings due to increased efficiency and reduced operational costs.
- Reduced storage space costs due to shared Master or Gold image as well as space-optimized boot images for the various LDoms.
- Enables High Availability of LDoms with Veritas Cluster Server monitoring the LDoms and their boot image from the Control Domain.

- Ability to create and deploy LDomS across any remote node in the cluster.

Creating the Boot_DG

The various boot images need to be stored in a diskgroup repository within whose boundary the Boot Image Manager operates. This diskgroup can store the various boot images of the LDomS hosted on this as well as on remote hosts in the cluster.

For a standalone environment the diskgroup would be local or private to the host from where we run the utility. For a clustered environment, it is recommended to create a shared diskgroup so that the gold image can be shared across multiple physical nodes.

It is possible to monitor the diskgroup containing the LDom boot image(s) and the LDomS themselves under VCS so that they can be monitored for any faults. However it must be kept in mind that since the boot images are in the same diskgroup, a fault in any one of the disks backing the snapshot volumes containing the boot disks can cause all the LDomS housed on this node to failover to another physical server in the SFCFS cluster. To increase the fault tolerance for this diskgroup, mirror all volumes across multiple enclosures making the volumes redundant and less susceptible to disk errors.

Example creating a shared boot diskgroup (boot_dg) with 3 devices:

```
# vxdbg -s init boot_dg <device_name_1>
# vxdbg -g boot_dg adddisk <device_name_2>
# vxdbg -g boot_dg adddisk <device_name_3>
```

Setting up the gold image

The Boot Image Manager creates Point-In-Time images based on a Master/Gold image. This image will serve as the basis for all boot images and needs to be setup first.

Setting up the golden image will broadly involve the following steps:

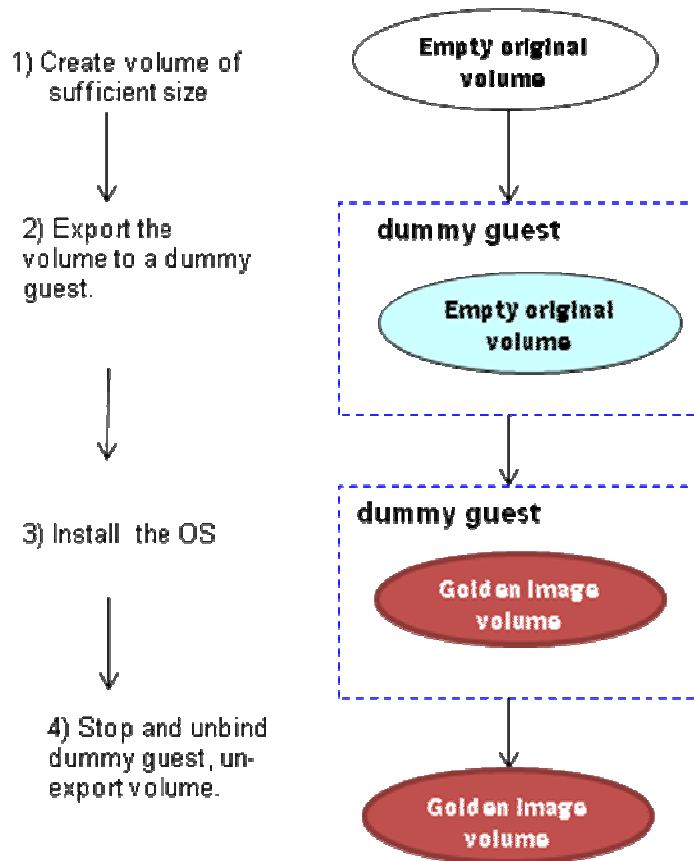


Figure 2: Process Flow for Gold Image Creation

If desired, you can install additional software packages after step 3 and before step 4.

Configuring a “dummy guest”

Start by creating a Volume to host the Gold Image

```
# vxassist -g boot_dg make gold-boot-disk-vol 15g
```

Add the volume as a virtual device to the virtual disk server previously configured

```
# ldm add-vdiskserverdevice /dev/vx/dsk/boot_dg/gold-boot-disk-vol gold-boot-disk-vol@vds0
```

The Dummy Guest (LDom) can be built in a variety of methods including Jumpstart, or manually using a Solaris installation DVD iso image. For the purposes of this example an iso image is used.

Configure the “Dummy LDom”

Architecting for a Resilient, Cost Effective Logical Domain Infrastructure

```
# ldm create gold_ldom                ' create an LDom named gold_ldom
# ldm set-vcpu 2 gold_ldom            '2 virtual CPU to gold_ldom
# ldm set-mem 2g gold_ldom            '2 G memory to gold_ldom
# ldm add-vnet vnet1 vsw0 gold_ldom   'setup virtual network device vnet1
# ldm add-vdisk vdisk1 gold-boot-disk-vol@vds0 gold_ldom 'create vdisk1 from virtual device
#ldm add-vdsdev /Solaris/sol-10-u7-ga-sparc-dvd.iso cdrom@vds0      'Create a virtual CD from ISO
#ldm add-vdisk cdrom cdrom@vds0 gold_ldom                          'setup CDROM from iso image
#ldm set-variable boot-device="disk cdrom:f" gold_ldom             ' set boot device to CDROM
#ldm bind gold_ldom                                                  ' bind the resources
#ldm add-spconfig gold_ldom                                          'save the config
```

Boot the LDom and manually install the Solaris OS (Console port can be found from '# ldm list-bindings')

```
# svcadm enable vntsd                'start the console service if not running
# ldm start gold_ldom                 ' start the domain
# telnet localhost <console_port>    'telnet to the console port for the LDom
```

Walk through the Solaris installer selecting the volume exposed in the previous step as the device on which to install the operating system. (Likely c0d0 if only device configured). After the installation is complete and the image is at the state you wish to start all LDom's from stop the Dummy LDom and unbind the resources.

Rapid Provisioning using Boot Image Manager (BIM)

The Boot Image Manager provides a simplified and quick virtual server provisioning mechanism.

Download and Installation:

To download the latest version of the BIM go to the following link:

<http://www.symantec.com/connect/downloads/boot-image-management-utility-ldom-logical-domains-environments-beta>

This version of the utility works with the 5.0MP3RP2 (and later) stack. Please refer to the README.txt file before installing the binaries. To install the BIM, follow the step-by-step instructions in the README.txt file which is downloadable from the link above.

Usage Syntax:

Architecting for a Resilient, Cost Effective Logical Domain Infrastructure

Boot Image Manager provisions and manages the various Logical Domains and their boot images using the `vxbootadm` command.

The `vxbootadm` utility helps you perform the following tasks:

- Create an LDom interactively or automatically using a description file.
- Add a new boot image to an existing LDom.
- Remove a snapshot boot image attached to a LDom.
- Destroy an LDom.
- List the cache object statistics for all the cache objects housing Space Optimized Boot Images of LDoms.
- List the Boot Image bindings of all the LDoms provisioned using the BIM.

`vxbootadm` takes the following form:

```
# vxbootadm keyword arg ...
```

Where *keyword* selects the task to perform. Each keyword has different arguments.

Following are the recognized keywords and arguments:

```
create [-d descfile]
```

```
remove snapshot-name
```

```
destroy ldom
```

```
cache list
```

```
list [ldom-name]
```

```
vxbootadm create -g diskgroup -v goldvolname -l ldomname -i IP_address -h hostname -s
```

```
snap-type [ -f xml_file ]
```

```
help [ keywords | options ]
```

Examples:

- The keyword `create` along with the argument “`-d descfile`” allows you to create and provision an LDom with a boot image using a description file.

If the named LDom already exists, then it just provisions the boot image for that LDom.

```
# vxbootadm create -d /opt/VRTS/bimutility/input.txt
```

The above file path is the default description file path which is shipped as part of the BIM binaries.

It contains the following fields which serve as input to the `vxbootadm` utility:

```
bash # cat /tmp/input.txt
#
# Sample input file for vxbootadm create -d <file> option.
# Please modify the parameters as per your needs.
#
#
# dg - diskgroup in which all the boot image volumes will be contained.
#
dg:Boot_dg
#
# gi_name - gold image volume name which resides in $dg
#           this volume is the source for the snapshot
#           being created using the create option.
#
gi_name:gvol
#
# ldom - logical domain name to be provisioned as part of the create operation.
#
ldom:ldom1
#
# ip - IP address of the target ldom to be provisioned.
#
ip:ww.xx.yy.zz
#
# hostname - fully qualified hostname of the target LDom to be provisioned.
#
hostname:sfguest1.engba.symantec.com
```

Architecting for a Resilient, Cost Effective Logical Domain Infrastructure

cont...

```
#
# snaptype - snapshot type, can be instant full (use "full")
#           or instant space optimised (use "so").
snaptype:so

#
# xmlfile - location of the xmlfile to be used in order to
#           create the target ldom if it does not exist.
#           The default path of the file is specified below.
#           this is an optional argument, hence can be left
#           blank as well.
#
xmlfile:/usr/sbin/default.xml

#
# target physical hostname - name of the physical host on whose Control Domain
#                           the target LDom should be hosted.
#
trgt_host:sfcontroll.engba.symantec.com
```

The 'trgt_host' option allows you to provision a Logical Domain on a remote node in the CVM cluster. If this option is used, the vxbootadm should be run from the MASTER node of the cluster.

The following usage of vxbootadm allows you to provision a guest LDom using command-line options:

```
# vxbootadm create -g ldom_dg -v gold-boot-disk-vol -l apacheserver -i /
10.180.192.12 -h sfprod1.engba.symantec.com -s so -x /tmp/apache1.xml
```

The following usage of vxbootadm allows you provision a guest LDom on a remote node (Slave) in the CVM cluster using the -r remote_host option from the Master node:

```
# vxdctl -c mode
mode: enabled: cluster active - MASTER
master: sft5240-05.engba.symantec.com

# vxbootadm create -g ldom_dg -v gold-boot-disk-vol -l webserver -i \
10.180.192.13 -h sft5240-06.engba.symantec.com -s so -r \
sftphys1.engba.symantec.com -x /tmp/web.xml
```

Storage Savings from Space Optimized Snapshots

With the large number of LDomS being housed per physical server, the boot images housed per server is also significant. A single bare-metal Solaris Boot Image needs around 7-8 GB of space at a minimum. Installing software stacks and application binaries on top of that would require additional space. For the sake of this discussion, let's say every boot image requires around 15 GB of space to be able to house a database application.

When a user needs to provision a new LDom, the boot image for that LDom can be created by doing a full-copy approach or by taking a full snapshot of the gold image. The result of using these approaches is highly inefficient use of storage. To illustrate the impact on storage, let's say a physical server farm houses 100 LDomS. If the gold image is 15 GB, then the Boot Images for all the LDomS would consume 100 x 15 GB or approximately 1.5 TB of capacity. And to make the boot image highly available, the user may want to mirror them across two enclosures, doubling the capacity to 3 TB. This amount of highly available, high performance storage will be very expensive, and will likely eliminate the cost advantages that server virtualization would otherwise provide. To add to it, backup and recovery of such capacity will also be an expensive task.

In order to address the above issue, the BIM makes use of Space-Optimized (SO) snapshots of the gold image as Boot Images of the various LDomS. SO snapshots do not make a full copy of the data in the gold image, rather they work on copy-on-write principle where only the changed blocks are stored on a cache object store which is backed by physical store. Typically the cache object is 5-20% of the size of the parent volume (the gold image volume in this case). The same cache object can be used to store changed blocks for multiple snapshot volumes.

Taking the previous example and assuming the higher end of 20% per snapshot, the storage capacity requirements with a single boot image would be 0.2 x 15 GB or around 3 GB, or if you desire redundancy then around 6 GB, thus bringing down the storage needs drastically when compared to traditional methods. To look at how this savings can scale, take the original example of 3 TB required to redundantly host 100 LDom's, that would now equate to ((100 X 6) + 15) or 615 GB or a 79.5% reduction in storage required.

Each snapshot held in the cache object will only contain changes made to the gold image to support that installation's boot environment. To achieve the best possible reduction software should always be installed to data disks rather than root file systems. Effort should be made to control changes to the gold image operating files (ex: system, hosts, passwd, etc)

Pooling storage with BIM + CVM across multiple nodes

Using shared CVM volumes to house the boot images provides the following benefits:

- Decrease the boot image storage capacity needs for multiple physical servers housing multiple LDom's. The same gold image volume can be shared across multiple physical servers if it is a CVM volume part of a shared diskgroup.
- Faster failover in case of a fault in the LDom service group. A shared CVM diskgroup will already be imported on the target physical node making the failover of the LDom much faster.
- Centralized provisioning for multiple physical servers part of the same CVM cluster.

This is achieved using the ability to provision guest LDom's on a remote host in the CVM cluster from the MASTER node. The `'-r remote_host'` option of the vxbootadm CLI is used for this purpose.

Architecting for a Resilient, Cost Effective Logical Domain Infrastructure

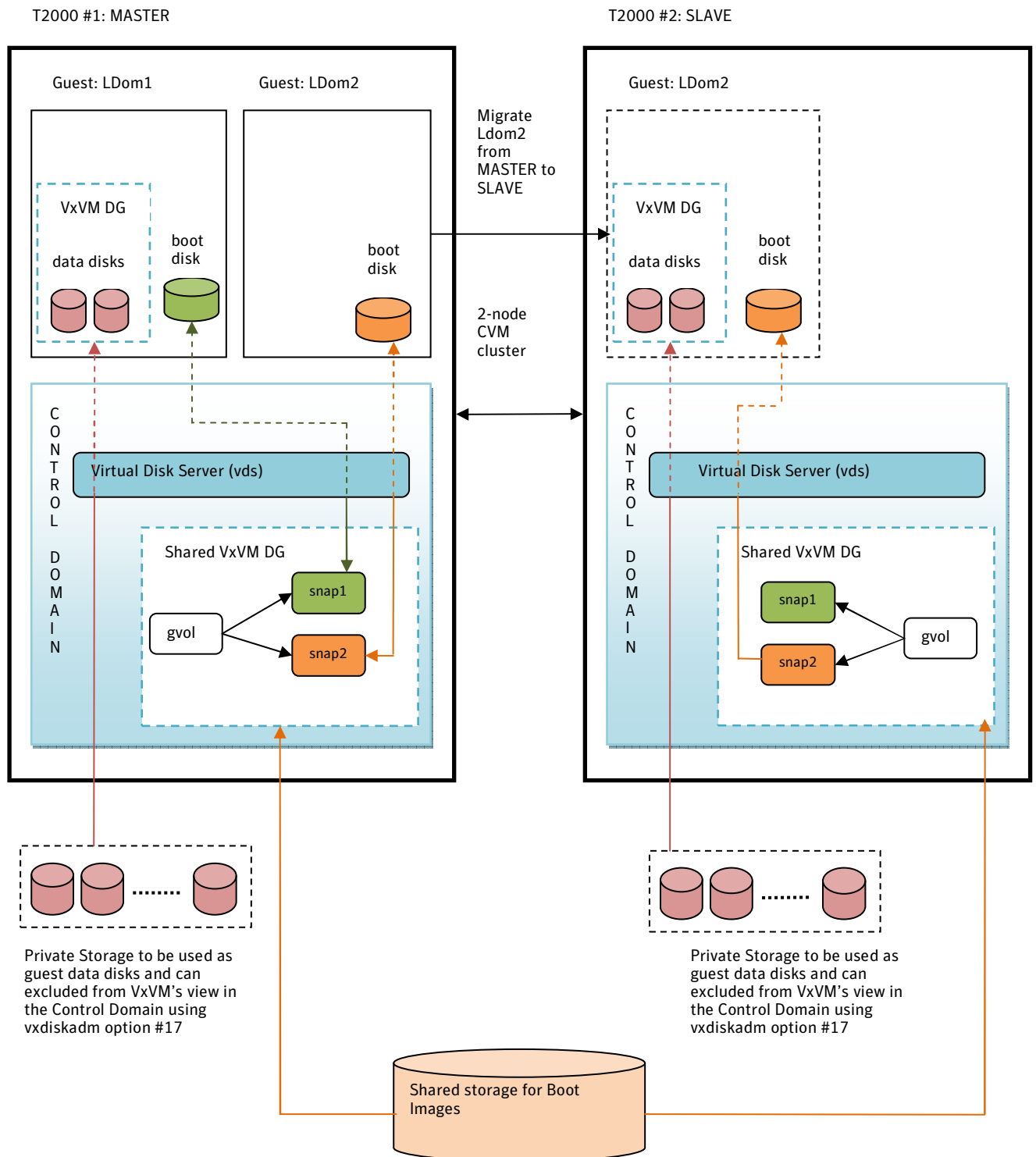


Figure 3: Configuration showing clustered BIM

Beyond the Boot Images

Considerations for Getting DATA Storage to Provisioned Guests (Choices choices and more choices)

Now that we can rapidly provision new LDom's using our Gold Boot Images and the BIM software, data devices must be considered. The LDom infrastructure allows numerous options to expose storage to guest domains, however not all of these options are equal in their effectiveness for general workloads. These options need to fit the administration model selected by the system administrator.

Typical design questions that will need to be answered are:

- Will there be different root administrators between the Control Domain and the individual Guest Domains? If so what will be the line of demarcation for the storage infrastructure (ie Who will manage what?) .
 - Will the Control Domain root admin responsibility end at the HBA port, the DMP device, or the Veritas Volume?
- Will Cluster File System be running in the Guest Domains?
- Will Guest Domains be moving between physical machines, or workloads moving between Guest Domains, or a combination of the both?
- Is there interest in using Cluster Volume Manager in the Control Domain to enable a faster failover of the guest domains, or will you be running VxVM / CVM in the Guest domains to perform application aware in-guest snapshot operations (ex: Oracle Database Snapshots)
- Will workloads be moving from current physical installations to a Guest Domain and if so have scripts been written to manage the local storage using Veritas commands.

Each of these answers will help determine what part of the storage management stack can be run at each layer of the LDom infrastructure. *In all data device configurations it is considered a BEST PRACTICE to host Boot Images in the Control Domain using the BIM tool.* Rather than explore every possible configuration option this paper will present three architectures that are most commonly implemented.

DATA access reference architectures:

- **Light Weight Guest Architecture**
 - Control Domain – DMP, VxVM or CVM to present volumes (or clustered volumes) to Guest
 - Guest Domain VxFS in Guest Domain for file system

This configuration offers:

- A light weight installation in the guest (only VxFS)
- The ability to utilize CVM in the Control domain for Fast Failover (described in a later section)
- Centralized DMP and Volume Management by the Control Domain root admin

Some Limitations of this architecture include:

- No VxVM snapshots in guest
- Limited thin storage support (no SmartMove / Reclamation)
- No SCSI 3 reservations in Guest (ie.. No I/O Fencing)
- No CFS in Guest

- **Full Stack in Guest**

- Control Domain – Devices directly passed from Control Domain to Guest Domain with no Veritas software in Data Device I/O path
 - VxVM / CVM will still manage and present volumes used as the boot devices from the Control Domain
- Guest Domain – Entire Veritas Storage Foundation stack

This configuration offers:

- Complete stack functionality in the guest
- Simplest administration model with all administration being performed in the guest
- Represent most similarity to physical server administration models
- Full support of scripts written in physical environments
- CFS support in Guest
 - I/O Fencing support (using Alternate I/O Domain)

Some Limitations of this architecture include:

- Cannot utilize CVM in the Control Domain for Fast Failover of data devices (can still use for boot images)

- **Centralized DMP**

- Control Domain – DMP for device presentation to Guest
 - VxVM / CVM will still manage and present volumes used as the boot devices from the Control Domain
- Guest Domain – Entire Veritas Storage Foundation stack

This configuration offers:

- Centralized Management of Multi-pathing by the Control Domain root admin
- Ability to run VxVM and VxFS in the Guest for legacy script support that uses both products.
- Full VxVM snapshot support

Some Limitations of this architecture include:

Architecting for a Resilient, Cost Effective Logical Domain Infrastructure

- Cannot utilize CVM in the Control Domain for Fast Failover of data devices (can still use for boot images)
- Only Active – Active array support (current as of this writing however this is likely to change so please verify HCL)
- No SCSI 3 reservations in the Guest Domain (ie.. No in guest I/O Fencing using coordinator lun's)
- CFS in Guest requires I/O fencing using Coordination Point Server based fencing

The latest details on each of these implementation methods can be found in the appropriate section of the *Application Note for Solaris Logical Domains* for the version of the Veritas software that is installed.

Architecting for a Resilient, Cost Effective Logical Domain Infrastructure

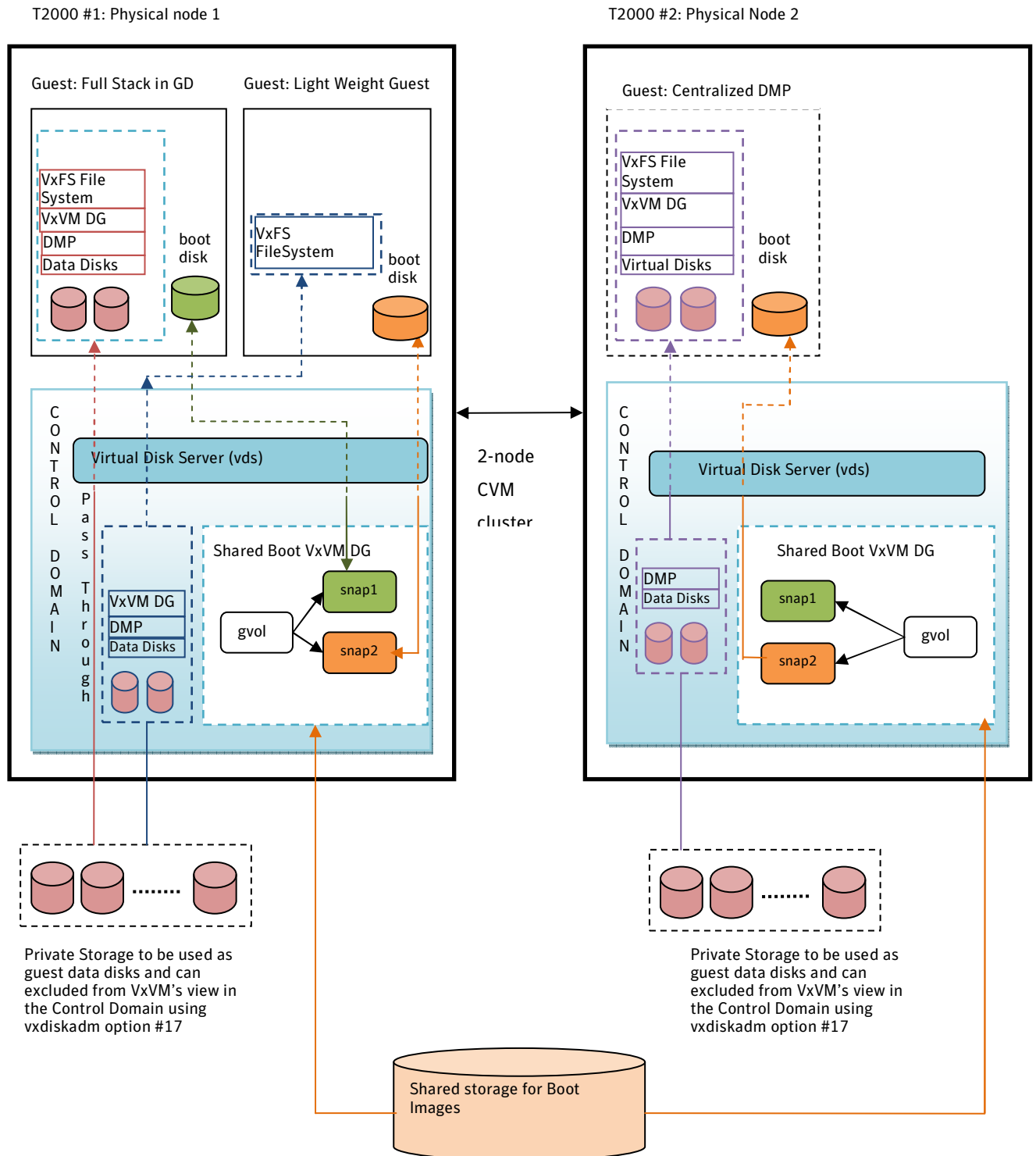


Figure 4: Examples of all Guest Architectures in a 2 node farm

Improving Application Resiliency with Veritas Cluster Server

LDom technology represents stacking multiple domains and instances on a single physical hardware device which in turn leads to multiple potential single points of failure. Thus application availability is at risk. Improving application resiliency in the event of hardware and / or software failure can be accomplished by implementing Veritas Cluster Server (VCS). VCS can be used both in the Control Domain as well as in the individual Guest Domains. Each installation point has its own advantages and both can work together to create a complete application support environment that offers more resiliency than a traditional architecture of a single physical node per application.

Managing LDom's in the Farm

VCS is implemented in the Control Domain to manage Guest Domain placement, provide farm level visibility, and ensure Guest Domains are balanced across physical assets in the farm in response to Control Domain detected resource failures.

VCS will manage Guest Domains as a single unit of control through the creation of a Service Group for each Guest Domain to be hosted.

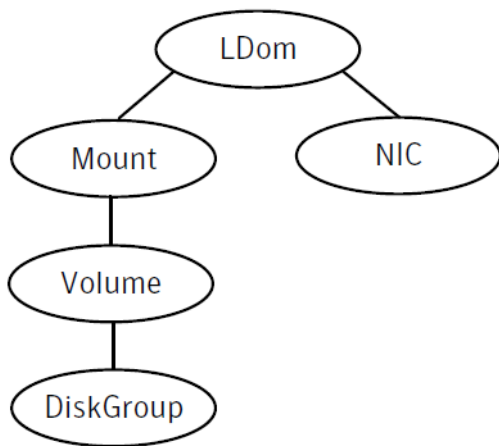


Figure 5: Example Resources in an LDom Service Group

The resources that are required for the Guest Domain to function will be configured by the system administrator in the Service Group. Examples of Control Domain resources that VCS would control include Boot Volumes, Data Devices, and Network Interfaces. VCS monitors the status of these resources that are required to start, run, and stop the Guest Domains and will provide automatic restart or move individual Guest Domains in response to failures. The system administrator should setup and configure Veritas Cluster Management Console which will provide Start / Stop / Monitor of individual Guest Domains from a common console across the entire farm.

VCS Workload Management (WLM) utilizing the load policy algorithm can control placement of an individual LDom configured to run on multiple physical nodes in active-active or active-passive configurations to ensure proper load balancing in the farm. Service Group dependencies are utilized to start Service Groups in the appropriate order for multi-tier application control. Further VCS Limits and Prerequisites can be used to ensure conflicting Logical Domains are not run on the same physical devices. For more information please review the VCS related information in the more information section.

VCS in the Control Domain can be configured to monitor or control applications running in Guest Domains through the combination of the remote group agent and the VCS in the Guest Domain architecture. This is covered in the next section.

Making the Farm Application Aware

Now that the Guest Domains are configured as VCS Service Groups attention must be turned to the application running within the Guest Domains. In many cases it is desired to monitor and manage the application state such that should the application fail within the Guest Domain, the application can be restarted or moved to another Domain or physical host. This is accomplished by installing Veritas Cluster Server in the Guest and configuring an application service group to start / stop / monitor and control the application within the Guest Domain. VCS can be configured to host the application in multiple Guest Domains in a failover or active-active (parallel) configuration. Additionally VCS can be configured to host the application in a physical to virtual or virtual to physical failover (or parallel) configuration.

Once the application service group is configured in the Guest Domain and the LDom Service Group is configured in the Control Domain these service groups can be connected using the Remote Group Agent (details in the Bundled Agents Reference Guide). This agent allows a resource to be configured that communicates with another VCS cluster to monitor, start or stop a service group in that remote cluster.

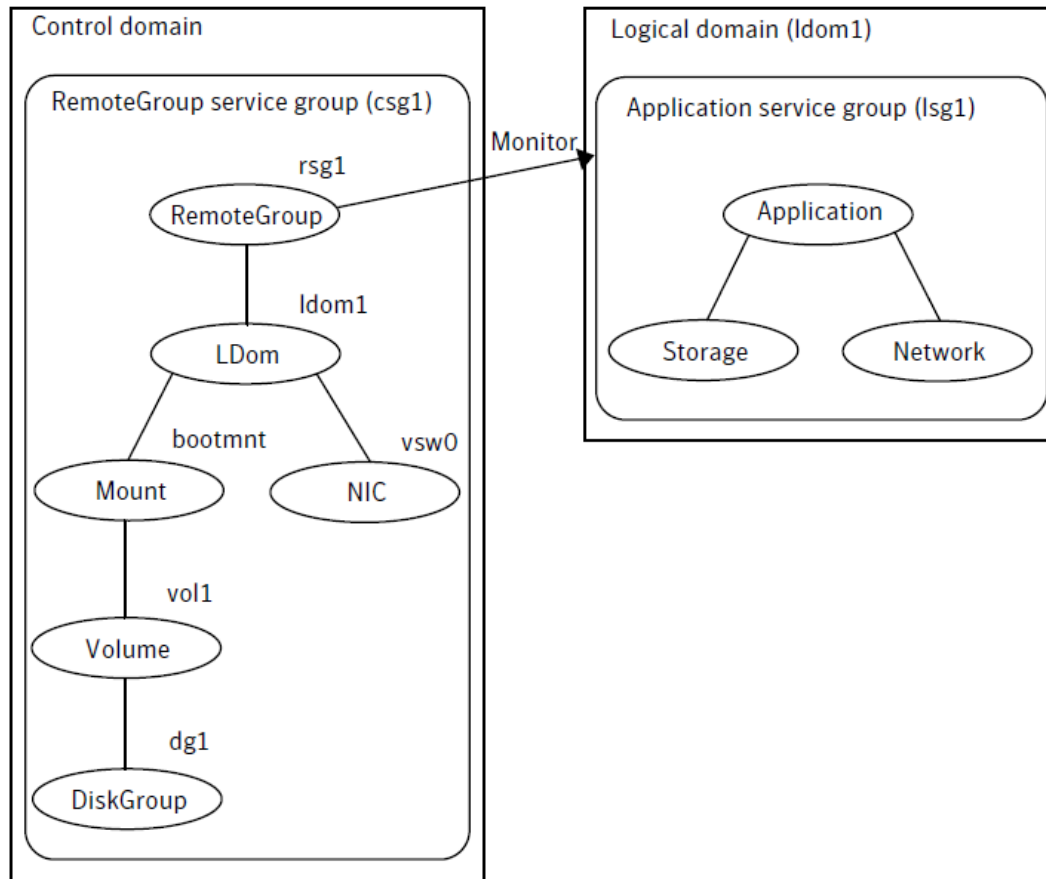


Figure 6: Example using Remote Group Agent to Control an Application

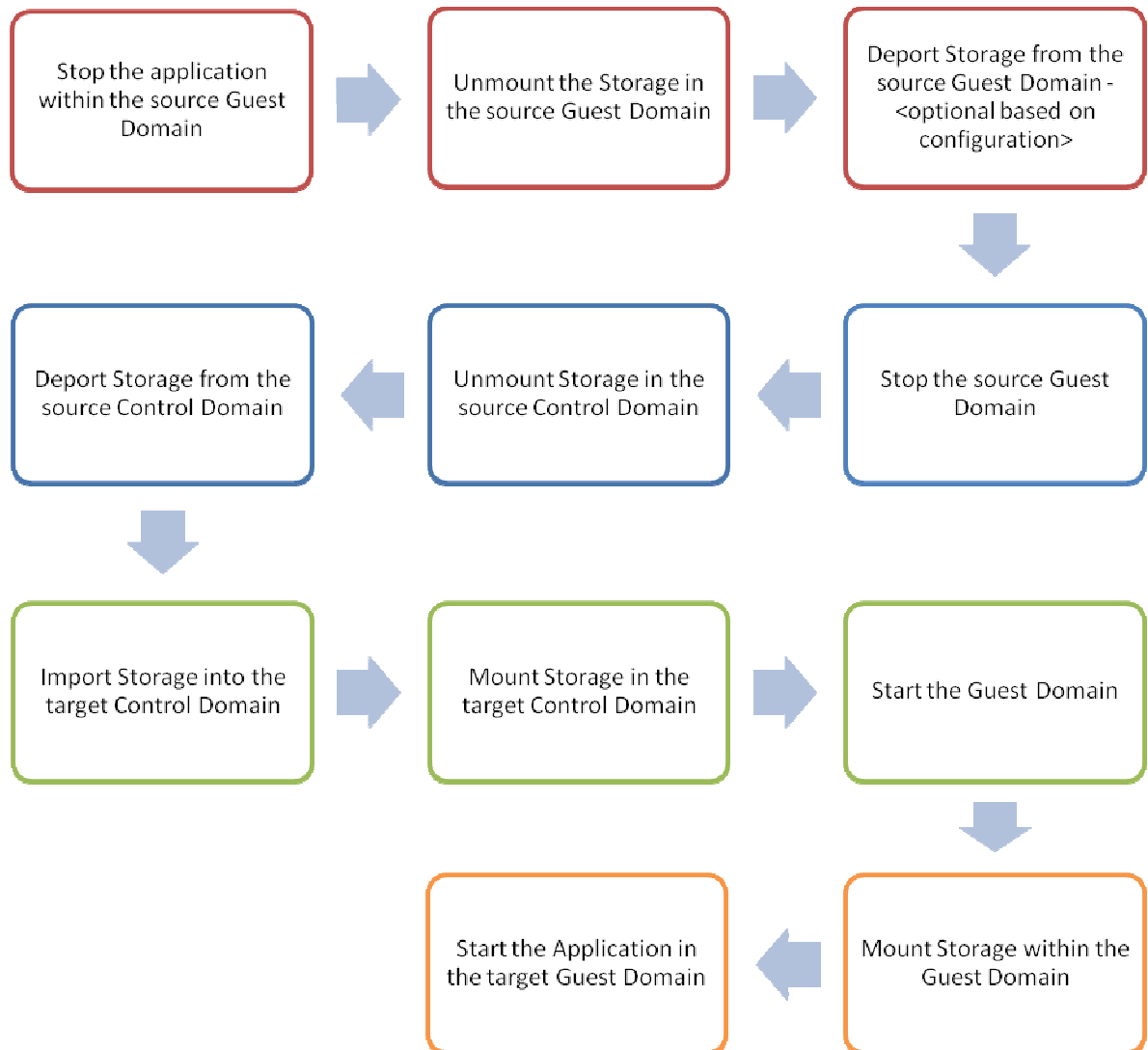
This complete implementation of multiple clusters can be managed with the Cluster Management Console. This console can provide visibility of the entire farm and allow administrators to better visualize the virtual and physical environments and their related interconnectivity.

Pooling Storage among Multiple Physical and Virtual Nodes and Advantages for Failover

Veritas Cluster Volume Manager and Cluster File System can be used within both the Control Domain and the Guest Domains to achieve faster failover architectures.

The operation of failing over a Guest Domain without Cluster Volume Manager or Cluster File System includes:

Note - For simplicity the following analysis ignores network resources as these resources start and stop rapidly, and multi-tier / single guest domain applications which would add some additional failover time associated with those stop and start actions.



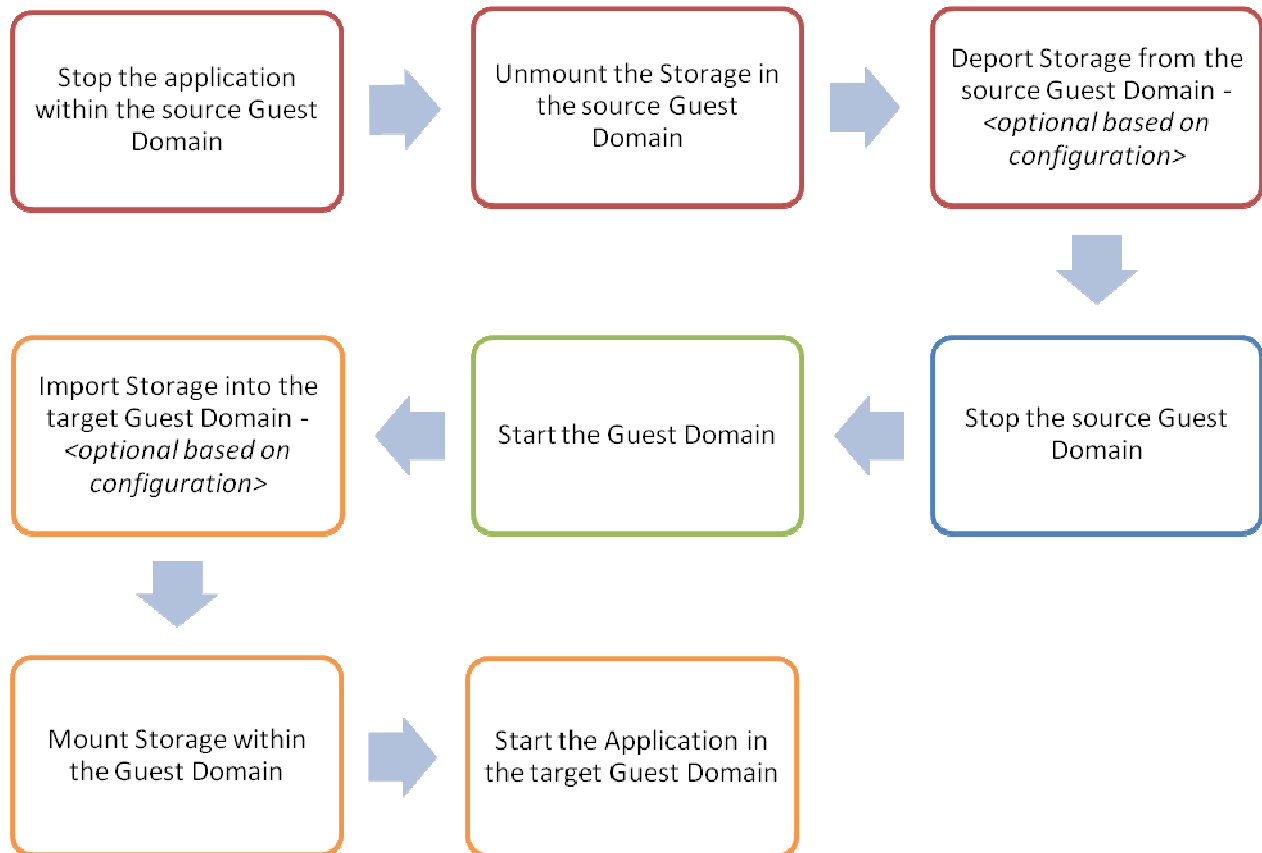
Fast Failover of Guest Domains

In order for a Guest domain to failover the storage hosting the boot image in a non-CVM implementation must be deported from the source physical node and imported on the target secondary node. This operation takes some amount of time and can be minimized through the implementation of Cluster Volume Manager which is configured to have all boot images available on all hosts within the farm at all times. Further in the “Light Weight Guest” architecture the Data Disk Groups can be imported on all hosts, and the Volumes can additionally be mounted in all Control Domains. This will expedite the Guest Domain failover and ensure all storage resources are online and ready to start the Guest immediately.

Architecting for a Resilient, Cost Effective Logical Domain Infrastructure

The resulting failover will now look like:

Note - For simplicity the following analysis ignores network resources as these resources start and stop rapidly, and multi-tier / single guest domain applications which would add some additional failover time associated with those stop and start actions.



Effectively removing the requirements to:

1. Unmount Storage in the source Control Domain
2. Deport Storage from the source Control Domain
3. Import Storage into the target Control Domain
4. Mount Storage in the target Control Domain

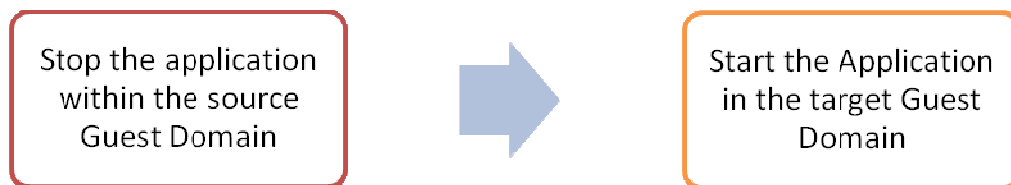
Fast Failover between Guest Domains (or to / from Physical Nodes)

Architecting for a Resilient, Cost Effective Logical Domain Infrastructure

In configurations where applications will be failed between two or more Guest Domains, Cluster File System can be utilized within those Guest Domains with all Guest Domains online at the same time to eliminate the time required to perform all storage operations and eliminate the Guest start up time.

The resulting failover thus becomes:

Note - For simplicity the following analysis ignores network resources as these resources start and stop rapidly, and multi-tier / single guest domain applications which would add some additional failover time associated with those stop and start actions.



Effectively removing the requirements to:

1. Unmount the Storage in the source Guest Domain
2. Deport Storage from the source Guest Domain - <optional based on configuration>
3. Stop the source Guest Domain
4. Unmount Storage in the source Control Domain
5. Deport Storage from the source Control Domain
6. Import Storage into the target Control Domain
7. Mount Storage in the target Control Domain
8. Start the Guest Domain
9. Mount Storage within the Guest Domain

This is the fastest failover architecture currently possible.

Customer Use Case Examples

The Boot Image Management Use Case:

Description: The Training Department was faced with a need to purchase new hardware to host classes for technology students, however the workloads hosted during these training sessions were fairly minimal and not at all performance dependant. They further desired a cost-effective rapid provisioning solution which would allow them to churn out multiple training images dynamically on as-need basis. Typically training sessions last for one week, and the following week a new class will begin. The systems need to be re-provisioned weekly to provide an OS image for each student as well as refreshed quickly and easily in the event a student makes an unrecoverable error. Each OS image is exactly the same in its base OS installation with Items like security lock down and common infrastructure software. Each class may have different binaries that will be needed, as well as some customer OS settings that will need to be applied following the base build.

Implementation Details: When looking at available hardware it was recognized that current market minimal server hardware provided significantly more horsepower than was required to host individual student environments by approximately a factor of 4. In an effort to reduce cost the server admin team decided to utilize Sun Logical Domain virtualization technology to provide Operating System environment isolation for each student and to stack 4 images on each physical node for a total of 4 physical nodes. To achieve the rapid provisioning needs the training admin utilized Storage Foundation Cluster Volume Manager and the BIM utility to build a 15GB Solaris boot image in a shared disk group (DG) which was shared across all 4 physical nodes and used to (re)-provision new student operating environments. Each OS image is provisioned as a Space Optimized Snapshot utilizing a shared cache object and as much as possible application binaries are installed to directories that are mount points with underlying shared storage to minimize cache object utilization. To add resiliency to the Golden Image the sys-admin mirrored the Gold Image with 3 total copies. One mirror was broken off to act as the online backup copy for rapid recovery in the event any root device corruption occurs. Finally a shared image was created on Cluster File System hosted on shared storage and exposed to all nodes to hold required student files to be used in each training class. This reduces the need for multiple copies as well as minimizes the size of the Gold Image. It also ensures that as the software requirement change over time it can be changed without having to modify the Gold Image.

Results: The training department was able to achieve this 4 x 4 implementation which enabled them to purchase $\frac{1}{4}$ the number of physical assets they would have had to acquire to host using dedicated OS instances by implementing LDomS. They were further able to achieve the rapid provisioning requirements with the implementation of Storage Foundation with the BIM utility. They were additionally able to reduce their storage requirements for boot images from 4 x 4 x 15G = 240 G to a net total of 61G (composed of a 15 G x 3 (mirror) + 16G of cache object storage (varies over time based on root installed software)) which equates in their environment to an approximate 75% reduction in storage for boot images. As an unintended by-product the Training Department was also able to save on software licensing costs associated with the software run within the training instances as each instance held fewer CPU's and rather than a need to license 16 physical nodes, they only needed to license 4.

The Production High Availability Use Case:

Description: The IT Department was faced with deploying four new business critical applications all with database back-ends. While they didn't expect a significant amount of daily throughput in any single given database, it was considered to be a design requirement to achieve the highest level of availability possible with a design MTTR (Mean Time To Recovery) for local availability of 15 minutes, an RPO for DR of 4 hours, and a "four 9's" of availability requirement including maintenance windows (99.99% yields 87 hours of annual downtime structured in this organization as one 7 hour maintenance window per month). The configuration under consideration were 4 x 2- physical node clusters for the database layer along with an additional requirement of 4 x 2 physical nodes for the application layer, however this was presenting a significant cost factor for the organization so alternative configurations were sought. This use case was designed to achieve maximum uptime for production applications while minimizing the User impact associated with Provisioning, Boot Image Management, Ldom Management, and Device Multipathing.

Implementation Details: Rather than implementing 16 physical nodes in a 4 x 2 + 4 x 2 configuration it was decided to implement a single 4 node cluster using Logical domains to host the individual workloads. Upon considering the workload from all of the applications and database instances it was determined that the databases consumed approximately twice the amount of system resources of a single application. Further the systems possessed enough resource to handle 2.5 database instances. Thus the farm was designed using Veritas Cluster Server's Service Group Workload Management by setting FailOverPolicy = Load and setting System Capacity = 500 and the Service Group Load attribute to 200 for databases and 100 for application Service Groups. This configuration results in a "soft limit" of 2 database and 1 application service group on any host at any given time, however in LDom's where the system resources must be dedicated to a particular LDom with no ability to handle a "soft limit" we need to include the ability to ensure a "hard" limit. To accomplish this the system administrator implemented Veritas Cluster Server Limit's and Pre-Requisites to control the service group placement. Each system was given a limit of 5, each database service group was given a pre-requisite limit of 2 and each application service group was given a limit of 1. Rather than specifying a defined location for Service Groups to start, the System Administrators wanted to utilize the Cluster Engine's capability to balance the load within the LDom farm in an automated fashion. Thus they set the AutoStartPolicy to Load as well to allow VCS to determine the best server upon which to start each Service Group by analyzing which server has the highest Available Capacity and sufficient Prerequisites.

Due to the stacking of diverse applications on a single set of hardware, the solution also required an architecture that supported separate root account control where the admin of the Control Domain was not necessarily the admin of the LDom Operating System(s). The Guest Domain root admin's installed and configured VCS in a single node fashion in each LDom setting up VCS to automatically start and monitor the database or application. The Control Domain root administrator then configured the LDom service groups with a critical resource utilizing the Remote Group agent to monitor the service groups running in the Control Domain.

In this environment it was determined that Application A must be restarted when Database A fails, thus Service Group dependencies are also set between the Database A and application A service groups to ensure that the application is restarted in the event of a database failure. This was accomplished using an *Online Global Firm* relationship where the database is the Child and the Application is the parent. Where *Online* indicates the parent must wait for the child to be online before it starts, *Global* indicates that the Child may be running on any system in the cluster, and *Firm* indicates that the parent will restart on child failure but the child will not restart on parent failure nor can the child be taken offline while the parent is online. More on Service Group Dependencies can be found in the VCS User Guide.

In an effort to achieve maximum availability and minimize the time associated with individual LDom failover the Control Domain administrator configured the LDom boot images using BIM to be Full Snapshots and used Cluster Volume Manager to share the boot image volumes as parallel VCS resources on all nodes to eliminate the time required to failover the boot images during the LDom failure. The Control Domain root then created volumes to hold application and database data and exposed them to the LDom's and shared the volumes via Cluster Volume manager on all nodes configuring them as parallel resources under VCS control. The Guest Domain root administrators saw these devices within their LDom as disk devices and simply created filesystems on them using VxFS for the applications and database to use.

Results: The customer was able to reduce the environment from a 16 node implementation down to a 4 node environment saving the costs by associated with the server hardware, the database and application licensing, fewer network ports, and fewer storage fiber-channel connectivity resources. Meanwhile the availability observed over the first several months of production was well within the original design requirements. In fact the maintenance windows have been shown to be significantly lower as they are able to easily evacuate a full node at a time for maintenance and are able to roll upgrades into the cluster.

Consolidated list of Recommended Best Practices

- Boot Images
 - Manage Boot Images in the Control Domain using the Boot Image Management utility with VxVM for rapid provisioning, roll back, and space savings. Use with CVM across nodes for greater space savings, and fast failover
 - Utilize Space Optimized Snapshots for environments that:
 - Will be re-provisioned often
 - Will be re-provisioned rather than upgraded
 - Lower Cost of implementation is desired
 - Do not require highest levels of availability
 - Always store Boot Gold Images and Cache Objects on redundant storage
 - Minimize the amount of post installation configuration that will be done in Boot Images to minimize post-installation tasks and reduce consumption of space in Space Optimized Snapshots
 - Preinstall software required by all nodes
 - Consider and pre-add all required Users, Hosts, etc/system settings etc.
 - Place application binaries, configuration files, and other items added after provisioning or files likely to change over time on Data Storage rather than allowing those items to reside in the boot file systems.
- Clustering
 - Utilize Cluster Management Console for visibility of the Entire Farm through a single console
 - Implement VCS in Control Domain as well as Guest Domains for end to end availability

- Utilize VCS workload management functionality in combination with Limits and Prerequisites to automatically balance load in the LDom farm.

Appendix

Where to go for more info:

Sun Beginners Guide to LDom's:

www.sun.com/blueprints/0207/820-0832.pdf

Sun LDom Admin Guide:

<http://docs.sun.com/app/docs/doc/820-4913-10>

BIM Symantec Connect Site:

<http://www.symantec.com/connect/downloads/boot-image-management-utility-ldom-logical-domains-environments-beta>

Veritas Application Support Note for Solaris Logical Domains:

<http://seer.entsupport.symantec.com/docs/307635.htm>

VCS User Guide

http://sfdoccentral.symantec.com/sf/5.0MP3/solaris/pdf/vcs_users.pdf

Bundled Agent Reference Guide:

http://sfdoccentral.symantec.com/sf/5.0MP3/solaris/pdf/vcs_bundled_agents.pdf

All Veritas Storage Foundations Documentation:

<http://sfdoccentral.symantec.com/>

All Storage Foundations for Solaris 5.0 MP3 documentatio:

http://sfdoccentral.symantec.com/Storage_Foundation_HA_50MP3_Solaris.html

About Symantec

Symantec is a global leader in providing security, storage and systems management solutions to help businesses and consumers secure and manage their information. Headquartered in Mountain View, Calif., Symantec has operations in 40 countries. More information is available at www.symantec.com.

For specific country offices and contact numbers, please visit our Web site. For product information in the U.S., call toll-free 1 (800) 745 6054.

Symantec Corporation
World Headquarters
350 Ellis Street
Mountain View, CA 94043 USA
Phone: +1 650-527-8000
www.symantec.com

Copyright © 2010 Symantec Corporation. All rights reserved.
Symantec and the Symantec logo are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

02/10