Sun™ Cluster 3.0 Administration
ES-333

Student Guide

Sun Microsystems, Inc.
UBRM05-104
500 Eldorado Blvd.
Broomfield, CO 80021
U.S.A.

February 2001, Revision A

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.

Table of Contents

Volume Management Using Solstice DiskSuite 8-1

About This Course

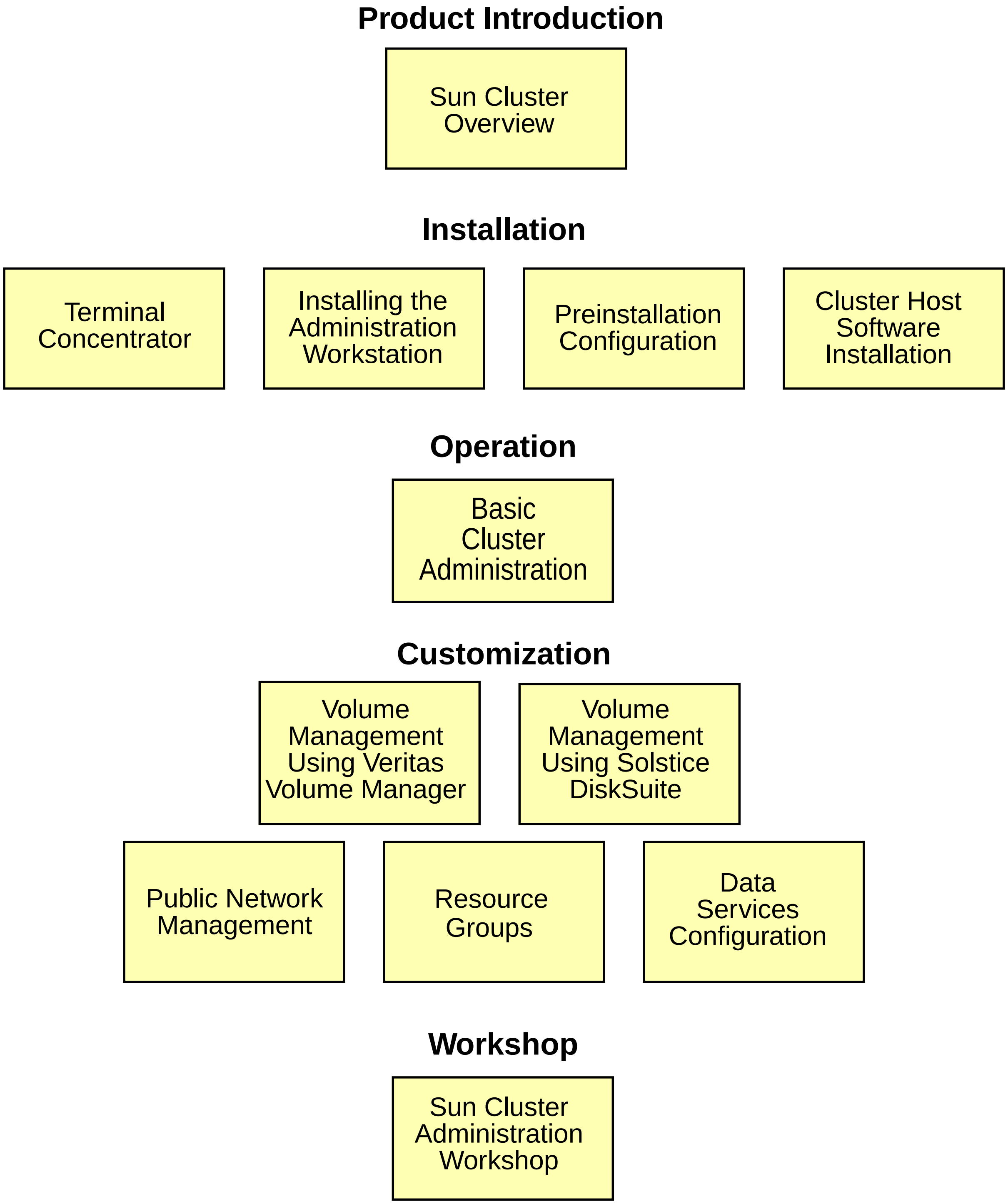
Course Goals

Upon completion of this course, you should be able to:

- Describe the major Sun™ Cluster components and functions
- Perform pre-installation configuration verification
- Configure the Terminal Concentrator
- Configure a cluster administration workstation
- Install the Sun Cluster 3.0 software
- Configure Sun Cluster 3.0 quorum devices
- Create network adapter failover (NAFO) groups
- Install Sun Cluster 3.0 data service software
- Configure global file systems
- Configure a failover data service resource group
- Configure a scalable data service resource group
- Configure the Sun Cluster 3.0 high-availability (HA) for network file system (NFS) failover data service
- Configure the Sun Cluster 3.0 HA for Apache scalable data service
- Use Sun Cluster administration tools

Course Map

The following course map enables you to see what you have accomplished and where you are going in reference to the course goals.



Product Introduction

Sun Cluster Overview

Installation

Terminal Concentrator

Installing the Administration Workstation

Preinstallation Configuration

Cluster Host Software Installation

Operation

Basic Cluster Administration

Customization

Volume Management Using Veritas Volume Manager

Volume Management Using Solstice DiskSuite

Public Network Management

Resource Groups

Data Services Configuration

Workshop

Sun Cluster Administration Workshop

Topics Not Covered

This course does not cover the topics shown on the overhead. Many of the topics listed on the overhead are covered in other courses offered by Sun Educational Services:

- Database installation and management – Covered in database vendor courses
- Network administration – Covered in SA-380: *Solaris 2.x Network Administration*
- Solaris administration – Covered in SA-238: *Solaris 8 Operating Environment System Administration I* and SA-288: *Solaris 8 Operating Environment System Administration II*
- Disk storage management – Covered in ES-220: *Disk Management With DiskSuite*, and ES-310: *Volume Manager with Sun StorEdge*

Refer to the Sun Educational Services catalog for specific information and registration.

How Prepared Are You?

To be sure you are prepared to take this course, can you answer yes to the following questions?

- Can you explain virtual volume management terminology, such as mirroring, striping, concatenation, volumes, and mirror synchronization?
- Can you perform basic Solaris™ Operating Environment administration tasks, such as using `tar` and `ufsdump` commands, creating user accounts, formatting disk drives, using `vi`, installing the Operating Environment, installing patches, and adding packages?
- Do you have prior experience with Sun hardware and the OpenBoot™ programmable read-only memory (PROM) technology?
- Are you familiar with general computer hardware, electro-static precautions, and safe handling practices?

Introductions

Now that you have been introduced to the course, introduce yourself to each other and the instructor, addressing the item shown in the bullets below.

- Name
- Company affiliation
- Title, function, and job responsibility
- Experience related to topics presented in this course
- Reasons for enrolling in this course
- Expectations for this course

How to Use Course Materials

To enable you to succeed in this course, these course materials employ a learning model that is composed of the following components:

- **Goals** – You should be able to accomplish the goals after finishing this course and meeting all of its objectives.
- **Objectives** – You should be able to accomplish the objectives after completing a portion of instructional content. Objectives support goals and can support other higher-level objectives.
- **Lecture** – The instructor will present information specific to the objective of the module. This information will help you learn the knowledge and skills necessary to succeed with the activities.
- **Activities** – The activities take on various forms, such as an exercise, self-check, discussion, and demonstration. Activities are used to facilitate mastery of an objective.
- **Visual aids** – The instructor might use several visual aids to convey a concept, such as a process, in a visual form. Visual aids commonly contain graphics, animation, and video.

Conventions

The following conventions are used in this course to represent various training elements and alternative learning resources.

Icons

Additional resources – Indicates other references that provide additional information on the topics described in the module.

Discussion – Indicates a small-group or class discussion on the current topic is recommended at this time.

Note – Indicates additional information that can help students but is not crucial to their understanding of the concept being described. Students should be able to understand the concept or complete the task without this information. Examples of notational information include keyword shortcuts and minor system adjustments.

Caution – Indicates that there is a risk of personal injury from a nonelectrical hazard, or risk of irreversible damage to data, software, or the operating system. A caution indicates that the possibility of a hazard (as opposed to certainty) might happen, depending on the action of the user.

Warning – Indicates that either personal injury or irreversible damage of data, software, or the operating system will occur if the user performs this action. A warning does not indicate potential events; if the action is performed, catastrophic events will occur.

Typographical Conventions

`Courier` is used for the names of commands, files, directories, programming code, and on-screen computer output; for example:

> Use `ls -al` to list all files.
>
> `system% You have mail`.

`Courier` is also used to indicate programming constructs, such as class names, methods, and keywords; for example:

> The `getServletInfo` method is used to get author information.
>
> The `java.awt.Dialog` class contains `Dialog` constructor.

`Courier bold` is used for characters and numbers that you type; for example:

> To list the files in this directory, type:
>
> `#` **`ls`**

`Courier bold` is also used for each line of programming code that is referenced in a textual description; for example:

> `1 import java.io.*;`
>
> **`2 import javax.servlet.*;`**
>
> `3 import javax.servlet.http.*;`
>
> Notice the `javax.servlet` interface is imported to allow access to its life cycle methods (Line 2).

`Courier italics` is used for variables and command-line placeholders that are replaced with a real name or value; for example:

> To delete a file, use the `rm` *`filename`* command.

`Courier italic bold` is used to represent variables whose values are to be entered by the student as part of an activity; for example:

> Type **`chmod a+rwx`** ***`filename`*** to grant read, write, and execute rights for filename to world, group, and users.

Palatino italics is used for book titles, new words or terms, or words that you want to emphasize; for example:

> Read Chapter 6 in the *User's Guide*.
>
> These are called *class* options.

Additional Conventions

Java™ programming language examples use the following additional conventions:

- Method names are not followed with parentheses unless a formal or actual parameter list is shown; for example:

  "The `doIt` method..." refers to any method called `doIt`.

  "The `doIt()` method..." refers to a method called `doIt` that takes no arguments.

- Line breaks occur only where there are separations (commas), conjunctions (operators), or white space in the code. Broken code is indented four spaces under the starting code.
- If a command used in the Solaris™ Operating Environment is different from a command used in the Microsoft Windows platform, both commands are shown; for example:

  If working in the Solaris Operating Environment

  ```
  $CD SERVER_ROOT/BIN
  ```

  If working in Microsoft Windows

  ```
  C:\>CD SERVER_ROOT\BIN
  ```

Module 1

Sun™ Cluster Overview

Objectives

Upon completion of this module, you should be able to:

- List the new Sun™ Cluster 3.0 features
- List the hardware elements that constitute a basic Sun Cluster system
- List the hardware and software components that contribute to the availability of a Sun Cluster system
- List the types of redundancy that contribute to the availability of a Sun Cluster system
- Identify the functional differences between failover, scalable, and parallel database cluster applications
- List the supported Sun Cluster 3.0 data services
- Explain the purpose of Disk ID (DID) devices
- Describe the relationship between system devices and the cluster global namespace
- Explain the purpose of resource groups in the Sun Cluster environment
- Describe the purpose of the cluster configuration repository
- Explain the purpose of each of the Sun Cluster fault monitoring mechanisms

Relevance

Discussion – The following questions are relevant to understanding the content of this module:

- Which can contribute more to system availability, hardware or software?
- Why is a highly available system usually more practical than a fault-tolerant system?

Additional Resources

Additional resources – The following references can provide additional information on the topics described in this module:

- *Sun Cluster 3.0 Installation Guide*, part number 806-1419
- *Sun Cluster 3.0 Hardware Guide*, part number 806-1420
- *Sun Cluster 3.0 Data Services Installation and Configuration Guide*, part number 806-1421
- *Sun Cluster 3.0 Data Service Developers Guide*, part number 806-1422
- *Sun Cluster 3.0 System Administration Guide*, part number 806-1423
- *Sun Cluster 3.0 Concepts*, part number 806-1424
- *Sun Cluster 3.0 Error Message Guide*, part number 806-1426
- *Sun Cluster 3.0 Release Notes*, part number 806-1428

Sun Cluster 3.0 Features

The Sun Cluster 3.0 software release has the following new features:

- Up to eight nodes

  Sun Cluster 3.0 supports up to eight cluster nodes.

- Cluster File System

  The Cluster File System allows mounting of cluster-wide user file systems (UFS) or High Sierra file systems (HSFS), allowing concurrent, continuous access to the file systems from any node in the cluster.

- Global device access

  Sun Cluster can access disk devices, tape drives, and compact disc read-only memory (CD-ROM) drives from any node in the cluster.

- Cluster networking (shared address)

  While each node retains its own publicly accessible Internet Protocol (IP) address, a global IP address can be configured for the applications on the cluster, where the data service requests received through the global address are distributed to different nodes in the cluster based on a selected load balancing policy.

- Scalable application support

  Sun Cluster 3.0 support scalable data services in which client requests are distributed to any number of cluster nodes. This is used in conjunction with the shared address feature.

- Sun Management Center-based monitoring

  Sun Cluster 3.0 nodes can be monitored using the Sun Management Center system management tool.

- Solaris 8 10/00 Operating Environment is supported

  The Solaris 8 10/00 platform release is recommended because the number of required patches has been significantly reduced.

- Two new installation methods

  You can now configure the first cluster node and automatically use it to assist in configuring all additional nodes. If you have an existing JumpStart™ server, you can also automatically add the Sun Cluster 3.0 installation to the JumpStart configuration.

Cluster Hardware Components

The minimum hardware components that are necessary for a cluster configuration include:

- One administration workstation
- One Terminal Concentrator
- Two hosts (up to eight)
- One or more public network interfaces per system (not shown)
- A private cluster transport interface
- Dual hosted, mirrored disk storage

The physical structure of a minimum cluster is shown in Figure 1-1.

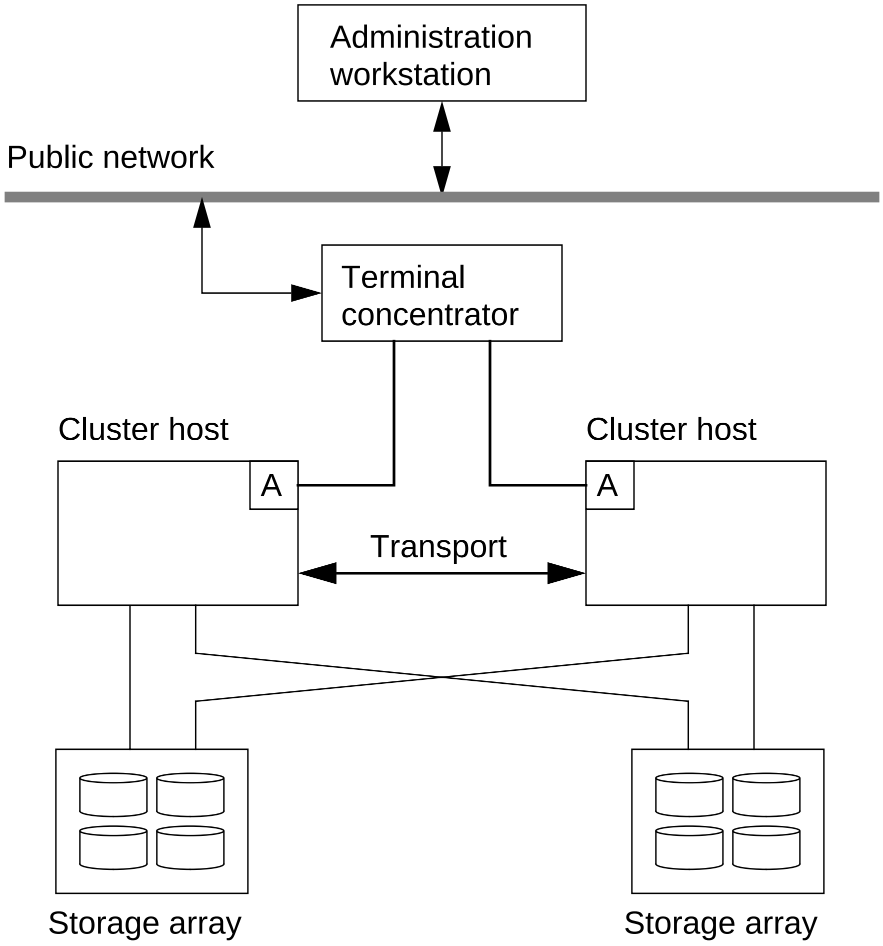


Figure 1-1 Cluster Physical Connections

Administration Workstation

The administration workstation can be any Sun workstation, providing it has adequate resources to support graphics and compute intensive applications. You can use cluster administration tools to monitor many clusters from one administration workstation.

Terminal Concentrator

The Sun terminal concentrator (TC) provides direct translation from the network to serial port interfaces. Each of the serial port outputs connects to a separate node in the cluster through serial port A. Because the nodes commonly do not have frame buffers, this is the only access path when the operating system is down.

Cluster Host Systems

A wide range of Sun hardware platforms are supported for use in the clustered environment. Mixed platform clusters are not supported.

Cluster Transport Interface

All nodes in a cluster are linked by a private cluster transport. The transport is redundant and can be used for the following purposes:

- Cluster-wide monitoring and recovery
- Parallel database lock and query information
- Global data access

Cluster Disk Storage

The Sun Cluster environment can use several Sun storage. They must all accept at least dual-host connections.

Note – Although some storage array models can physically accept up to four host system connections, the Sun Cluster 3.0 release supports a maximum of two host connections to a storage array.

Sun Cluster High Availability Features

The Sun Cluster system focuses on providing reliability, availability, and scalability. Part of the reliability and availability is inherent in the cluster hardware and software.

High Availability Hardware Design

Many of the supported cluster hardware platforms have the following features that contribute to maximum uptime:

- Hardware is interchangeable between models.
- Redundant system board power and cooling modules.
- The systems contain automatic system reconfiguration; failed components, such as the central processing unit (CPU), memory, and input/output (I/O), can be disabled at reboot.
- Several disk storage options support hot swapping of disks.

Sun Cluster High Availability Software

The Sun Cluster software has monitoring and control mechanisms that can initiate various levels of cluster reconfiguration to help maximize application availability.

Software RAID Technology

The Veritas Volume Manager and Sun Solstice DiskSuite™ software provide redundant array of independent disks (RAID) protection in redundant mirrored volumes.

Controller-based RAID Technology

The Sun StorEdge™ A3500 uses controller-based RAID technology that is sometimes referred to as hardware RAID.

Sun Cluster Data Service Support

Each of the Sun Cluster data services (Sun Cluster 3.0 Agents), provides a control and monitoring framework that enables a standard application to be highly-available or scalable.

Some of the data services can be configured for either failover or scalable operation.

Highly Available and Scalable Data Service Support

The Sun Cluster software provides preconfigured components that support the following high-availability (HA) data services:

- Sun Cluster HA for Oracle (failover)
- Sun Cluster HA for iPlanet (failover or scalable)
- Sun Cluster HA for Netscape Directory Server (failover)
- Sun Cluster HA for Apache (failover or scalable)
- Sun Cluster HA for DNS (failover)
- Sun Cluster HA for NFS (failover)

Parallel Database Support

The Sun Cluster software provides support for the Oracle Parallel Database application.

High-Availability Strategies

To help provide the level of system availability required by many customers, the Sun Cluster system uses the following strategies:

- Redundant servers
- Redundant data
- Redundant public network access
- Redundant private communications (transport)
- Multi-host storage access

The illustration in Figure 1-2 shows the location and relationship of the high-availability strategies.

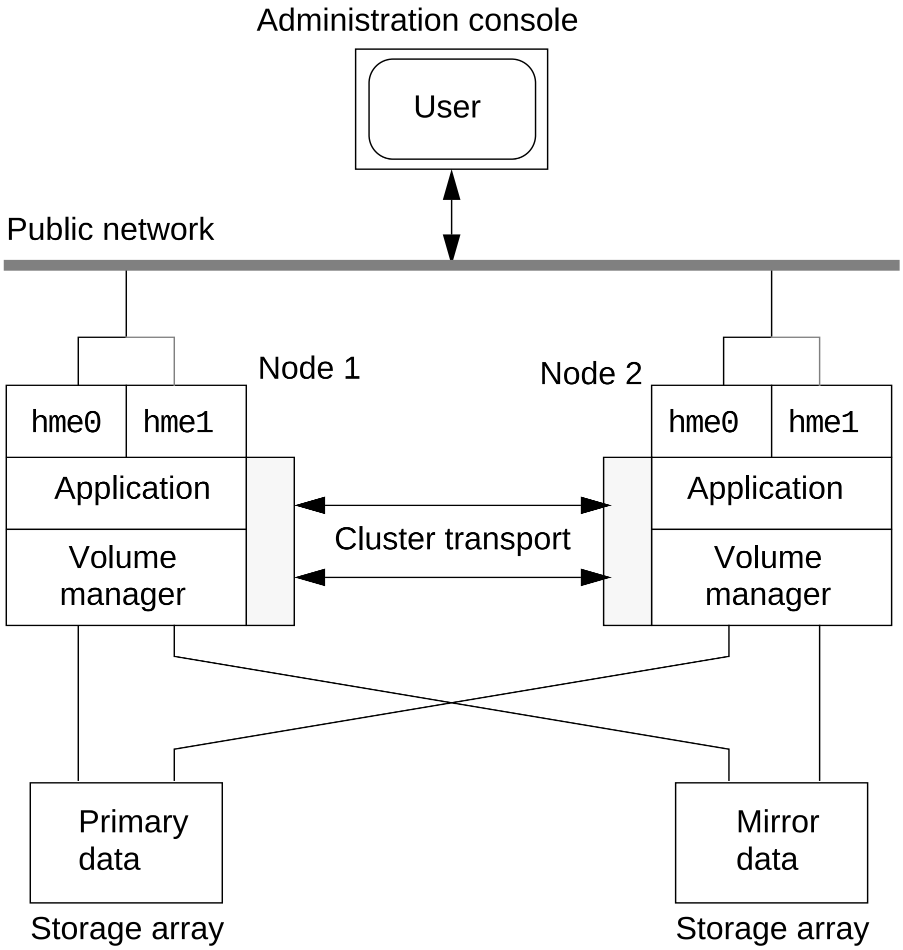


Figure 1-2 Cluster Availability Strategies

Redundant Servers

The Sun Cluster system consists of two to eight interconnected systems that are referred to as cluster host systems or nodes. The systems can be any of a range of Sun platforms and use off-the-shelf non-proprietary hardware.

Note – You cannot mix systems that use peripheral component interconnect (PCI) bus technology, such as the Sun Enterprise™ 450 server, with SBus technology systems, such as the Sun Enterprise 3500 server.

Redundant Data

A Sun Cluster system can use either of two virtual volume management packages to provide data redundancy. The use of data mirroring provides a backup in the event of a disk drive or storage array failure.

Redundant Public Network Interfaces

The Sun Cluster system provides a proprietary feature, public network management (PNM), that can transfer user I/O from a failed network interface to a predefined backup interface. The switch to the backup interface is transparent to cluster applications and users.

Redundant Transport Interface

The cluster transport interface consists of dual high-speed private node-to-node communication interfaces. The cluster software uses only one of the interfaces at a time. If the primary interface fails, the cluster software automatically switches to the backup. This is transparent to cluster applications.

Sun Cluster High-Availability Failover

The Sun Cluster high-availability failover features include the following:

- Failover applications
- Node fault monitoring
- Network fault monitoring
- Data service fault monitoring

Failover Applications

As shown in Figure 1-3, a failover application (data service) runs on a single cluster node. If there is a node failure, a designated backup node takes over the application that was running on the failed node.

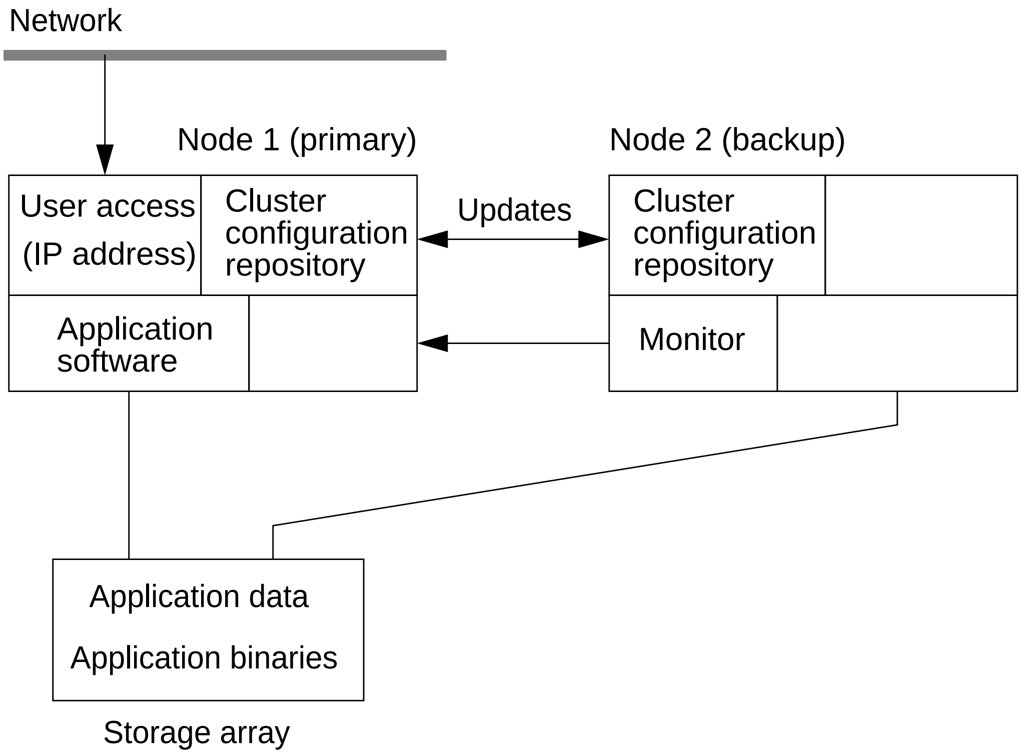


Figure 1-3 Failover Data Service Features

Node Fault Monitoring

The cluster membership monitor (CMM) is kernel-resident on each node and detects major cluster status changes, such as loss of communication between one or more nodes. The CMM instances communicate with one another across the private high-speed network interfaces by sending regular heartbeat messages. If the heartbeat from any node is not detected within a defined timeout period, it is considered as having failed and a cluster reconfiguration is initiated to renegotiate cluster membership.

Note – The cluster membership negotiation process is relatively complex and is described in Module 4, "Preinstallation Configuration."

Network Fault Monitoring

Both the public network adapter failover (NAFO) interfaces and the cluster transport interfaces are monitored for potential failures.

Public Network Management

The PNM daemon, `pnmd`, monitors the functionality of NAFO group network interfaces and can transparently switch to backup interfaces in the event of a failure.

Cluster Transport Monitoring

The cluster transport interfaces are monitored on each node. If the active interface on any node is determined to be inoperative, all nodes switch to the backup interface and attempt to reestablish communication.

Data Service Fault Monitoring

Each Sun-supplied data service, such HA for NFS, has predefined fault monitoring routines associated with it. When a resource group is brought online with its associated resources, the resource group management (RGM) software automatically starts the appropriate fault monitoring processes. The data service fault monitors are referred to as *probes*.

The fault monitor probes verify that the data service is functioning correctly and providing its intended service. Typically, data service fault monitors perform two functions:

- Monitoring for the abnormal exit of data service processes
- Checking the health of the data service

Data Service Process Monitoring

The Process Monitor Facility (PMF) monitors the data service process. On abnormal exit, the PMF invokes an action script supplied by the data service to communicate the failure to the data service fault monitor. The probe then updates the status of the data service as “Service daemon not running” and takes action. The action can involve just restarting the data service locally or failing over the data service to a secondary cluster node.

Data Service Health Monitoring

A health monitoring probe typically behaves as a data service client and performs regular tests. Each data service requires different kinds of testing. For example, to test the health of the HA for NFS data service, its health probes check to make sure the exported file systems are available and functional. The HA for DNS health monitor regularly uses the `nslookup` command to query the naming service for host or domain information.

Cluster Configuration Repository

General cluster configuration information is stored in global configuration files collectively referred to as the cluster configuration repository (CCR). The CCR must be kept consistent between all nodes and is a critical element that enables each node to be aware of its potential role as a designated backup system.

Caution – Never attempt to modify any of the CCR-related files. The files contain timestamp and checksum information that is critical to the operation of the cluster software. The CCR information is automatically modified as the result of administrative command execution and cluster status changes.

The CCR structures contain the following types of information:

- Cluster and node names
- Cluster transport configuration
- The names of Veritas disk groups
- List of nodes that can master each disk group
- Data service operational parameter values (timeouts)
- Paths to data service control routines
- Disk ID device configuration
- Current cluster status

The CCR is accessed when error or recovery situations occur or when there has been a general cluster status change, such a a node leaving or joining the cluster.

Sun Cluster Scalable Services

The Sun Cluster scalable services rely on the following components:

- Disk ID (DID) devices
- Global devices
- Global file systems (PXFS)

A scalable data service application is designed to distribute an application workload between two or more cluster nodes. As shown in Figure 1-4, the same Web page server software is running on all nodes in a cluster.

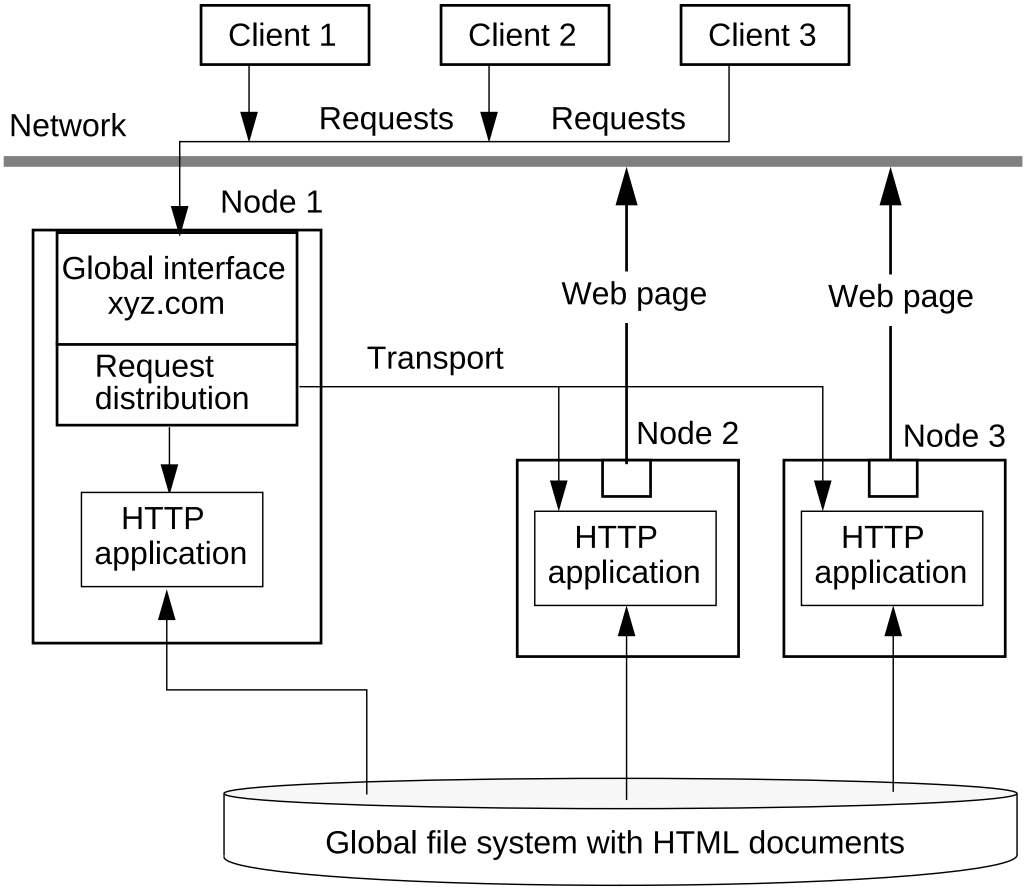


Figure 1-4 Scalable Data Service Features

A scalable data service configuration operates as follows:

- Multiple nodes in the cluster answer incoming Web page requests in parallel.
- A designated node receives all requests and distributes them among the other nodes using the private cluster transport.
- If a node crashes, the cluster framework keeps the data service running with the remaining nodes.
- A single copy of the Hypertext Markup Language (HTML) documents can be placed on a globally accessible cluster file system.

Disk ID Devices

An important component of the Sun Cluster global device technology is the Disk ID (DID) pseudo driver. During the Sun Cluster installation, the DID driver probes devices on each node and creates a unique DID device name for each disk or tape device.

As shown in Figure 1-5, a disk drive in a storage array can have a different logical access path from attached nodes but is globally known by a single DID device number. A third node that has no array storage still has unique DID devices assigned for its boot disk and CD-ROM.

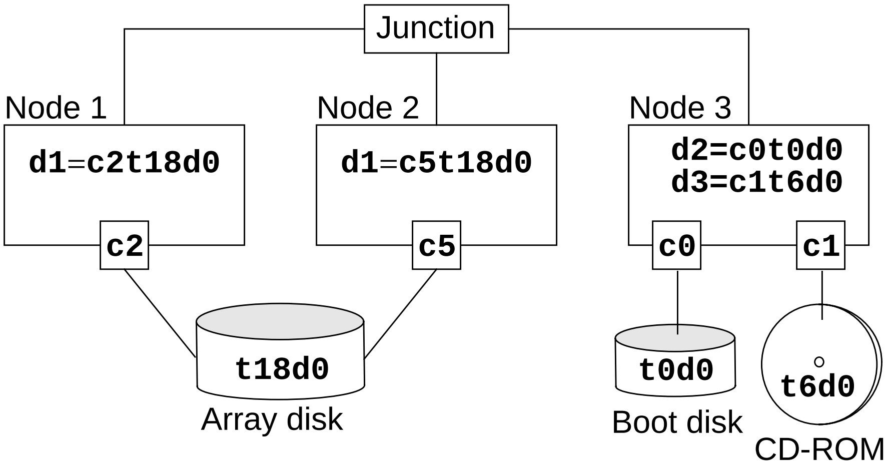


Figure 1-5 DID Driver Devices

The DID device driver would create the following DID instance names for the devices shown in Figure 1-5:

- `/dev/did/rdsk/d1`
- `/dev/did/rdsk/d2`
- `/dev/did/rdsk/d3`

Global Devices

Sun Cluster uses *global devices* to provide cluster-wide, highly available access to any device in a cluster, from any node, without regard to where the device is physically attached. In general, if a node fails while providing access to a global device, Sun Cluster automatically discovers another path to the device and redirects the access to that path. Sun Cluster global devices include disks, CD-ROMs, and tapes. However, disks are the only supported multiported global devices.

The Sun Cluster mechanism that enables global devices is the *global namespace*. The global namespace includes the `/dev/global/` hierarchy as well as the volume manager namespace. The global namespace reflects both multihost disks and local disks (and any other cluster device, such as CD-ROMs and tapes), and provides multiple failover paths to the multihost disks. Each node physically connected to multihost disks provides a path to the storage for any node in the cluster.

In Sun Cluster, each of the local device nodes, including volume manager namespace, are replaced by symbolic links to device nodes in the `/global/.devices/node@`*`nodeID`* file system, where *`nodeID`* is an integer that represents a node in the cluster (node1, node2, node3). Sun Cluster continues to present the volume manager devices, as symbolic links, in their standard locations as well. The global namespace is available from any cluster node.

Typical global namespace relationships for a *`nodeID`* of 1 are shown in Table 1-1. They include a standard disk device, a DID disk device, a Veritas Volume Manager volume and a Solstice DiskSuite metadevice.

Table 1-1 Global Namespace

Local Node Namespace	Global Namespace
`/dev/dsk/c0t0d0s0`	`/global/.devices/node@1/dev/dsk/c0t0d0s0`
`/dev/did/dsk/d0s0`	`/global/.devices/node@1/dev/did/dsk/d0s0`
`/dev/md/nfs/dsk/d0`	`/global/.devices/node@1/dev/md/nfs/dsk/d0`
`/dev/vx/dsk/nfs/v0`	`/global/.devices/node@1/dev/vx/dsk/nfs-dg/v0`

Global Device Links

In the Sun Cluster global device environment, each node can essentially access all of the devices attached to any other cluster node. Even a node with no local data storage can access another node's storage through the cluster interconnect.

- All nodes see each other's `/global/.devices` file system mounts.

  ```
  # mount |grep /global/.devices
  /global/.devices/node@1 on /dev/vx/dsk/rootdisk14vol
  /global/.devices/node@2 on /dev/vx/dsk/rootdisk24vol
  /global/.devices/node@3 on /dev/vx/dsk/rootdisk34vol
  ...
  ...
  /global/.devices/node@8 on /dev/vx/dsk/rootdisk84vol
  ```

- The local `/dev` and `/devices` structures on each node are linked or copied into that node's global structure.

The illustration in Figure 1-6 demonstrates the basic global device structure. The paths shown do not represent the actual path details but accurately depict the concepts.

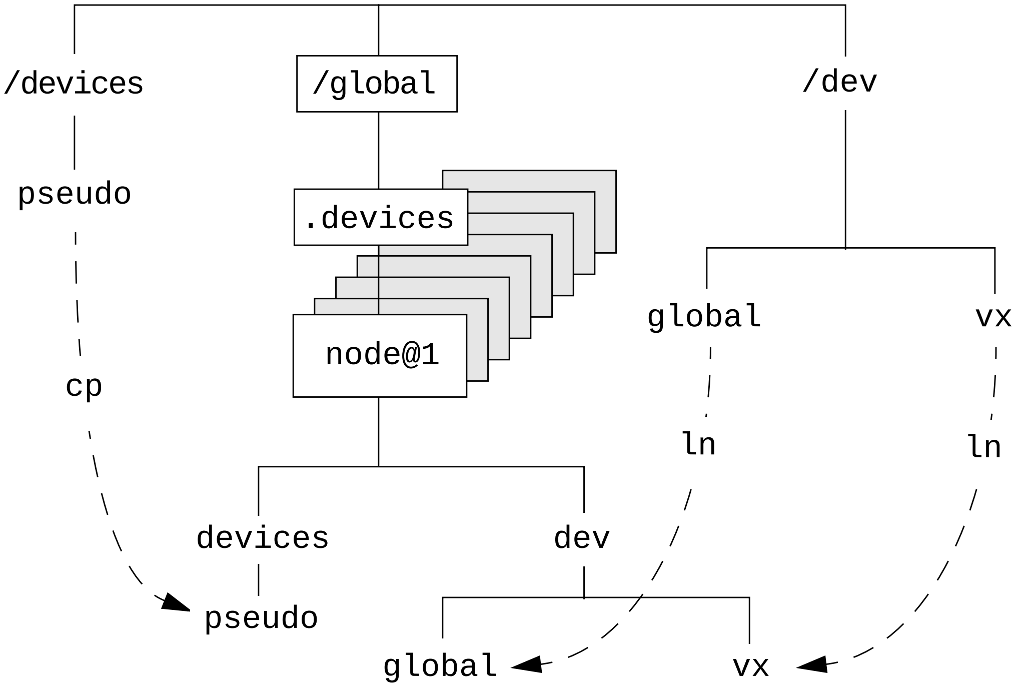


Figure 1-6 Global Device Links

Cluster File Systems

Cluster file systems are dependent on global devices (disks, tapes, CD-ROMs) with physical connections to one or more nodes.

To make a global file system, you create a standard file system on a virtual volume (Veritas or Solstice DiskSuite) and mount the volume on a mount point in the `/global` directory using special global mount options. A typical mount command is as follows:

```
# mount -g dev/vx/dsk/nfs-dg/vol-01 /global/nfs
```

The equivalent mount entry in the `/etc/vfstab` file is:

```
/dev/vx/dsk/nfs-dg/vol-01 /dev/vx/rdsk/nfs-dg/vol-01 \
/global/nfs  ufs 2 yes global,logging
```

After the file system is mounted, it is available on all nodes in the cluster.

Note – Veritas Volume Manager disk groups must be registered with the cluster framework software before any disk group structures can be made globally available.

Proxy File System

The cluster file system is based on the proxy file system (PXFS), which has the following features:

- PXFS makes file access locations transparent. A process can open a file located anywhere in the global file system structure, and processes on all nodes can use the same path name to locate a file.
- PXFS uses coherency protocols to preserve the UNIX® file access semantics even if the file is accessed concurrently from multiple nodes.
- PXFS provides continuous access to data, even when failures occur. Applications do not detect failures as long as a path to disks is still operational. This guarantee is maintained for raw disk access and all file system operations.

Note – PXFS is not a distinct file system type. That is, clients see the underlying file system type (for example, `ufs`).

Resource Groups

At the heart of any Sun Cluster highly available data service is the concept of a resource group. Resource group definitions are created by the cluster administrator and are associated with a particular data service such as Sun Cluster HA for NFS.

A resource group definition provides all of the necessary information for a designated backup system to take over the data services of a failed node. This includes the following:

- The IP address/host name (logical host name) that users use to access the data service application
- The path to data and administrative resources
- The data service type that is to be started
- A list of participating nodes (primary/backup or scalable group)

Figure 1-7 shows the major components of a resource group definition.

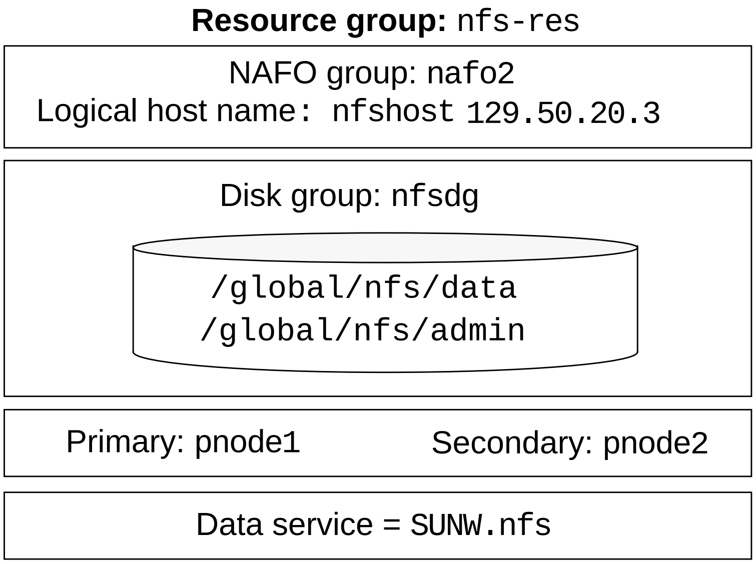


Figure 1-7 Resource Group Components

Note – Resource group configuration information is maintained in the globally available CCR database.

Parallel Database Application

The Oracle Parallel Server (OPS) configuration is characterized by two nodes that access a single database image. OPS configurations are throughput applications. When a node fails, an application does not move to a backup system.

OPS uses a distributed lock management (DLM or IDLM) scheme to prevent simultaneous data modification by two hosts. The lock ownership information is transferred between cluster hosts across the cluster transport system.

When a failure occurs, most of the recovery work is performed by the OPS software, which resolves incomplete database transactions.

As shown in Figure 1-8, the Sun Cluster software performs a relatively minor role. It initiates a portion of the database recovery process.

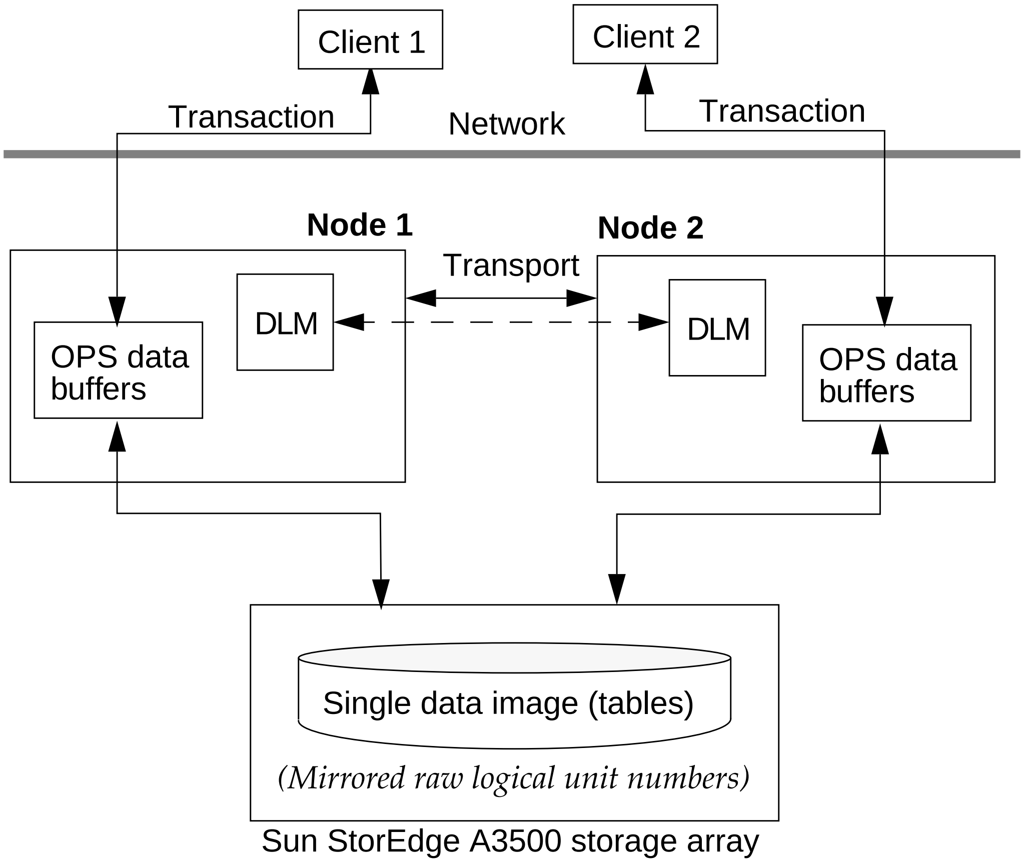


Figure 1-8 Parallel Database Features

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ❑ List the new Sun Cluster 3.0 features
- ❑ List the hardware elements that constitute a basic Sun Cluster system
- ❑ List the hardware and software components that contribute to the availability of a Sun Cluster system
- ❑ List the types of redundancy that contribute to the availability of a Sun Cluster system
- ❑ Identify the functional differences between failover, scalable, and parallel database cluster applications
- ❑ List the supported Sun Cluster 3.0 data services
- ❑ Explain the purpose of Disk ID (DID) devices
- ❑ Describe the relationship between system devices and the cluster global namespace
- ❑ Explain the purpose of resource groups in the Sun Cluster environment
- ❑ Describe the purpose of the cluster configuration repository
- ❑ Explain the purpose of each of the Sun Cluster fault monitoring mechanisms

Think Beyond

What are some of the most common problems encountered during cluster installation?

How do you install a cluster? What do you need to do first?

Do you need to be a database expert to administer a Sun Cluster system?

Module 2

Terminal Concentrator

Objectives

Upon completion of this module, you should be able to:

- Describe the main features of the Sun Cluster administrative interface
- List the main functions of the terminal concentrator operating system
- Verify the correct TC cabling
- Configure the TC IP address
- Configure the TC to self-load
- Verify the TC port settings
- Configure a TC default router if necessary
- Verify that the TC is functional
- Use the TC `help`, `who`, and `hangup` commands
- Describe the purpose of the `telnet send brk` command

Relevance

Discussion – The following questions are relevant to your learning the material presented in this module:

- Why is this hardware covered so early in the course?
- Does this information apply to a Sun Enterprise 10000-based cluster?

Additional Resources

Additional resources – The following references can provide additional information on the topics described in this module:

- *Sun Cluster 3.0 Installation Guide*, part number 806-1419
- *Sun Cluster 3.0 Hardware Guide*, part number 806-1420
- *Sun Cluster 3.0 Data Services Installation and Configuration Guide*, part number 806-1421
- *Sun Cluster 3.0 Data Service Developers Guide*, part number 806-1422
- *Sun Cluster 3.0 System Administration Guide*, part number 806-1423
- *Sun Cluster 3.0 Concepts*, part number 806-1424
- *Sun Cluster 3.0 Error Message Guide*, part number 806-1426
- *Sun Cluster 3.0 Release Notes*, part number 806-1428

Cluster Administration Interface

The TC is a hardware interface, consisting of several components that provide the only access path to *headless* cluster host systems when these systems are halted or before any operating system software is installed.

Note – If the cluster host systems do not have a keyboard and frame buffer, they are said to be *headless*.

As shown in Figure 2-1, the cluster administration interface is a combination of hardware and software that enable you to monitor and control one or more clusters from a remote location.

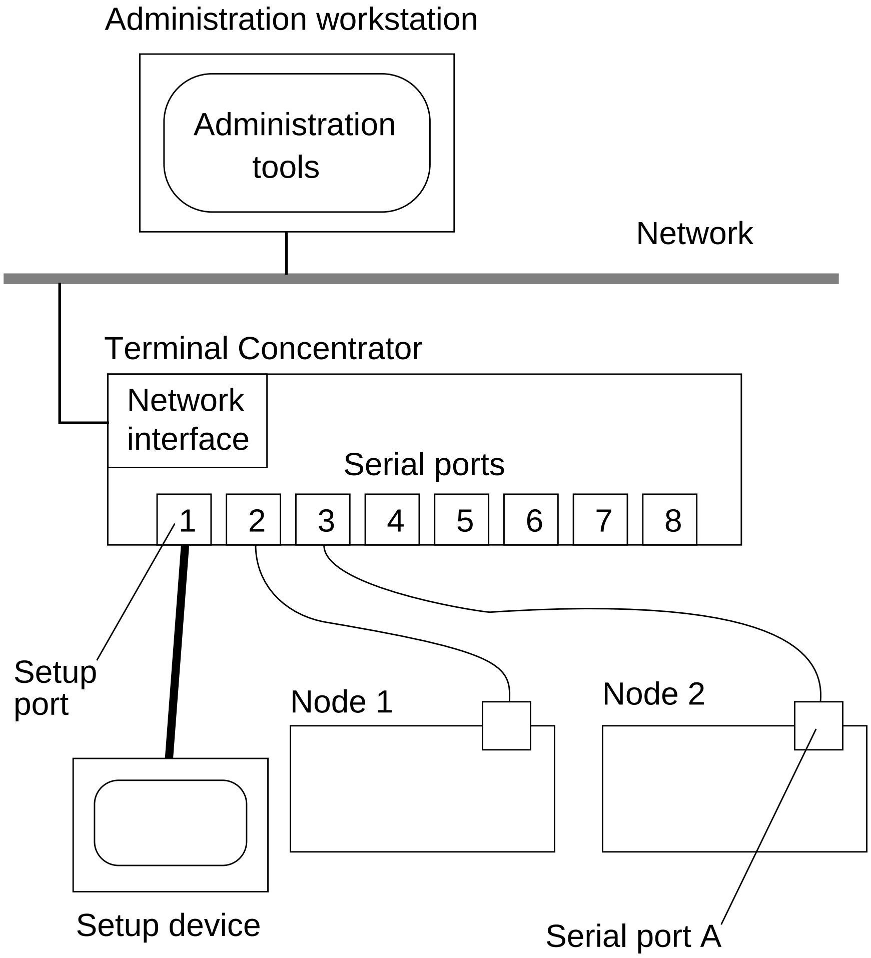


Figure 2-1 Cluster Administration Interface

Major Elements

The relationship of the following elements is shown in Figure 2-1.

Administration Workstation

The administration workstation can be any Sun SPARC™ workstation, providing it has adequate resources to support graphics and computer-intensive applications, such as the Sun Management Center software.

Administration Tools

There are several administration tools but only one of them, the Cluster Console, is functional when the cluster host systems are at the OpenBoot™ Programmable Read-Only Memory (OBP) prompt.

Cluster Console is a tool that automatically links to each node in a cluster through the TC. A text window is provided for each node. The connection is functional even when the headless nodes are at the ok prompt level. This is the only path available to boot the cluster nodes or initially load the operating system software when the systems are headless.

Terminal Concentrator

The TC provides translation between the local area network (LAN) environment and the serial port interfaces on each node. The nodes do not have display monitors so the serial ports are the only means of accessing each node to run local commands.

Cluster Host Serial Port Connections

The cluster host systems do not have a display monitor or keyboard. The system firmware senses this when power is turned on and directs all system output through serial port A. This is a standard feature on all Sun systems.

Terminal Concentrator Overview

The TC used in the Sun Cluster systems has its own internal operating system and resident administration programs. The TC firmware is specially modified for Sun Cluster installation.

Note – If any other TC is substituted, it *must not* send an abort signal to the attached host systems when it is powered on.

As shown in Figure 2-2, the TC is a self-contained unit with its own operating system.

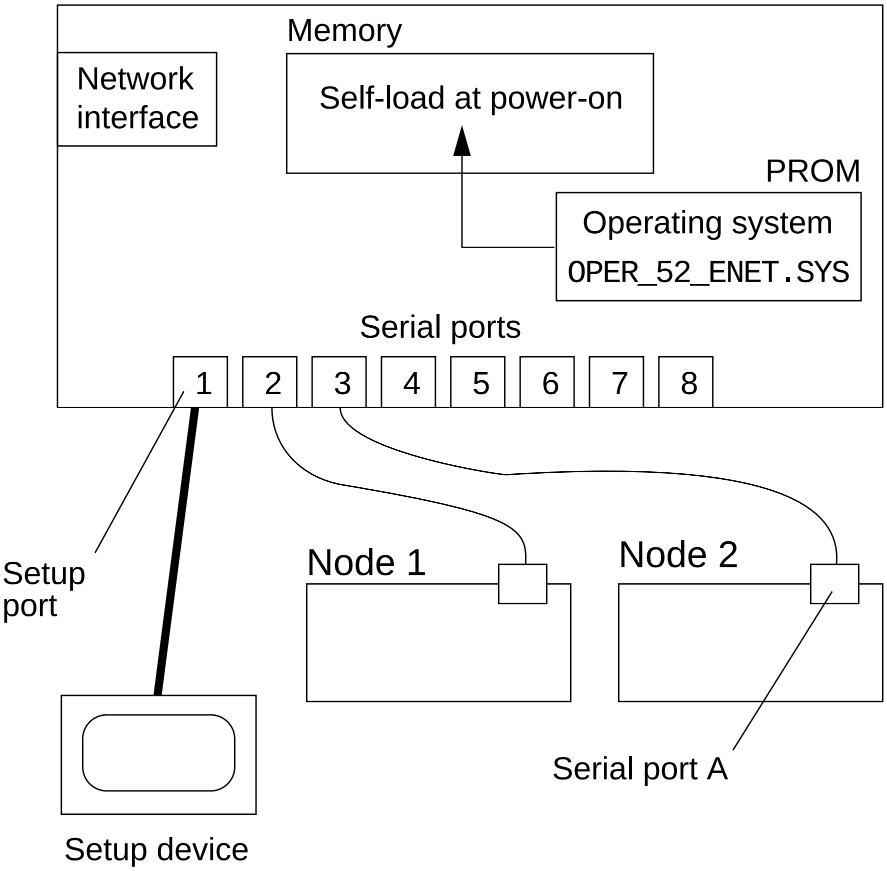


Figure 2-2 Terminal Concentrator Functional Diagram

Caution – If the programmable read-only memory (PROM) operating system is older than version 52, it must be upgraded.

Operating System Load

You can set up the TC to load its operating system either internally from the resident PROM, or externally from a server. In the cluster application, it is always set to load internally. Placing the operating system on an external server can decrease the reliability of the terminal server.

When power is first applied to the TC, it performs the following steps:

1. Runs a PROM-based self-test and displays error codes.
2. Loads a resident PROM-based operating system into the TC memory.

Setup Port

Serial port 1 on the TC is a special purpose port that is used only during initial setup. It is used primarily to set up the IP address and load sequence of the TC. You can access port 1 from either a `tip` connection or from a locally connected terminal.

Terminal Concentrator Setup Programs

You must configure the TC non-volatile random-access memory (NVRAM) with the appropriate IP address, boot path, and serial port information. You use the following resident programs to specify this information:

- `addr`
- `seq`
- `image`
- `admin`

Terminal Concentrator Setup

The TC must be configured for proper operation. Although the TC setup menus seem simple, they can be confusing, and it is easy to make a mistake. You can use the default values for many of the prompts.

Connecting to Port 1

To perform basic TC setup, you must connect to its setup port. Figure 2-3 shows a tip hardwire connection from the administration workstation but you can also connect an American Standard for Information Interchange (ASCII) terminal to the setup port.

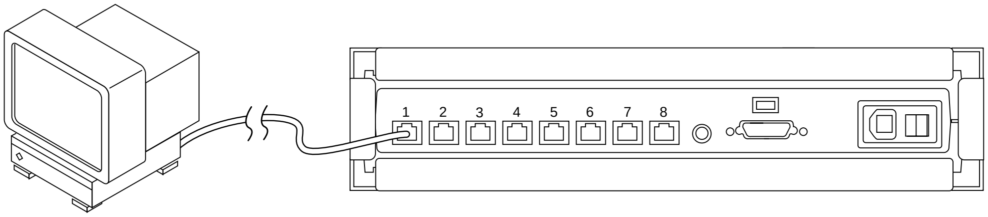


Figure 2-3 Setup Connection to Port 1

Enabling Setup Mode

To enable Setup mode, press the TC Test button shown in Figure 2-4 until the TC power indicator begins to blink rapidly, then release the Test button and press it again briefly.

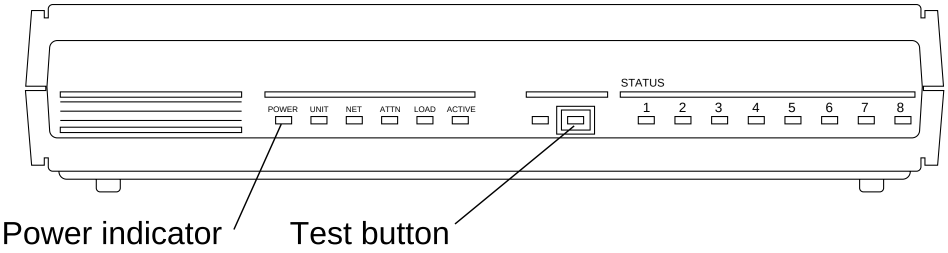


Figure 2-4 Terminal Concentrator Test Button

After you have enabled Setup mode, a `monitor::` prompt should appear on the setup device. Use the `addr`, `seq`, and `image` commands to complete the configuration.

Setting the Terminal Concentrator IP Address

The following example shows how to use the `addr` program to set the IP address of the TC. Usually this is set correctly when your cluster arrives, but you should always verify this.

```
monitor:: addr

Enter Internet address [192.9.22.98]::
129.150.182.100
Enter Subnet mask [255.255.255.0]::
Enter Preferred load host Internet address
[192.9.22.98]:: 129.150.182.100
Enter Broadcast address [0.0.0.0]::
129.150.182.255
Enter Preferred dump address [192.9.22.98]::
129.150.182.100
Select type of IP packet encapsulation
(ieee802/ethernet) [<ethernet>]::
    Type of IP packet encapsulation: <ethernet>

Load Broadcast Y/N [Y]:: y
```

Setting the Terminal Concentrator Load Source

The following example shows how to use the `seq` program to specify the type of loading mechanism to be used.

```
monitor:: seq

Enter a list of 1 to 4 interfaces to attempt to use
for downloading code or upline dumping. Enter them
in the order they should be tried, separated by
commas or spaces. Possible interfaces are:

    Ethernet: net
    SELF:  self

Enter interface sequence [self]::
```

The `self` response configures the TC to load its operating system internally from the PROM when you turn on the power. The PROM image is currently called `OPER_52_ENET.SYS`.

Enabling the self-load feature negates other setup parameters that refer to an external load host and dump host, but you must still define them during the initial setup sequence.

Note – Although you can load the TC’s operating system from an external server, this introduces an additional layer of complexity that is prone to failure.

Specify the Operating System Image

Even though the self-load mode of operation negates the use of an external load and dump device, you should still verify the operating system image name as shown by the following:

```
monitor:: image

    Enter Image name [“oper_52_enet”]::
    Enter TFTP Load Directory [“9.2.7/”]::
    Enter TFTP Dump path/filename
[“dump.129.150.182.100”]::

monitor::
```

Note – Do not define a dump or load address that is on another network because you receive additional questions about a gateway address. If you make a mistake, you can press Control-C to abort the setup and start again.

Setting the Serial Port Variables

The TC port settings must be correct for proper cluster operation. This includes the `type` and `mode` port settings. Port 1 requires different `type` and `mode` settings. You should verify the port settings before installing the cluster host software. The following is an example of the entire procedure:

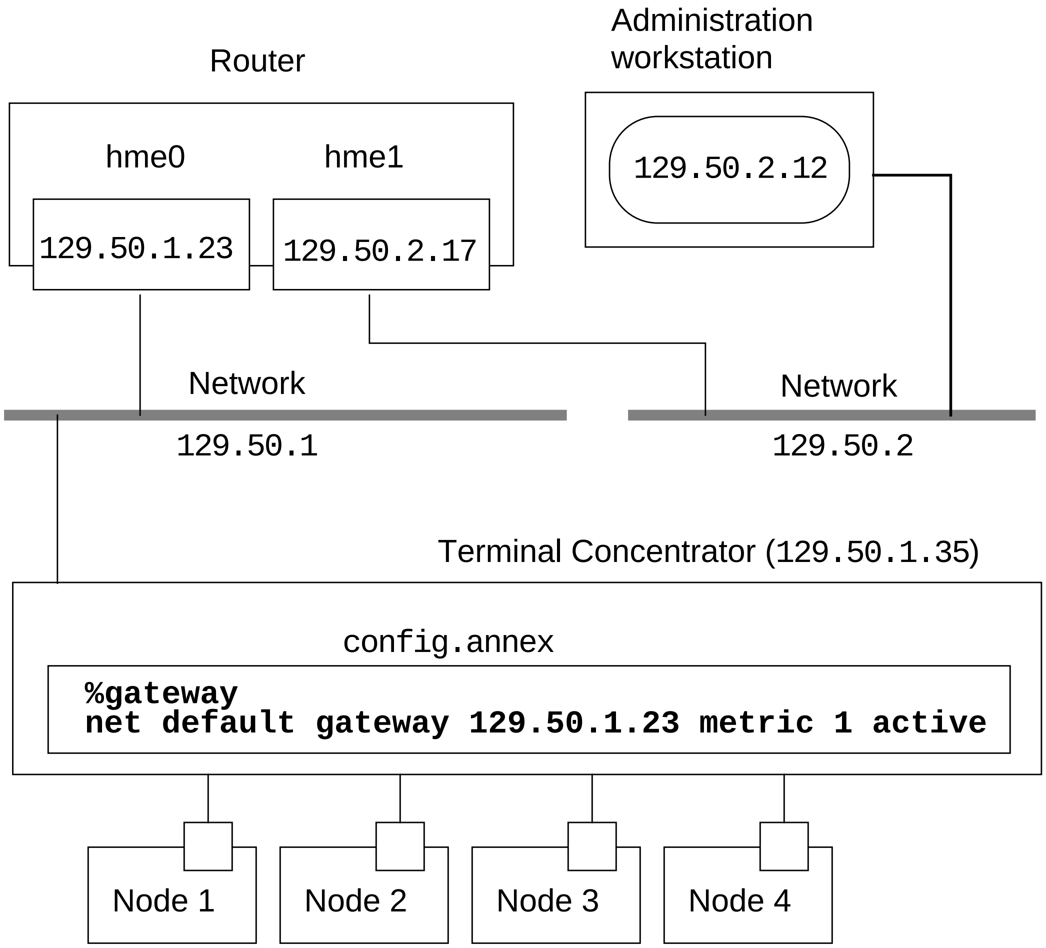
```
admin-ws# telnet terminal_concentrator_name
Trying terminal concentrator IP address
Connected to sec-tc.
Escape character is '^]'.
Rotaries Defined:
    cli                              -
Enter Annex port name or number: cli
Annex Command Line Interpreter  *  Copyright 1991
Xylogics, Inc.
annex: su
Password: type the password
annex# admin
Annex administration MICRO-XL-UX R7.0.1, 8 ports
admin: show port=1 type mode
Port 1:
type: hardwired     mode: cli
admin:set port=1 type hardwired mode cli
admin:set port=2-8 type dial_in mode slave
admin:set port=1-8 imask_7bits Y
admin: quit
annex# boot
bootfile: <CR>
warning: <CR>.
```

Note – Do not perform this procedure through the special setup port but through public network access.

Terminal Concentrator Routing

If you access a TC from a host that is on a different network, the TC's internal routing table can overflow. If the TC routing table overflows, network connections can be intermittent or lost completely.

As shown in Figure 2-5, you can correct this problem by setting a default route within the TC.



```
%gateway
net default gateway 129.50.1.23 metric 1 active
```

Figure 2-5 Terminal Concentrator Routing Table

Creating a Terminal Concentrator Default Route

To create a default route for the TC, you must edit a electrically erasable programmable read-only memory (EEPROM) file in the TC named `config.annex`. The following is a summary of the general process.

```
admin-ws# telnet tc1.central
Trying 192.9.200.1 ...
Connected to 192.9.200.1.
Escape character is ’^]’.
[Return] [Return]
Enter Annex port name or number: cli
...
annex: su
Password: root_password
annex: edit config.annex
(Editor starts)
Ctrl-W:save and exit Ctrl-X:exit Ctrl-F:page down
Ctrl-B:page up
%gateway
net default gateway 192.9.200.2 metric 1 active ^W
annex# admin set annex routed n
You may need to reset the appropriate port, Annex
subsystem or
reboot the Annex for changes to take effect.
annex# boot
```

Note – You must enter an IP routing address appropriate for your site. While the TC is rebooting, the node console connections are not available.

Using Multiple Terminal Concentrators

A single TC can provide serial port service to a maximum of seven systems. If you have eight nodes you must use two TCs for a single cluster.

The maximum length for a TC serial port cable is approximately 348 feet. As shown in Figure 2-6, it might be necessary to have cluster host systems separated by more that the serial port cable limit. You might need a dedicated TC for each node in a cluster.

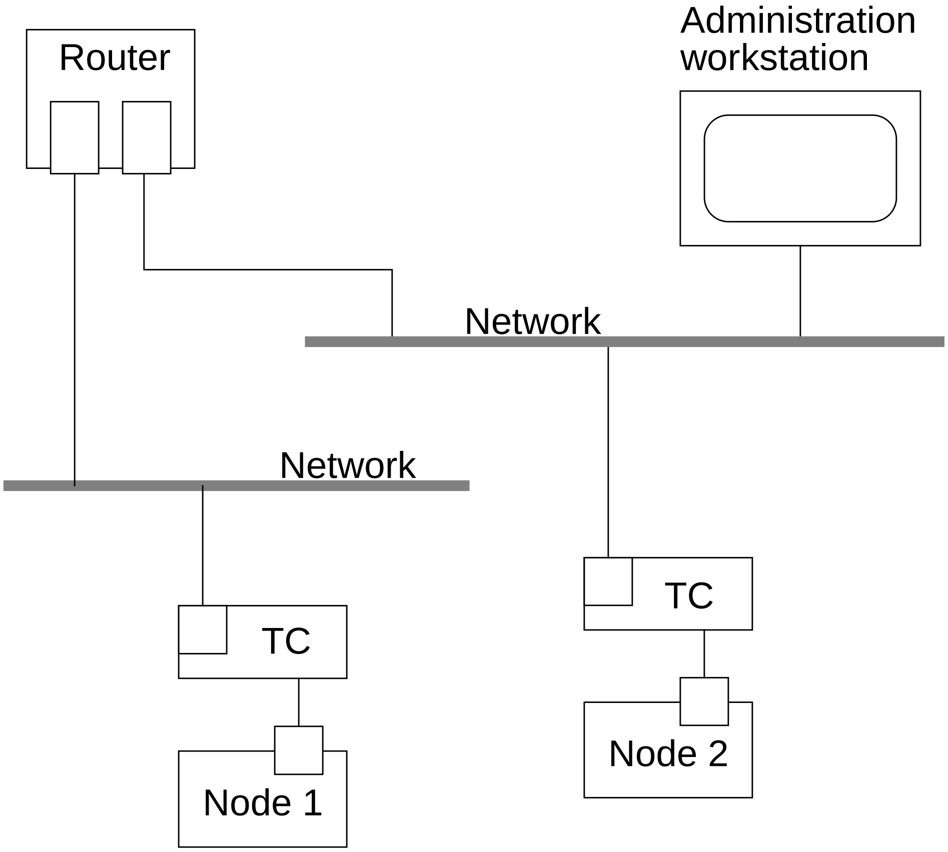


Figure 2-6 Multiple Terminal Concentrators

Terminal Concentrator Troubleshooting

Occasionally it is useful to be able to manually manipulate the TC. The commands to do this are not well documented in the cluster manuals.

Manually Connecting to a Node

If the `cconsole` tool is not using the TC serial ports, you can use the `telnet` command to connect to a specific serial port as follows:

```
# telnet tc_name 5002
```

You can then log in to the node attached to port 5002. After you have finished and logged out of the node, you must break the `telnet` connection with the Control-] keyboard sequence and then type `quit`. If you do not, the serial port remains locked and cannot be used by other applications, such as the `cconsole` tool.

Using the `telnet` Command to Abort a Node

If you have to abort a cluster node, you can either use the `telnet` command to connect directly to the node and use the Control-] keyboard sequence, or you can use the Control-] keyboard sequence in a cluster console window. Once you have the `telnet` prompt, you can abort the node with the following command:

```
telnet > send brk
ok
```

Note – You might have to repeat the command multiple times.

Connecting to the Terminal Concentrator CLI

As shown, you can use the `telnet` command to connect directly to the TC, and then use the resident command line interpreter to perform status and administration procedures.

```
# telnet IPaddress
Trying 129.146.241.135...
Connected to 129.146.241.135
Escape character is '^]'.

Enter Annex port name or number: cli

Annex Command Line Interpreter * Copyright 1991
Xylogics, Inc.
annex:
```

Using the Terminal Concentrator `help` Command

After you connect directly into a terminal concentrator, you can get online help as shown.

```
annex: help
annex: help hangup
```

Identifying and Resetting a Locked Port

If a node crashes, it can leave a `telnet` session active that effectively locks the port from further use. You can use the `who` command to identify which port is locked, and then use the `admin` utility to reset the locked port. The command sequence is as follows:

```
annex: who
annex: su
Password:
annex# admin
Annex administration MICRO-XL-UX R7.0.1, 8 ports
admin : reset 6
admin : quit
annex# hangup
```

Notes:

Exercise: Configuring the Terminal Concentrator

In this exercise, you complete the following tasks:

- Verify the correct TC cabling
- Configure the TC IP address
- Configure the TC to self-load
- Verify the TC port settings
- Verify that the TC is functional

Preparation

Before starting this lab, record the name and IP address assignment for your terminal concentrator. Ask your instructor for assistance.

TC Name:____________________

TC IP address:____________________

Note – During this exercise, when you see italicized names such as *`IPaddress`*, *`enclosure_name`*, *`node1`*, or *`clustername`* imbedded in a command string, substitute the names appropriate for your cluster.

Task – Verifying the Network and Host System Cabling

A TC can have one or two Ethernet connections, depending on its age. All TC generations have the same serial port connections.

Before you begin to configure the TC, you must verify that the network and cluster host connections are correct.

1. Inspect the rear of the TC and make sure it is connected to a public network.

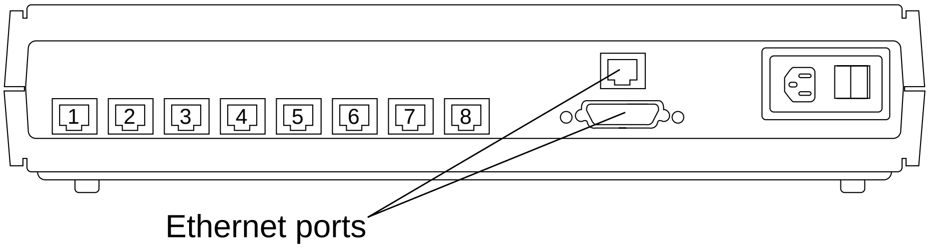


Figure 2-7 Terminal Concentrator Network Connection

2. Verify that the serial ports are properly connected to the cluster host systems. Each output should go to serial port A on the primary system board of each cluster host system.

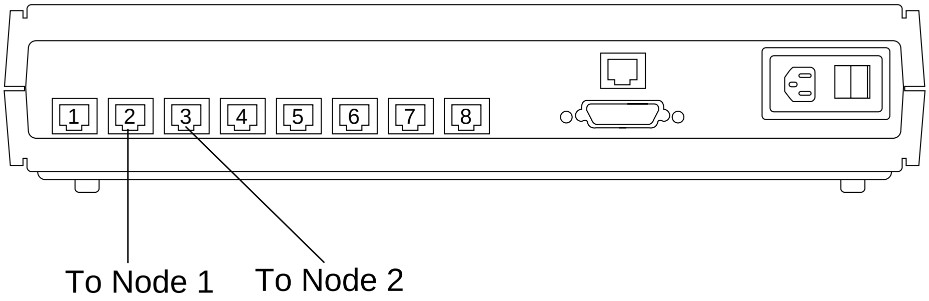


Figure 2-8 Concentrator Serial Port Connections

Note – In clusters with more than two nodes, there are additional serial port connections to the TC.

To set up the TC, you can either connect a "dumb" terminal to serial port 1 or use the `tip hardwire` command from a shell on the administration workstation.

If you are using a local terminal connection, continue with the next section, "Task – Connecting a Local Terminal." If you are using the administration workstation, proceed to the "Task – Connecting Tip Hardwire" on page 2-21.

Task – Connecting a Local Terminal

Perform the following procedure only if you are connecting to the TC with a dumb terminal.

1. Connect the local terminal to serial port 1 on the back of the TC using the cable supplied.

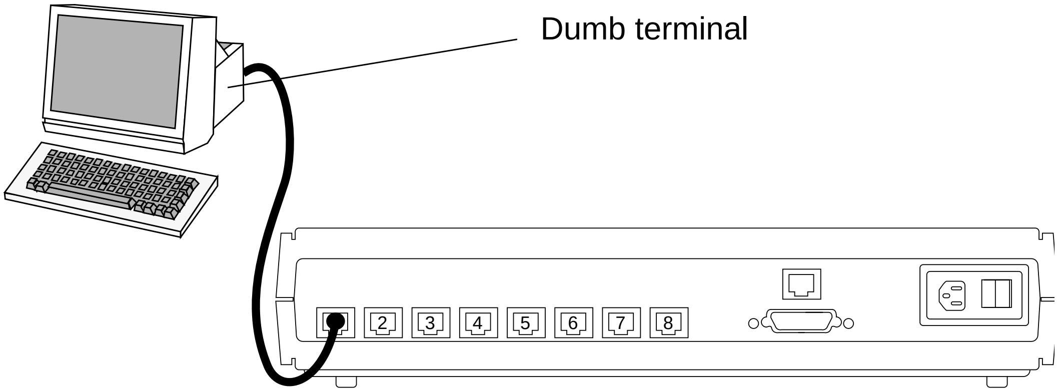


Figure 2-9 Concentrator Local Terminal Connection

Note – Do not use a cable length of more than 500 feet. Use a null modem cable.

2. Verify that the local terminal operating parameters are set to 9600 baud, 7 data bits, no parity, and 1 stop bit.
3. Proceed to the "Task – Achieving Setup Mode" on page 2-22.

Task – Connecting Tip Hardwire

Perform the following procedure only if you are using the tip connection method to configure the TC. If you have already connected to the TC with a dumb terminal, skip this procedure.

1. Connect serial port B on the administration workstation to serial port 1 on the back of the TC using the cable supplied.

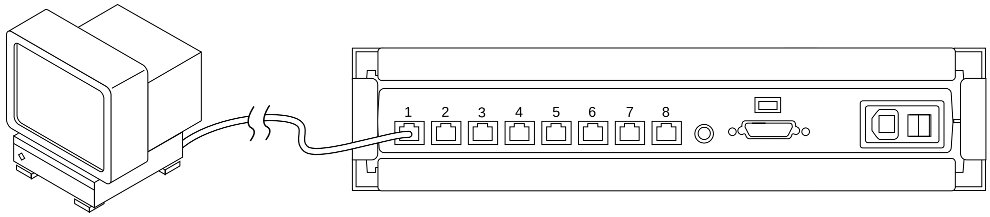


Figure 2-10 Concentrator to Tip Hardware Connection

Note – Do not use a cable length over 500 feet. Use a null modem cable.

2. Verify that the `hardwire` entry in the `/etc/remote` file matches the serial port you are using:

```
hardwire:\
:dv=/dev/term/b:br#9600:el=^C^S^Q^U^D:ie=%$:oe=^D:
```

The baud rate must be 9600.

The serial port designator must match
the serial port you are using.

3. Open a shell window on the administration workstation and make the `tip` connection by typing the following command:

```
# tip hardwire
```

Task – Achieving Setup Mode

Before the TC configuration can proceed, you must first place it in its Setup mode of operation. Once in Setup mode, the TC accepts configuration commands from a serial device connected to port 1.

1. To enable Setup mode, press and hold the TC Test button until the TC power indicator begins to blink rapidly, then release the Test button and press it again briefly.

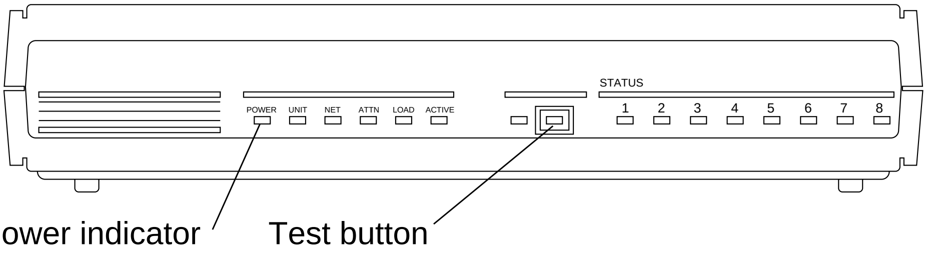


Figure 2-11 Enabling Setup Mode on the Terminal Concentrator

2. After the TC completes its power-on self-test, you should see the following prompt on the shell window or on the local terminal:

```
monitor::
```

Note – It can take a minute or more for the self-test process to complete.

Task – Configuring the IP Address

Verify that the TC Internet address and preferred load host address are set to your assigned value. This address must not conflict with any other network systems or devices.

1. To configure the Terminal Concentrator IP address using the `addr` command, type **`addr`** at the `monitor::` prompt.

```
monitor:: addr

Enter Internet address [192.9.22.98]::
129.150.182.101

Enter Subnet mask [255.255.255.0]::

Enter Preferred load host Internet address
[192.9.22.98]::
129.150.182.101

***Warning: Local host and Internet host are the
same***

Enter Broadcast address [0.0.0.0]::
129.150.182.255

Enter Preferred dump address [192.9.22.98]::
129.150.182.101

Select type of IP packet encapsulation
(ieee802/ethernet) [<ethernet>]::
    Type of IP packet encapsulation: <ethernet>

Load Broadcast Y/N [Y]:: n

monitor::
```

Task – Configuring the TC to Self-Load

When the TC is turned on, you must configure it to load a small operating system. You can use the `seq` command to define the location of the operating system and the `image` command to define its name.

1. To configure the TC to load from itself instead of trying to load from a network host, type the **`seq`** command at the `monitor::` prompt.

   ```
   monitor:: seq
   ```

   Enter a list of 1 to 4 interfaces to attempt to use for downloading code or upline dumping. Enter them in the order they should be tried, separated by commas or spaces. Possible interfaces are:

   ```
      Ethernet: net
      SELF:  self

      Enter interface sequence [self]::
   ```

2. To configure the TC to load the correct operating system image, type the **`image`** command at the `monitor::` prompt.

   ```
      monitor:: image
      Enter Image name [“oper.52.enet”]::
      Enter TFTP Load Directory [“9.2.7/”]::

      Enter TFTP Dump path/filename
   [“dump.129.150.182.101”]::
   ```

3. If you used a direct terminal connection, disconnect it from the TC when finished.

4. If you used the `tip hardwire` method, break the tip connection by typing the **`~.`** sequence in the shell window.

Task – Verifying the Self-Load Process

Before proceeding, you must verify that the TC can complete its self-load process and that it answers to its assigned IP address.

1. Turn off the TC power for at least 10 seconds and then turn it on again.
2. Observe the light-emitting diodes (LEDs) on the TC front panel. After the TC completes its power-on self-test and load routine, the front panel LEDs should look like the following:

Table 2-1 LED Front Panel Settings

Power (Green)	**Unit (Green)**	**Net (Green)**	**Attn (Amber)**	**Load (Green)**	**Active (Green)**
ON	ON	ON	OFF	OFF	Intermittent blinking

Note – It takes at least one minute for the process to complete. The Load light extinguishes after the internal load sequence is complete.

Verifying the Terminal Concentrator Pathway

Complete the following steps on the administration workstation from a shell or command tool window:

1. Test the network path to the TC using the following command:

```
# ping IPaddress
```

Note – Substitute the IP address of your TC for *IPaddress*.

Task – Verifying the TC Port Settings

You must set the TC port variable *type* to dial_in for each of the eight TC serial ports. If it is set to hardwired, the cluster console might be unable to detect when a port is already in use. There is also a related variable called *imask_7bits* that you must set to Y.

You can verify and, if necessary, modify the *type*, *mode*, and *imask_7bits* variable port settings, with the following procedure.

1. On the administration workstation, use the telnet command to connect to the TC. Do not use a port number.

```
# telnet IPaddress
Trying 129.146.241.135...
Connected to 129.146.241.135
Escape character is '^]'.
```

2. Enable the command line interpreter, use the su command to get to the root account, and start the admin program.

```
Enter Annex port name or number: cli

Annex Command Line Interpreter * Copyright 1991
Xylogics, Inc.

annex: su
Password:
annex# admin
Annex administration MICRO-XL-UX R7.0.1, 8 ports

admin :
```

Note – By default, the superuser password is the TC IP address. This includes the periods in the IP address.

3. Use the show command to examine the current setting of all ports.

```
admin : show port=1-8 type mode
```

4. Perform the following procedure to change the port settings and to end the TC session.

```
admin:set port=1 type hardwired mode cli
admin:set port=1-8 imask_7bits Y
admin:set port=2-8 type dial_in mode slave
admin: quit
annex# boot
bootfile: <CR>
warning: <CR>
Connection closed by foreign host.
```

Note – It takes at least one minute for the process to complete. The Load light extinguishes after the internal load sequence is complete.

Task – Terminal Concentrator Troubleshooting

1. On the administration workstation, use the `telnet` command to connect to the TC. Do not use a port number.

```
# telnet IPaddress
Trying 129.146.241.135...
Connected to 129.146.241.135
Escape character is '^]'.
```

2. Enable the command line interpreter.

```
Enter Annex port name or number: cli

Annex Command Line Interpreter * Copyright 1991
Xylogics, Inc.

annex:
```

3. Practice using the `help` and `who` commands.

4. End the session with the `hangup` command.

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ❑ Describe the main features of the Sun Cluster administrative interface
- ❑ List the main functions of the TC operating system
- ❑ Verify the correct TC cabling
- ❑ Configure the TC IP address
- ❑ Configure the TC to self-load
- ❑ Verify the TC port settings
- ❑ Configure a TC default router if necessary
- ❑ Verify that the TC is functional
- ❑ Use the TC `help`, `who`, and `hangup` commands
- ❑ Describe the purpose of the `telnet send brk` command

Think Beyond

Is there a significant danger if the TC port variables are not set correctly?

Is the TC a single point of failure? What would happen if it failed?

Module 3

Installing the Administration Workstation

Objectives

Upon completion of this module, you should be able to:

- List the Sun Cluster administration workstation functions
- Install the Sun Cluster console software on the administration workstation
- Set up the administration workstation environment
- Configure the Sun Cluster console software

Relevance

Discussion – The following questions are relevant to understanding this module's content:

1. How important is the administration workstation during the configuration of the cluster host systems?

Additional Resources

Additional resources – The following references can provide additional information on the topics described in this module:

- *Sun Cluster 3.0 Installation Guide*, part number 806-1419
- *Sun Cluster 3.0 Hardware Guide*, part number 806-1420
- *Sun Cluster 3.0 Data Services Installation and Configuration Guide*, part number 806-1421
- *Sun Cluster 3.0 Data Service Developers Guide*, part number 806-1422
- *Sun Cluster 3.0 System Administration Guide*, part number 806-1423
- *Sun Cluster 3.0 Concepts*, part number 806-1424
- *Sun Cluster 3.0 Error Message Guide*, part number 806-1426
- *Sun Cluster 3.0 Release Notes*, part number 806-1428

Sun Cluster Console Software

As shown in Figure 3-1, the Sun Cluster console software is installed on the administration workstation. The Sun Cluster framework and data service software is installed on each of the cluster host systems along with appropriate virtual volume management software.

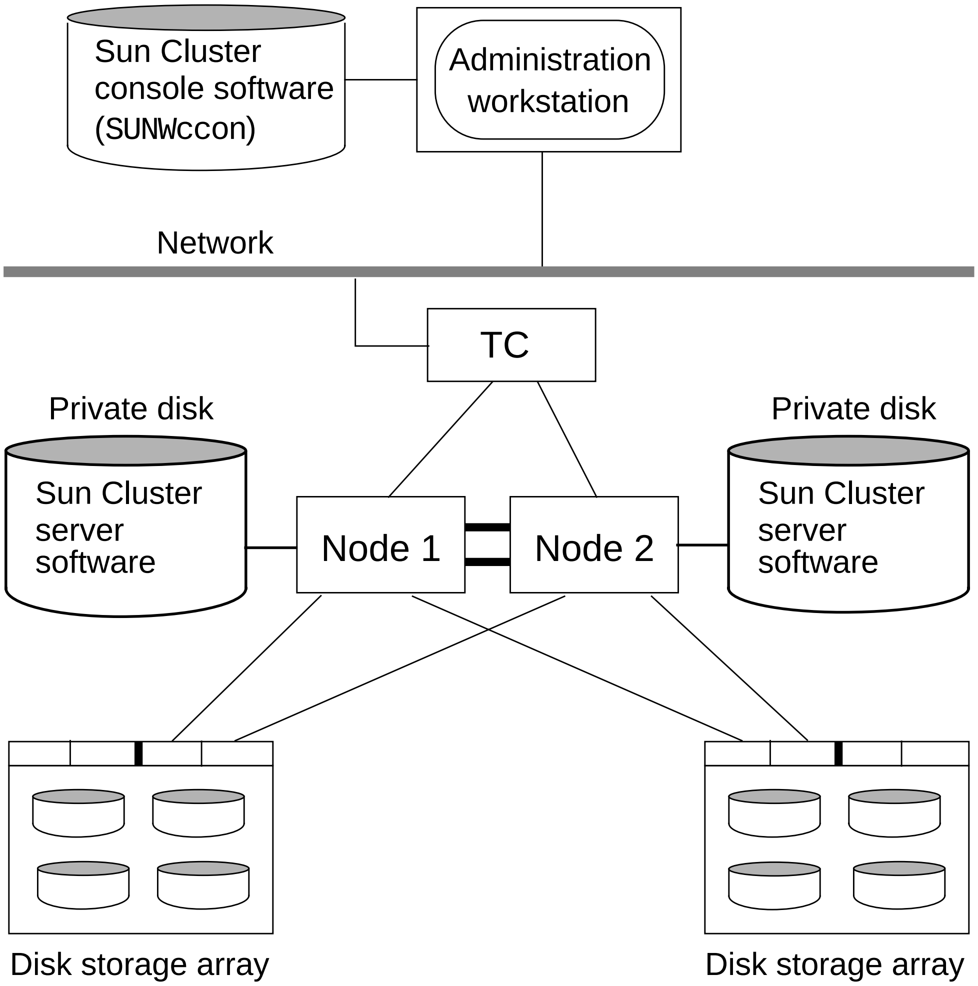


Figure 3-1 Sun Cluster Console Software

Console Software Installation

The administration workstation console software is contained in a single package: `SUNWccon`. The `SUNWccon` package is installed manually from the the Sun Cluster 3.0 software distribution CD.

Sun Cluster Console Tools

Use the cluster administration tools to manage a cluster. They provide many useful features, including:

- Centralized tool bar
- Command-line interface to each cluster host

The console tool programs are:

- `ccp`, `cconsole`, `crlogin`, and `ctelnet`

You can start the cluster administration tools manually or by using the `ccp` program tool bar.

The Cluster Control Panel

As shown in Figure 3-2, the Cluster Control Panel provides centralized access to three variations of the cluster console tool.

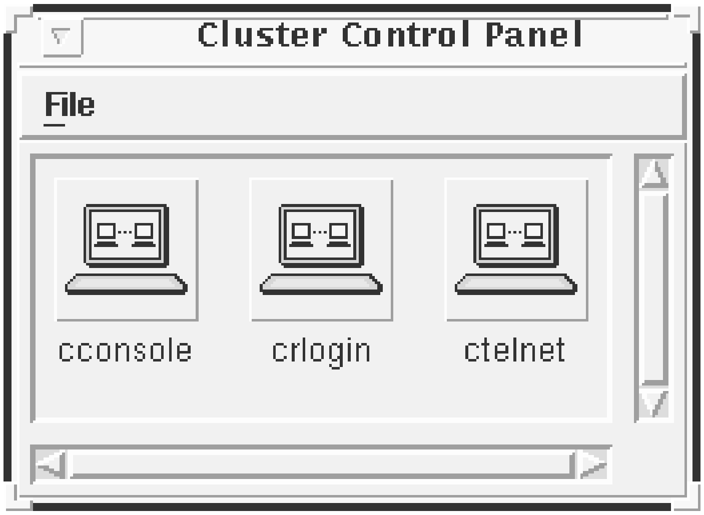


Figure 3-2 Cluster Control Panel

Starting the Cluster Control Panel

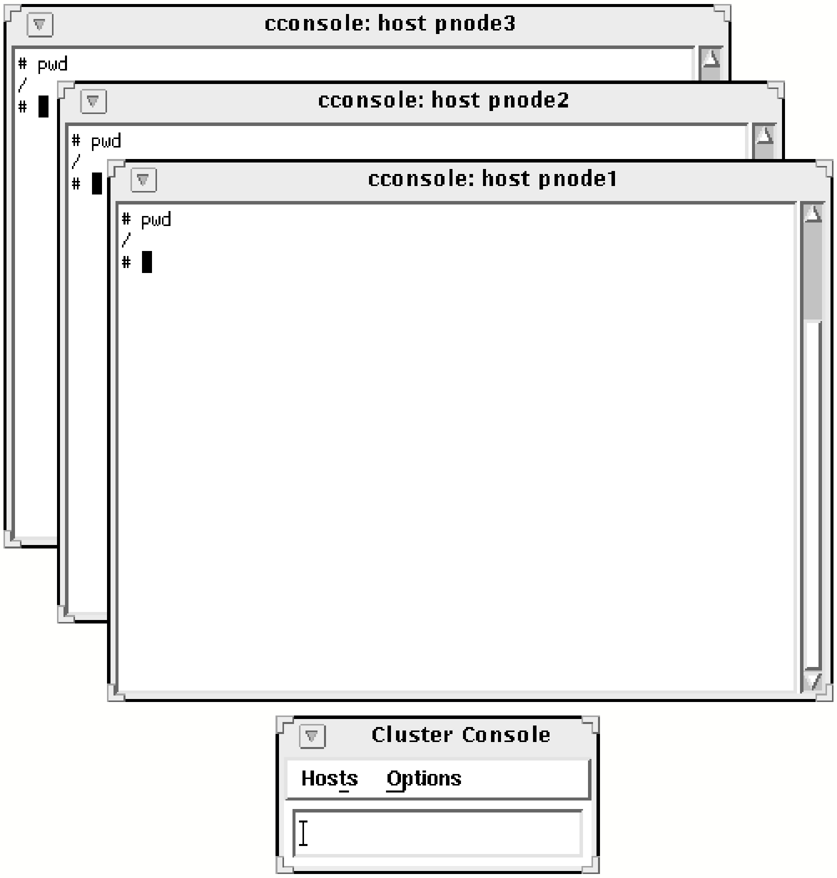
To start the Cluster Control Panel, type the following command:

```
# /opt/SUNWcluster/bin/ccp [clustername] &
```

Cluster Console

The cluster console tool uses the TC to access the cluster host systems through serial port interfaces. The advantage of this is that you can connect to the cluster host systems even when they are halted. This is essential when booting headless systems and can be useful during initial cluster host configuration.

As shown in Figure 3-3, the cluster console tool uses `xterm` windows to connect to each of the cluster host systems.



```
# pwd
/
#
```

```
# pwd
/
#
```

```
# pwd
/
#
```

Figure 3-3 Cluster Console Windows

Manually Starting the cconsole Tool

As shown, you can use the cconsole tool manually to connect to a single cluster node or to the entire cluster.

```
# /opt/SUNWcluster/bin/cconsole node1 &
# /opt/SUNWcluster/bin/cconsole my-cluster &
# /opt/SUNWcluster/bin/cconsole node3 &
```

Cluster Console Host Windows

There is a host window for each node in the cluster. You can enter commands in each host window separately.

The host windows all appear to be vt220 terminals. Set the TERM environment variable to vt220 to use the arrow and other special keys.

Cluster Console Common Window

The common window shown in Figure 3-4 enables you to enter commands to all host system windows at the same time. All of the windows are tied together, so when you move the common window, the host windows follow. The Options menu allows you to ungroup the windows, move them into a new arrangement, and group them again.

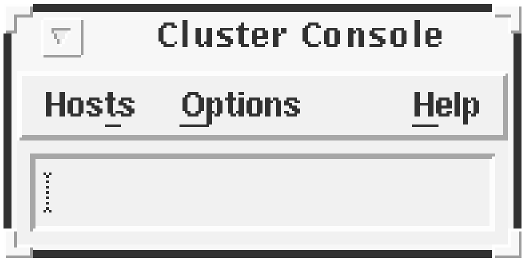


Figure 3-4 Cluster Console Common Window

Cluster Console Window Variations

There are three variations of the cluster console tool that each use a different method to access the cluster hosts. They all look and behave the same way.

- Cluster console (`cconsole`)

  The `cconsole` program accesses the host systems through the TC interface. The Solaris Operating Environment does not have to be running on the cluster host systems.

  Only one connection at a time can be made through the TC to serial port A of a cluster node. You cannot start a second instance of `cconsole` for the same cluster.

  For Sun Enterprise 10000 domains, a `telnet` connection is made to the `ssp` account on the domain's system service processor (SSP), then a `netcon` session is established.

- Cluster console (`crlogin`)

  The `crlogin` program accesses the host systems through the public network using the `rlogin` command. The Solaris Operating Environment must be running on the cluster host systems.

- Cluster console (`ctelnet`)

  The `ctelnet` program accesses through the public network using the `telnet` command. The Solaris Operating Environment must be running on the cluster host systems.

Cluster Console Tools Configuration

All of the necessary information needed for the cluster administration tools that run on the administration workstation is configured in two administration workstation files. The files are:

```
/etc/clusters
/etc/serialports
```

When you install the Sun Cluster client software on the administration workstation, two blank files are created. You must edit the files and supply the necessary information.

Configuring the /etc/clusters File

The /etc/clusters file contains the name of a cluster followed by the names of node that are part of that cluster.

The following is a typical entry in the /etc/clusters file:

```
sc-cluster sc-node1 sc-node2
```

The single-line entry defines a cluster named sc-cluster that has two nodes named sc-node1 and sc-node2.

Note – The cluster name is purely arbitrary, but it should agree with the name you use when you install the server software on each of the cluster host systems.

You can define many different clusters in a single /etc/clusters file, so you can administer several clusters from a single administration workstation.

Configuring the `/etc/serialports` File

The `/etc/serialports` file defines the terminal concentrator path to each node defined in the `/etc/clusters` file. You must enter the paths to all nodes in all of your clusters in this file.

The following are typical entries in the `/etc/serialports` file:

```
sc-node1 sc-tc 5002
sc-node2 sc-tc 5003
```

There is a line for each cluster host that describes the name of each host, the name of the terminal concentrator, and the terminal concentrator port to which each host is attached.

For the Sun Enterprise 10000 server, the `/etc/serialports` entries for each cluster domain are configured with the domain name, the SSP name, and (always) the number 23, which represents the `telnet` port.

```
sc-10knode1 sc10k-ssp23
sc-10knode2 sc10k-ssp23
```

Note – When upgrading the cluster software, the `/etc/serialports` and `/etc/clusters` files are overwritten. You should make a backup copy before starting an upgrade.

Multiple Terminal Concentrator Configuration

If you have widely separated nodes or an eight-node cluster, you might need more than one TC to manage a single cluster. The following examples demonstrate how the /etc/clusters and /etc/serialports files might appear for an eight-node cluster.

The following entry in the /etc/clusters file would represent the nodes in an eight-node cluster:

```
sc-cluster sc-node1 sc-node2 sc-node3 sc-node4 \
sc-node5 sc-node6 sc-node7 sc-node8
```

The single-line entry defines a cluster named sc-cluster that has eight nodes.

The following entries in the /etc/serialports file define the TC paths to each node in an eight-node cluster:

```
sc-node1 sc-tc1 5002

sc-node2 sc-tc1 5003

sc-node3 sc-tc1 5004

sc-node4 sc-tc1 5005

sc-node5 sc-tc2 5002

sc-node6 sc-tc2 5003

sc-node7 sc-tc2 5004

sc-node8 sc-tc2 5005
```

There is a line for each cluster host that describes the name of the host, the name of the terminal concentrator, and the terminal concentrator port to which the host is attached.

Exercise: Configuring the Administration Workstation

In this exercise, you complete the following tasks:

- Install and configure the Sun Cluster console software on an administration workstation
- Configure the Sun Cluster administration workstation environment for correct Sun Cluster console software operation
- Start and use the basic features of the Cluster Control Panel and the Cluster Console

Preparation

This lab assumes that the Solaris 8 Operating Environment software is already installed on all of the cluster systems.

Record the following information about your assigned cluster before proceeding with this exercise:

Table 3-1 Cluster Names and Addresses

System	Name	IP Address
Administration workstation		
Terminal concentrator		
Node 1		
Node 2		
Node 3		

Ask your instructor for assistance with completing the information. The names and addresses might already be in the `/etc/hosts` files as the result of a JumpStart configuration.

Note – During this exercise, when you see italicized names such as *`IPaddress`*, *`enclosure_name`*, *`node1`*, or *`clustername`* imbedded in a command string, substitute the names appropriate for your cluster.

Task – Updating Host Name Resolution

Even though your site might use NIS or DNS to resolve host names, it can be beneficial to resolve the names locally on the administration workstation and cluster hosts. This can be valuable in the case of naming service failures. The `cconsole` program will not start unless it can first resolve the host names in the `/etc/clusters` file.

1. If necessary, edit the `/etc/hosts` file on your administrative workstation and add the IP addresses and names of the TC and the host systems in your cluster.

2. Verify that the `/etc/nsswitch.conf` file entry for `hosts` has `files` listed first.

```
hosts:      files nis
```

Task – Installing the Cluster Console Software

1. Log in to your administration workstation as user `root`.

2. Move to the `Sun Cluster 3.0 packag`es directory.

Note – Either load the Sun Cluster 3.0 CD or move to the location provided by your instructor.

3. Verify that you are in the correct location.

```
# ls
SUNWccon   SUNWscid   SUNWscscn
SUNWmdm    SUNWscidx  SUNWscshl
SUNWrsmop  SUNWscman  SUNWscssv
SUNWscdev  SUNWscr    SUNWscu
SUNWscfab  SUNWscsal  SUNWscvm
SUNWsci    SUNWscsam
```

4. Install the cluster console software package.

```
# pkgadd -d . SUNWccon
```

Task – Verifying the Administration Workstation Environment

1. Verify that the following search paths and variables are present in the `/.profile` file:

```
PATH=$PATH:/opt/SUNWcluster/bin
MANPATH=$MANPATH:/opt/SUNWcluster/man
EDITOR=/usr/bin/vi
export PATH MANPATH EDITOR
```

Note – Create the `.profile` file in the `/` directory if necessary and add the changes.

2. Execute the `.profile` file to verify changes that have been made.

```
# ./.profile
```

Task – Configuring the `/etc/clusters` File

The `/etc/clusters` file has a single line entry for each cluster you intend to monitor. The entries are in the form:

clustername host1name host2name host3name host4name

Sample `/etc/clusters` **File**

```
sc-cluster  pnode1      pnode2      pnode3
```

1. Edit the `/etc/clusters` file and add a line using the cluster and node names assigned to your system.

Task – Configuring the `/etc/serialports` File

The `/etc/serialports` file has an entry for each cluster host describing the connection path. The entries are in the form:

```
hostname    tcname     tcport
```

Sample `/etc/serialports` File

```
pnode1    cluster-tc      5002
pnode2    cluster-tc      5003
pnode3    cluster-tc      5004
```

1. Edit the `/etc/serialports` file and add lines using the node and TC names assigned to your system.

Note – When you upgrade the cluster software, the `/etc/serialports` and `/etc/clusters` files might be overwritten. Make a backup copy of these files before starting an upgrade.

Task – Starting the `cconsole` Tool

This section provides a good functional verification of the TC in addition to the environment configuration.

1. Make sure power is on for the TC and all of the cluster hosts.
2. Start the `cconsole` application on the administration workstation.

```
# cconsole clustername &
```

Note – Substitute the name of your cluster for *`clustername`*.

3. Place the cursor in the `cconsole` Common window and press Return several times. You should see a response on all of the cluster host windows. If not, ask your instructor for assistance.

Note – The `cconsole` Common window is useful for simultaneously loading the Sun Cluster software on all of the cluster host systems.

4. If the cluster host systems are not booted, boot them now.

   ```
   ok boot
   ```

5. After all cluster host systems have completed their boot, log in as user `root`.
6. Practice using the Common window Group Term Windows feature under the Options menu. You can ungroup the `cconsole` windows, rearrange them, and then group them together again.

Task – Using the ccp Control Panel

The ccp control panel can be useful if you need to use the console tool variations `crlogin` and `ctelnet`.

1. Start the `ccp` tool (`#` **`ccp`** ***`clustername`*** **`&`**).
2. Practice using the `crlogin` and `ctelnet` console tool variations.
3. Quit the `crlogin`, `ctelnet`, and `ccp` tools.

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

❑ List the Sun Cluster administration workstation functions

❑ Install the Sun Cluster console software on the administration workstation

❑ Set up the administration workstation environment

❑ Configure the Sun Cluster console software

Think Beyond

What is the advantage of the `/etc/clusters` and `/etc/serialports` files?

What is the impact on the cluster if the administrative workstation is not available? What alternatives could you use?

Module 4

Preinstallation Configuration

Objectives

Upon completion of this module, you should be able to:

- List the Sun Cluster boot disk requirements
- Physically configure a cluster topology
- Configure a supported cluster interconnect system
- Identify single points of failure in a cluster configuration
- Identify the quorum devices needed for selected cluster topologies
- Verify storage firmware revisions
- Physically configure a public network group

Relevance

Discussion – The following question is relevant to your learning the material presented in this module:

- Why is so much preinstallation planning required for an initial software installation?

Additional Resources

Additional resources – The following references can provide additional information on the topics described in this module:

- *Sun Cluster 3.0 Installation Guide*, part number 806-1419
- *Sun Cluster 3.0 Hardware Guide*, part number 806-1420
- *Sun Cluster 3.0 Data Services Installation and Configuration Guide*, part number 806-1421
- *Sun Cluster 3.0 Data Service Developers Guide*, part number 806-1422
- *Sun Cluster 3.0 System Administration Guide*, part number 806-1423
- *Sun Cluster 3.0 Concepts*, part number 806-1424
- *Sun Cluster 3.0 Error Message Guide*, part number 806-1426
- *Sun Cluster 3.0 Release Notes*, part number 806-1428

Cluster Server Configuration

The servers you use in a Sun Cluster 3.0 configuration have a number of general software and hardware requirements. These are *required* to qualify for support. The requirements include both hardware and software in the following areas:

- Boot device configuration and restrictions
- Minimum number of CPU modules and memory

Boot Device Restrictions

With the Sun Cluster 3.0 release, there are several restrictions on boot devices including:

- A shared storage device cannot be used as a boot device. If a storage device is connected to more than one host, it is shared.
- The Solaris 8 Operating Environment is supported with the following restrictions:
    - Sun Cluster 3.0 requires a minimum of End User level support. Data service applications are likely to require a higher level of support such as Developer or Entire Distribution.
    - Alternate Pathing (AP) is not supported.
    - Dynamic reconfiguration (DR) is not supported.
- The boot disk partitions have several requirements:
    - Swap space must be twice the size of memory with a minimum size of 750 Mbytes.
    - There must be a 100-Mbyte `/globaldevices` file system.
    - The Solstice DiskSuite application requires a 10-Mbyte partition for replicas. This should be slice 7 and should start at cylinder 0.

Note – The 100-Mbyte `/globaldevices` file system is modified during the Sun Cluster 3.0 installation. It is automatically renamed to be `/global/.devices`.

Boot Disk JumpStart Profile

If you decide to configure your cluster servers using the JumpStart utility, the following JumpStart profile represents a starting point for boot disk configuration:

```
install_type       initial_install
system_type        standalone
partitioning       explicit
cluster            SUNWCall
usedisk            c0t0d0
filesys            c0t0d0s1 1024 swap
filesys            c0t0d0s4 100 /globaldevices
filesys            c0t0d0s7 10
filesys            c0t0d0s0 free /
```

Server Hardware Restrictions

All cluster servers must meet the following hardware requirements:

- Each server must have a minimum of 512-Mbytes of memory.
- Each server must have a minimum of two CPUs.
- Servers in a cluster can be heterogeneous with restrictions:
    - Only Sun Enterprise 220R/250/450 systems can be mixed.
    - Only Sun Enterprise 3500/4500/5500/6500 systems can be mixed.
    - Sun Enterprise 10000 servers should have a minimum of two system boards in each domain.

Cluster Topology Configurations

You can configure a Sun Cluster system in several ways, called topologies. Topology configurations are determined by the types of disk storage devices used in a cluster and how they are physically connected to the cluster host systems.

Sun Cluster 3.0 supports the following topologies:

- Clustered Pairs topology
- N+1 topology
- Pair + N

Clustered Pairs Topology

A clustered pairs topology is two or more pairs of nodes operating under a single cluster administrative framework. The nodes in a pair are backups for one another. In this configuration, failover occurs only between a pair. However, all nodes are connected by the private networks and operate under Sun Cluster software control.

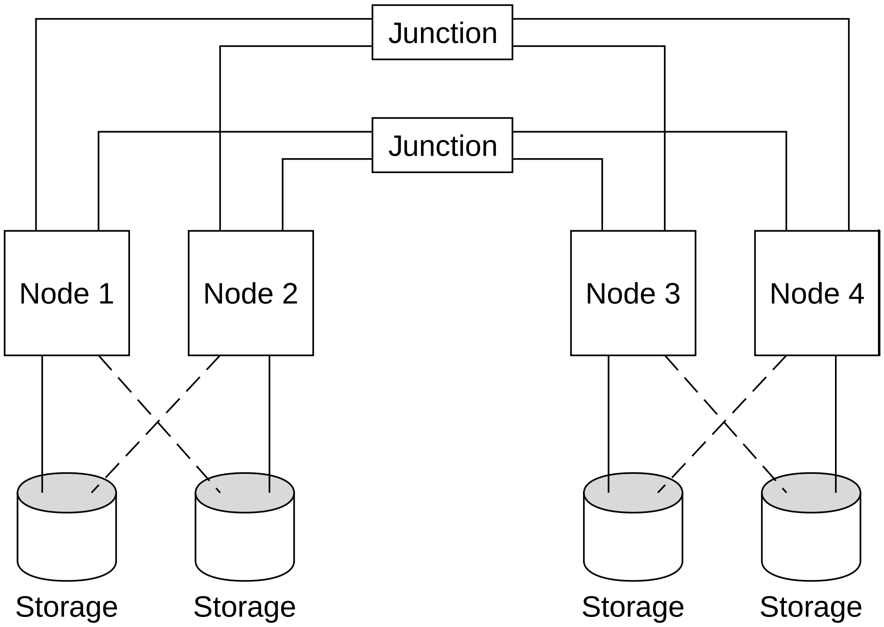


Figure 4-1 Clustered Pairs Topology Configuration

The features of Clustered Pairs configurations are:

- Nodes are configured in pairs. Possible configurations include either two, four, six, or eight nodes.
- Each pair has shared storage. Storage is connected to both nodes in the pair.

The benefits of Clustered Pairs configurations are:

- All nodes are part of the same cluster, simplifying administration.
- Since each pair has its own storage, no one node needs to be of significantly higher capacity than the others.
- The cost of the cluster interconnect is spread across all the nodes.
- This configuration is well suited for failover data services.

The limitations of Clustered Pairs configurations are:

- Each node in a pair must not be running at maximum capacity or it cannot handle the additional load of a failover.

Pair+N Topology

The Pair+N topology includes a pair of nodes directly connected to shared storage and an additional set of nodes that must use the cluster interconnect to access shared storage because they have no direct connection themselves. This configuration can be expanded to a total of eight nodes.

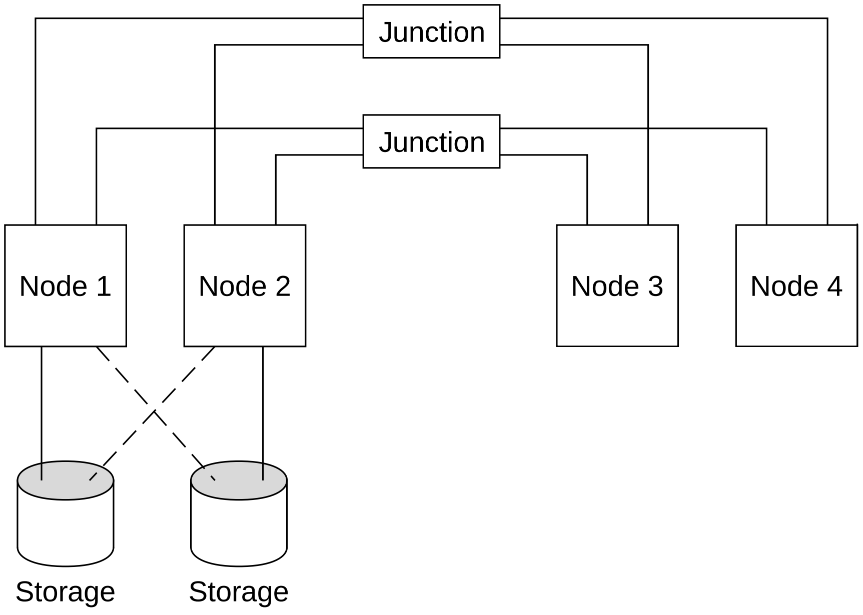


Figure 4-2 Pair+N Topology

The features of Pair+N configurations are:

- All shared storage is dual-hosted and connected to a single pair.
- Additional cluster nodes are used to support scalable data services.
- Possible configurations include a total of three to eight nodes in the cluster.
- There are common redundant interconnects between all nodes.
- All nodes in this configuration are still configured with volume managers.

The benefits of Pair+N configuration are:

- This configuration is well suited for scalable data services.

The limitations of Pair+N configurations are:

- There can be heavy data traffic on the cluster interconnects.

N+1 Topology

The N+1 topology, shown in Figure 4-3, provides one system to act as the backup for every other system in the cluster. All of the secondary paths to the storage devices are connected to the redundant or secondary system, which can be running a normal workload of its own.

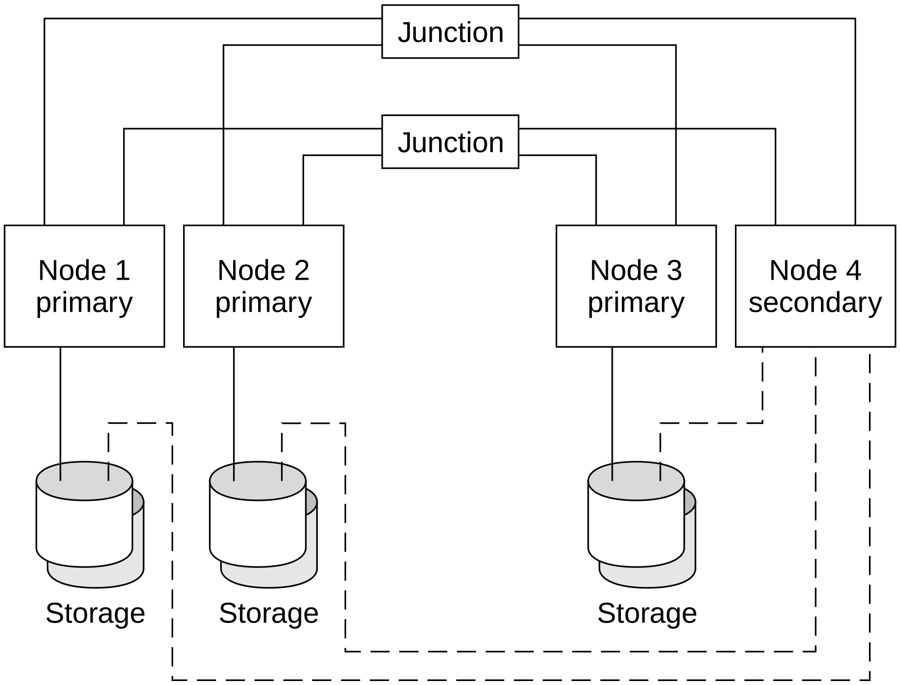


Figure 4-3 N+1 Topology Configuration

The features of N+1 configurations are:

- The secondary node is the only node in the configuration that is physically connected to all the multihost storage.

The benefits of N+1 configurations are:

- The backup node can take over without any performance degradation.
- The backup node is more cost effective because it does not require additional data storage.
- This configuration is best suited for failover data services.

The limitations of N+1 configurations are:

- If there is more than one primary node failure, the secondary node can be overloaded.

Storage Configuration

Each of the supported storage devices have configuration rules that you must follow. Some of the restrictions are common to any installation of the storage device and others are unique to the Sun Cluster 3.0 environment.

You must follow the storage configuration rules to qualify for Sun Cluster 3.0 support.

Multipack Configuration

Multipack configurations are relatively simple. The main configuration limitation is they cannot be daisy chained. Figure 4-4 illustrates a typical Sun Cluster 3.0 Multipack configuration.

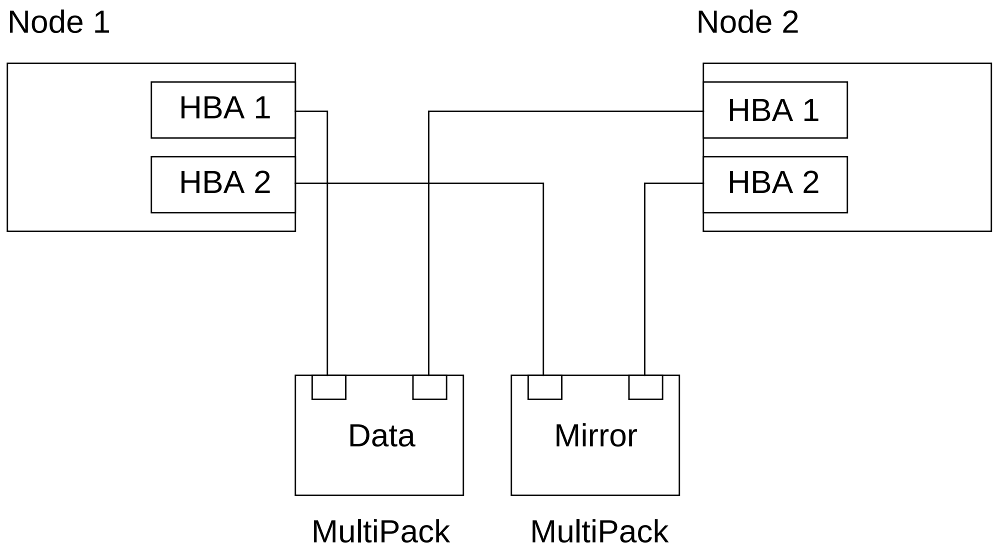


Figure 4-4 Multipack Configuration

Note – You must change the SCSI initiator ID on one of the HBAs connected to a MultiPack. The process is complex and should be performed by someone who is familiar with OpenBoot™ PROM `nvramrc` programming.

Sun StorEdge D1000 System Configuration

Sun StorEdge D1000 system configurations have configuration restrictions similar to the MultiPack storage:

- Daisy chaining of Sun StorEdge D1000 systems is not supported.
- A single Sun StorEdge D1000 system, in a split-bus configuration, is not supported.

Figure 4-5 illustrates a typical Sun Cluster 3.0 D1000 system configuration.

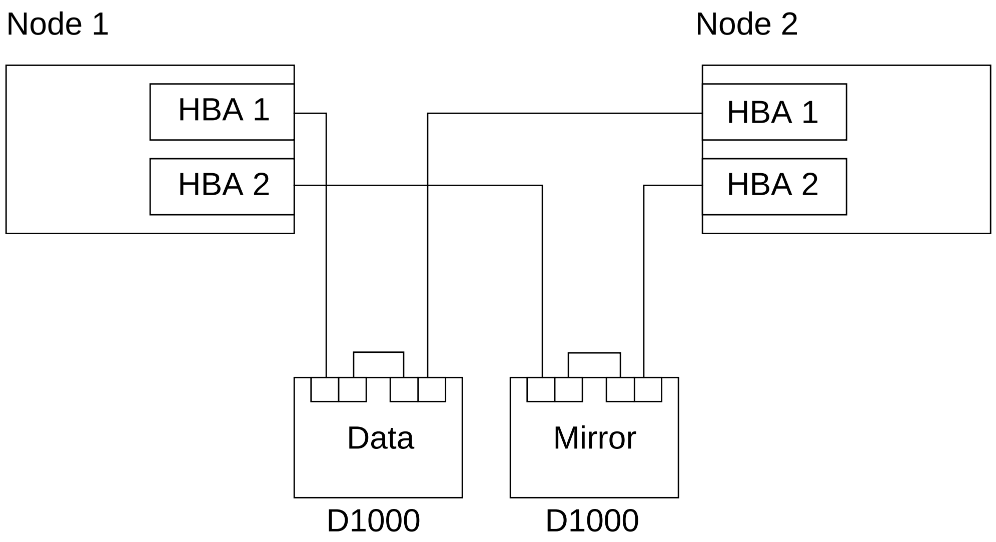


Figure 4-5 Sun StorEdge D1000 System Configuration

Sun StorEdge A3500 System Configuration

The configuration rules for using the Sun StorEdge A3500 storage array in the Sun Cluster 3.0 environment are as follows:

- Daisy chaining of the controller modules is not supported.
- A Sun StorEdge A3500 storage array with a single controller module is supported.
- The Sun StorEdge A3500 Lite system is supported.
- The Sun StorEdge A3500FC system is not supported.
- It is required to connect the two SCSI ports of a controller module to different HBAs on a node.
- The Sun StorEdge A3500 system disks cannot be used as a quorum devices.

Figure 4-6 illustrates a typical Sun Cluster 3.0 A3500 system configuration.

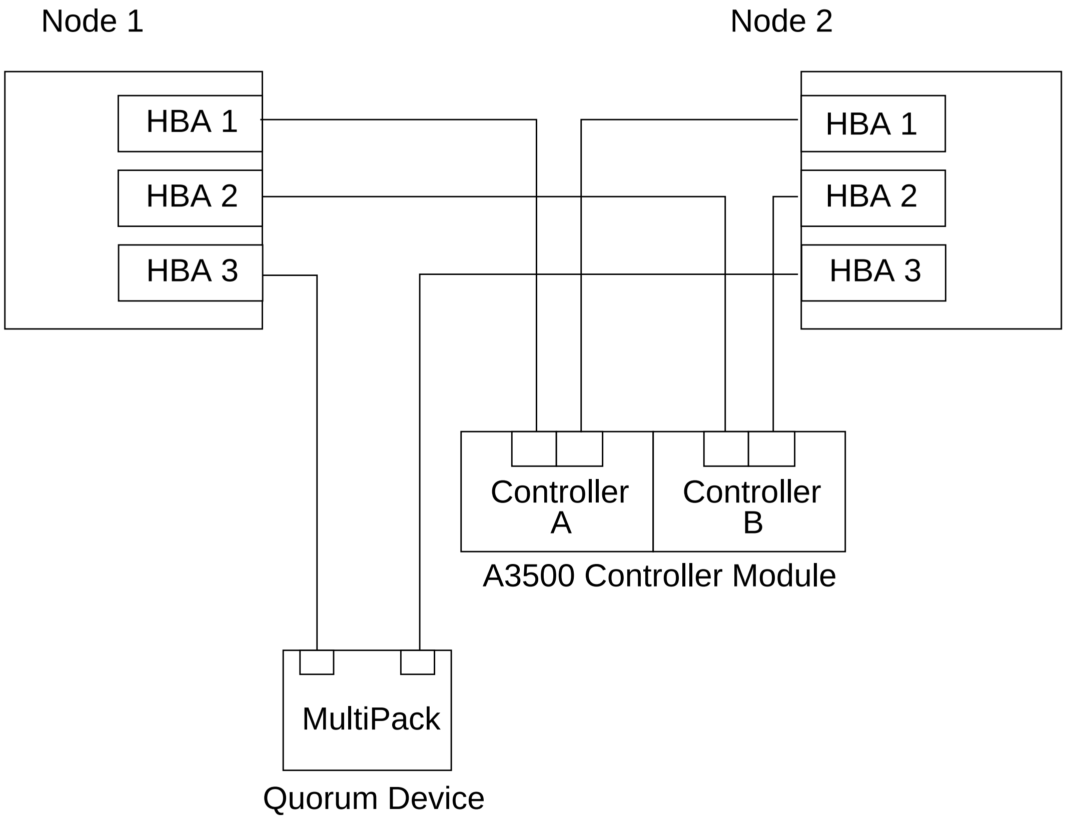


Figure 4-6 Sun StorEdge A3500 System Configuration

Sun StorEdge A5x00 System Configuration

The Sun StorEdge A5000/5100/5200 storage arrays have two primary restrictions when used in the Sun Cluster 3.0:

- A maximum of two host connections per loop.

  A full-loop configured Sun StorEdge A5x00 array can have only two host system connections and each connection must be made through a different Sun StorEdge A5x00 system interface board.

  A split-loop configured Sun StorEdge A5x00 system can have two host system connections to each IB.

- Peripheral component interconnect (PCI)-based FC-100 interface boards must be connected to Sun StorEdge A5x00 storage arrays through FCAL hubs (hub-attached).

  SBus-based FC-100 interface boards are attached directly to Sun StorEdge A5x00 storage arrays (direct-attached).

- Daisy chaining of Sun StorEdge A5x00 storage arrays is not supported.

Direct-Attached Full-Loop Sun StorEdge A5x00 System Configuration

The configuration shown in Figure 4-7 is typical of a simple two-node failover cluster. It could also be used in a Pair+N cluster.

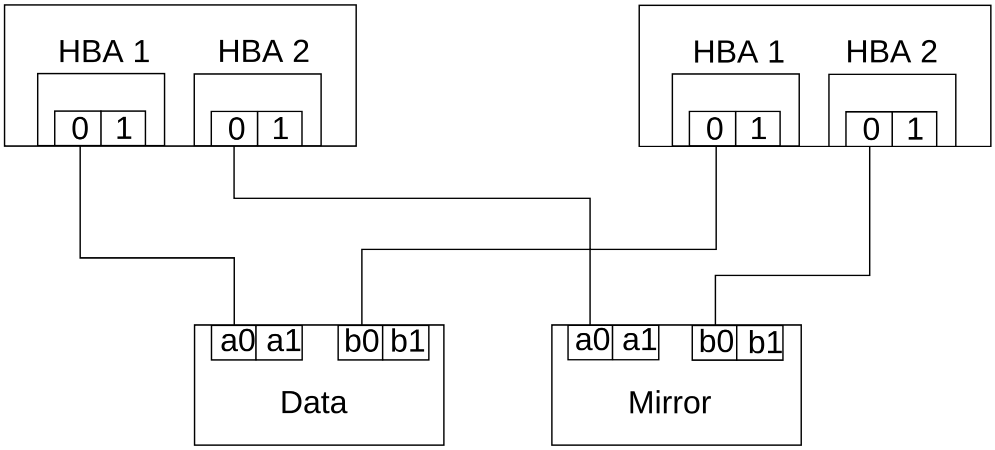


Figure 4-7 Direct-Attached Full-Loop Sun StorEdge A5x00 System

Direct-Attached Split-Loop Sun StorEdge A5x00 System Configuration

The direct-attached split-loop Sun StorEdge A5x00 system configuration shown in Figure 4-8 is useful when a smaller amount of storage is needed.

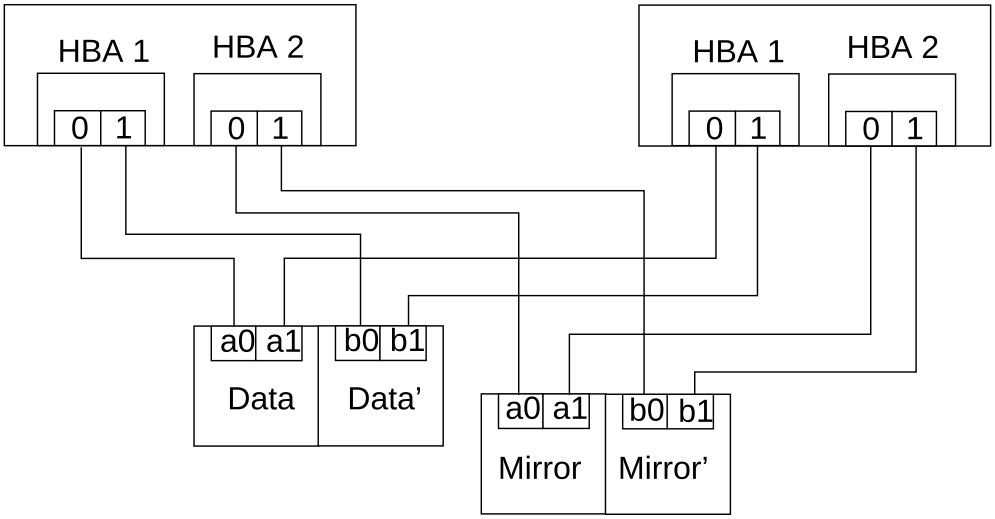


Figure 4-8 Direct-Attached Split-Loop Sun StorEdge A5x00 System

Hub-Attached Full-Loop Sun StorEdge A5x00 System Configuration

The hub-attached full-loop Sun StorEdge A5x00 system configuration show in Figure 4-9 is a common configuration for PCI-based failover clusters.

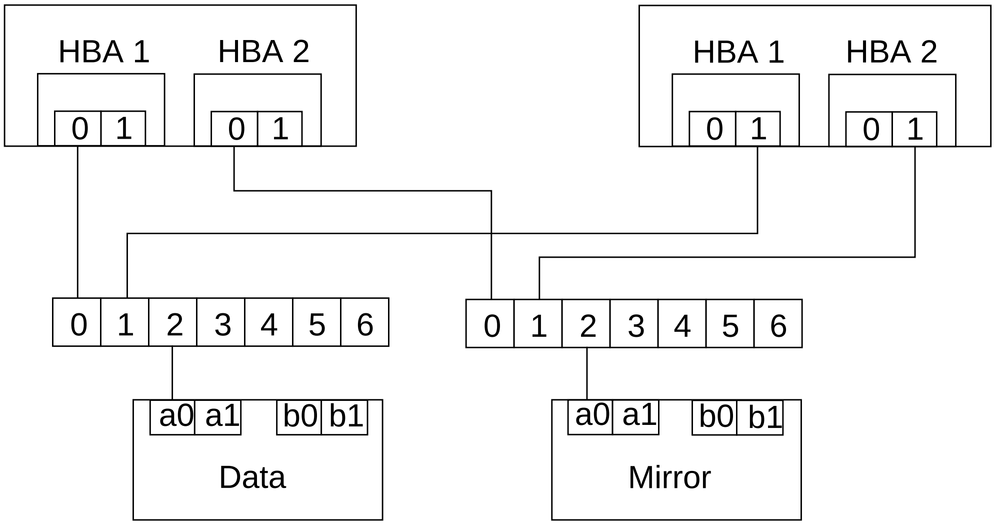


Figure 4-9 Hub-Attached Full-Loop Sun StorEdge A5x00 System

Cluster Interconnect Configuration

There are two variations of cluster interconnects: point-to-point and junction-based. The junctions must be switches and not hubs.

Point-to-Point Cluster Interconnect

In a two-node cluster, the interconnect interfaces can be directly connected using crossover cables. A point-to-point interconnect configuration using 100Base-T interfaces is illustrated in Figure 4-10.

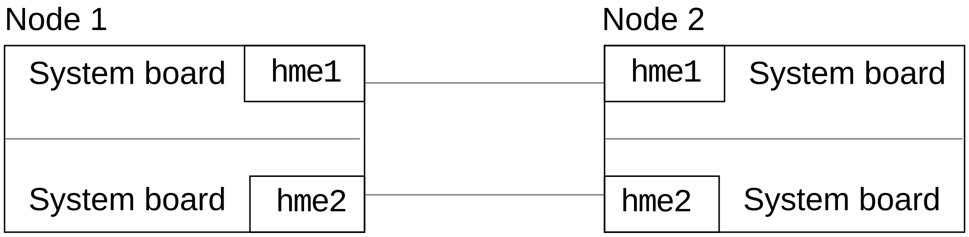


Figure 4-10 Point-to-Point Cluster Interconnect

During the Sun Cluster 3.0 installation, you must furnish the names of the end-point interfaces for each cable.

Caution – If you furnish the wrong interconnect interface names during the initial Sun Cluster installation, the first node installs without errors, but when you try to install the second node, the installation will hang indefinitely. You have to correct the cluster configuration error on the first node and then restart the installation on the second node.

Junction-based Cluster Interconnect

In cluster configurations that are greater than two nodes, you must join the interconnect interfaces using switches. Two-node cluster interconnects can also use switches. This can be in preparation for expanding the number of nodes at a later time. A typical junction-based interconnect is illustrated in Figure 4-11.

During the Sun Cluster 3.0 software installation, you are asked whether the interconnect system uses junctions. If you answer yes, you are asked to furnish names for each of the switches.

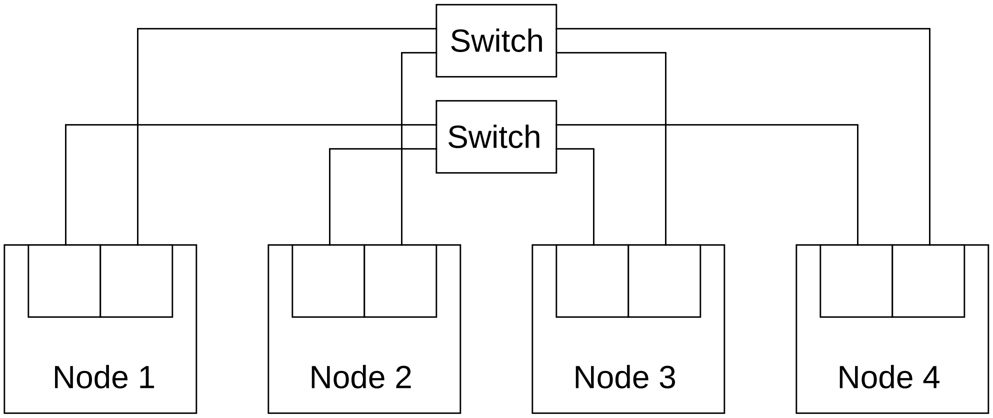


Figure 4-11 Junction-based Cluster Interconnect

Note – If you specify more than two nodes during the initial portion of the Sun Cluster software installation, the use of junctions is assumed.

Cluster Transport Interface Addresses

During the Sun Cluster software installation, the cluster interconnect are assigned IP addresses based on a base address of `172.16.0.0`. If necessary, you can override the default address but this is not recommended. Uniform addresses can be a benefit during problem isolation.

Identifying Cluster Transport Interfaces

Identifying network interfaces is not a simple task. Several steps are required to accurately determine the logical name of each interface on a system. A general procedure follows.

1. Look for network interfaces with the `prtconf` command. Typical instances you might see are `network, instance #0`, `SUNW,hme, instance #1`, and `SUNW,hme, instance #2`.

2. Verify which interfaces are already up.

```
# ifconfig -a
lo0:
flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4>
mtu 8232 index 1 inet 127.0.0.1 netmask ff000000
hme0:
flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
mtu 1500 index 2 inet 129.200.9.2 netmask ffffff00
broadcast 129.200.9.255 ether 8:0:20:96:3:86
```

3. Bring up the unplumbed interfaces for testing.

```
# ifconfig hme1 plumb
# ifconfig hme1 up
# ifconfig hme2 plumb
# ifconfig hme2 up
```

4. Verify the new interfaces are up.

```
# ifconfig -a
```

5. Test each of the new interfaces while plugging and unplugging them from an active network.

```
# ping -i hme1 pnode2
pnode2 is alive
# ping -i hme2 pnode2
pnode2 is alive
```

6. After you have identified the new network interfaces, bring them down again.

```
# ifconfig hme1 down
# ifconfig hme2 down
# ifconfig hme1 unplumb
# ifconfig hme2 unplumb
```

Eliminating Single Points of Failure

A single point of failure is any hardware or software configuration item that can completely eliminate access to data if it fails.

An example of a software-oriented single point of failure is creating RAID 1 mirroring within a single storage array. If the array has a major failure, all access to the data is lost.

There are also practices that are designed to increase data availability but are not related to single points of failure.

Each of the following rules describe a best practice that should be used whenever possible in a clustered configuration.

- RAID 1 mirrors should reside in different storage arrays.

  If an array fails, one of the mirrors is still available.

- Host bus adapters should be distributed across system boards.

  A single system board failure should not disable access to both copies of mirrored data.

- Order equipment with optional redundancy if possible.

  Many system and storage array models have optional redundant power supplies and cooling. Attach each power supply to a different power circuit.

- Redundant cluster interconnects are *required*, not optional.
- Uninterruptible power supply (UPS) systems and/or local power generators can increase overall cluster availability.

  Although expensive, UPS systems and/or local power generators can be worthwhile in critical cluster applications.

Cluster Quorum Device Configuration

Because cluster nodes share data and resources, the cluster must take steps to maintain data and resource integrity. The concept of quorum voting is used to control cluster membership.

Each node in a cluster is assigned a vote. To form a cluster and for the cluster to remain up, a majority of votes must be present. In order to form a two-node cluster, for instance, a majority of votes would be two. Without some modification to a two-node cluster, both nodes would have to be booted before a cluster could form. An additional problem is that the cluster cannot continue if a single node fails. This is a single point of failure that defeats the high-availability requirement.

This problem is resolved by assigning a vote to a disk drive called a *quorum device* which is assigned a single vote. Now when a single node tries to come up, it reserves the quorum device. There is now a majority of two votes out of three possible.

Quorum devices are also used for another purpose: *failure fencing*. As shown in Figure 4-12, if interconnect communication between nodes ceases, either because of a complete interconnect failure or a node crashing, each node must assume the other is still functional. This is called *split-brain* operation. Two separate clusters cannot be allowed to exist because of potential data corruption. Each node tries to establish a cluster by gaining another quorum vote. Both nodes attempt to reserve the designated quorum device. The first node to reserve the quorum disk establishes a majority and remains as a cluster member. The node that fails the race to reserve the quorum device aborts the Sun Cluster software because it does not have a majority of votes.

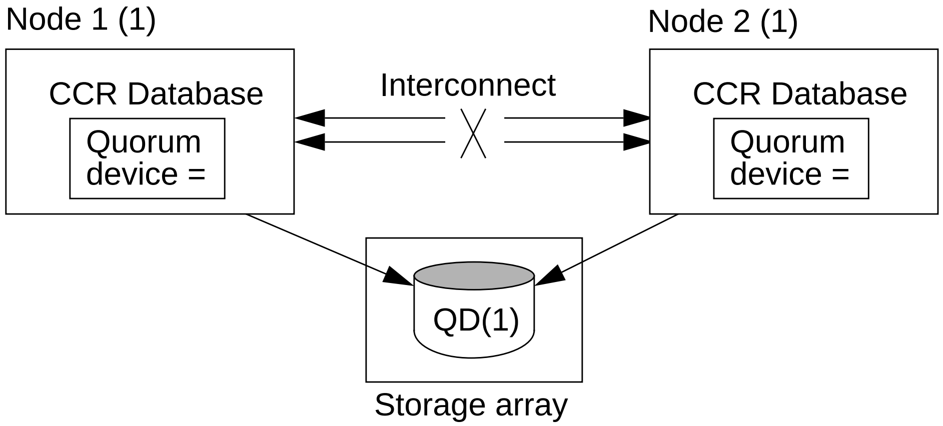


Figure 4-12 Failure Fencing

Quorum Device Rules

The general rules for quorum devices are:

- A quorum device must be available to both nodes in a two-node cluster.
- Quorum device information is maintained globally in the cluster configuration repository (CCR) database.
- A quorum device can contain user data.
- The *maximum* number of votes contributed by quorum devices should be the number of node votes minus one (N-1)

  If the number of quorum devices equals or exceeds the number of nodes, the cluster cannot come up if too many quorum devices fail.
- Quorum devices are not required in clusters with greater than two nodes but they are recommended for higher cluster availability.
- Quorum devices are manually configured after the Sun Cluster software installation is complete.
- Quorum devices are configured using DID devices and are available only to directly attached nodes.

Two-Node Cluster Quorum Devices

As shown in Figure 4-13, a two-node cluster needs a single quorum disk. The total votes are three. With the quorum disk, a single node can start clustered operation with a majority of votes (2).

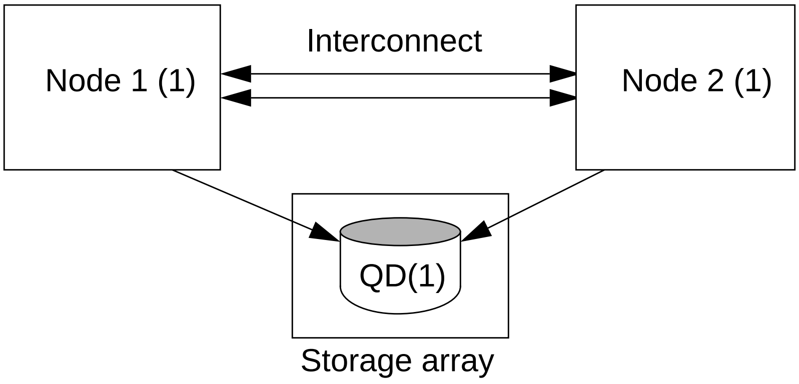


Figure 4-13 Two-Node Cluster Quorum Devices

Clustered Pair Quorum Devices

In a clustered pairs configuration shown in Figure 4-14, there are always an even number of cluster nodes (2, 4, 6, 8). The nodes in each pair usually provide data service failover backup for one another.

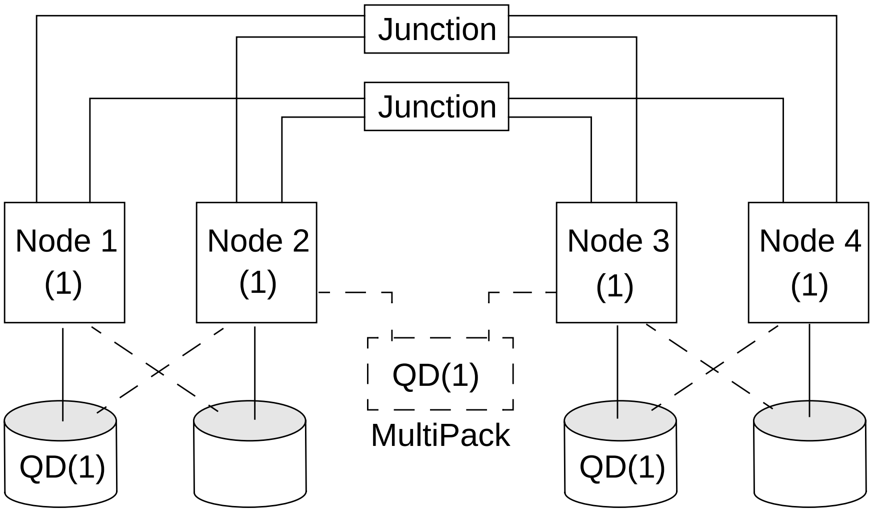


Figure 4-14 Clustered Pair Quorum Devices

There are many possible split-brain scenarios. Not all of the possible split-brain combinations allow the continuation of clustered operation. The following is true for a clustered pair configuration:

- There are six possible votes.
- A quorum is four votes.
- If both quorum devices fail, the cluster can still come up.

  The nodes wait until all are present (booted).
- If Nodes 1 and 2 fail, there are not enough votes for Nodes 3 and 4 to continue running

  A token quorum device between Nodes 2 and 3 can eliminate this problem. A MultiPack could be used for this purpose.
- A node in each pair can fail and there are still four votes.

Pair+N Quorum Devices

Figure 4-15 illustrates a typical quorum disk configuration in a Pair+2 configuration. Three quorum disks are used. You use this configuration for scalable data services.

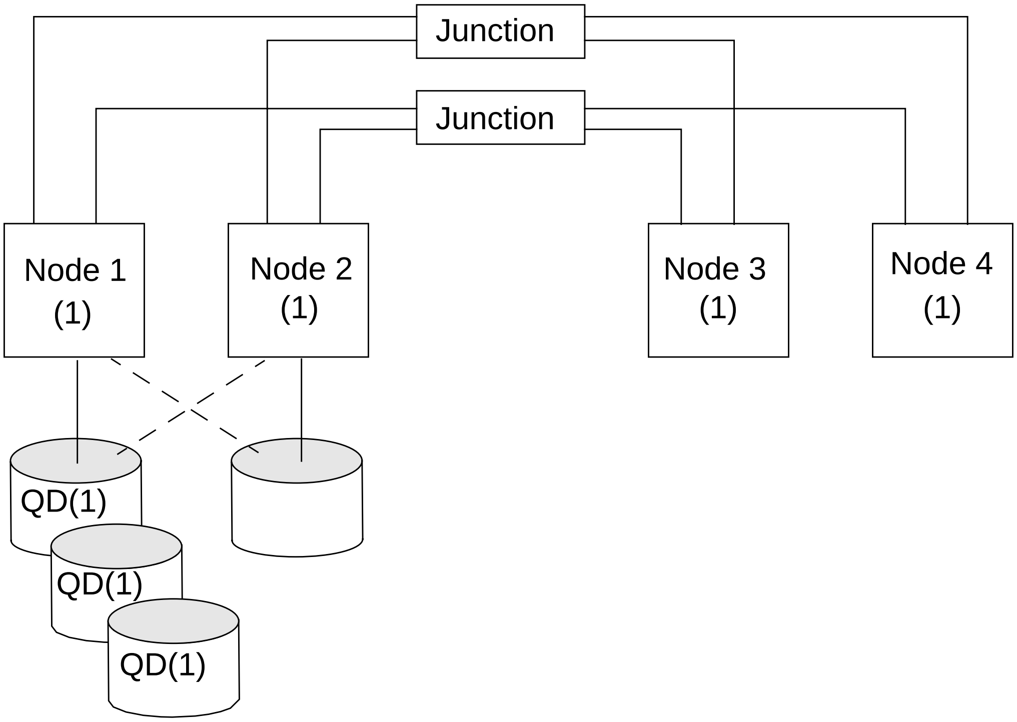


Figure 4-15 Pair+N Quorum Devices

The following is true for the Pair+N configuration shown in Figure 4-15:

- There are three quorum disks.
- There are seven possible votes.
- A quorum is four votes.
- Nodes 3 and 4 do not have access to any quorum devices.
- Nodes 1 or 2 can start clustered operation by themselves.
- Up to 3 nodes can fail (1, 3, and 4 or 2, 3, and 4) and clustered operation can continue.

N+1 Quorum Disks

The N+1 configuration shown in Figure 4-16, requires a different approach. Node 3 is the failover backup for both Node 1 and Node 2.

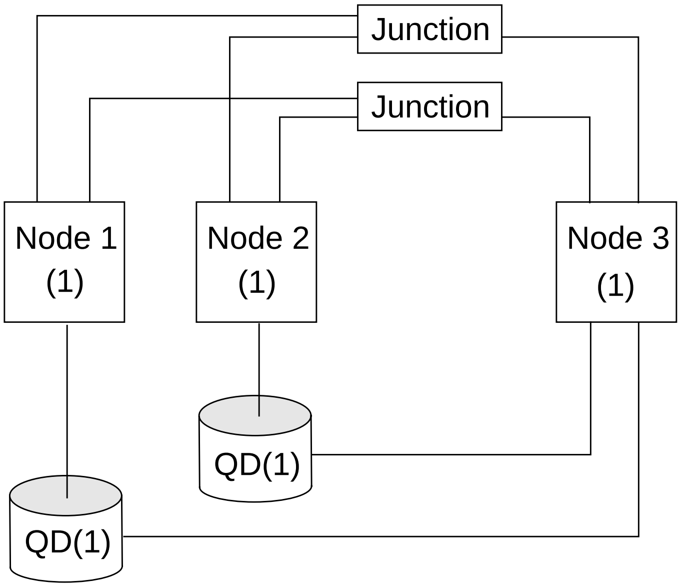


Figure 4-16 N+1 Quorum Devices

The following is true for the N+1 configuration shown in Figure 4-16:

- There are five possible votes.
- A quorum is three votes.
- If Node 1 and 2 fail, Node 3 can continue.

Public Network Configuration

The PNM software creates and manages designated groups of local network adapters commonly referred to as NAFO groups. If a cluster host network adapter fails, its associated IP address is transferred to a backup adapter in its group.

As shown in Figure 4-17, the PNM daemon (`pnmd`) continuously monitors designated network adapters on a single node. If a failure is detected, `pnmd` uses information in the CCR and the `pnmconfig` file to initiate a failover to a healthy adapter in the backup group.

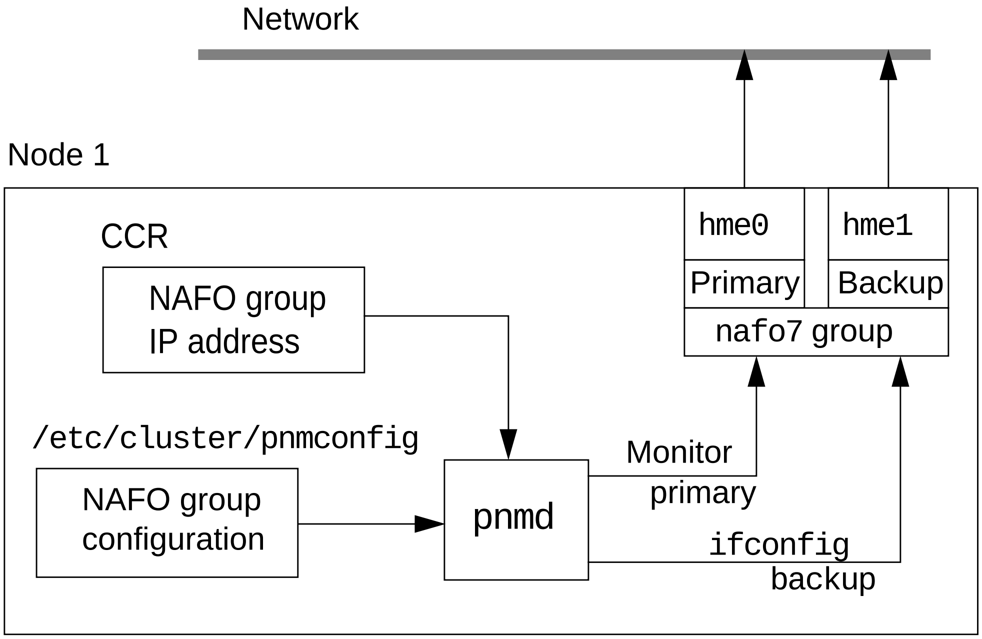


Figure 4-17 Public Network Management Components

The adapters within a group must meet the following criteria:

- All adapters in a NAFO group must be of the same media type.
- The adapters can be 100-Mbit/second or Gigabit Ethernet.

Identifying Storage Array Firmware

Only the Sun StorEdge A5x00 storage arrays have significant amounts of firmware. A typical Sun StorEdge A5x00 system installation has firmware in the following locations:

- In each host bus adapter (HBA)
- In each array interface board (IB)
- In many array disk drive models

Most of the firmware revisions can be verified using the `luxadm` command. The `luxadm` command is a standard Solaris Operating Environment command.

Note – The firmware can be updated using the `luxadm` command or in some cases special programs supplied with firmware patches. The firmware upgrade process is complex and can permanently disable storage array interfaces unless performed correctly. It is advisable to contact your Sun field representative about any firmware related questions.

Identifying Attached Sun StorEdge A5x00 Storage Arrays

The `luxadm probe` option displays basic information about all attached Sun StorEdge A5x00 storage arrays.

```
# /usr/sbin/luxadm probe
Found Enclosure(s):
SENA          Name:AA  Node WWN:5080020000034ed8
  Logical Path:/dev/es/ses1
  Logical Path:/dev/es/ses3
SENA          Name:BB  Node WWN:5080020000029e70
  Logical Path:/dev/es/ses6
  Logical Path:/dev/es/ses7
```

Identifying Host Bus Adapter Firmware

The `luxadm` command can display information about HBA firmware revisions for the FC/S and FC100/S fiber channel cards. Currently, the FC100/P (PCI-based) fiber channel firmware can not be checked or downloaded.

The command to check HBA card firmware is:

```
# luxadm fcal_s_download -p
```

Note – The older FC/S SBus cards are not supported for use with Sun Cluster 3.0 and Sun StorEdge A5x00 storage arrays.

An example of the command output follows.

```
# /usr/sbin/luxadm fcal_s_download -p

  Found Path to 0 FC/S Cards
  Complete

  Found Path to 0 FC100/S Cards
  Complete

  Found Path to 2 FC100/P, FC100/2P Devices

 Opening Device: /devices/pci@6,4000/SUNW,ifp@3:devctl
  Detected FCode Version:
  No version available for this FCode

 Opening Device: /devices/pci@6,4000/SUNW,ifp@4:devctl
  Detected FCode Version:
  No version available for this FCode
  Complete
```

Identifying Sun StorEdge A5x00 Interface Board Firmware

The Sun StorEdge A5x00 storage arrays can have two interface boards. Their firmware revisions levels can be checked as follows:

```
# luxadm display enclosure_name
```

Note – Sun StorEdge A5x00 storage arrays are usually assigned unique names from their LED display panel.

A typical `luxadm display` output follows.

```
# luxadm display BB

                                   SENA
                                 DISK STATUS
SLOT   FRONT DISKS      (Node WWN)         REAR DISKS    (Node WWN)
0     On (O.K.)        20000020370c2d2b   Not Installed
1     On (O.K.)        20000020370cbc90   Not Installed
2      Not Installed                         On (O.K.)   20000020370c9085
3     On (O.K.)        20000020370d6570   Not Installed
4      Not Installed                         On (O.K.)   20000020370caa26
5     On (O.K.)        20000020370d6940   Not Installed
6     Not Installed                      Not Installed
                                SUBSYSTEM STATUS
FW Revision:1.09   Box ID:1   Node WWN:5080020000029e70   Enclosure
Name:BB
Power Supplies (0,2 in front, 1 in rear)
. . . . .
. . . . .
. . . . .
Loop  configuration
        Loop A is configured as a single loop.
        Loop B is configured as a single loop.
Language        USA English
```

Note – Although there can be two interfaces boards in a Sun StorEdge A5x00 storage array, the firmware displays as a single value and both boards load automatically during firmware upgrade procedures.

Exercise: Preinstallation Preparation

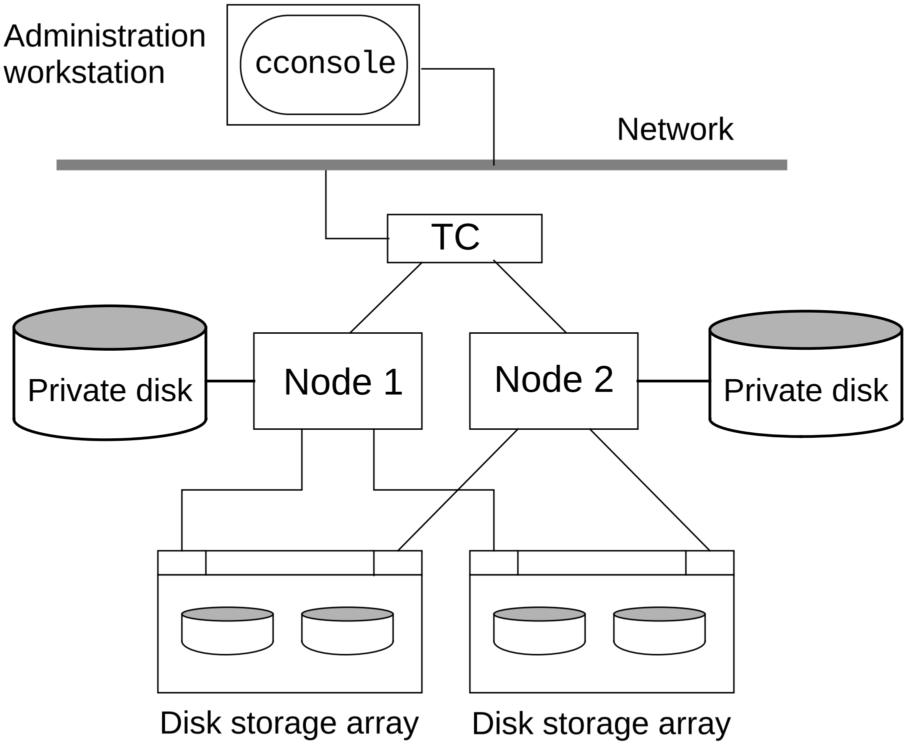
In this exercise, you complete the following tasks:

- Configure a cluster topology
- Identify quorum devices needed
- Verify the cluster interconnect cabling
- Verify storage array firmware revisions

Preparation

To begin this exercise, you must be connected to the cluster hosts through the `cconsole` tool and you logged into them as user `root`.

Your assigned cluster should be similar to the configuration shown in the following illustration:



Note – During this exercise, when you see italicized names such as *`IPaddress`*, *`enclosure_name`*, *`node1`*, or *`clustername`* imbedded in a command string, substitute the names appropriate for your cluster.

Task – Verifying the Solaris Operating Environment

In this section, you verify that the boot disk is correctly partitioned on all nodes.

1. Type the **`/etc/prtconf`** command on each node and record the size of physical memory (**`/etc/prtconf |grep Memory`**).

   Node 1 memory:_______________

   Node 2 memory:_______________

   Node 3 memory:_______________

2. Type the **`df -kl`** command on each node and verify the following are true:

   - Swap space is at least twice the size of memory
   - There is a 100-Mbyte `/globaldevices` file system

Task – Identifying a Cluster Topology

1. Record the desired topology configuration of your cluster.

Topology Configuration	
Number of nodes	
Number of storage arrays	
Types of storage arrays	

2. Verify that the storage arrays in your cluster are properly connected for your target topology. Recable the storage arrays if necessary.

Task – Selecting Quorum Devices

1. Record the estimated number of quorum devices you must configure after the cluster host software installation.

   Estimated number of quorum devices: __________

Note – Please consult with your instructor if you are not sure about your quorum device configuration.

2. Type the **`format`** command and record the logical path to a suitable quorum disk drive in one of your storage arrays.

Quorum Disk(s): ________________________________

2. Type Control-D to cleanly exit the `format` utility.

Task – Verifying the Cluster Interconnect Configuration

Configuring a Point-to-Point Ethernet Interconnect

Skip this section if your cluster interconnect is not point-to-point.

1. Ask your instructor for assistance in determining the logical names of your cluster interconnect interfaces.
2. Complete the form in Figure 4-18 if your cluster uses an Ethernet-based point-to-point interconnect configuration.

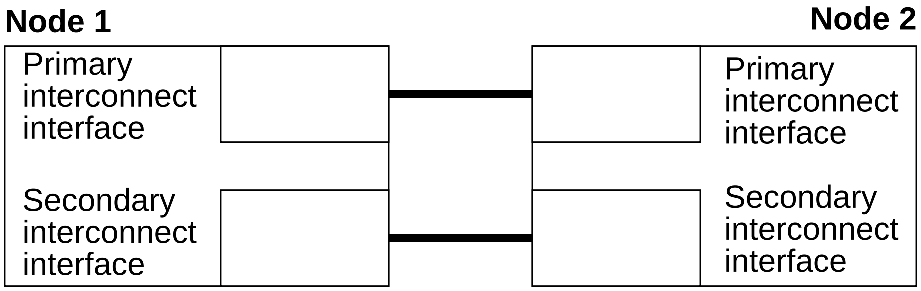

Node 1
Node 2
Primary interconnect interface
Primary interconnect interface
Secondary interconnect interface
Secondary interconnect interface

Figure 4-18 Ethernet Interconnect Point-to-Point Form

Configuring a Switch-Based Ethernet Interconnect

1. Complete the form in Figure 4-19 if your cluster uses an Ethernet-based cluster interconnect with switches. Record the logical names of the cluster interconnect interfaces (`hme2`, `qfe1`, ...)

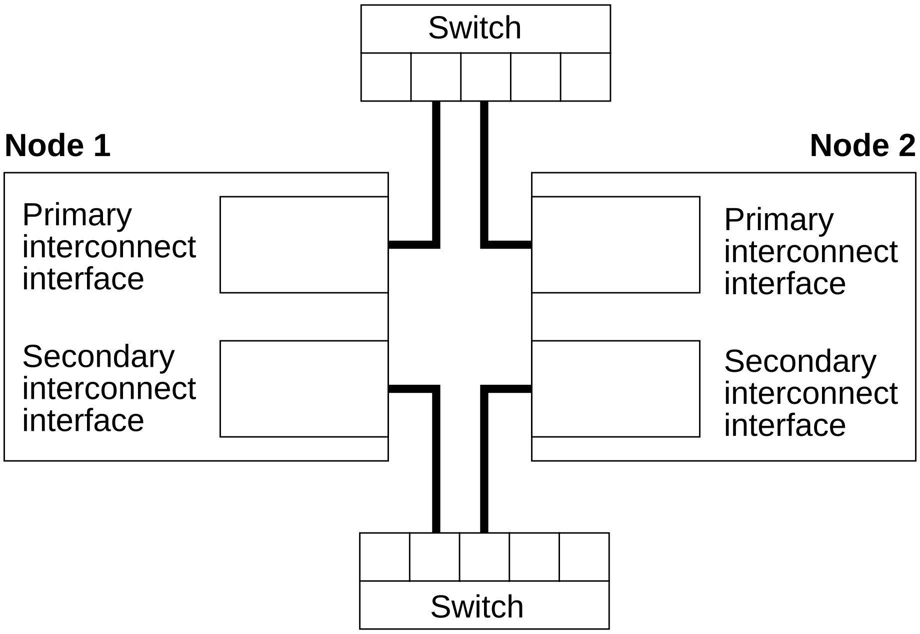

Switch
Node 1
Node 2
Primary interconnect interface
Secondary interconnect interface
Primary interconnect interface
Secondary interconnect interface
Switch

Figure 4-19 Ethernet Interconnect With Hubs Form

2. Verify that each Ethernet interconnect interface is connected to the correct hub.

Note – If you have any doubt about the interconnect cabling, consult with your instructor now. Do not continue this lab until you are confident that your cluster interconnect system is cabled correctly.

Task – Verifying Storage Array Firware Revisions

1. If your assigned cluster uses Sun StorEdge A5x00 storage arrays, use the **`luxadm probe`** command to identify the names of attached storage arrays. Record the enclosure names.

   Enclosure name: __________

   Enclosure name: __________

2. Use the **`luxadm fcal_s_download -p`** command to display the firmware revision of any SBus HBA cards. Record the HBA firmware revisions.

   HBA firmware revision: __________

   HBA firmware revision: __________

3. Use the **`luxadm display`** *`enclosure_name`* command to display the firmware revision of each Sun StorEdge A5x00 storage array.

   Array name: __________Firmware: __________

   Array name: __________Firmware: __________

Task – Selecting Public Network Interfaces

Ask for help from your instructor in identifying two public network interfaces on each node that can be used in PNM NAFO groups.

1. Record the logical names of two potential PNM Ethernet interfaces on each node.

System	Primary NAFO interface	Backup NAFO interface
Node 1		
Node 2		

Note – It is important that you are sure about the logical name of each NAFO interface (`hme2`, `qfe3`, etc...).

2. Verify that the target NAFO interfaces on each node are connected to a public network.

Exercise Summary

Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

❑ List the Sun Cluster boot disk requirements

❑ Physically configure a cluster topology

❑ Configure a supported cluster interconnect system

❑ Identify single points of failure in a cluster configuration

❑ Identify the quorum devices needed for selected cluster topologies

❑ Verify storage firmware revisions

❑ Physically configure a public network group

Think Beyond

What additional preparation might be necessary before installing the Sun Cluster host software?

Module 5

Cluster Host Software Installation

Objectives

Upon completion of this module, you should be able to:

- Install the Sun Cluster host system software
- Correctly interpret configuration questions during the Sun Cluster software installation
- Perform postinstallation configuration

Relevance

Discussion – The following questions are relevant to understanding this module's content:

- What configuration issues might control how the Sun Cluster software is installed?
- What type of postinstallation tasks might be necessary?
- What other software might you need to finish the installation?

Additional Resources

Additional resources – The following references can provide additional information on the topics described in this module:

- *Sun Cluster 3.0 Installation Guide*, part number 806-1419
- *Sun Cluster 3.0 Hardware Guide*, part number 806-1420
- *Sun Cluster 3.0 Data Services Installation and Configuration Guide*, part number 806-1421
- *Sun Cluster 3.0 Data Service Developers Guide*, part number 806-1422
- *Sun Cluster 3.0 System Administration Guide*, part number 806-1423
- *Sun Cluster 3.0 Concepts*, part number 806-1424
- *Sun Cluster 3.0 Error Message Guide*, part number 806-1426
- *Sun Cluster 3.0 Release Notes*, part number 806-1428

Sun Cluster Software Summary

Sun Cluster software is installed on a Sun Cluster hardware platform. The complete Sun Cluster 3.0 software collection shown in Figure 5-1, consists of the following CDs:

- Sun Cluster 3.0 CD
- Sun Cluster 3.0 Data Services CD
- Veritas Volume Manager 3.0.4 CD
- Solstice DiskSuite 4.2.1 (on the Solaris 8 Software 2 of 2)
- Sun Management Center CD-ROM

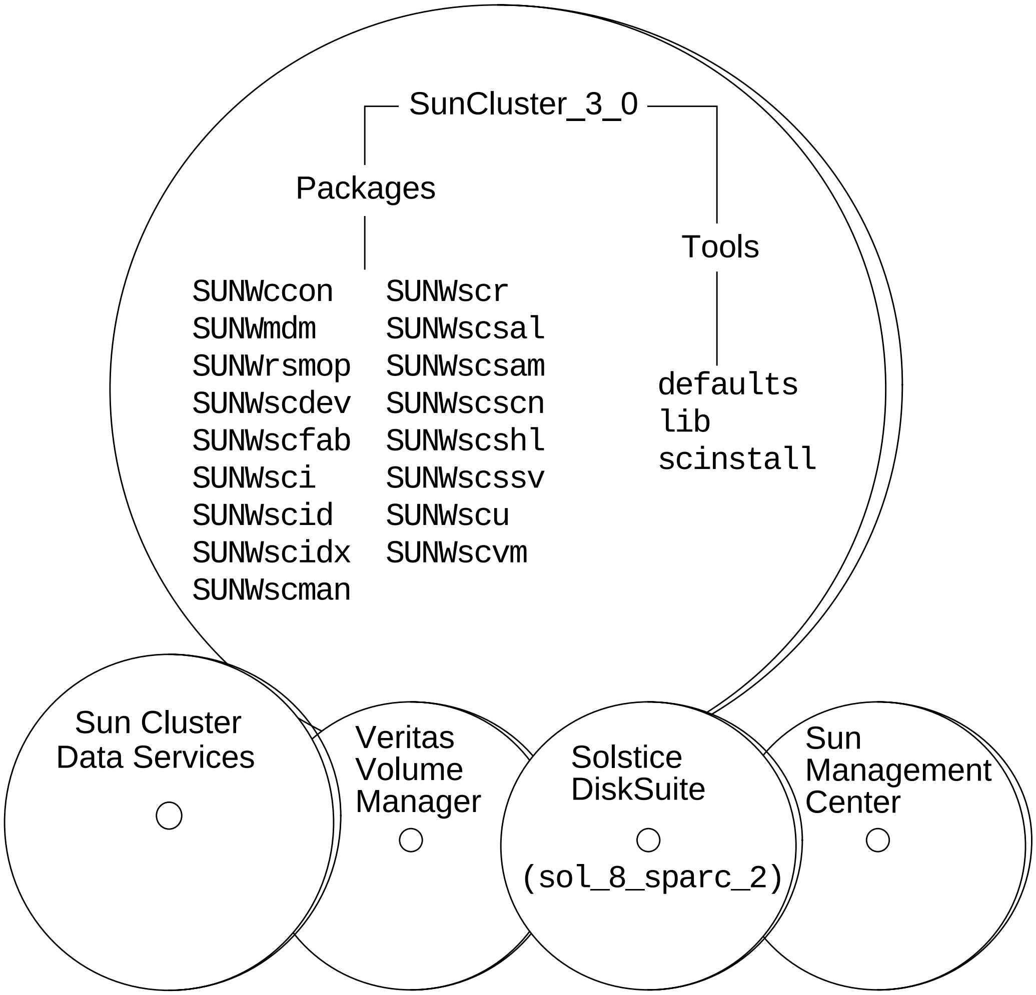


Figure 5-1 Sun Cluster CD Collection

Sun Cluster Software Distribution

As shown in Figure 5-2, you install the Sun Cluster server software on each of the cluster host systems along with the appropriate data service software and virtual volume management software.

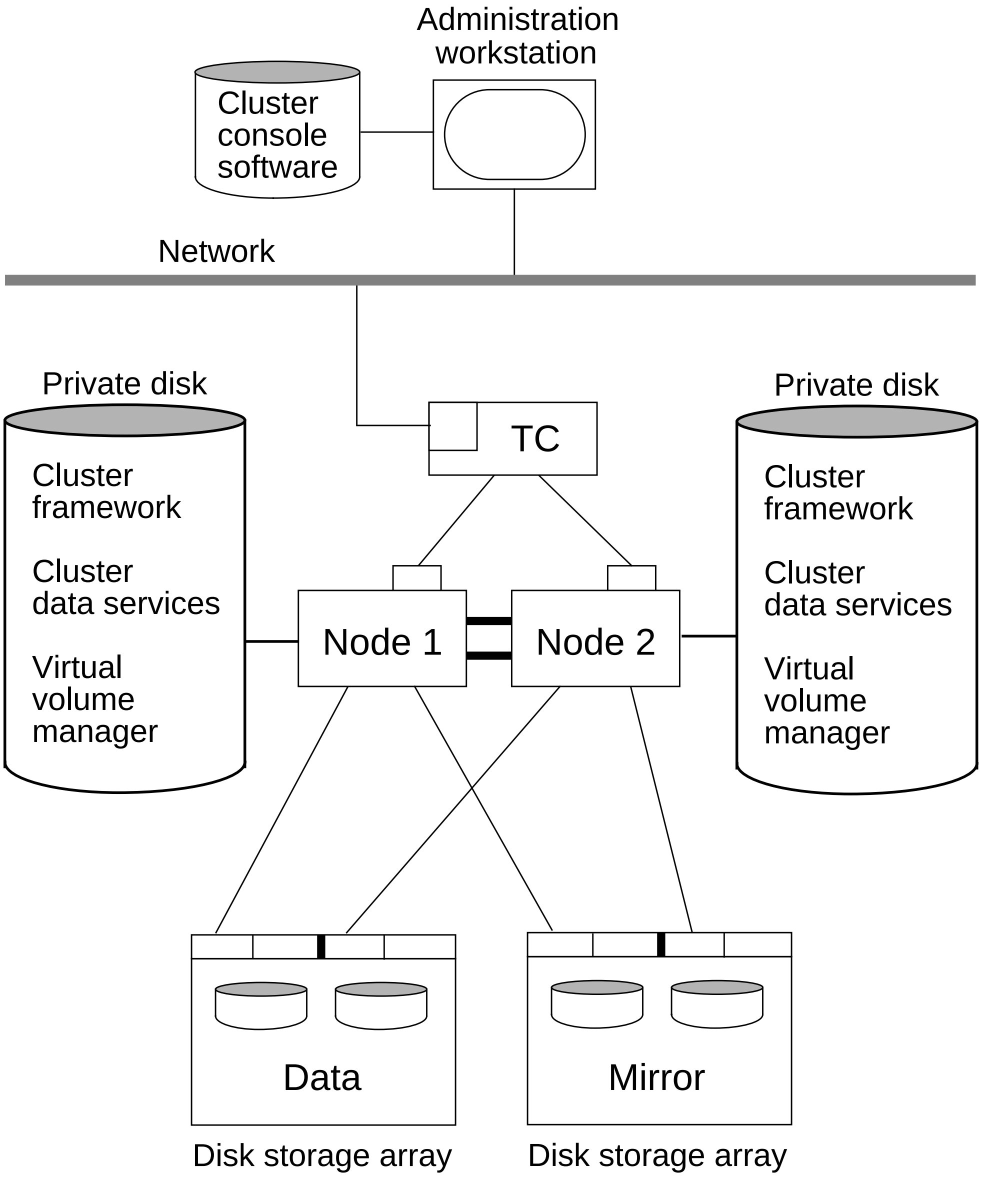


Figure 5-2 Cluster Software Distribution

Note – Oracle Parallel Server installations in the Sun Cluster 3.0 environment are installed exclusively on Sun StorEdge A3500 arrays and do not use virtual volume managers to provide RAID protection of data.

Sun Cluster Framework Software

The Sun Cluster 3.0 CD-ROM contains the following framework software packages:

- SUNWccon – Sun Cluster Console
- SUNWmdm – Solstice DiskSuite support (Mediator software)
- SUNWrsmop – Sun Cluster remote shared memory operations interface
- SUNWscdev – Sun Cluster Developer support
- SUNWscfab – SGML Documentation
- SUNWsci – Sun Cluster SCI Dolphin Driver
- SUNWscid – Sun Cluster SCI DLPI Driver
- SUNWscidx – Sun Cluster SCI DLPI Driver
- SUNWscman – Sun Cluster Manual Pages
- SUNWscr – Sun Cluster Framework software
- SUNWscsal – Sun Cluster SyMON agent library
- SUNWscsam – Sun Cluster SyMON modules
- SUNWscscn – Sun Cluster SyMON console add-on
- SUNWscshl – Sun Cluster SyMON help add-on
- SUNWscssv – Sun Cluster SyMON server add-on
- SUNWscu – Sun Cluster, (Usr)
- SUNWscvm – Sun Cluster VxVM Support

The software package, SUNWccon, SUNWscfab, SUNWscscn, SUNWscshl, and SUNWscssv are not part of the cluster framework software. The include administration console software and Sun Management Center agent packages.

Sun Cluster 3.0 Agents

The following data service support packages are available on the Sun Cluster 3.0 Agents CD:

- SunCluster_Data_Service_Answer_Book_3.0
- SunCluster_HA_Apache_3.0
- SunCluster_HA_DNS_3.0
- SunCluster_HA_NFS_3.0
- SunCluster_HA_Netscape_HTTP_3.0
- SunCluster_HA_Netscape_LDAP_3.0
- SunCluster_HA_Oracle_3.0
- SunCluster_Oracle_Parallel_Server_3.0

Note – The Oracle Parallel Server support supplies the Sun Cluster 2.2 version of the cluster membership monitor (CMM) that runs in user-mode. Support for Oracle distributed lock management (UDLM/IDLM) is also supplied along with special support for Oracle Parallel Server to work with the Sun StorEdge A3500 hardware RAID.

Virtual Volume Management Software

The Solstice DiskSuite software is available on the *Solaris 8 Software 2 of 2* CD in the `sol_8_sparc_2/Solaris_8/EA/products` directory.

The *Veritas Volume Manager 3.0.4* is shipped with the Sun Cluster 3.0 installation kit.

Sun Cluster 3.0 Licensing

No license keys are required for the Sun Cluster software. If you use MultiPacks, you need a special license for Veritas Volume Manager.

Veritas Volume Manager is automatically licensed when used with most Sun storage arrays.

You need a special license to use the Sun Management Center software.

Sun Cluster Installation Process

The Sun Cluster 3.0 host system (node) installation process is completed in several major steps. The general process is:

1. Repartition boot disks to meet Sun Cluster 3.0
2. Install the Solaris Operating Environment software
3. Configure the cluster host systems environment
4. Install Solaris 8 Operating Environment patches
5. Install hardware-related patches
6. Install Sun Cluster 3.0 software on the first cluster node
7. Install Sun Cluster 3.0 on the remaining nodes
8. Install any Sun Cluster patches
9. Perform postinstallation checks and configuration

Note – It is not necessary to include virtual volume management installation and data service installation as indicated in the *Sun Cluster 3.0 Installation Guide.*

Configuring the Sun Cluster Node Environment

You can configure the Sun Cluster environment before you install the Sun Cluster 3.0 software providing you have documented your intended cluster configuration.

You should configure the user *root* environment on each cluster node and should also configure local network name resolution.

Configuring the User root Environment

The root login environment should include the following search path and man page information:

```
PATH=$PATH:/usr/cluster/bin:/etc/vx/bin:/opt/VRTSvmsa/b
in

MANPATH=$MANPATH:/usr/cluster/man:/usr/share/man:/opt/V
RTSvxvm/man:/opt/VRTSvmsa/man
```

Note – Some of the path information depends on which virtual volume management software your cluster uses.

Configuring Network Name Resolution

The names of cluster nodes should be resolved locally so that the cluster is disabled in the event of a naming service failure. Following is a recommended configuration for the /etc/nsswitch.conf files. The example shows only partial contents.

```
passwd:     files nis
group:      files nis
...
hosts:      cluster files nis
...
netmasks:   cluster files nis
```

Note – Several changes to the /etc/nsswitch.conf file are mandatory and are performed automatically during the Sun Cluster software installation.

Installing Sun Cluster Node Patches

The Sun Cluster nodes might require patches in the following areas:

- Solaris Operating Environment patches
- Storage Array interface firmware patches
- Storage Array disk drive firmware patches
- Veritas Volume Manager 3.0.4 patches
- Solstice DiskSuite 4.2.1 patches

Some patches, such as those for Veritas Volume Manager and Solstice DiskSuite, cannot be installed until after the volume management software installation is completed.

Patch Installation Warnings

Before installing any patches, always do the following:

- Make sure all cluster nodes have the same patch levels.
- Do not install any firmware-related patches without qualified assistance.
- Always obtain the most current patch information.
- Read all patch README notes carefully.

Obtaining Patch Information

You should always consult with your Sun field representative about possible patch installations. You can obtain current patch information as follows:

- Consult `http://sunsolve.com`
- Use the SunSolve℠ PatchPro tool interactively
- Read current Sun Cluster release notes

Installing the Sun Cluster 3.0 Software

You can use two methods to install the Sun Cluster 3.0 software on the cluster nodes:

- Interactive installation using the `scinstall` installation interface
- Automatic JumpStart installation (requires a pre-existing Solaris JumpStart server)

Installing Sun Cluster Interactively

The Sun Cluster installation program, `scinstall`, is located on the Sun Cluster 3.0 CD in the `SunCluster_3_0/Tools` directory. When you start the program without any options, it prompts you for cluster configuration information that is stored for use later in the process.

Before starting the installation process you must have the following information at hand:

- The cluster name.
- The names of all nodes that will be part of this cluster.
- The node authentication mode during installation (DES).
- The cluster transport network address and netmask if you do not want to use the default address and netmask.

  You cannot change the private network address after the cluster has successfully formed.
- The cluster transport adapters.
- Global devices file-system name.
- Whether you want an automatic reboot after installation.

Note – When you finish answering the prompts, the `scinstall` command line equivalent generated from your input displays for confirmation. This information in stored in the cluster installation log file in the `/var/cluster/logs/install` directory.

The Initial scinstall Menu

Although the Sun Cluster software can be installed on all nodes in parallel, you can complete the installation on the first node for practice and then run scinstall on all other nodes in parallel. The additional nodes get some basic configuration information from the first, or sponsoring node, that was configured.

As shown in the following example, you use option 1 when establishing the first node in a new cluster. Use option 2 on all other nodes.

```
# ./scinstall
*** Main Menu ***

    Please select from one of the following (*) options:

      * 1) Establish a new cluster using this machine as
           the first node
      * 2) Add this machine as a node in an established
           cluster
        3) Configure a cluster to be JumpStarted from this
           install server
        4) Add support for a new data service to this
           cluster node
        5) Print release information for this cluster node

      * ?) Help with menu options
      * e) Exit

    Option:  1
```

The balance of the initial node cluster software installation is excerpted in the following sections with comments for each section.

Note – You can type Control-D at any time to return to the main scinstall menu and restart the installation. Your previous answers to questions become the default answers.

Supplying Cluster and Node Names

Initially, the `scinstall` program requests the name of the cluster and the name of nodes that will be added. The name of the initial node is already known. An example of the dialogue follows.

```
What is the name of the cluster you want to establish?
planets

>>> Cluster Nodes <<<

    This release of Sun Cluster supports a total of up
to 8 nodes.Please list the names of the other nodes
planned for the initial cluster configuration. List one
node name per line. When finished, type Control-D:

    Node name:  mars
    Node name (Ctrl-D to finish):  ^D

    This is the complete list of nodes:

        venus
        mars

    Is it correct (yes/no) [yes]?  yes
```

Note – Use local `/etc/hosts` name resolution to prevent installation difficulties and to increase cluster availability by eliminating the cluster's dependence on a naming service.

Selecting DES Authentication

As shown in the following example, you can select DES authentication for use only during the remainder of the cluster installation. This can prevent unauthorized modification of the CCR contents during installation. This is not very likely but might be a requirement in high security environments.

```
Do you need to use DES authentication (yes/no) [no]? no
```

Note – If you must use DES authentication for security reasons, you must completely configure DES authentication on all nodes before starting the cluster installation. DES authentication is used only during the node installation process.

Configuring the Cluster Interconnect

The following examples summarize the process of configuring the cluster interconnect. You must approve the default network address (`172.16.0.0`) and netmask (`255.255.0.0`). If the default base address is in use elsewhere at your site, you have to supply a different address and/or netmask.

If your cluster is a two-node cluster, you are asked if switches are used. The connections can be point-to-point in a two-node cluster, but you can also use switches. If the cluster is greater than two nodes, switches are assumed. The switches are assigned arbitrary names.

You must also furnish the names of the cluster transport adapters.

```
Is it okay to accept the default network address (yes/no) [yes]?  yes
Is it okay to accept the default netmask (yes/no) [yes]?  yes

Does this two-node cluster use transport junctions (yes/no) [yes]?  yes

What is the name of the first junction in the cluster [switch1]? sw1
What is the name of the second junction in the cluster [switch2]? sw2

What is the name of the first cluster transport adapter ?  hme1
Name of the junction to which “hme1” is connected [sw1]? sw1
Okay to use the default for the “hme1” connection [yes]?  yes

What is the name of the second cluster transport adapter [hme2]?  hme2
Name of the junction to which “hme2” is connected [sw2]? sw2
Use the default port for the “hme2” connection [yes]? yes
```

Caution – If you do not specify the correct interfaces when installing the first cluster node, the installation completes without errors. When you install the Sun Cluster software on the second node, it is not able to join the cluster because of the incorrect configuration data. You have to manually correct the problem on the first node before continuing.

Configuring the Global Devices File System

Normally, a /globaldevices file system already exists on the boot disk of each cluster node. The scinstall utility asks if you wish to use the default global file system. You can also use a different file system or have the scinstall utility create one for you.

The first example is for an existing /globaldevices file system.

```
The default is to use /globaldevices.
Is it okay to use this default (yes/no) [yes]?  yes
```

The second example shows how you can use a file system other than the default /globaldevices.

```
Is it okay to use this default (yes/no) [yes]?  no
Do you want to use an already existing file system (yes/no)
[yes]?  yes
What is the name of the file system  /sparefs
```

Warning – If you use a file system other than the default, it must be an empty file system containing only the lost+found directory.

The third example shows how to create a new /globaldevices file system using an available disk partition.

```
Is it okay to use this default (yes/no) [yes]?  no
Do you want to use an already existing file system (yes/no)
[yes]?  no
What is the name of the disk partition you want to use
/dev/dsk/c0t0d0s4
```

Note – It is recommended that you use the default /globaldevices file system. Standardization helps during problem isolation.

Selecting Automatic Reboot

To complete the basic installation of the first node, you must decide whether you want the system to reboot automatically.

```
Do you want scinstall to re-boot for you (yes/no) [yes]?
yes
```

The reboot question should be considered because you might need to install Sun Cluster patches before rebooting.

Note – If for any reason the reboot fails, you can use a special boot option (`ok boot -x`) to disable the cluster software startup until you can fix the problem.

Confirming the Final Configuration

Before starting the Sun Cluster installation, the `scinstall` utility displays the command line equivalent of the installation for your approval. This information is recorded in the installation log file that can be examined in the `/var/cluster/logs/install` directory.

A typical display is as follows:

```
Your responses indicate the following options to scinstall:

      scinstall -i \
           -C planets \
           -N venus \
           -T node=venus,node=mars,authtype=sys \
           -A trtype=dlpi,name=hme1 -A trtype=dlpi,name=hme2 \
           -B type=switch,name=hub1 -B type=switch,name=hub2 \
           -m endpoint=:hme1,endpoint=hub1 \
           -m endpoint=:hme2,endpoint=hub2

    Are these the options you want to use [yes]? yes

    Do you want to continue with the install (yes/no) [yes]? yes
```

Installation Operations

During the final installation process, the `scinstall` utility performs the following operations on the first cluster node:

- Installs cluster software packages
- Disables routing on the node (`touch /etc/notrouter`)
- Creates and installation log (`/var/cluster/logs/install`)
- Reboots the node
- Creates the Disk ID devices during the reboot

Note – It is normal to see some DID error messages during the reboot. You should not see any such messages during later system reboots. A typical error message is: `did_instances, no such file`.

Installing Additional Nodes

After you complete the Sun Cluster software installation on the first node for practice, you can run scinstall in parallel on all remaining cluster nodes. The additional nodes are placed in *install mode* so they do not have a quorum vote. Only the first node has a quorum vote.

As the installation on each new node completes, each node reboots and comes up in install mode without a quorum vote. If you reboot the first node at this point, all the other nodes would panic because they cannot obtain a quorum. You can, however, reboot the second or later nodes freely. They should come up and join the cluster without errors.

Cluster nodes remain in install mode until you use the `scsetup` command to reset the install mode.

You must perform postinstallation configuration to take the nodes out of install mode and also to establish quorum disk(s).

Note – If you must reboot the first node before performing postinstallation procedures, you can first shut down the entire cluster using the `scshutdown` command. You can also shutdown single nodes using the standard `init 0` command.

Postinstallation Configuration

Postinstallation can include a number of complex tasks such as installing a volume manager and or database software. There are less complex tasks that must be completed first.

This section focuses on the following postinstallation tasks:

- Taking the cluster nodes out of install mode
- Defining quorum disks

Resetting Install Mode of Operation

Before a new cluster can operate normally, the install mode attribute must be reset on all nodes. You can do this in a single step using the `scsetup` utility.

Resetting Install Mode Using the `scsetup` Utility

The `scsetup` utility is a menu-driven interface that prompts for quorum device information the first time it is run on a new cluster installation. Once the quorum device is defined, the install mode attribute is reset for all nodes.

Note – The path to the quorum disk(s) must be furnished in a DID device format. You must first identify the appropriate DID devices with the `scdidadm` command.

Resetting Install Mode Using the `scconf` Command

Use the `scconf` command as follows to disable or enable install mode:

- `scconf -c -q reset` (reset install mode)
- `scconf -c -q installmode` (enable install mode)

Note – Use the `scconf -p` command to verify install mode. The `scstat` command might not display the current install mode status.

Most of the informational text has been omitted in the following `scsetup` example for clarity.

```
# /usr/cluster/bin/scsetup
  >>> Initial Cluster Setup <<<

    This program has detected that the cluster
“installmode” attribute is set ...

    Please do not proceed if any additional nodes have yet
to join the cluster.

    Is it okay to continue (yes/no) [yes]?  yes

Which global device do you want to use (d<N>)?  d2

Is it okay to proceed with the update (yes/no) [yes]?  yes

scconf -a -q globaldev=d2

Do you want to add another quorum disk (yes/no)?  no
Is it okay to reset “installmode” (yes/no) [yes]?  yes

scconf -c -q reset
Cluster initialization is complete.
```

Note – Although it appears that the `scsetup` utility uses two simple `scconf` commands to define the quorum device and reset install mode, the process is more complex. The `scsetup` utility perform numerous verification checks for you. It is recommended that you *do not* use `scconf` manually to perform these functions.

Configuring Network Time Protocol

The Network Time Protocol (NTP) configuration file, `/etc/inet/ntp.conf`, must be modified on all cluster nodes. You must remove all private host name entries that are not being used by the cluster. Also, if you changed the private host names of the cluster nodes, update this file accordingly.

You can also make other modifications to meet your NTP requirements.

You can verify the current `ntp.conf` file configuration as follows:

```
# more /etc/inet/ntp.conf |grep clusternode
peer clusternode1-priv prefer
peer clusternode2-priv
peer clusternode3-priv
peer clusternode4-priv
peer clusternode5-priv
peer clusternode6-priv
peer clusternode7-priv
peer clusternode8-priv
```

If your cluster will ultimately have two nodes, the entries for nodes 3, 4, 5, 6, 7, and 8 should be removed from the `ntp.conf` file on all nodes.

Until the `ntp.conf` file configuration is corrected, you see boot error messages similar to the following:

```
Dec  2 17:55:28 pnode1 xntpd[338]: couldn’t resolve
‘clusternode3-priv’, giving up on it
```

Postinstallation Verification

When you have completed the Sun Cluster software installation on all nodes, verify the following information:

- DID device configuration
- General CCR configuration information

Verifying DID Devices

Each attached system sees the same DID devices but might use a different logical path to access them. You can verify the DID device configuration with the `scdidadm` command. The following `scdidadm` output demonstrates how a DID device can have a different logical path from each connected node.

```
# scdidadm -L
1        devsys1:/dev/rdsk/c0t0d0       /dev/did/rdsk/d1
2        devsys1:/dev/rdsk/c2t37d0      /dev/did/rdsk/d2
2        devsys2:/dev/rdsk/c3t37d0      /dev/did/rdsk/d2
3        devsys1:/dev/rdsk/c2t33d0      /dev/did/rdsk/d3
3        devsys2:/dev/rdsk/c3t33d0      /dev/did/rdsk/d3
4        devsys1:/dev/rdsk/c2t52d0      /dev/did/rdsk/d4
4        devsys2:/dev/rdsk/c3t52d0      /dev/did/rdsk/d4
5        devsys1:/dev/rdsk/c2t50d0      /dev/did/rdsk/d5
5        devsys2:/dev/rdsk/c3t50d0      /dev/did/rdsk/d5
6        devsys1:/dev/rdsk/c2t35d0      /dev/did/rdsk/d6
6        devsys2:/dev/rdsk/c3t35d0      /dev/did/rdsk/d6
7        devsys1:/dev/rdsk/c3t20d0      /dev/did/rdsk/d7
7        devsys2:/dev/rdsk/c2t20d0      /dev/did/rdsk/d7
8        devsys1:/dev/rdsk/c3t18d0      /dev/did/rdsk/d8
8        devsys2:/dev/rdsk/c2t18d0      /dev/did/rdsk/d8
9        devsys1:/dev/rdsk/c3t1d0       /dev/did/rdsk/d9
9        devsys2:/dev/rdsk/c2t1d0       /dev/did/rdsk/d9
10       devsys1:/dev/rdsk/c3t3d0       /dev/did/rdsk/d10
10       devsys2:/dev/rdsk/c2t3d0       /dev/did/rdsk/d10
11       devsys1:/dev/rdsk/c3t5d0       /dev/did/rdsk/d11
11       devsys2:/dev/rdsk/c2t5d0       /dev/did/rdsk/d11
12       devsys2:/dev/rdsk/c0t0d0       /dev/did/rdsk/d12
```

Note – Devices `d1` and `d12` are the local boot disks for each node.

Verifying General Cluster Status

The `scstat` utility displays the current status of various cluster components. You can use it to display the following information:

- The cluster name and node names
- Names and status of cluster members
- Status of resource groups and related resources
- Cluster interconnect status

The following `scstat-q` command option displays the cluster membership and quorum vote information.

```
# /usr/cluster/bin/scstat -q
Quorum
  Current Votes:                      3
  Votes Configured:                   3
  Votes Needed:                       2
    Node Quorum
      Node Name:                      venus
        Votes Configured:             1
        Votes Contributed:            1
        Status:                       Online

      Node Name:                      mars
        Votes Configured:             1
        Votes Contributed:            1
        Status:                       Online

    Device Quorum
    Quorum Device Name:               /dev/did/rdsk/d2s2
    Votes Configured:                 1
    Votes Contributed:                1
    Nodes Having Access:
      venus                           Enabled
      mars                            Enabled
    Owner Node:                       venus
    Status:                           Online
```

Verifying Cluster Configuration Information

Cluster configuration information is stored in the CCR on each node. You should verify that the basic CCR values are correct. The `scconf -p` command displays general cluster information along with detailed information about each node in the cluster.

The following `scconf` output is for the first node added to a new two-node cluster.

```
# scconf -p
Cluster name:                          codev
Cluster ID:                            0x3A297CD9
Cluster install mode:                  enabled
Cluster private net:                   172.16.0.0
Cluster private netmask:               255.255.0.0
Cluster new node authentication:       unix
Cluster new node list:                 pnode1 pnode2
Cluster nodes:                         pnode1

Cluster node name:                     pnode1
  Node ID:                             1
  Node enabled:                        yes
  Node private hostname:               clusternode1-priv
  Node quorum vote count:              1
  Node reservation key:                0x3A297CD900000001
  Node transport adapters:             hme1 hme2

Node transport adapter:                hme1
    Adapter enabled:                   no
    Adapter transport type:            dlpi
    Adapter property:                  device_name=hme
    Adapter property:                  device_instance=1
    Adapter property:          dlpi_heartbeat_timeout=10000
    Adapter property:                  1000
    Adapter property:                  nw_bandwidth=80
    Adapter property:                  bandwidth=10
    Adapter port names:                0

    Adapter port:                      0
      Port enabled:                    no

Node transport adapter:                hme2
    Adapter enabled:                   no
    Adapter transport type:            dlpi
    Adapter property:                  device_name=hme
    Adapter property:                device_instance=2
    Adapter property:                10000
    Adapter property:                1000
    Adapter property:                nw_bandwidth=80
    Adapter property:                bandwidth=10
    Adapter port names:              0

    Adapter port:                    0
      Port enabled:                  no

Cluster transport junctions:         switch1 switch2

Cluster transport junction:          switch1
  Junction enabled:                  no
  Junction type:                     switch
  Junction port names:               1

  Junction port:                     1
    Port enabled:                    no

Cluster transport junction:          switch2
  Junction enabled:                  no
  Junction type:                     switch
  Junction port names:               1

  Junction port:                     1
    Port enabled:                    no


Cluster transport cables

                    Endpoint      Endpoint      State
                    --------      --------      -----
  Transport cable:  pnode1:hme1@0 switch1@1     Disabled
  Transport cable:  pnode1:hme2@0 switch2@1     Disabled


Quorum devices:     <NULL>
```

Correcting Minor Configuration Errors

When you install the Sun Cluster software, some common mistakes are:

- Using the wrong cluster name
- Using incorrect cluster interconnect interface assignments

You can resolve these simple mistakes using the `scsetup` utility.

You can run the `scsetup` utility on any cluster node and is used perform the following tasks:

- Add or remove quorum disks
- Add, remove, enable, or disable cluster transport components
- Register or unregister Veritas disk groups
- Add or remove node access from a Veritas disk group
- Change the cluster private hostnames
- Prevent or permit the addition of new nodes
- Change the name of the cluster

Note – The `scsetup` utility is strongly recommended for use instead manual commands such as `scconf`. It is less prone to errors and performs complex verification in some cases before proceeding.

Exercise: Installing the Sun Cluster Server Software

In this exercise, you complete the following tasks:

- Configure environment variables
- Install the Sun Cluster server software
- Perform post-installation configuration

Preparation

Obtain the following information from your instructor:

1. Ask your instructor about the location of the Sun Cluster 3.0 software. Record the location.

   Software location: ______________________________

Note – During this exercise, when you see italicized names such as *IPaddress*, *enclosure_name*, *node1*, or *clustername* imbedded in a command string, substitute the names appropriate for your cluster.

Task – Verifying the Boot Disk

Perform the following steps on all nodes to verify that the boot disks have a 100-Mbyte `/globaldevices` partition on slice 4 and a small partition on slice 7 for use by Solstice DiskSuite replicas.

1. Type the mount command and record the logical path to the boot disk on each node (typically `c0t0d0`).

   Node 1 boot device: ________________

   Node 2 boot device: ________________

2. Use the `prtvtoc` command to verify each boot disk meets the Sun Cluster partitioning requirements.

```
# /usr/sbin/prtvtoc /dev/rdsk/c0t0d0s2
```

Task – Verifying the Environment

Perform the following steps on both nodes.

1. Verify that the `/.profile` file on each cluster node contains the following environment variables:

```
PATH=$PATH:/usr/cluster/bin:/etc/vx/bin:/opt/VRTSvmsa/b
in

MANPATH=$MANPATH:/usr/cluster/man:/usr/share/man:/opt/V
RTSvxvm/man:/opt/VRTSvmsa/man

TERM=vt220

export PATH MANPATH TERM
```

Note – If necessary, create the `.profile` login file as follows:
`cp /etc/skel/local.profile ./.profile`.

2. If you edit the file, verify the changes work by logging out and in again as user root.
3. On both nodes, create a `.rhosts` file in the root directory. Edit the file and add a single line with a plus (+) sign.
4. On both cluster nodes, edit the `/etc/default/login` file and comment out the `CONSOLE=/dev/console` line.

Note – The `.rhosts` and `/etc/default/login` file modifications shown here can be a security risk in some environments. They are used here to simplify some of the lab exercises.

Task – Updating the Name Service

1. Edit the `/etc/hosts` file on the administrative workstation and all cluster nodes, and add the IP addresses and host names of the administrative workstation and cluster nodes.
2. If you are using NIS or NIS+, add the IP addresses and host names to the name service.

Note – Your lab environment might already have all of the IP addresses and host names entered in the `/etc/hosts` file.

Task – Establishing a New Cluster

Perform the following steps to establish the first node in your cluster:

1. In the `cconsole` window, log in to Node 1 as user root.
2. Change to the location of the Sun Cluster 3.0 software furnished by your instructor.
3. Change to the `SunCluster_3.0/Tools` directory.
4. Start the `scinstall` script on Node 1 only.
5. As the installation proceeds, make the following choices:

   a. Select option 1, *Establish a new cluster*.

   b. Furnish your assigned cluster name.

   c. Furnish the name of the second node that will be added later.

   d. Verify the list of node names.

   e. Reply **`no`** to using DES authentication.

   f. Unless your instructor has stated otherwise, accept the default cluster transport base address and netmask values.

   g. Configure the cluster transport based on your cluster configuration. Accept the default names for switches if your configuration uses them.

   h. Accept the default global device file system.

   i. Reply **`yes`** to the automatic reboot question.

Note – The `scinstall -F` option is for the global devices file system. This entry should be blank if you accepted the default (`/globaldevices`). The command line output is copied into the cluster installation log file in `/var/cluster` directory.

   j. Examine the `scinstall` command options for correctness. Accept them if they seem appropriate. The options should look similar to the following:

```
scinstall -i
-C codev
-F
-T node=pnode1,node=pnode2,authtype=sys
-A trtype=dlpi,name=hme1 -A trtype=dlpi,name=hme2
```

```
-B type=switch,name=switch1 -B type=switch,name=switch2
-m endpoint=:hme1,endpoint=switch1
-m endpoint=:hme2,endpoint=switch2
```

6. Observe the following messages during the node reboot:

```
/usr/cluster/bin/scdidadm: Could not load DID instance
list.
Cannot open /etc/cluster/ccr/did_instances.

Booting as part of a cluster

NOTICE: CMM: Node pnode1 (nodeid = 1) with votecount =
1 added.
NOTICE: CMM: Node pnode1: attempting to join cluster.
NOTICE: CMM: Cluster has reached quorum.
NOTICE: CMM: Node pnode1 (nodeid = 1) is up; new
incarnation number = 975797536.
NOTICE: CMM: Cluster members: pnode1 .
NOTICE: CMM: node reconfiguration #1 completed.
NOTICE: CMM: Node pnode1: joined cluster.


Configuring DID devices

Configuring the /dev/global directory (global devices)
Dec  2 17:55:28 pnode1 xntpd[338]: couldn't resolve
'clusternode2-priv', giving up on it

Dec  2 17:57:31 pnode1 Cluster.PMF.pmfd: Error opening
procfs control file </proc/492/ctl> for tag <scsymon>:
No such file or directory
The
```

Task – Adding a Second Cluster Node

Perform the following steps to complete the creation of a two-node cluster.

1. In the `cconsole` window, log in to Node 2 as user `root`.
2. Change to the location of the Sun Cluster 3.0 software furnished by your instructor.
3. Change to the `SunCluster_3.0/Tools` directory.
4. Start the `scinstall` script on Node 2 only.

5. As the installation proceeds, make the following choices:

   a. Select option 2, *Add this machine as a node in an established cluster*.

   b. Furnish the name of a sponsoring node.

   c. Furnish the name of the cluster.

      Type **`scconf -p`** on the first node (the sponsoring node) if you have forgotten the name of the cluster.

   d. Answer the cluster interconnect questions as required.

   e. Select the default for the global device directory.

   f. Reply **`yes`** to the automatic reboot question.

   g. Examine and approve the `scinstall` command line options. It should look similar to the following:

```
scinstall -i
      -C scdev
      -N pnode1
      -A trtype=dlpi,name=hme1 -A trtype=dlpi,name=hme2
      -m endpoint=:hme1,endpoint=switch1
      -m endpoint=:hme2,endpoint=switch2
```

Note – You see interconnect-related errors on Node 1 until Node 2 completes the first portion of its reboot operation.

Task – Configuring a Quorum Device

Perform the following steps to finish initializing your new two-node cluster.

1. On either node, type the **`scstat -q`** command.

   The second node is still in install mode and has no votes. Only the first node has a vote. No quorum device has been assigned.

2. On Node 1, type the **`scdidadm -l`** command and record the DID device you intend to configure as a quorum disk.

   Quorum disk: _______ (d4, d6, etc.)

Caution – Pay careful attention. The first few DID devices might be local disks such as the boot disk and a CD-ROM (target 6). Examine the standard logical path to make sure the DID device you select is a disk in a storage array.

3. On Node 1, type the **`scsetup`** command and supply the name of the DID device you selected in the previous step. You should see output similar to the following.

```
scconf -a -q globaldev=d12
Dec  3 22:29:13 pnode1 cl_runtime: NOTICE: CMM: Cluster
members: pnode1 pnode2 .
Dec  3 22:29:13 pnode1 cl_runtime: NOTICE: CMM: node
reconfiguration #4 completed.
```

4. Do *not* add a second quorum disk.

5. Reply **`yes`** to the reset installmode question.

   You should see a “Cluster initialization is complete” message.

Now that install mode has been reset and a quorum device defined, the `scsetup` utility displays its normal menu selections. Type **`q`** to quit the `scsetup` utility.

Task – Configuring the Network Time Protocol

Perform the following steps on both nodes to complete the NTP configuration.

1. On both nodes, edit the `/etc/inet/ntp.conf` file and remove configuration entries for node instances that are not configured. In a two-node cluster, you should remove the following lines:

   ```
   peer clusternode3-priv
   peer clusternode4-priv
   peer clusternode5-priv
   peer clusternode6-priv
   peer clusternode7-priv
   peer clusternode8-priv
   ```

2. On both nodes type the **`scstat -q`** command.

   You should see three quorum votes present and a quorum device.

3. On both nodes type the **`scdidadm -L`** command.

   Each shared (dual-ported) DID device should show a logical path from each cluster node.

4. On either node, type the **`scconf -p`** command.

   The cluster status, node names, transport configuration, and quorum device information should be complete.

Testing Basic Cluster Operation

Perform the following steps to verify the basic cluster software operation.

Note – You are using commands that have not yet been presented in the course. If you have any questions, please consult with your instructor.

1. Log in to each of your cluster host systems as user root.
2. On Node 1, shut down all cluster nodes.

```
# scshutdown -y -g 15
```

Note – The `scshutdown` command completely shuts down all cluster nodes, including the Solaris Operating Environment.

3. Boot Node 1, it should come up and join the cluster.
4. Boot Node 2, it should come up and join the cluster.

Exercise Summary

Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

❑ Install the Sun Cluster host system software

❑ Correctly interpret configuration questions during the Sun Cluster software installation

❑ Perform postinstallation configuration

Think Beyond

How can you can you add a new node to an existing cluster?

What could happen if you did not configure any quorum devices?

Module 6

Basic Cluster Administration

Objectives

Upon completion of this module, you should be able to:

- Perform basic cluster startup and shutdown operations
- Boot nodes in non-cluster mode
- Place nodes in a maintenance state
- Verify cluster status from the command line

Relevance

Discussion – The following questions are relevant to your understanding of the module's content:

- What needs to be monitored in the Sun Cluster environment?
- How current does the information need to be?
- How detailed does the information need to be?
- What types of information are available?

Additional Resources

Additional resources – The following references can provide additional information on the topics described in this module:

- *Sun Cluster 3.0 Installation Guide*, part number 806-1419
- *Sun Cluster 3.0 Hardware Guide*, part number 806-1420
- *Sun Cluster 3.0 Data Services Installation and Configuration Guide*, part number 806-1421
- *Sun Cluster 3.0 Data Service Developers Guide*, part number 806-1422
- *Sun Cluster 3.0 System Administration Guide*, part number 806-1423
- *Sun Cluster 3.0 Concepts*, part number 806-1424
- *Sun Cluster 3.0 Error Message Guide*, part number 806-1426
- *Sun Cluster 3.0 Release Notes*, part number 806-1428

Cluster Status Commands

There are several cluster status commands. Some of the commands have uses other than status reporting.

Checking Status Using the scstat Command

Without any options, the scstat command displays general information for all cluster nodes. You can use options to restrict the status information to a particular type of information and/or to a particular node.

The following command displays the cluster transport status for a single node.

```
# scstat -W -h pnode2

-- Cluster Transport Paths --

                    Endpoint      Endpoint      Status
                    --------      --------      ------
  Transport path:   pnode1:hme2   pnode2:hme2   Path online
  Transport path:   pnode1:hme1   pnode2:hme1   Path online
```

Checking Status Using the sccheck Command

The sccheck command verifies that all of the basic global device structure is correct on all nodes. Run the sccheck command after installing and configuring a cluster, as well as after performing any administration procedures that might result in changes to the devices, volume manager, or Sun Cluster configuration.

You can run the command without options or direct it to a single node. You can run it from any active cluster member. There is no output from the command unless errors are encountered. Typical sccheck command variations follow.

```
# sccheck
# sccheck -h pnode2
```

Checking Status Using the `scinstall` Utility

During the Sun Cluster software installation, the `scinstall` utility is copied into the `/usr/cluster/bin` directory. You can run the `scinstall` utility with options that display the Sun Cluster revision and/or the names and revision of installed packages. The displayed information is for the local node only. A typical `scinstall` status output follows.

```
# scinstall -pv
SunCluster 3.0
SUNWscr:        3.0.0,REV=2000.10.01.01.00
SUNWscdev:      3.0.0,REV=2000.10.01.01.00
SUNWscu:        3.0.0,REV=2000.10.01.01.00
SUNWscman:      3.0.0,REV=2000.10.01.01.00
SUNWscsal:      3.0.0,REV=2000.10.01.01.00
SUNWscsam:      3.0.0,REV=2000.10.01.01.00
SUNWscvm:       3.0.0,REV=2000.10.01.01.00
SUNWmdm:        4.2.1,REV=2000.08.08.10.01
#
```

Caution – Use the `scinstall` utility carefully. It is possible to create serious cluster configuration errors using the `scinstall` utility.

Cluster Control

Basic cluster control includes starting and stopping clustered operation on one or more nodes and booting nodes in non-cluster mode.

Starting and Stopping Cluster Nodes

The Sun Cluster software starts automatically during a system boot operation. Use the `init` command to shut down a single node. You use the `scshutdown` command to shut down all nodes in the cluster.

Before shutting down a node, you should switch resource groups to the next preferred node and then run `init 0` on the node.

Note – After an initial Sun Cluster installation, there are no configured resource groups to worry about.

Shutting Down a Cluster

You can shut down the entire cluster with the `scshutdown` command from any active cluster node. A typical cluster shutdown example follows.

```
# scshutdown -y -g 30
Broadcast Message from root (???) on pnode1 Wed Dec 20
17:43:17...
 The cluster scdev will be shutdown in  30 seconds
Dec 20 17:43:38 pnode1 cl_runtime: WARNING: CMM: Monitoring
disabled.
INIT: New run level: 0
The system is coming down.  Please wait.
System services are now being stopped.
/etc/rc0.d/K05initrgm: Calling scswitch -S (evacuate)
Print services stopped.
Dec 20 17:43:54 pnode1 syslogd: going down on signal 15
The system is down.
syncing file systems... done
Program terminated
ok
```

Note – Similar messages appear on all active cluster nodes.

Booting Nodes in Non-Cluster Mode

Occasionally, you might want to boot a node without it joining in clustered operation. A common reason would be installing software patches. Some patches cannot be installed on a active cluster member. An example follows.

```
ok boot -x
Rebooting with command: boot -x
Boot device:/pci@1f,4000/scsi@3/disk@0,0  File and args: -x
SunOS Release 5.8 Version Generic_108528-03 64-bit
Copyright 1983-2000 Sun Microsystems, Inc.All rights
reserved.
configuring IPv4 interfaces: hme0.
Hostname: pnode2
Not booting as part of a cluster
The system is coming up.  Please wait.
Starting IPv4 routing daemon.
starting rpc services: rpcbind done.
Setting netmask of hme0 to 255.255.255.0
Setting default IPv4 interface for multicast: add net
224.0/4: gateway pnode2
syslog service starting.
Print services started.
Dec 20 17:51:02 pnode2 xntpd[195]: couldn’t resolve
‘clusternode1-priv’, giving up on it
Dec 20 17:51:02 pnode2 xntpd[195]: couldn’t resolve
‘clusternode2-priv’, giving up on it
volume management starting.
The system is ready.

pnode2 console login:
```

Other active cluster nodes display transport-related errors because they try to establish a transport path to the node running in non-cluster mode. Typical errors on active nodes follow.

```
# Dec 4 21:09:42 pnode1 cl_runtime: WARNING: Path
pnode1:hme1 - pnode2:hme1 initiation encountered errors,
errno = 62. Remote node may be down or unreachable through
this path.
Dec 4 21:09:42 pnode1 cl_runtime: WARNING: Path pnode1:hme2
- pnode2:hme2 initiation encountered errors, errno = 62.
Remote node may be down or unreachable through this path.
```

Placing Nodes in Maintenance Mode

If you anticipate a node will be down for an extended period, you can place the node in a maintenance state from an active cluster node. The maintenance state disables the node's quorum vote. You cannot place an active cluster member in a maintenance state. A typical command follows.

```
# scconf -c -q node=pnode2,maintstate
```

The `scstat` command shows that the possible vote for `pnode2` is now set to 0.

You can reset the maintenance state for a node either by rebooting the node or with the `scconf -c -q reset` command string.

Monitoring With the Sun Management Center

The Sun Management Center (SMC) software is a central monitoring utility that gathers and displays status information about configured client systems. The SMC software has three functional sections, which are:

- Server software
- Console software
- Agent software

Figure 6-1 illustrates the SMC software relationship in a Sun Cluster environment.

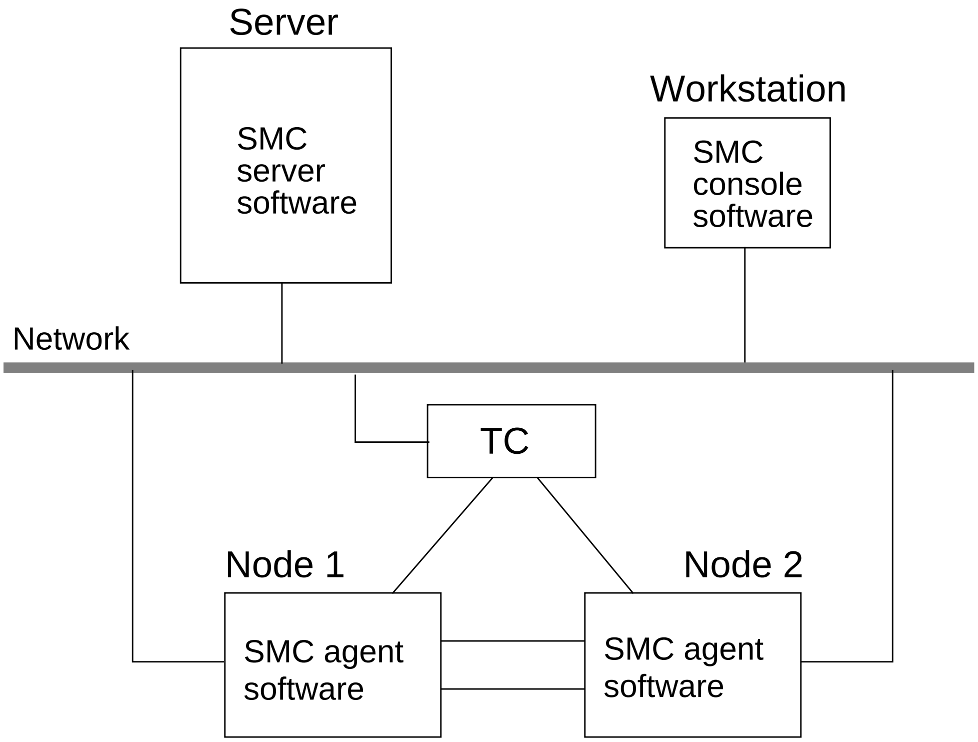


Figure 6-1 Sun Management Center Software Distribution

The cluster agent packages, `SUNWscsam` and `SUNWscsam`, are automatically installed on the nodes during the initial Sun Cluster installation.

You can install the SMC server and console software on the same system as long as it has at least 128-Mbytes of memory and 50-Mbytes of available disk space in the `/opt` directory. You can choose a different directory other than `/opt`. The SMC software is licensed.

SMC Server Software

The server software usually resides on a centralized system. The server software is the majority of the SMC software.

Server Software Function

The function of the SMC server software is to accept user requests from the console software and pass the requests to the appropriate agent software. When the agent response is relayed back to the console interface, the server software interprets the response and forwards an appropriate graphical display.

SMC Console Software

The SMC console software is the user interface. You can configure the console software on several different workstations for different users.

SMC Agent Software

The agent software is installed on each Sun Cluster node. Agent software is unique to the installation. Sun Cluster agents are designed to gather Simple Network Management Protocol (SNMP) status information on command and transfer it to the SMC server.

The Sun Cluster-supplied modules for SMC enable you to graphically display cluster resources, resource types, and resource groups. They also enable you to monitor configuration changes and check the status of cluster components.

Note – The Sun Cluster-supplied SMC packages include server, console, and agent software. They are installed in all three locations.

Exercise: Performing Basic Cluster Administration

In this exercise, you complete the following tasks:

- Perform basic cluster startup and shutdown operations
- Boot nodes in non-cluster mode
- Place nodes in a maintenance state
- Verify cluster status from the command line

Preparation

All nodes should be joined in the cluster and the `cconsole` tool should be running on the administration workstation.

Note – During this exercise, when you see italicized names such as *`IPaddress`*, *`enclosure_name`*, *`node1`*, or *`clustername`* imbedded in a command string, substitute the names appropriate for your cluster.

Task – Verifying Basic Cluster Status

Perform the following steps to verify the basic status of your cluster.

1. Use the `scstat` command to verify the current cluster membership.

   ```
   # scstat -q
   ```

2. Record the quorum configuration from the previous step.

   Quorum votes possible: _____

   Quorum votes needed: _____

   Quorum votes present: _____

3. Verify that the global device structures on all nodes are correct.

   ```
   # sccheck
   ```

4. Verify the revision of the currently installed Sun Cluster software on each cluster node.

   ```
   # scinstall -pv
   ```

Task – Starting and Stopping Cluster Nodes

Perform the following steps to start and stop configured cluster nodes.

1. Verify that both nodes are active cluster members.
2. Shut down Node 2.

   ```
   # init 0
   ```

Note – You might see RPC-related errors because the NFS server daemons, `nfsd` and `mountd`, are not running. This is normal if you are not sharing file systems in `/etc/dfs/dfstab`.

3. Join Node 2 into the cluster again by performing a boot operation.
4. When both nodes are members of the cluster, run `scshutdown` on one node to shut down the entire cluster.

   ```
   # scshutdown -y -g 60 Log off now!!
   ```

5. Join Node 1 into the cluster by performing a boot operation.

6. When Node 1 is in clustered operation again, verify the cluster quorum configuration again.

```
# scstat -q | grep “Quorum votes”
  Quorum votes possible:      3
  Quorum votes needed:        2
  Quorum votes present:       2
```

7. Leave Node 2 down for now.

Task – Placing a Node in Maintenance State

Place a node in the maintenance state by performing the following steps.

1. On Node 1, use the `scconf` command to place Node 2 into a maintenance state.

```
# scconf -c -q node=node2,maintstate
```

Note – Substitute the name of your node.

2. Verify the cluster quorum configuration again.

```
# scstat -q | grep “Quorum votes”
```

Note – The number of *possible* quorum votes should be reduced by one.

3. Boot Node 2 again. This should reset its maintenance state. You should see the following message on both nodes:

```
NOTICE: CMM: Votecount changed from 0 to 1 for node
pnode2
```

4. Verify the cluster quorum configuration again. The number of possible quorum votes should be back to normal.

Task – Booting Nodes in Non-cluster Mode

Perform the following steps to boot a cluster node so that it does not participate in clustered operation.

1. Shut down Node 2.
2. Boot Node 2 in non-cluster mode.

```
ok boot -x
```

Note – You should see a message similar to: `Not booting as part of a cluster`. You can also add the single-user mode option: `boot -sx`.

3. Verify the quorum status again.
4. Return Node 2 to clustered operation.

```
# init 6
```

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

❑ Perform basic cluster startup and shutdown operations

❑ Boot nodes in non-cluster mode

❑ Place nodes in a maintenance state

❑ Verify cluster status from the command line

Think Beyond

What strategies can you use to simplify administering a cluster with 8 nodes and 200 storage arrays?

What strategies can you use to simplify administering a large installation of 20 clusters?

Module 7

Volume Management Using Veritas Volume Manager

Objectives

Upon completion of this module, you should be able to:

- Explain the disk space management technique used by Veritas Volume Manager
- Describe the Veritas Volume Manager initialization process
- Describe how Veritas Volume Manager groups disk drives
- Install and initialize Veritas Volume Manager
- Perform Veritas Volume Manager postinstallation configuration
- Use the basic Veritas Volume Manager status commands
- Register Veritas Volume Manager disk groups
- Create global file systems
- Perform basic device group management

Relevance

Discussion – The following questions are relevant to your learning the material presented in this module:

- Which Veritas Volume Manager features are the most important to clustered systems?
- Are there any Veritas Volume Manager feature restrictions when it is used in the Sun Cluster environment?

Additional Resources

Additional resources – The following references can provide additional information on the topics described in this module:

- *Sun Cluster 3.0 Installation Guide*, part number 806-1419
- *Sun Cluster 3.0 Hardware Guide*, part number 806-1420
- *Sun Cluster 3.0 Data Services Installation and Configuration Guide*, part number 806-1421
- *Sun Cluster 3.0 Data Service Developers Guide*, part number 806-1422
- *Sun Cluster 3.0 System Administration Guide*, part number 806-1423
- *Sun Cluster 3.0 Concepts*, part number 806-1424
- *Sun Cluster 3.0 Error Message Guide*, part number 806-1426
- *Sun Cluster 3.0 Release Notes*, part number 806-1428

Disk Space Management

Veritas Volume Manager manages data in a non-partitioned environment. Veritas Volume Manager manages disk space by maintaining tables that associate a list of contiguous disk blocks with a data volume structure. A single disk drive can potentially be divided into hundreds of independent data regions.

As shown in Figure 7-1, Veritas Volume Manager maintains detailed configuration records that equate specific blocks on one or more disk drives with virtual volume structures.

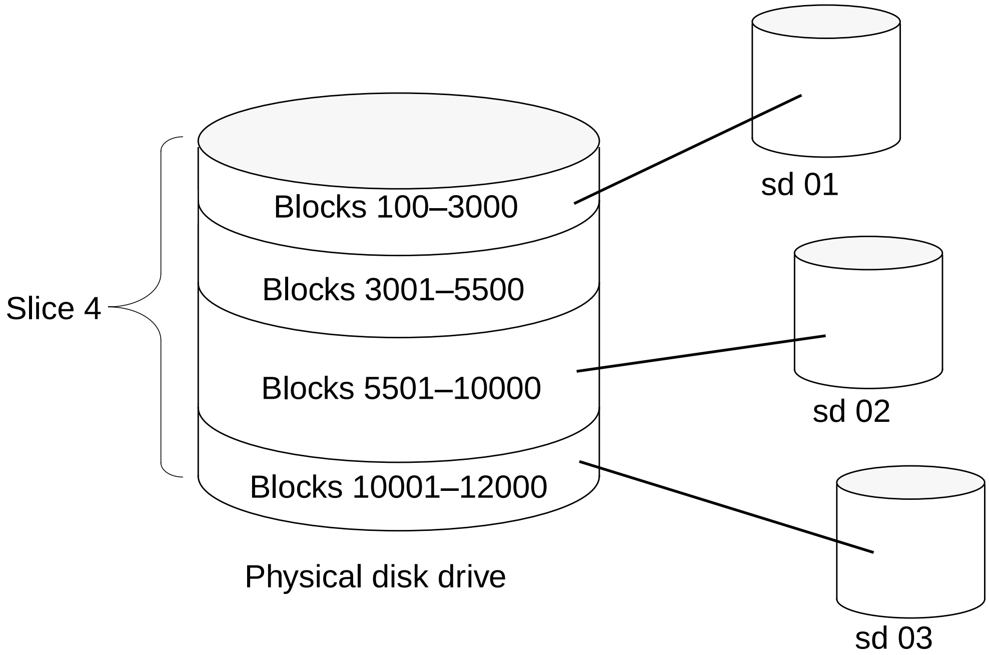


Figure 7-1 Veritas Volume Manager Space Management

Veritas Volume Manager divides a disk into a single slice and then allocates portions of the slice to data structures named *subdisks*. Subdisks are the basic storage space used to create Veritas Volume Manager volumes.

Veritas Volume Manager Disk Initialization

When a physical disk drive is initialized by Veritas Volume Manager, it is divided into two sections called the *private* region and the *public* region.

The private and public regions are used for different purposes.

- The private region is used for configuration information.
- The public region is used for data storage.

As shown in Figure 7-2, the private region is small. It is usually configured as slice 3 on the disk and is, at most, a few cylinders in size.

The public region is the rest of the disk drive. It is usually configured as slice 4 on the disk.

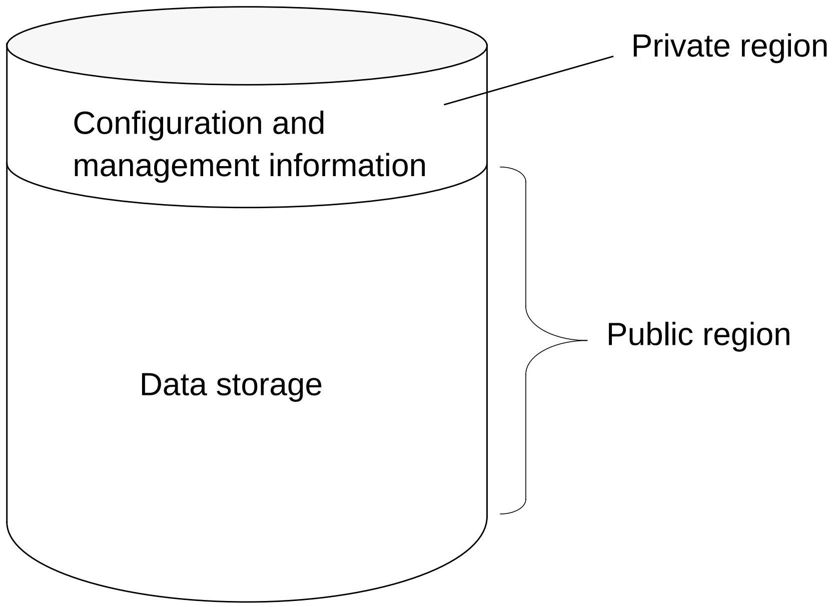


Figure 7-2 Veritas Volume Manager Disk Initialization

Note – You must specially initialize all disk drives that Veritas Volume Manager uses unless they have existing data you want to preserve. To preserve existing data, you *encapsulate* the disk. The Veritas Volume Manager encapsulation process is described in the next section of this module.

Private Region Contents

The size of the private region, by default, is 1024 sectors (512K-bytes). You can enlarge it if a large number of Veritas Volume Manager objects are anticipated in a disk group. If you anticipate having more than 2000 Veritas Volume Manager objects in a disk group, increase the size of the private region on the disks that are added to the disk group. The private region contents are:

- Disk header

  Two copies of the file that defines and maintains the host name of the current disk group owner, a unique DID, disk geometry information, and disk group association information.

- Table of contents

  The disk header points to this linked list of blocks.

- Configuration database

  This database contains persistent configuration information for all of the disks in a disk group. It is usually referred to as the `configdb` record.

- Disk group log

  This log is composed of kernel-written records of certain types of actions, such as transaction commits, plex detaches resulting from I/O failures, dirty region log (DRL) failures, first write to a volume, and volume close. The Veritas Volume Manager software uses this information after a crash or clean reboot to recover the state of the disk group just before the crash or reboot.

Note – A Veritas Volume Manager object is a volume, a plex, a subdisk, or a disk group. Volumes are created from plexes. A plex is built from one or more subdisks. A single mirrored volume in a disk group consists of six objects: two subdisks, two plexes, the volume, and the disk group.

Private and Public Region Format

The private and public region format of an initialized Veritas Volume Manager disk can be verified with the `prtvtoc` command. As shown in the following example, slice 2 is defined as the entire disk. Slice 3 has been assigned tag 15 and is 2016 sectors in size. Slice 4 has been assigned tag 14 and is the rest of the disk.

In this example, the private region is the first two cylinders on the disk. The disk is a 1.05-Gbyte disk and a single cylinder has 1008 sectors or blocks, which does not meet the 1024 sector minimum size for the private region. This is calculated by using the `nhead=14` and `nsect=72` values for the disk found in the `/etc/format.dat` file.

```
# prtvtoc /dev/rdsk/c2t4d0s2

                             First    Sector    Last
 Partition  Tag  Flags    Sector     Count    Sector
      2       5    01          0   2052288   2052287
      3      15    01          0      2016      2015
      4      14    01       2016   2050272   2052287
```

Initialized Disk Types

By default, Veritas Volume Manager initializes disk drives with the type Sliced. There are other possible variations. The three types of initialized disks are:

- Simple – Private and public regions are on the same partition.
- Sliced – Private and public regions are on different partitions (default).
- `nopriv` – The disk does not have a private region.

Note – You should not use the `nopriv` format. It is normally used only for RAM disk storage on non-Sun systems.

Veritas Volume Manager Disk Groups

Veritas Volume Manager uses the term *disk group* to define a related collection of disk drives. The disk groups are given unique names and current ownership is assigned to a single cluster node.

The term "dg" is commonly used when referring to a disk group.

As shown in Figure 7-3, Veritas Volume Manager disk groups are owned by an individual node and the `hostname` of that node is written onto private regions on the physical disks.

Even though another node is physically connected to the same array, it cannot access data in the array that is part of a disk group it does not own. During a node failure, the disk group ownership can be transferred to another node that is physically connected to the array. To take ownership of a disk group, a node uses a Veritas Volume Manager command to *import* the disk group. This is the backup node scheme used by all of the supported high-availability data services.

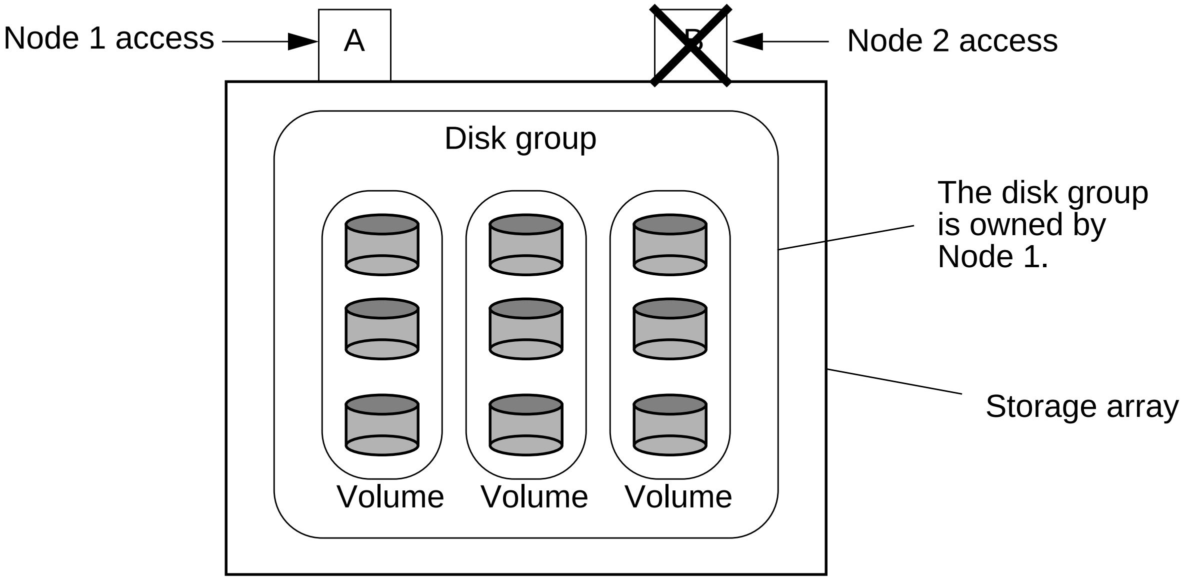


Figure 7-3 Veritas Volume Manager Disk Group Ownership

Veritas Volume Manager Status Commands

Although the graphical user interface (GUI) for Veritas Volume Manager furnishes useful visual status information, the most reliable and the quickest method of checking status is from the command line. Command line status tools are easy to use in script files, `cron` jobs, and remote logins.

Checking Volume Status

The Veritas Volume Manager `vxprint` command is the easiest way to check the status of all volume structures. The following `vxprint` sample output shows the status of two plexes in a volume as bad. One of the plexes is a log.

```
# vxprint

Disk group: sdg0

TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS  STATE
dg sdg0         sdg0         -        -        -       -

dm disk0        c4t0d0s2     -        8368512  -       -
dm disk7        c5t0d0s2     -        8368512  -       -

v  vol0         fsgen        ENABLED  524288   -       ACTIVE
pl vol0-01      vol0         DISABLED 525141   -       IOFAIL
sd disk0-01     vol0-01      ENABLED  525141   0       -
pl vol0-02      vol0         ENABLED  525141   -       ACTIVE
sd disk7-01     vol0-02      ENABLED  525141   0       -
pl vol0-03      vol0         DISABLED LOGONLY  -       IOFAIL
sd disk0-02     vol0-03      ENABLED  5        LOG     -
```

Note – You can use the `vxprint -ht vol0` command to obtain a detailed analysis of the volume. This gives you all the information you need, including the physical path to the bad disk.

You can also use the `vxprint` command to create a backup configuration file that is suitable for re-creating the entire volume structure. This is useful as a worst-case disaster recovery tool.

Checking Disk Status

When disk drives fail, the Veritas Volume Manager software can lose contact with a disk and no longer displays the physical path with the `vxprint -ht` command. At those times, you must find the media name of the failed disk from the `vxprint` command and then use the `vxdisk list` command to associate the media name with the physical device.

```
#vxdisk list


DEVICE    TYPE    DISK    GROUP    STATUS
c0t0d0s2  sliced  -       -        error
c0t1d0s2  sliced  disk02  rootdg   online
-         -       disk01  rootdg   failed was:c0t0d0s2
```

When a disk fails and becomes detached, the Veritas Volume Manager software cannot currently find the disk but still knows the physical path. This is the origin of the `failed was` status, which means the disk has failed and the physical path was the value displayed.

Saving Configuration Information

You can also use the `vxprint` and `vxdisk` commands to save detailed configuration information that is useful in disaster recovery situations. The output of the following commands should be copied into a file and stored on tape. You should also keep a printed copy of the files.

```
# vxprint -ht > filename
# vxdisk list > filename
```

Optimizing Recovery Times

In the Sun Cluster environment, data volumes are frequently mirrored to achieve a higher level of availability. If one of the cluster hosts system fails while accessing a mirrored volume, the recovery process might involve several steps, including:

- Synchronizing mirrors
- Checking file systems

Mirror synchronization can take a long time and must be completed before you can check file systems. If your cluster uses many large volumes, the complete volume recovery process can take hours.

You can expedite mirror synchronization by using the Veritas Volume Manager dirty region logging feature.

You can expedite file system recovery by using the Solaris 8 Operating Environment UNIX file system (UFS) logging feature.

Note – You can also expedite file system recovery by using the Veritas VxFS file system software. VxFS is a licensed product.

Dirty Region Logging

A DRL is a Veritas Volume Manager log file that tracks data changes made to mirrored volumes. The DRL speeds recovery time when a failed mirror needs to be synchronized with a surviving mirror.

- Only those regions that have been modified need to be synchronized between mirrors.
- Improper placement of DRLs can negatively affect performance.

A volume is divided into regions and a bitmap (where each bit corresponds to a volume region) is maintained in the DRL. When a write to a particular region occurs, the respective bit is set to on. When the system is restarted after a crash, this region bitmap is used to limit the amount of data copying that is required to recover plex consistency for the volume. The region changes are logged to special log subdisks linked with each of the plexes associated with the volume.

The Solaris Operating Environment UFS Logging

UFS logging is a standard feature of the Solaris 8 Operating Environment. All Sun Cluster global file systems are mounted using the `logging` mount option.

If logging is specified, then logging is enabled for the duration of the mounted file system. Logging is the process of storing transactions (changes that make up a complete UFS operation) in a log before the transactions are applied to the file system. After a transaction is stored, the transaction can be applied to the file system. This prevents file systems from becoming inconsistent, therefore eliminating the need to run `fsck`. And, because `fsck` can be bypassed, logging reduces the time required to reboot a system if it crashes or after an unclean halt. The default behavior is no logging.

The log is allocated from free blocks on the file system, and is sized approximately 1 Mbyte per 1 Gbyte of file system, up to a maximum of 64 Mbytes. Logging can be enabled on any UFS, including `root` (`/`). The log created by UFS logging is continually flushed as it fills up. The log is totally flushed when the file system is unmounted or as a result of the `lockfs -f` command.

The Veritas VxFS File System

The VxFS file system provides recovery only seconds after a system failure by using a tracking feature called intent logging. Intent logging is a logging scheme that records pending changes to the file system structure. During system recovery from a failure, the intent log for each file system is scanned and operations that were pending are completed. The file system can then be mounted without a full structural check of the entire system. When the disk has a hardware failure, the intent log might not be enough to recover. In such cases, a full `fsck` check must be performed, but often, when failure is due to software rather than hardware, a system can be recovered in seconds.

Note – The Veritas VxFS feature is a separately licensed Veritas product and is not directly related to the Veritas Volume Manager product.

Veritas Volume Manager Installation Overview

The basic Veritas Volume Manager installation process is relatively simple and consists of the following:

- Ensuring Dynamic Multipathing is not enabled
- Installing Veritas Volume Manager packages
- Matching `vxio` driver major number after installing Veritas Volume Manager

Disabling Veritas Volume Manager Dynamic Multipathing

The dynamic multipath (DMP) driver is a Veritas Volume Manager product feature. It is used only with fiber-optic interface storage arrays. As shown in Figure 7-4, the DMP driver can access the same storage array through more than one path. The DMP driver automatically configures multiple paths to the storage array if they exist. Depending on the storage array model, the paths are used for load-balancing in a primary/backup mode of operation.

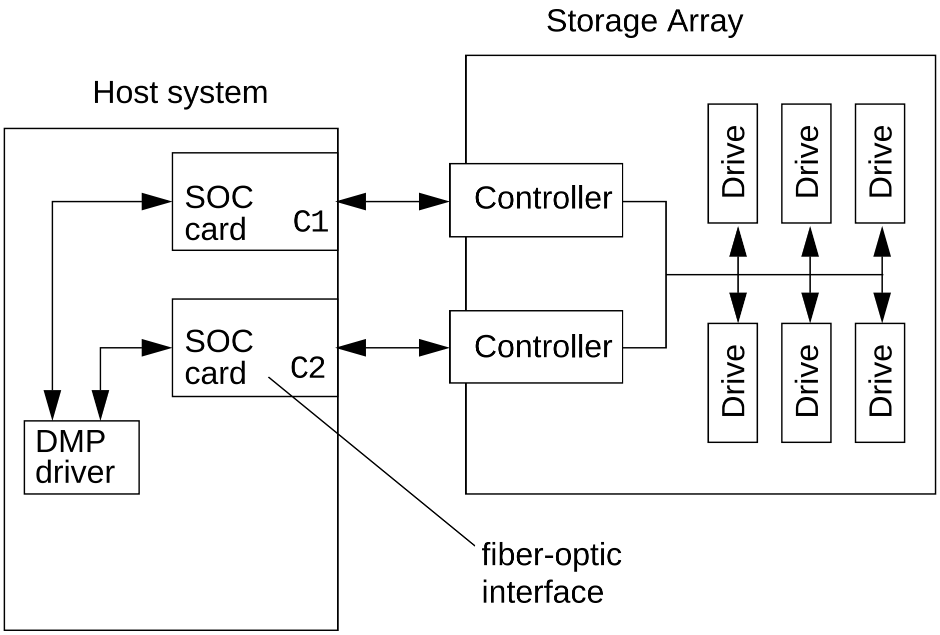


Figure 7-4 Dynamic Multipath Driver

The DMP feature is not compatible with the cluster operation so you must permanently disable it. During Veritas Volume Manager software installation, the DMP feature is automatically configured unless you take steps to prevent this from happening.

To prevent Veritas Volume Manager from enabling the DMP feature, perform the following procedure *before* installing the Veritas Volume Manager software:

1. Symbolically link `/dev/vx/dmp` to `/dev/dsk` and `/dev/vx/rdmp` to `/dev/rdsk`.

```
# mkdir /dev/vx
# ln -s /dev/dsk /dev/vx/dmp
# ln -s /dev/rdsk /dev/vx/rdmp
```

Installing the Veritas Volume Manager Software

The Veritas Volume Manager software is installed on each node using the `pkgadd` utility. Not all of the packages are necessary. A summary of the packages follows.

`VRTSVvmdev`	Header and library files
`VRTSvmdoc`	Veritas PDF/PostScript™ format documents
`VRTSvmman`	Veritas manual pages
`VRTSvmsa`	Volume Manager Storage Administrator graphical interface
`VRTSvxvm`	Veritas binaries and related files

Run the `pkgadd` command on all nodes to install the Veritas Volume Manager software.

```
# pkgadd -d . VRTSvmdev VRTSvmman VRTSvxvm
```

Note – The Veritas document package `VRTSvmdoc` and the graphic management tool in `VRTSvmsa` are not required. The manual pages package, `VRTSvmman`, is not required, but you should always install it.

Matching the vxio Driver Major Numbers

During software installation, device drivers are assigned a major number in the /etc/name_to_major file. Unless these numbers are the same on HA for NFS primary and backup host systems, the HA for NFS users receive “Stale file handle” error messages after a HA for NFS logical host migrates to a backup system. This effectively terminates the user session and destroys the high-availability feature.

It makes no difference what the major numbers are as long as they agree on both host systems attached to a disk group/storage array and are unique. Check all nodes associated with a HA for NFS logical host as follows:

```
# grep vxio /etc/name_to_major
vxio 45
```

Make sure that the number is unique in all of the files. Change one so that they all match or, if that is not possible, assign a completely new number in all of the files.

If your boot disk is not encapsulated, you can stop all activity on the nodes and edit the /etc/name_to_major files so they all agree.

Note – If your boot disk has been encapsulated, the process is more complex. Consult the *Sun Cluster Software Installation Guide* for detailed instructions.

Caution – You must stop the Sun Cluster software before making changes to the vxio driver major number.

Initializing the rootdg Disk Group

After the basic Veritas Volume Manager installation is complete, you must establish a rootdg disk group on each cluster node. The rootdg disk group is private to each node.

The Veritas Volume Manager software cannot start until a private disk group named rootdg is created on a node.

There are two ways to satisfy this requirement:

- Initialize any storage array disk and add it to the rootdg disk group
- Encapsulate the system boot disk

Creating a Simple rootdg Disk Group

If you do not want to encapsulate the boot disk on each of your cluster nodes, you can create a rootdg disk group that contains a single disk drive in a storage array. An example of such a configuration is shown in Figure 7-5.

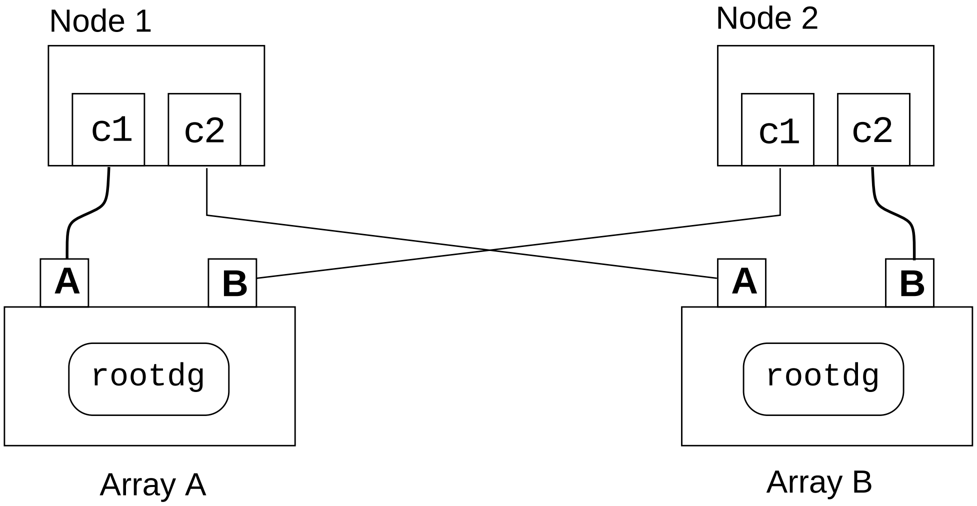


Figure 7-5 Storage Array rootdg Disk Group

The only difficulty with this configuration is that Veritas Volume Manager scans all attached disks during startup to find configuration information. Each node's own rootdg disk displays error messages about finding another rootdg disk with the wrong ownership information. Ignore the error.

Encapsulating the System Boot Disk

If you want to preserve existing data on a disk, you can choose to *encapsulate* the disk instead of initializing it. The Veritas Volume Manager encapsulation process preserves existing file systems such as those on your system boot disk.

When you install the Veritas Volume Manager software on a system, you can place your system boot disk under Veritas Volume Manager control using the `vxinstall` program.

Preferred Boot Disk Configuration

Although there are many possible boot disk variations, the preferred boot disk configuration is shown in Figure 7-6.

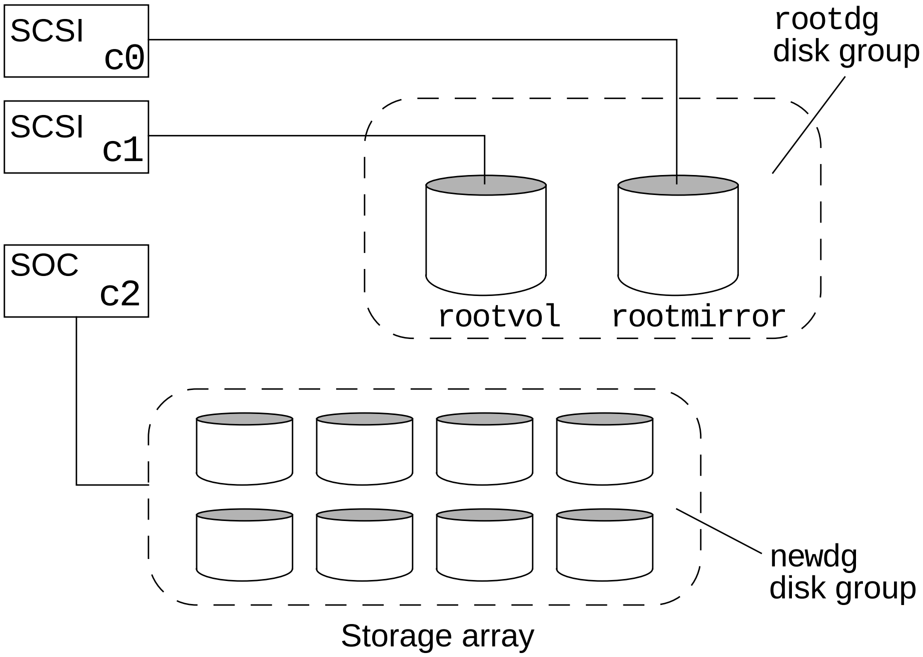


Figure 7-6 Preferred Boot Disk Configuration

The preferred configuration has the following features:

- The boot disk and mirror are on separate interfaces
- The boot disk and mirror are not in a storage array
- Only the boot disk and mirror are in the `rootdg` disk group

Prerequisites for Boot Disk Encapsulation

For the boot disk encapsulation process to succeed, the following prerequisites must be met:

- The disk must have at least two unused slices
- The boot disk should not have any slices in use other than the following:
  - `root`
  - `swap`
  - `var`
  - `usr`

An additional prerequisite that is desirable but not mandatory:

- There should be at least 1024 sectors of unused space on the disk. Practically, this is at least two full cylinders at either the beginning or end of the disk.

Note – This is needed for the private region. If necessary, Veritas Volume Manager takes the space from the end of the swap partition.

Primary and Mirror Configuration Differences

When you encapsulate your system boot disk, the location of all data remains unchanged even though the partition map is modified.

When you mirror the encapsulated boot disk, the location of the data on the mirror is different from the original boot disk.

During encapsulation, a copy of the system boot disk partition map is made so that the disk can later be returned to a state that allows booting directly from a slice again.

The mirror of the boot disk cannot be easily returned to a sliced configuration. The original boot disk can be returned to its unencapsulated format with the assistance of Veritas programs, such as the `vxunroot` utility.

The /etc/vfstab File

The original boot device mount information is commented out and retained before the new boot disk path names are configured. The following /etc/vfstab file entries are typical for an encapsulated boot disk with a single partition root file system and a swap partition.

```
#device            device          mount   FS     fsck   mount   mount
#to mount          to fsck         point   type   pass   at boot options
/dev/vx/dsk/swapvol-               -       swap   -      no
/dev/vx/dsk/rootvol/dev/vx/rdsk/rootvol /  ufs    1      no      -
#
#NOTE: volume rootvol (/) encapsulated partition c0t0d0s0
#NOTE: volume swapvol (swap) encapsulated partition c0t0d0s1
```

Sun Cluster Boot Disk Encapsulation Issues

Encapsulating the boot disk on the cluster nodes is a complex procedure because of the Sun Cluster global file systems. By design, Veritas file systems are globally available in the Sun Cluster 3.0 environment.

The `vxio` major number is the same on all nodes and the minor number for a file system is also seen to be the same by all nodes.

The rootdg file systems are private to each node and access must be restricted to the local node.

In the Sun Cluster 3.0 environment, the `rootdg` file systems are created as global and can have the same major and minor numbers.

Effectively you have multiple `rootdg` file systems that have the same name, the same major number, and the same minor number. They are also globally available. This is a severe problem and you must correct it before you start the cluster operation.

Change the `rootdg` file system names and minor numbers on each node. Special steps must also be taken to ensure the `/global` file systems will still be available on each node.

Sun Cluster Encapsulation Process

The general process for encapsulating the boot disks in a Sun Cluster 3.0 environment is as follows.

1. Install the Veritas Volume Manager software on Node 1.
2. Use `vxinstall` to encapsulate the boot disk on Node 1.
   a. Choose a unique name for the boot disk.
   b. Do not accept automatic reboot.
   c. Do not add any other disks to the `rootdg` disk group.
3. Change the `/global` mount entry in `vfstab` to use the original logical device path instead of the DID device path.
4. Repeat previous steps on all cluster nodes.
5. Shutdown the cluster nodes with `scshutdown.`
6. Boot all nodes in non-cluster mode.

7. Bypass any /global fsck errors on nodes by pressing Control-D.

   One node will successfully mount its /global file system.

   Veritas Volume Manager finishes the basic boot disk encapsulation process during the reboot on each node.

8. Unmount the one successful /global file system.

   ```
   # umount /global/.devices/node@nodeid
   ```

9. Re-minor the rootdg disk group on each cluster node.

   ```
   # vxdg reminor rootdg 100
   ```

   Use a different minor number on each node, such as 100 and 200.

10. Verify the root disk volume minor numbers are unique on each node.

    ```
    # ls -l /dev/vx/dsk/rootdg
    total 0
    brw------- 1 root root 55,100 Apr 4 10:48 rootdiska3vol
    brw------- 1 root root 55,101 Apr 4 10:48 rootdiska7vol
    brw------- 1 root root 55,  0 Mar 30 16:37 rootvol
    brw------- 1 root root 55,  7 Mar 30 16:37 swapvol
    ```

    The swap volume is automatically renumbered after a reboot.

11. If there are not separate /var or /usr file systems, then shut down the cluster, and reboot each node in cluster mode.

Note – If the encapsulated boot disks had a separate /var or /usr partition before encapsulation, you must perform additional work before rebooting. Consult Appendix B of the *Sun Cluster 3.0 Installation Guide* for additional information.

12. After you reboot the nodes, you can mirror the boot disk.

Note – Mirroring the boot disk also requires additional configuration that can be found in Appendix B of the *Sun Cluster 3.0 Installation Guide*.

Registering Veritas Volume Manager Disk Groups

After you create a new Veritas Volume Manager disk group, you must register the disk group using either the scsetup utility or the scconf command. The scsetup utility is recommended.

When a Veritas Volume Manager disk group is registered in the Sun Cluster environment, it is referred to as a *disk device group*.

Until a Veritas Volume Manager disk group is registered the cluster does detect it, and you cannot build file systems on volumes in the disk group. The newfs utility cannot find the volume path (/dev/vx/rdsk/nfsdg/vol-01). The Veritas Volume Manager utilities, such as vxprint and vxdisk, show the disk group is present and the volumes are enabled and active.

The scstat -D command does not recognize disk groups until they are registered and become *disk device groups*.

When you register a disk group, you must associate it with a list of attached nodes. The first node in the list is the primary node. When the cluster is first booted, each node imports disk groups for which it is the assigned primary.

The secondary node takes over the disk group in the event the primary node fails.

It is also possible to establish a failback policy. If failback is enabled, the disk group always migrates back to the primary node as soon as the node becomes available.

A typical scconf command to register a disk group is as follows:

```
# scconf -a -D type=vxvm,name=webdg, \
nodelist=pnode2:pnode1, \
preferenced=true,failback=enabled
```

Caution – Do not use Veritas commands to deport and import disk groups in the Sun Cluster 3.0 environment. Use the scsetup utility to register and unregister Veritas Volume Manager disk groups, to take them online and offline, and to synchronize volume changes within a disk group.

Device Group Policies

When you first register a Veritas Volume Manager disk group with the Sun Cluster framework, you can associate a list of nodes with the disk group. You can change the node list and associated policies with the **scconf -c** command. An example follows.

```
# scconf -c -D name=nfsdg,nodelist=pnode1:pnode2
```

You can apply a preferred node policy to the node list along with a failback policy.

- Preferred node policy

  If the `preferenced` option is set to `true`, the first node (primary) in the list is the preferred owner of the disk group. It imports the disk group when it boots. The second node in the list (secondary) does not import the disk group when it boots. If the primary node fails, the secondary node takes ownership of the disk group.

  ```
  # scconf -c -D name=nfsdg,nodelist=pnode1:pnode2 \
  ,preferenced=true
  ```

  If the `preferenced` option is set to `false`, any node in the node list that starts clustered operation imports the disk group and places the device group online. If both node are booted simultaneously, either one might import any or all of the available disk groups.

  ```
  # scconf -c -D name=nfsdg,nodelist=pnode1:pnode2 \
  ,preferenced=false
  ```

- Failback policy

  If the primary node fails, the secondary node automatically takes ownership of the disk group. If the failback policy is true, the disk group automatically migrates back (fail back) to the primary node when it boots again.

  ```
  # scconf -c -D name=nfsdg,nodelist=pnode1:pnode2 \
  ,preferenced=true,failback=enabled
  ```

  If the `preferenced` option is set to `false`, the failback feature automatically disables.

Exercise: Configuring Volume Management

In this exercise, you complete the following tasks:

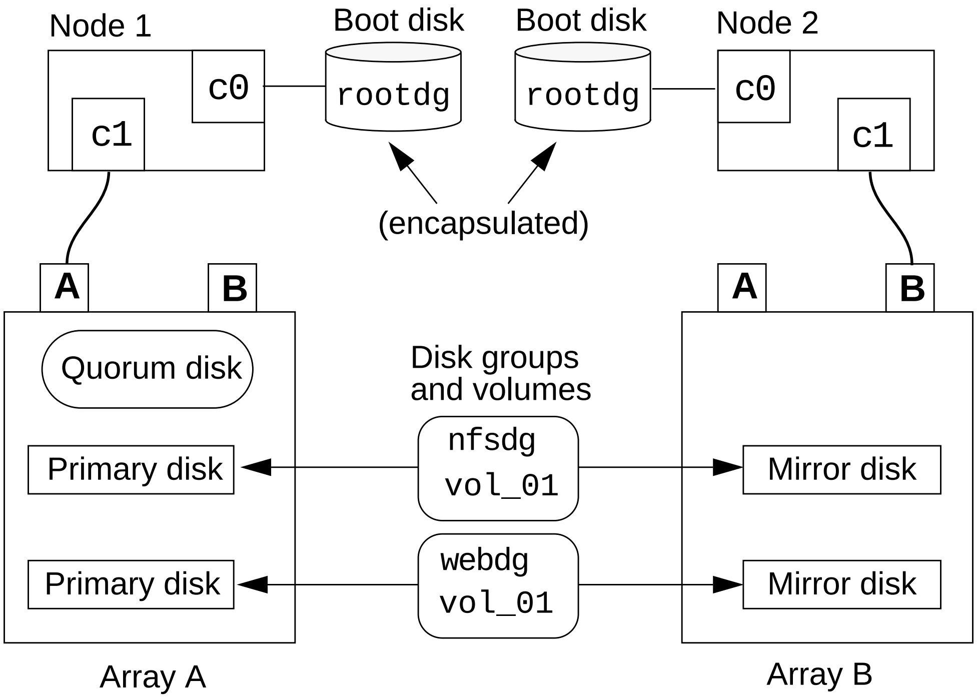
- Install the Veritas Volume Manager software
- Initialize the Veritas Volume Manager software
- Create demonstration volumes
- Perform Sun Cluster disk group registration
- Create global file systems
- Modify device group policies

Preparation

Record the location of the Veritas Volume Manager software you will install during this exercise.

Location: ______________________________

During this exercise, you create a private `rootdg` disk group for each node and two data service disk groups. Each data service disk group contains a single mirrored volume, as shown in the following illustration.



Note – During this exercise, when you see italicized names such as *IPaddress*, *enclosure_name*, *node1*, or *clustername* imbedded in a command string, substitute the names appropriate for your cluster.

Task – Selecting Disk Drives

Before proceeding with this exercise, you must have a clear picture of the disk drives that are used throughout this course. You have already configured one disk drive as a quorum disk. In this exercise, you must identify the boot disks to be encapsulated and two disks in each storage array for use in the two demonstration disk groups, nfsdg and webdg.

1. Use the scstat -q, scdidadm -l, and format commands to identify and record the logical addresses of disks for use during this exercise. Use the format: c0t2d0.

   Quorum disk: _________

   Node 1 Node 2

   Boot disks: _________ _________

   Array A Array B

   nfsdg **disks:** _________ _________

   webdg **disks:** _________ _________

Task – Disabling Dynamic Multipathing

Perform the following steps to prevent Veritas Volume Manager from enabling DMP during installation.

1. On both nodes, symbolically link /dev/vx/dmp to /dev/dsk and /dev/vx/rdmp to /dev/rdsk.

```
# mkdir /dev/vx
# ln -s /dev/dsk /dev/vx/dmp
# ln -s /dev/rdsk /dev/vx/rdmp
```

Task – Installing the Veritas Volume Manager Software

1. Move to the location of the Veritas Volume Manager software on both nodes.
2. Verify you are in the correct location on both nodes.

   ```
   # ls
   VRTSvmdev   VRTSvmdoc   VRTSvmman   VRTSvmsa
   VRTSvxvm
   ```

3. Run the pkgadd command on both nodes to begin the installation.

   ```
   # pkgadd -d . VRTSvmdev VRTSvmman VRTSvxvm
   ```

Note – The Veritas document package VRTSvmdoc and the graphic management tool in VRTSvmsa are not used in this course.

   a. Select your Solaris Operating Environment version. It should be the Solaris 8 10/00 Operating Environment.

   b. Answer yes to the Continue installation question.

   c. Answer yes to the setuid/setgid question.

   d. Answer yes to the continue with VRTSvxvm question.

4. After the installation completes, verify that the Veritas binaries and manual pages are installed as follows:

   ```
   # ls /usr/sbin/vx*
   # ls /etc/vx/bin
   # ls /opt/VRTSvxvm/man
   ```

Task – Installing Patches

This is the appropriate time to install any necessary Veritas Volume Manager patches.

1. Check with your instructor to see if you need to install any Veritas Volume Manager patches.

Task – Verifying the vxio Driver Major Numbers

Perform the following steps to ensure that the vxio driver major numbers are the same on all cluster nodes and that the number is unique.

1. Verify and record the vxio driver major numbers on all nodes.

```
# grep vxio /etc/name_to_major
```

Node 1 vxio major number: _______

Node 2 vxio major number: _______

2. If the numbers are not the same, edit the /etc/name_to_major file on one of the nodes and modify the major number so they match.

Caution – Before you modify the vxio major number on a node, you must first make sure the new number is not already in use on that node. If it is, you will have to select a unique and unused number for both nodes.

Task – Encapsulating the Boot Disk

Both node should be active cluster members. Perform the following steps to encapsulate the system boot disk on each node.

Note – There is an alternate section, "Task – Optional rootdg Disk Group Initialization" on page 7-30 that you can be use to initialize the rootdg disk group. It creates a rootdg disk group for each node using storage array disks.

---

1. Run vxinstall on *both* nodes to encapsulate their boot disks.

```
# /usr/sbin/vxinstall
```

a. Select Custom Installation (**2**).

b. Answer yes (**y**) to Encapsulate Boot Disk.

c. Enter a unique disk name on each node (rootdisk1, rootdisk2)

d. Select Leave these disk alone (4) until you get to the final summary of your choices.

e. Reply yes (**y**) to the boot disk encapsulation.

f. Reply no (**n**) to Shutdown and reboot now.

2. On Node 1, change the `/global` mount entry in `vfstab` to use the logical device paths that were used in the original `/globaldevices` mount entry.

   After modification, the change on Node 1 should look similar to the following:

```
/dev/dsk/c0t0d0s4 /dev/rdsk/c0t0d0s4 /global/.devices/node@1 ufs 2 no global
```

3. On Node 2, change the `/global` mount entry in `vfstab` to use the logical device paths that were used in the original `/globaldevices` mount entry.

   The changes on Node 2 should look similar to the following:

```
/dev/dsk/c0t0d0s4 /dev/rdsk/c0t0d0s4 /global/.devices/node@2 ufs 2 no global
```

4. Shut down the cluster with **`scshutdown -y -g 15`**.

   The default for the grace period (**`-g`**) is 60 seconds.

5. Boot Node 1 in non-cluster mode (`ok` **`boot -x`**).

Note – The Veritas Volume Manager software initiates a reboot after it finishes encapsulating the boot disk.

6. Boot Node 2 in non-cluster mode. During the automatic reboot, bypass any `/global` `fsck` errors by pressing Control-D.

   One node successfully mounts its `/global` file system.

7. Unmount the one successful `/global` file system. This should be on Node 1.

```
# umount /global/.devices/node@1
```

8. Check the `rootdg` minor numbers on each node and compare them.

```
# ls -l /dev/vx/dsk/rootdg
total 0
brw------- 1 root root 171,5 Dec 21 16:47 rootdisk24vol
brw------- 1 root root 171,6 Dec 21 16:47 rootdisk27vol
brw------- 1 root root 171,0 Dec 21 16:47 rootvol
brw------- 1 root root 171,7 Dec 21 16:47 swapvol
```

9. Re-minor the rootdg disk group on Node 1.

```
# vxdg reminor rootdg 100
```

10. Re-minor the rootdg disk group on Node 2.

```
# vxdg reminor rootdg 200
```

11. Verify the root disk volume minor numbers are unique on each node.

```
# ls -l /dev/vx/dsk/rootdg
total 0
brw------- 1 root root 55,100 Apr 4 10:48 rootdiska3vol
brw------- 1 root root 55,101 Apr 4 10:48 rootdiska7vol
brw------- 1 root root 55,  0 Mar 30 16:37 rootvol
brw------- 1 root root 55,  7 Mar 30 16:37 swapvol
```

Note – The rootvol and swapvol minor numbers are automatically renumbered after a reboot.

12. Shut down the cluster and reboot each node in cluster mode. During the reboot. the following error message displays:

```
VxVM starting special volumes ( swapvol )...
/dev/vx/dsk/swapvol: No such device or address
```

Caution – If your boot disk had a separated /var/ or /usr partition before encapsulation, you must perform additional steps before rebooting. Consult Appendix B of the *Sun Cluster 3.0 Installation Guide* for additional information.

Task – Optional rootdg Disk Group Initialization

Do not perform this optional rootdg configuration unless your instructor has directed you to use this procedure instead of boot-disk encapsulation.

1. Select a rootdg disk in each storage array. *Do not use the quorum disk drive or disks you previously selected for the demonstration disk groups.*

2. On each node, manually configure the rootdg disk using the disk you selected for that particular node.

   ```
   # vxconfigd -m disable
   # vxdctl init
   # vxdg init rootdg
   # vxdctl add disk logical_address type=sliced
   # vxdisksetup -i logical_address
   # vxdg adddisk logical_address
   # vxdctl enable
   # rm /etc/vx/reconfig.d/state.d/install-db
   ```

3. Shut down and reboot Node 1.

   ```
   # init 6
   ```

4. After Node 1 has completed its reboot operation, shut down and reboot Node 2.

5. Verify that you see the following Veritas Volume Manager messages when the nodes reboot.

   ```
   VxVM starting in boot mode...
   VxVM general startup...
   ```

Note – You will see warnings such as `vxvm:vxconfigd: WARNING: Disk c2t32d0s2 names group rootdg, but group ID differs` when the nodes boot. This means that there are other rootdg disk groups present that do not belong to this node.

6. Verify that the rootdg disk group is operational on both nodes.

   ```
   # vxprint
   Disk group: rootdg
   TY NAME ASSOC KSTATE LENGTH PLOFFS STATE TUTIL0 PUTIL0
   dg rootdg  rootdg  -  -  -  -  -  -
   dm c2t52d0 c2t52d0s2  -  17678493   -  -  -  -
   ```

Task – Configuring Demonstration Volumes

Perform the following steps to configure two demonstration disk group, each containing a single mirrored volume.

1. On Node 1, create the nfsdg disk group with your previously selected logical disk addresses.

   ```
   # vxdiskadd disk01 disk02
   Which disk group [<group>,none,list,q,?]
   (default: rootdg) nfsdg
   Create a new group named nfsdg? [y,n,q,?]
   (default: y) y
   Use default disk names for these disks? [y,n,q,?]
   (default: y) y
   Add disks as spare disks for nfsdg? [y,n,q,?] (default:
   n) n
   ```

2. Verify the name and ownership of the disks in the nfsdg disk group.

   ```
   # vxdisk list
   ```

3. On Node 1, verify that the new nfsdg disk group is globally linked.

   ```
   # cd /dev/vx/dsk/nfsdg
   # pwd
   /global/.devices/node@1/dev/vx/dsk/nfsdg
   ```

4. On Node 1, create a 500-Mbyte mirrored volume in the nfsdg disk group.

   ```
   # vxassist -g nfsdg make vol-01 500m layout=mirror
   ```

5. On Node 2, create the webdg disk group with your previously selected logical disk addresses.

   ```
   # vxdiskadd disk03 disk04
   Which disk group [<group>,none,list,q,?]
   (default: rootdg) webdg
   ```

6. On Node 2, create a 500-Mbyte mirrored volume in the webdg disk group.

   ```
   # vxassist -g webdg make vol-01 500m layout=mirror
   ```

7. Type the vxprint command on both nodes. Notice that each node does not see the disk group that was created on a different node.

Task – Registering Demonstration Disk Groups

Perform the following steps to register the two new disk groups with the Sun Cluster framework software.

1. On Node 1, use the `scconf` utility to register the `nfsdg` disk group.

```
# scconf -a -D type=vxvm,name=nfsdg,nodelist=node1:node2
```

Note – Put the local node (Node 1) first in the node list.

2. On Node 2, register the `webdg` disk group.

```
# scconf -a -D type=vxvm,name=webdg,nodelist=node2:node1
```

Note – Put Node 2 first in the node list.

3. From either node, verify the status of the disk groups.

```
# scstat -D
```

Note – Until a disk group is registered, you cannot create file systems on the volumes. Even though `vxprint` shows the disk group and volume as being active, the `newfs` utility cannot detect it.

Alternate Method to Register Disk Groups

You can also use the `scsetup` utility, option 3, to register disk groups.

```
# scsetup
```

Note – The `scsetup` utility is recommended for many cluster-related tasks. It is menu-driven and is less prone to user errors.

Task – Creating a Global nfs File System

Perform the following steps on Node 1 to create and mount a demonstration file system on the nfsdg disk group volume.

1. On Node 1, create a file system on vol-01 in the nfsdg disk group.

```
# newfs /dev/vx/rdsk/nfsdg/vol-01
```

2. On *both* Node 1 and Node 2, create a global mount point for the new file system.

```
# mkdir /global/nfs
```

3. On *both nodes*, add a mount entry in the /etc/vfstab file for the new file system with the global and logging mount options.

```
/dev/vx/dsk/nfsdg/vol-01 /dev/vx/rdsk/nfsdg/vol-01 \
/global/nfs  ufs 2 yes global,logging
```

Note – Do not use the line continuation character (\) in the vfstab file.

4. On Node 1, mount the /global/nfs file system.

```
# mount /global/nfs
```

5. Verify that the file system is mounted and available on *both* nodes.

```
# mount
# ls /global/nfs
lost+found
```

Task – Creating a Global `web` File System

Perform the following steps on Node 2 to create and mount a demonstration file system on the `webdg` disk group volume.

1. On Node 2, create a file system on `vol_01` in the `webdg` disk group.

   ```
   # newfs /dev/vx/rdsk/webdg/vol-01
   ```

2. On *both nodes*, create a global mount point for the new file system.

   ```
   # mkdir /global/web
   ```

3. On *both nodes*, add a mount entry in the `/etc/vfstab` file for the new file system with the `global` and `logging` mount options.

   ```
   /dev/vx/dsk/webdg/vol-01 /dev/vx/rdsk/webdg/vol-01 \
   /global/web ufs 2 yes global,logging
   ```

Note – Do not use the line continuation character (\) in the `vfstab` file.

4. On Node 2, mount the `/global/web` file system.

   ```
   # mount /global/web
   ```

5. Verify that the file system is mounted and available on *both* nodes.

   ```
   # mount
   # ls /global/web
   lost+found
   ```

Task – Testing Global File Systems

Perform the following steps to confirm the general behavior of globally available file systems in the Sun Cluster 3.0 environment.

1. On Node 2, move into the `/global/nfs` file system.

   ```
   # cd /global/nfs
   ```

2. On Node 1, try to unmount the `/global/nfs` file system. You should get an error that the file system is busy.
3. On Node 2, move out of the `/global/nfs` file system (`cd /`) and try to unmount it again on Node 1.
4. Mount the `/global/nfs` file system on Node 1.
5. Try unmounting and mounting `/global/nfs` from both nodes.

Task – Managing Disk Device Groups

In the Sun Cluster 3.0 environment, Veritas disk groups become *disk device groups* when they are registered. In most cases, they should *not* be managed using Veritas commands. Some administrative tasks are accomplished using a combination of Sun Cluster and Veritas commands. Common tasks are:

- Adding volumes to a disk device group
- Removing volume from a disk device group

Adding a Volume to a Disk Device Group

To add a volume to an existing device group, perform the following steps:

1. Make sure the *device group* is online (to Sun Cluster).

   ```
   # scstat -D
   ```

Note – You can bring it online to a selected node as follows:

```
        # scswitch -z -D nfsdg -h node1
```

2. Determine which node is currently mastering the related Veritas *disk group* (has it imported).

   ```
   # vxprint
   ```

3. On Node 1, create a 50-Mbyte test volume in the nfsdg disk group.

   ```
   # vxassist -g nfsdg make testvol 50m layout=mirror
   ```

4. Perform the following steps to register the changes to the nfsdg disk group configuration.

   a. Start the scsetup utility.

   b. Select menu item 3, Device groups and volumes.

   c. Select menu item 2, Synchronize volume information.

   d. Supply the name of the disk group and quit scsetup when the operation is finished.

Note – The command line equivalent is:

```
        scconf -c -D name=nfsdg,sync
```

Removing a Volume From a Disk Device Group

To remove a volume from a disk device group, perform the following steps on the node that currently has the related disk group imported:

1. Unmount any file systems that are related to the volume.

2. On Node 1, recursively remove the test volume, testvol, from the nfsdg disk group.

   ```
   # vxedit -g nfsdg -rf rm testvol
   ```

3. Register the nfsdg disk group configuration change with the Sun Cluster framework.

Note – You can use either the scsetup utility or scconf as follows:

```
        scconf -c -D name=nfsdg,sync
```

Migrating Device Groups

The `scconf -p` command is the best method of determining current device group configuration parameters. Verify device group behavior by performing the following steps:

1. Verify the current demonstration device group configuration.

   ```
   # scconf -p |grep group
   Device group name:                        webdg
     Device group type:                      VxVM
     Device group failback enabled:          no
     Device group node list:                 pnode2, pnode1
     Device group ordered node list:         yes
     Diskgroup name:                         webdg
   Device group name:                        nfsdg
     Device group type:                      VxVM
     Device group failback enabled:          no
     Device group node list:                 pnode1, pnode2
     Device group ordered node list:         yes
     Diskgroup name:                         nfsdg
   ```

2. Shut down Node 1. The `nfsdg` disk group should automatically migrate to Node 2 (verify this with the `vxprint` command).

Note – The migration is initiated by Node 1 during shutdown when you see the message: `/etc/rc0.d/K05initrgm: Calling scswitch -S (evacuate)`.

3. Boot Node 1. The `nfsdg` disk group should remain mastered by Node 2.

4. Use the `scswitch` command from either node to migrate the `nfsdg` disk group back to Node 1.

   `# scswitch -z -D nfsdg -h` *`node1`* (use your node name)

Note – The device group is only part of the resource groups that can migrate between cluster nodes. There are other components, such as IP addresses and data services that also migrate as part of a resource group. This section is intended only as a basic introduction to resource group behavior in the cluster environment.

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

- ❑ Explain the disk space management technique used by Veritas Volume Manager
- ❑ Describe the Veritas Volume Manager initialization process
- ❑ Describe how Veritas Volume Manager groups disk drives
- ❑ Install and initialize Veritas Volume Manager
- ❑ Perform Veritas Volume Manager postinstallation configuration
- ❑ Use the basic Veritas Volume Manager status commands
- ❑ Register Veritas Volume Manager disk groups
- ❑ Create global file systems
- ❑ Perform basic device group management

Think Beyond

Where does Veritas Volume Manager recovery fit into the Sun Cluster environment?

Is Veritas Volume Manager required for high-availability functionality?

Module 8

Volume Management Using Solstice DiskSuite

Objectives

Upon completion of this module, you should be able to:

- Explain the disk space management technique used by Solstice DiskSuite
- Describe the Solstice DiskSuite initialization process
- Describe how Solstice DiskSuite groups disk drives
- Use Solstice DiskSuite status commands
- Describe the Solstice DiskSuite software installation process
- Install and initialize Solstice DiskSuite
- Perform Solstice DiskSuite postinstallation configuration
- Create global file systems
- Perform basic device group management

Relevance

Discussion – The following questions are relevant to your learning the material presented in this module:

- Which Solstice DiskSuite features are the most important to clustered systems?
- What relationship does Solstice DiskSuite have to normal cluster operation?
- Are there any restrictions on Solstice DiskSuite features when it is used in the Sun Cluster environment?

Additional Resources

Additional resources – The following references can provide additional information on the topics described in this module:

- *Sun Cluster 3.0 Installation Guide*, part number 806-1419
- *Sun Cluster 3.0 Hardware Guide*, part number 806-1420
- *Sun Cluster 3.0 Data Services Installation and Configuration Guide*, part number 806-1421
- *Sun Cluster 3.0 Data Service Developers Guide*, part number 806-1422
- *Sun Cluster 3.0 System Administration Guide*, part number 806-1423
- *Sun Cluster 3.0 Concepts*, part number 806-1424
- *Sun Cluster 3.0 Error Message Guide*, part number 806-1426
- *Sun Cluster 3.0 Release Notes*, part number 806-1428

Disk Space Management

The Solstice DiskSuite software manages disk space by associating standard UNIX partitions with a data volume structure. A single disk drive can be divided into only seven independent data regions, which is the UNIX partition limit for each physical disk.

Solstice DiskSuite Disk Space Management

As shown in Figure 8-1, Solstice DiskSuite manages virtual volume structures by equating standard UNIX disk partitions with virtual volume structures.

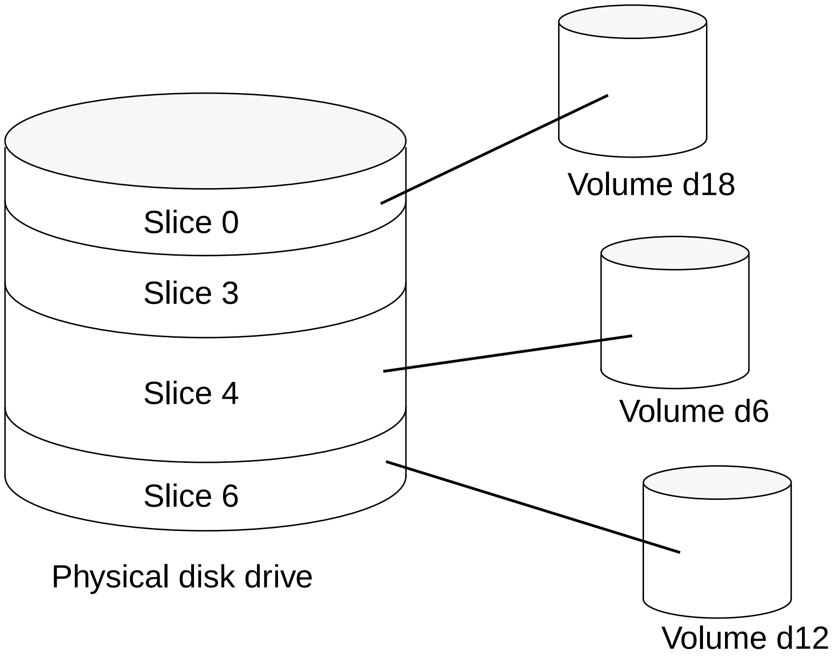


Figure 8-1 Solstice DiskSuite Space Management

Note – Slice 7 on the system boot disk should be reserved for Solstice DiskSuite state database storage.

Solstice DiskSuite Initialization

Disk drives that are to be used by Solstice DiskSuite do not need special initialization. Standard UNIX partitions are used without any modification.

Solstice DiskSuite needs several small databases in which to store volume configuration information along with some error and status information. These are called state databases and can be replicated on one or more disk drives as shown in Figure 8-2. Another common term for the state databases is replicas.

By default, Solstice DiskSuite requires a minimum of three copies of the state database.

The replicas are placed on standard unused partitions by a special command, `metadb`. The default size for each replica is 517 Kbytes (1034 disk blocks).

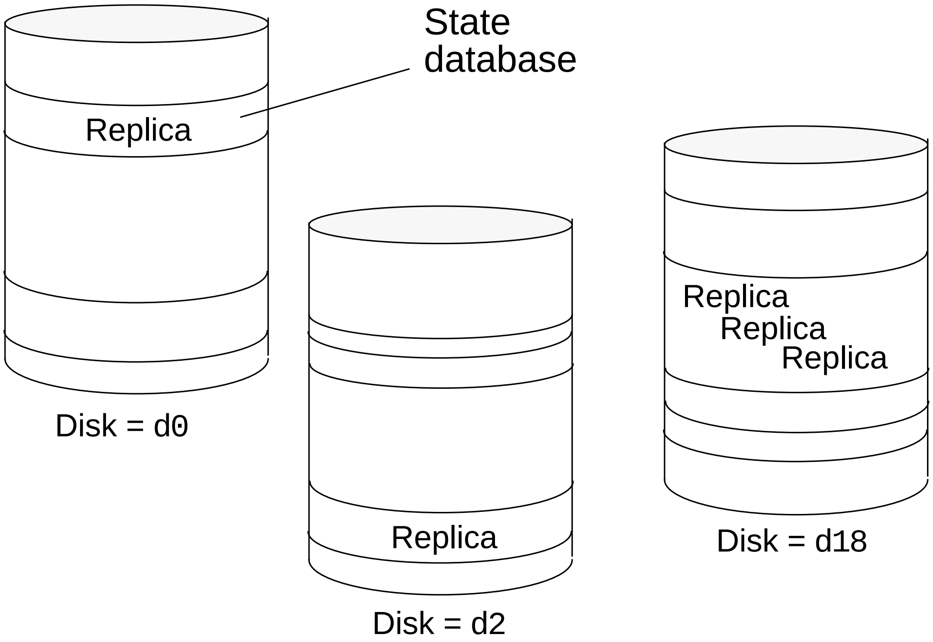


Figure 8-2 Solstice DiskSuite Replica Distribution

Note – Several replicas can be created in a single partition.

Replica Configuration Guidelines

At least one replica is required to start the Solstice DiskSuite software. A minimum of three replicas is recommended. Solstice DiskSuite 4.2 allows a maximum of 50 replicas.

The following guidelines are recommended:

- For one drive – Put all three replicas in one slice
- For two to four drives – Put two replicas on each drive
- For five or more drives – Put one replica on each drive

Use your own judgement to gauge how many replicas are required (and how to best distribute the replicas) in your storage environment.

Note – You cannot store replicas on the `root`, `swap`, or `/usr` partitions, or on partitions containing existing file systems or data.

If multiple controllers exist on the system, replicas should be distributed as evenly as possible across all controllers. This provides redundancy in case a controller fails and also helps balance the load. If multiple disks exist on a controller, at least two of the disks on each controller should store a replica.

Do not place more than one replica on a single disk unless that is the only way to reach the minimum requirement of three replicas.

Solstice DiskSuite Disk Grouping

Disk groups are an arbitrary collection of physical disks that allow a backup host to assume a workload. The disk groups are given unique names and ownership is assigned to a single cluster host system.

Solstice DiskSuite uses the term *diskset* to define a related collection of disk drives.

A *shared diskset* is a grouping of two hosts and disk drives that are accessible by both hosts and have the same device names on both hosts. Each host can have *exclusive* access to a shared diskset; they cannot access the same diskset simultaneously.

Note – It is important to stress that the hosts do not "share" the disk drives in a shared diskset. They can take turns having exclusive access to a shared diskset, but they cannot concurrently access the drives.

Disksets facilitate moving disks between host systems, and are an important piece in enabling high availability. Disksets also enable you to group storage by department or application.

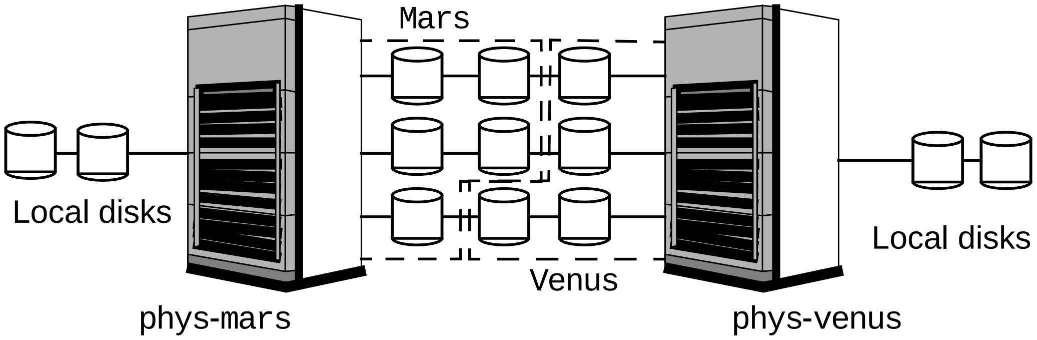


Figure 8-3 Shared Disksets

- Each host must have a *local diskset* that is separate from the shared diskset.
- There is one state database for each shared diskset and one state database for the local diskset.

Adding Disks to a Diskset

You use the `metaset` command to both create new empty disksets and to add disk drives into and existing diskset. In the following example, both functions are performed at the same time.

```
# metaset -s nfsds -a /dev/did/rdsk/d2 \
/dev/did/rdsk/d5
```

Note – You should evenly distribute the disks in each diskset across at least 2 arrays to accommodate mirroring of data. Make sure to use the DID device name instead of the actual disk drive names.

When a disk is added to a diskset, it is automatically repartitioned as follows:

- A small portion of the drive (starting at cylinder 0) is placed into slice 7 to be used for metadevice state database replicas (usually at least 2Mbytes)
- The rest of the drive is placed into slice 0

The drive will *not* be repartitioned if slice 7 has the following characteristics:

- Starts at cylinder 0
- At least 2Mbytes (large enough to hold a state database)
- Has the `V_UNMT` flag set (unmountable flag)
- Not read-only

Dual-String Mediators

Solstice DiskSuite has two kinds of state databases. Initially, a local state database is replicated on each local boot disk. The local replicas are private to each host system.

When a shared diskset is created, a different set of replicas are created that are unique to the diskset. Each shared diskset has its own set of replicas.

If there is any inconsistency in the database replicas, DiskSuite uses what the majority (half + 1) of the database replicas contain. This is called a replica quorum. If a majority of the replica databases cannot be contacted, Solstice DiskSuite effectively shuts down.

When a diskset is configured in a dual-string (that is, two disk arrays) configuration, as shown in Figure 8-4, Solstice DiskSuite splits the number of replicas for each diskset evenly across both arrays.

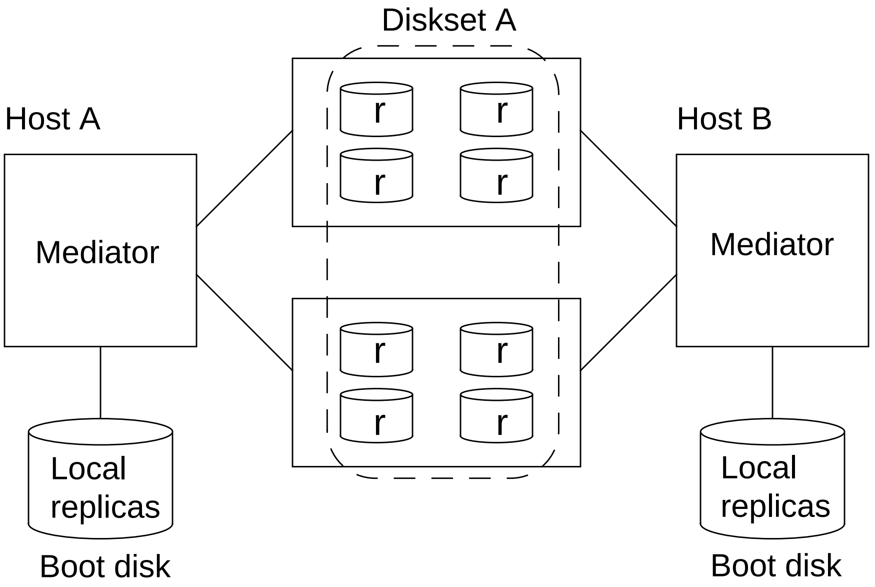


Figure 8-4 Diskset Replica Placement

If one of the disk arrays fails, exactly half of the disk replicas are available and a majority cannot be reached. If exactly half of the replicas are available, the information in the mediators (namely the state database commit count) allows Solstice DiskSuite to determine the integrity of the remaining replicas, effectively casting the tie-breaking vote, allowing DiskSuite to continue as if it had a replica quorum.

Metatrans Devices

After a system panic or power failure, the `fsck` utility checks UFS file systems at boot time. The entire file system must be checked, and this can be time consuming.

Solstice DiskSuite offers a feature called *UFS Logging*, sometimes referred to as “journaling.” UFS logging takes the (logically) synchronous nature of updating a file system and makes it asynchronous. Updates to a file system are not made directly to the disk or partition containing the file system. Instead, user data is written to the device containing the file system, but the file system disk structures are not modified—they are logged instead. Updates to the file system structure are applied to the file system when: the log is detached and the file system is idle for 5 seconds, the log fills, or the device is cleared.

Any changes made to the file system by unfinished system calls are discarded, ensuring that the file system is in a consistent state. This means that logging file systems do not have to be checked at boot time, speeding the reboot process.

As shown in Figure 8-5, a *metatrans device* is used to log a UFS file system. A metatrans device has two components: a *master device* and a *logging device*. The master device contains the UFS file system and has the same on-disk file system format as a non-logged UFS system. The logging device contains the log of file system transactions.

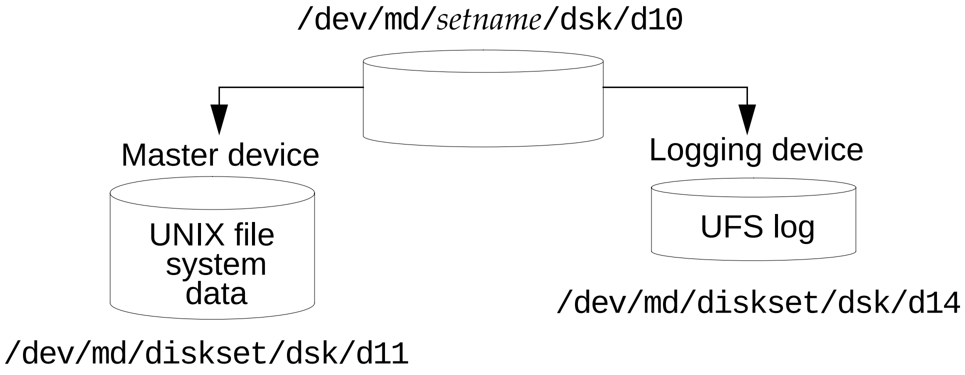


Figure 8-5 Solstice DiskSuite UFS Logging

Both the master and logging devices must be mirrored to prevent data loss in the event of a device failure. Losing data in a log because of device errors can leave a file system in an inconsistent state, and user intervention might be required for repair.

Metatrans Device Structure

Figure 8-6 illustrates a typical metatrans device structure. All components are mirrored.

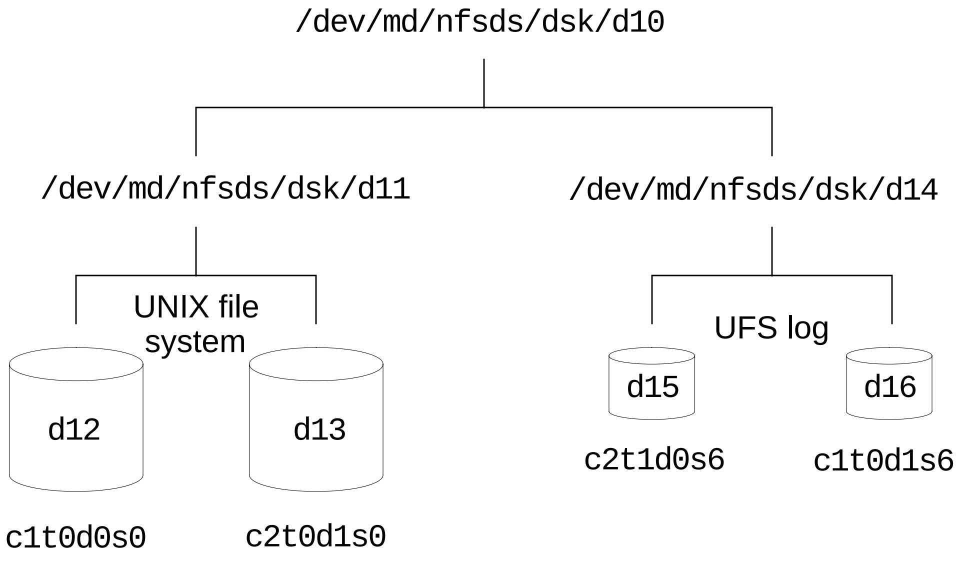


Figure 8-6 Metatrans Device Structure

Solstice DiskSuite Status

Although the GUI for Solstice DiskSuite furnishes useful visual status information, there are times when the images might not update completely due to window interlocks or system loads.

The most reliable and the quickest method of checking status is from the command line. Command-line status tools have the additional benefits of being easy to use in script files, `cron` jobs, and remote logins.

Checking Volume Status

The following `metastat` command output, is for a mirrored metadevice, `d0`, and is used with the Solstice DiskSuite volume manager.

```
# metastat d0
d0: Mirror
Submirror 0: d80
State: Okay
Submirror 1: d70
State: Resyncing  Resyncin progress: 15% done
Pass: 1
Read option: roundrobin (default)
Write option: parallel (default)
Size: 2006130 blocks
```

Note – You can also use the `metastat` command to create a backup configuration file that is suitable for recreating the entire volume structure. This is useful as a worst-case disaster recovery tool.

Checking Mediator Status

You use the `medstat` command to verify the status of mediators in a dual-string storage configuration.

```
# medstat -s nfsds
Mediator       Status     Golden
pnode1         Ok         No
pnode2         Ok         No
```

Checking Replica Status

The status of the state databases is important, and you can verify it using the `metadb` command, as shown in the following example. The `metadb` command can also be used to initialize, add, and remove replicas.

```
# metadb
flags      first blk   block count
 a u       16          1034
/dev/dsk/c0t3d0s5
 a u       16          1034
/dev/dsk/c0t2d0s0
 a u       16          1034
/dev/dsk/c0t2d0s1
 o - replica active prior to last configuration change
 u - replica is up to date
 l - locator for this replica was read successfully
 c - replica's location was in /etc/opt/SUNWmd/mddb.cf
 p - replica's location was patched in kernel
 m - replica is master, this is replica selected as
input
 W - replica has device write errors
 a - replica is active, commits are occurring
 M - replica had problem with master blocks
 D - replica had problem with data blocks
 F - replica had format problems
 S - replica is too small to hold current data base
 R - replica had device read errors
```

The status flags for the replicas shown in the previous example indicate that all of the replicas are active and up to date.

Recording Solstice DiskSuite Configuration Information

Archive diskset configuration information using the `metastat -p` command option. The configuration information is output in a format that you can later use to automatically rebuild your diskset volumes.

```
# metastat -s nfsds -p
nfsds/d100 -m nfsds/d0 nfsds/d10 1
nfsds/d0 1 1 /dev/did/dsk/d10s0
nfsds/d10 1 1 /dev/did/dsk/d28s0
nfsds/d101 -m nfsds/d1 nfsds/d11 1
nfsds/d1 1 1 /dev/did/dsk/d10s1
nfsds/d11 1 1 /dev/did/dsk/d28s1
nfsds/d102 -m nfsds/d2 nfsds/d12 1
nfsds/d2 1 1 /dev/did/dsk/d10s3
nfsds/d12 1 1 /dev/did/dsk/d28s3
```

Solstice DiskSuite Installation Overview

Installing and configuring Solstice DiskSuite for use in a Sun Cluster 3.0 environment consists of the following steps:

1. Install the appropriate packages for Solstice DiskSuite.
2. Modify the `md.conf` file appropriately.
3. Reboot all nodes in the cluster.
4. Create `/.rhosts` files or add `root` to group 14.
5. Initialize the local state databases.
6. Create disksets to house data service data.
7. Add drives to each diskset.
8. Partition disks in the disksets.
9. Create the metadevices for each diskset.
10. Configure dual string mediators if appropriate.

Solstice DiskSuite Postinstallation

After you install the Solstice DiskSuite software, you must make a number of postinstallation configuration modification before Solstice DiskSuite can operate.

Modifying the `md.conf` File

Based on your planned implementation, you might need to update DiskSuite's kernel configuration file: `/kernel/drv/md.conf`. There are two variables that might need to be updated. The modifications are summarized as follows:

Variable	**Default Value**	**Description**
`nmd`	128	Maximum number of metadevices. DiskSuite uses this setting to limit the *names* of the metadevices as well. If you are going to have 100 metadevices, but you want to name them d1000 through d1100, you need to set this to 1101, not 100. The maximum value for `nmd` is 8192.
`md_nsets`	4	Maximum number of disksets. This number should be set to the number of disksets you plan to create in your cluster (probably equal to the number of logical hosts you plan to have assuming one diskset per logical host). The maximum value for `md_nsets` is 32.

Note – This file must be kept identical on all nodes of the cluster. Changes to this file take effect after you perform a reconfiguration reboot.

Enabling Solstice DiskSuite Node Access

Solstice DiskSuite requires root level access on each cluster node. There are two ways to satisfy this requirement:

- Create `/.rhosts` files listing the names of all the cluster nodes on each node of the cluster
- Add root to the `sysadmin` group on all nodes of the cluster

Initializing Local State Database Replicas

Before you can perform any Solstice DiskSuite configuration tasks, such as creating disksets on the multihost disks or mirroring the root (`/`) file system, you must create the metadevice state database replicas on the local (private) disks on each cluster node. The local disks are separate from the multihost disks. The state databases located on the local disks are necessary for basic Solstice DiskSuite operation.

A typical command to place three state database replicas on slice 7 of a system boot disk is as follows:

```
# metadb -a -c 3 -f c0t0d0s7
```

Creating Disksets for the Data Services

On one of the cluster nodes, you create the empty disksets that you need to support the data services. You must specify the names to two hosts that will access the diskset. An example follows:

```
# metaset -s nfsds -a -h pnode1 pnode1
```

Adding Disks to a Diskset

Add disks from different storage arrays to enable mirroring across arrays to maintain high availability. An example follows:

```
# metaset -s nfsds -a /dev/did/rdsk/d7 \
/dev/did/rdsk/d15
```

Configuring Metadevices

Solstice DiskSuite metadevices can be configured in a file, /etc/lvm/md.tab, and automatically created by running the metainit command against the file.

Metadevices can also be created from the command line in a series of steps. The command sequence to create a mirrored metadevice (d100) in the diskset nfsds using two DID disk d7 and d15 is as follows:

1. Create a submirror, d0, on slice 0 of DID device d7, in the diskset named nfsds.

   ```
   # metainit -s nfsds nfsds/d0 1 1 /dev/did/rdsk/d7s0
   ```

2. Create another submirror, d10, on slice 0 of DID device d15, in the diskset named nfsds.

   ```
   # metainit -s nfsds nfsds/d10 1 1
   /dev/did/rdsk/d15s0
   ```

3. Create a metadevice, d100, and add the d0 submirror to it.

   ```
   # metainit -s nfsds nfsds/d100 -m nfsds/d0
   ```

4. Attach the second submirror, d10, to the metadevice d100.

   ```
   # metattach -s nfsds nfsds/d100 nfsds/d10
   ```

5. Verify the status of the metadevice d100.

   ```
   # metastat d100
   d100: Mirror
   Submirror 0: d0
   State: Okay
   Submirror 1: d10
   State: Resyncing  Resync in progress: 15% done
   Pass: 1
   Read option: roundrobin (default)
   Write option: parallel (default)
   Size: 2006130 blocks
   ```

Note – Remember that the size of the volumes are defined by the size of the DID device partitions.

Configuring Dual-String Mediators

If you created any disksets using exactly two disk arrays (which are connected to two cluster nodes), you must configure dual-string mediators. The following rules apply when configuring dual string mediators:

- Disksets using dual strings and two hosts must be configured with exactly two mediator hosts, and these hosts must be the same two hosts used for the diskset.
- A diskset cannot have more than two mediator hosts.
- Mediators cannot be configured for disksets that do not meet the two-string, two-host criteria.

Note – Mediators are not only for use in two-node clusters. Clusters having more than two nodes can also benefit from the use of mediators, depending on the topology and how the disksets are constructed.

The process to configure dual-string mediators is as follows:

1. Start the cluster software on both nodes.
2. Determine the host name of both mediator hosts (nodes).

   ```
   # hostname
   ```

3. Use the `hastat` command to determine the current master of the diskset you are configuring for mediators.
4. Configure the mediators using the `metaset` command on the host that is currently mastering the diskset.

   ```
   # metaset -s nfsds -a -m pnode1
   # metaset -s nfsds -a -m pnode2
   ```

5. Check the mediator status using the `medstat` command.

   ```
   # medstat -s nfsds
   ```

Exercise: Configuring Solstice DiskSuite

In this exercise, you complete the following tasks:

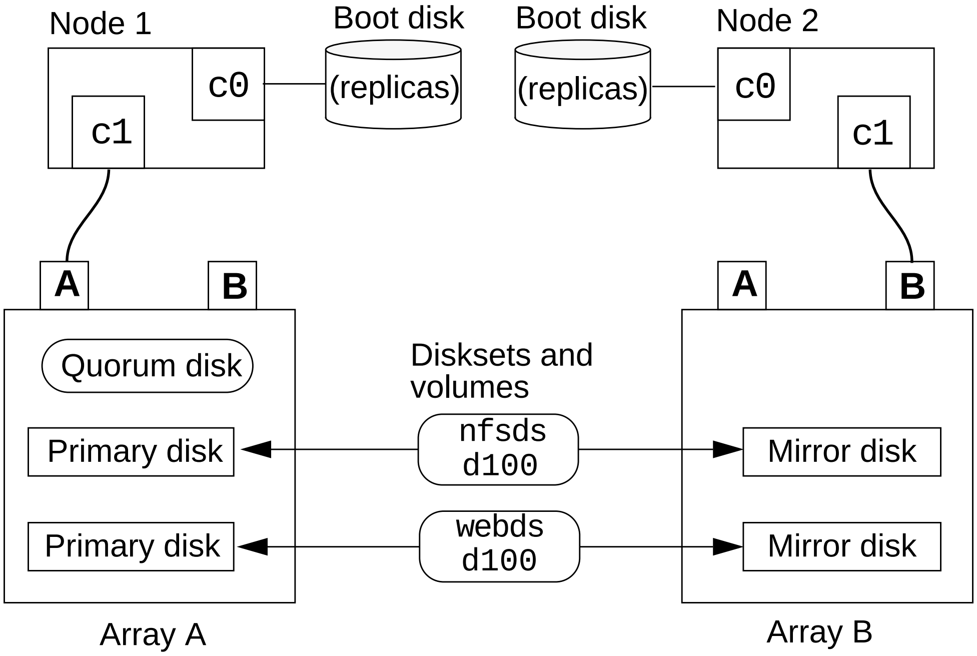
- Install the Solstice DiskSuite volume manager
- Initialize the Solstice DiskSuite volume manager
- Create and manage disksets
- Create dual-string mediators if appropriate
- Create global file systems

Preparation

Ask your instructor about the location of the software that is needed during this exercise. Record the location of the Solstice DiskSuite software.

Solstice DiskSuite location: ________________________

Each of the cluster host boot disks must have a small unused slice that you can use for a state database during the Solstice DiskSuite installation.During this exercise you create two data service disksets that each contain a single mirrored volume as shown in the following illustration.



Note – During this exercise, when you see italicized names such as *IPaddress*, *enclosure_name*, *node1*, or *clustername* imbedded in a command string, substitute the names appropriate for your cluster.

Task – Installing the Solstice DiskSuite Software

1. On both nodes, move to the location of the Solstice DiskSuite software.

2. If you are in the correct location you should see the following files:

```
# ls
SUNWmdg   SUNWmdja  SUNWmdnr  SUNWmdnu  SUNWmdr
SUNWmdu   SUNWmdx
```

3. Run the pkgadd command on both cluster host systems to begin the volume manager installation.

```
# pkgadd -d ‘pwd‘
The following packages are available:
1  SUNWmdg      Solstice DiskSuite Tool
2  SUNWmdja     Japanese localization
3  SUNWmdnr     Log Daemon Configuration Files
4  SUNWmdnu     Log Daemon
5  SUNWmdr      Solstice DiskSuite Drivers
6  SUNWmdu      Solstice DiskSuite Commands
7  SUNWmdx      Solstice DiskSuite Drivers(64-bit)

Select package(s) you wish to process (or 'all' to
process all packages). (default: all) [?,??,q]:
```

Note – Install only the SUNWmdg, SUNWmdr, SUNWmdu, and SUNWmdx packages.

4. Install any current Solstice DiskSuite patches on all cluster host systems. For Solaris 8 10/00 and Solstice DiskSuite 4.2.1, this is 108693-04 or later.

5. Stop the Sun Cluster software on all nodes.

6. Reboot all of the cluster hosts after you install the Solstice DiskSuite patch.

Task – Initializing the Solstice DiskSuite State Databases

Before you can use Solstice DiskSuite to create disksets and volumes, you must initialize the state database and create one or more replicas.

Configure the system boot disk on each cluster host with a small unused partition. This should be slice 7.

1. On each node in the cluster, verify that the boot disk has a small unused slice available for use. Use the `format` command to verify the physical path to the unused slice. Record the paths of the unused slice on each cluster host. A typical path is `c0t0d0s7.`

   Node 1 Replica Slice:_____________

   Node 2 Replica Slice:________________

Warning – You must ensure that you are using the correct slice. A mistake can corrupt the system boot disk. Check with your instructor.

---

2. On each node in the cluster, use the `metadb` command to create three replicas on the unused boot disk slice.

```
# metadb -a -c 3 -f replica_slice
```

Warning – Make sure you reference the correct slice address on each node. You can destroy your boot disk if you make a mistake.

---

3. On both nodes, verify that the replicas are configured and operational.

```
# metadb
flags    first blk   block count
 a  u      16          1034      /dev/dsk/c0t0d0s7
 a  u      1050        1034      /dev/dsk/c0t0d0s7
 a  u      2084        1034      /dev/dsk/c0t0d0s7
```

Task – Selecting Solstice DiskSuite Demo Volume Disk Drives

1. Use the scdidadm command on Node 1 to list all of the available DID drives.

```
# scdidadm -l
```

2. Record the logical path and DID path numbers of four disks that you will use to create the demonstration volumes. Remember to mirror across arrays.

Note – You need to record only the last portion of the DID path. The first part is the same for all DID devices: /dev/did/rdsk.

Diskset	**Volumes**	**Primary disk**	**Mirror disk**
example	*d400*	*c2t3d0* *d4*	*c3t18d0* *d15*
nfsds	d100		
webds	d100		

Caution – All disks in a diskset must be the same geometry (they must all be the same type of disk).

Task – Configuring the Solstice DiskSuite Demonstration Disksets

Perform the following steps to create demonstration disksets and volumes for use in later exercises.

1. On Node 1, create the `nfsds` diskset and configure the nodes that are physically connected to it.

   ```
   # metaset -s nfsds -a -h node1 node2
   ```

2. On Node 1, create the `webds` diskset and configure the nodes that are physically connected to it.

   ```
   # metaset -s webds -a -h node1 node2
   ```

3. Add the primary and mirror disks to the `nfsds` diskset.

   ```
   # metaset -s nfsds -a /dev/did/rdsk/primary \
   /dev/did/rdsk/mirror
   ```

4. Add the primary and mirror disks to the `webds` diskset.

   ```
   # metaset -s webds -a /dev/did/rdsk/primary \
   /dev/did/rdsk/mirror
   ```

5. One Node 1, start the `format` utility and repartition each of your diskset disks and reduce the size of slice 0 to approximately 500-Mbytes. The slices must be identical in size on each primary and mirror pair.

Note – Make sure the first few cylinders are not already mapped to slice 7 for the local state databases.

6. Verify the status of the new disksets.

   ```
   # metaset -s nfsds
   # metaset -s webds
   ```

Task – Configuring Solstice DiskSuite Demonstration Volumes

Perform the following steps on Node 1 to create a 500-Mbyte mirrored volume in each diskset.

Diskset nfsds

1. Create a submirror on each of your disks in the nfsds diskset.

```
# metainit -s nfsds nfsds/d0 1 1 /dev/did/rdsk/primarys0
# metainit -s nfsds nfsds/d1 1 1 /dev/did/rdsk/mirrors0
```

2. Create a metadevice, d100, and add the d0 submirror to it.

```
# metainit -s nfsds nfsds/d100 -m nfsds/d0
```

3. Attach the second submirror, d1, to the metadevice d100.

```
# metattach -s nfsds nfsds/d100 nfsds/d1
```

Diskset webds

1. Create a submirror on each of your disks in the webds diskset.

```
# metainit -s webds webds/d0 1 1 /dev/did/rdsk/primarys0
# metainit -s webds webds/d1 1 1 /dev/did/rdsk/mirrors0
```

2. Create a metadevice, d100, and add the d0 submirror to it.

```
# metainit -s webds webds/d100 -m webds/d0
```

3. Attach the second submirror, d1, to the metadevice d100.

```
# metattach -s webds webds/d100 webds/d1
```

4. Verify the status of the new volumes.

```
# metastat
```

Task – Configuring Dual-String Mediators

If your cluster is a dual-string configuration, you must configure mediation for both of the disksets you have created.

1. Make sure the cluster software is running on the cluster hosts.
2. Use the `metaset` command to determine the current master of the disksets you are configuring for mediators.

Note – Both disksets should be mastered by Node 1.

3. Configure the mediators using the `metaset` command on the host that is currently mastering the diskset.

```
# metaset -s nfsds -a -m node1
# metaset -s nfsds -a -m node2
#
# metaset -s webds -a -m node1
# metaset -s webds -a -m node2
```

4. Check the mediator status using the `medstat` command.

```
# medstat -s nfsds
# medstat -s webds
```

Task – Creating a Global `nfs` File System

Perform the following steps on Node 1 to create a global file system in the `nfsds` diskset.

1. On Node 1, create a file system on `d100` in the `nfsds` diskset.

   ```
   # newfs /dev/md/nfsds/rdsk/d100
   ```

2. On *both* Node 1 and Node 2, create a global mount point for the new file system.

   ```
   # mkdir /global/nfs
   ```

3. On *both nodes*, add a mount entry in the `/etc/vfstab` file for the new file system with the `global` and `logging` mount options.

   ```
   /dev/md/nfsds/dsk/d100 /dev/md/nfsds/rdsk/d100 \
   /global/nfs  ufs 2 yes global,logging
   ```

Note – Do not use the line continuation character (\) in the `vfstab` file.

4. On Node 1, mount the `/global/nfs` file system.

   ```
   # mount /global/nfs
   ```

5. Verify that the file system is mounted and available on *both* nodes.

   ```
   # mount
   # ls /global/nfs
   lost+found
   ```

Task – Creating a Global web File System

Perform the following steps on Node 1 to create a global file system in the webds diskset.

1. On Node 1, create a file system on d100 in the webds diskset.

```
# newfs /dev/md/webds/rdsk/d100
```

2. On *both* Node 1 and Node 2, create a global mount point for the new file system.

```
# mkdir /global/web
```

3. On *both nodes*, add a mount entry in the /etc/vfstab file for the new file system with the global and logging mount options.

```
/dev/md/webds/dsk/d100 /dev/md/webds/rdsk/d100 \
/global/nfs ufs 2 yes global,logging
```

Note – Do not use the line continuation character (\) in the vfstab file.

4. On Node 1, mount the /global/web file system.

```
# mount /global/web
```

5. Verify that the file system is mounted and available on *both* nodes.

```
# mount
# ls /global/web
lost+found
```

Task – Testing Global File Systems

Perform the following steps to confirm the general behavior of globally available file systems in the Sun Cluster 3.0 environment.

1. On Node 2, move into the `/global/nfs` file system.

   ```
   # cd /global/nfs
   ```

2. On Node 1, try to unmount the `/global/nfs` file system. You should get an error that the file system is busy.
3. On Node 2, move out of the `/global/nfs` file system (`cd /`) and try to unmount it again on Node1.
4. Mount the `/global/nfs` file system on Node 1.
5. Try unmounting and mounting `/global/nfs` from both nodes.

Task – Managing Disk Device Groups

Perform the following steps to migrate a disk device group (diskset) between cluster nodes.

1. Make sure the *device groups* are online (to Sun Cluster).

```
# scstat -D
```

Note – You can bring a device group online to a selected node as follows:

```
      # scswitch -z -D nfsds -h node1
```

2. Verify the current demonstration device group configuration.

```
pnode1# scconf -p |grep group
Device group name:                       webds
  Device group type:                     SDS
  Device group failback enabled:         no
  Device group node list:                pnode1, pnode2
  Device group ordered node list:        yes
  Diskset name:                          webds
Device group name:                       nfsds
  Device group type:                     SDS
  Device group failback enabled:         no
  Device group node list:                pnode1, pnode2
  Device group ordered node list:        yes
  Diskset name:                          nfsds
```

3. Shut down Node 1. The `nfsds` and `webds` disksets should automatically migrate to Node 2 (verify with the `scstat -D` command).

Note – The migration is initiated by Node 1 during shutdown when you see the message: `/etc/rc0.d/K05initrgm: Calling scswitch -S (evacuate)`.

4. Boot Node 1. The `both` disksets should remain mastered by Node 2.

5. Use the `scswitch` command from either node to migrate the `nfsds` diskset to Node 1.

```
# scswitch -z -D nfsds -h node1
```

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

❑ Explain the disk space management technique used by Solstice DiskSuite

❑ Describe the Solstice DiskSuite initialization process

❑ Describe how Solstice DiskSuite groups disk drives

❑ Use Solstice DiskSuite status commands

❑ Describe the Solstice DiskSuite software installation process

❑ Install and initialize Solstice DiskSuite

❑ Perform Solstice DiskSuite postinstallation configuration

❑ Create global file systems

❑ Perform basic device group management

Think Beyond

Where does Solstice DiskSuite fit into the high-availability environment?

What planning issues are required for Solstice DiskSuite in the high-availability environment?

Is use of the Solstice DiskSuite required for high-availability functionality?

Module 9

Public Network Management

Objectives

Upon completion of this module, you should be able to:

- List the main features of the Sun Cluster PNM software
- Explain the basic PNM fault monitoring mechanism
- Describe the three NAFO group states
- Configure a NAFO group

Relevance

Discussion – The following questions are relevant to understanding this module's content:

- What happens if a fully functional cluster node loses its network interface to a public network?

Additional Resources

Additional resources – The following references can provide additional information on the topics described in this module:

- *Sun Cluster 3.0 Installation Guide*, part number 806-1419
- *Sun Cluster 3.0 Hardware Guide*, part number 806-1420
- *Sun Cluster 3.0 Data Services Installation and Configuration Guide*, part number 806-1421
- *Sun Cluster 3.0 Data Service Developers Guide*, part number 806-1422
- *Sun Cluster 3.0 System Administration Guide*, part number 806-1423
- *Sun Cluster 3.0 Concepts*, part number 806-1424
- *Sun Cluster 3.0 Error Message Guide*, part number 806-1426
- *Sun Cluster 3.0 Release Notes*, part number 806-1428

Public Network Management

The PNM software is a Sun Cluster package that provides the mechanism for monitoring public network adapters and failing-over IP addresses from one adapter to another when a fault is detected.

Network adapters are organized into NAFO groups. IP addresses are assigned to a NAFO group using regular name service procedures. If a NAFO group network adapter fails, its associated IP address is transferred to the next backup adapter in the group.

Only one adapter in a NAFO group can be active (plumbed) at any time.

Each NAFO backup group on a cluster host is given a unique name, such as `nafo12`, during creation. A NAFO group can consist of any number of network adapter interfaces but usually contains only a few.

As shown in Figure 9-1, the PNM daemon (`pnmd`) monitors designated network adapters on a single node. If a failure is detected, `pnmd` uses information in the CCR and the `pnmconfig` file to initiate a failover to a healthy adapter in the backup group.

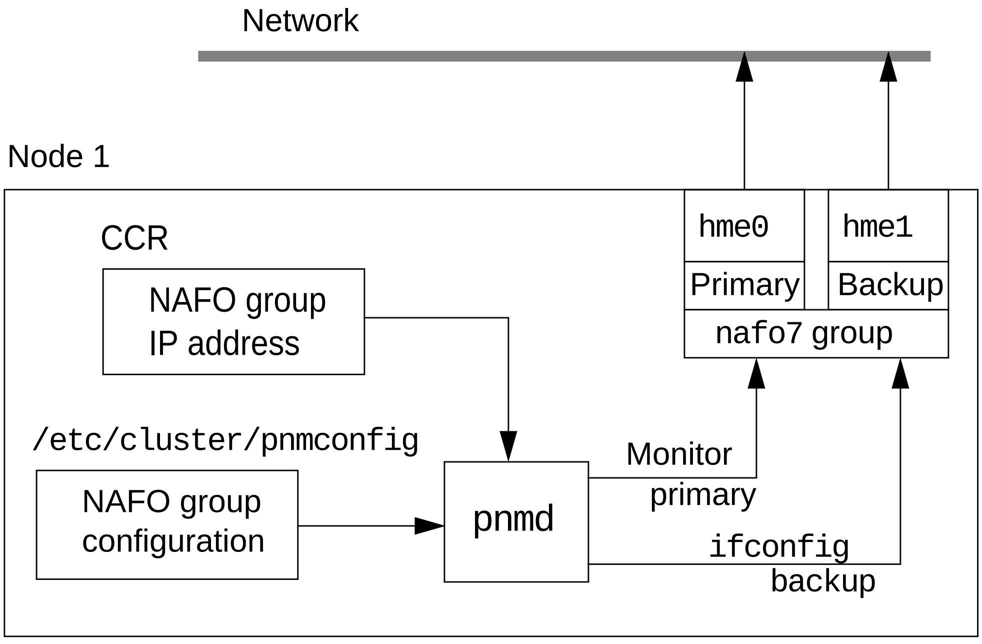


Figure 9-1 PNM Components

Supported Public Network Interface Types

PNM supports the following public network interfaces:

- SBus and PCI bus 100-Mbit/second Ethernet
- Quad Ethernet cards (100-Mbit/second)
- Gigabit Ethernet

Global Interface Support

Sun Cluster 3.0 also provides a global interface feature. A global interface is a single network interface for incoming requests from all clients. It is strongly recommended that the NAFO group for global interfaces are configured with redundant adapters.

Note – The global interface feature is only used by the scalable data services.

Configuring NAFO Groups

Use the `pnmset` command to create and modify network adapter backup groups. You can create several NAFO groups at the same time or one at a time.

Only `root` can run the `pnmset` command.

Pre-Configuration Requirements

Before you use the `pnmset` command to create a new NAFO group, you must make sure `pnmset` can resolve the IP address and logical host name that will be associated with the NAFO group. You must perform the following steps:

1. Set the EEPROM parameter `local-mac-address?` to `false` on all nodes to prevent problems between PNM and the Solaris Operating Environment Network Interface Group feature.
2. Ensure that all adapters in a NAFO group are of the same media type.
3. Ensure that there is a `/etc/hostname.`*xxx* file for each NAFO group.
4. The host name must be resolved either in the local `/etc/hosts` file or through a naming service.

Note – If there is *not* a `/etc/hostname.`*xxx* file for an interface in a NAFO group, `pnmset` uses the node's primary IP address and host name to test the NAFO group interfaces. This means the primary Ethernet interface for the node is taken down for a short time while `pnmset` performs tests on the proposed NAFO group interfaces.

Configuring Backup Groups

The `pnmset` program prompts you for the following information:

- The number of PNM backup groups you want to configure.
- The backup group number.

  The group number is arbitrary. The total number of groups cannot exceed 256. If the group already exists, its configuration is overwritten with new information.
- A list of network adapters to be configured in the group.

  The backup group should contain a minimum of two interfaces. If reconfiguring an existing group, you can add more interfaces.

Sample NAFO Group Configuration

The following is a transcript of the creation of a single NAFO backup group.

```
# pnmset
In the following, you will be prompted to do configuration
for network adapter failover

Do you want to continue ... [y/n]: y

How many NAFO groups to configure [1]: 1

Enter NAFO group number [0]: 2

Enter space-separated list of adapters in nafo2: qfe0 qfe1

Checking configuration of nafo2:

Testing active adapter qfe0...
Testing adapter qfe1...

NAFO configuration completed

# pnmset -p
current configuration is:
nafo2 qfe0 qfe1
```

Modifying Existing PNM Configurations

After PNM is initially configured, you can no longer use pnmset without special options. There are different options to add new NAFO groups or to modify an existing NAFO group.

Adding an Adapter to an Existing NAFO Group

The following is an example of using the pnmset command to add another adapter to an existing NAFO group.

```
# pnmset -c nafo2 -o add qfe2
```

Creating an Additional NAFO Group

After you have created an initial NAFO group on a node, you must use the following form of the pnmset command to create additional NAFO groups.

```
# pnmset -c nafo12 -o create qfe2 qfe3
```

Note – You must first create a /etc/hostname.*xxx* file for the new NAFO group with a new host name. You must also make changes to resolve the new IP address for the logical host.

Consult the pnmset manual page for additional information about adding and modifying NAFO backup groups.

PNM Status Commands

There are several status commands that are useful for checking general NAFO group status and identifying active adapters.

The pnmstat Command

The following shows how to use the pnmstat command to check the status of all local backup groups:

```
# pnmstat -l
group   adapters        status  fo_time act_adp
nafo1   qfe0:qfe1       OK      NEVER   qfe0
```

Note – The NAFO group shown has never experienced a problem, so the failover time (fo_time) shows NEVER.

The pnmptor Command

The following shows how to use the pnmptor command to identify which adapter is active in a given backup group:

```
# pnmptor nafo1
qfe0
```

The pnmrtop Command

The following shows how to use the pnmrtop command to determine which backup group contains a given adapter:

```
# pnmrtop qe0
nafo1
```

The PNM Monitoring Process

The PNM daemon is based on an Remote Procedure Call (RPC) client-server model. It is started at boot time in an `/etc/rc3.d` script and killed in `/etc/rc0.d.`

PNM uses the CCR for storing state information for the results of the adapter monitoring test results on the various hosts. Data services can query the status of the remote adapters at any time using the data service API framework.

When monitoring for network faults, the PNM software must determine where the failure is before taking action. The fault could be a general network failure and not a local adapter. PNM can use the cluster private transport interface to find out if other nodes are also experiencing network access problems. If other nodes (peers) see the problem, then the problem is probably a general network failure, and there is no need to take action.

If the detected fault is determined to be the fault of the local adapter, notify the network failover component to begin an adapter failover, which is transparent to the highly available data services. A backup network adapter is activated to replace the failed network adapter, and the associated IP address is configured on the new interface. This avoids having to move the entire server workload to another server because of the loss of a single network adapter.

If there are no more operational adapters in the NAFO group for a data service, the Sun Cluster API framework then uses NAFO status information to determine:

- Whether to migrate the data service
- Where to migrate the data service

PNM Monitoring Routines

PNM uses three general routines to manage NAFO groups. The routines perform the following functions:

- Monitor active NAFO interfaces
- Evaluate a suspected failure
- Fail over to a new adapter or host

NAFO Group Status

During testing and evaluation, a NAFO group status can transition through several states depending on the results of evaluation and failover. The possible states and associated actions are as follows:

OK State

After monitoring and testing, the current adapter appears to be active. No action is taken, so monitoring continues.

DOUBT State

The current adapter seems to be inactive. Perform further testing and switch to a backup adapter, if necessary. Adapters in the NAFO group are tested sequentially to find an active adapter to replace the currently failed interface.

If a backup adapter is found, the IP address is configured on the new adapter, and the NAFO group status returns to the OK state.

DOWN State

There are no active adapters in the NAFO group. If appropriate, take action to move the IP address and associated resources to a different cluster node.

PNM Parameters

During testing and evaluation of NAFO groups, the PNM monitoring process uses the ping command in increasingly invasive levels as follows:

- First ping ALLROUTERS multicast (224.0.0.2)
- Then ping ALLHOSTS multicast (224.0.0.1)
- Finally, try broadcast ping (255.255.255.255)

Tests are retried a number of times, and there are timeout values for responses. Several configurable PNM parameters can be set in the /etc/cluster/pnmparams file. The parameters can be modified to help ensure that unneeded adapter or data service failovers are not performed because of temporary network performance problems. The parameters are shown in Table 9-1.

Table 9-1 Public Network Management Tunable Parameters

Parameter	Description
inactive_time	The number of seconds between successive probes of the packet counters of the current active adapter. The default is 5.
ping_timeout	The time-out value in seconds for the ALL_HOST_MULTICAST and subnet broadcast ping commands. The default is 4.
repeat_test	The number of times to do the ping sequence before declaring that the active adapter is faulty and failover is triggered. The default is 3.
slow_network	The number of seconds waited after each ping sequence before checking packet counters for any change. The default is 2.
warmup_time	The number of seconds to wait after failover to a backup adapter before resuming fault monitoring. This allows extra time for any slow driver or port initialization. The default is 0. The qfe interface can take 15 seconds or more to initialize with certain switches. Extra warmup time might be needed during initial pnmd startup or during failovers to another host.

Exercise: Configuring the NAFO Groups

In this exercise, you complete the following task:

- Create a NAFO group on each cluster host system

Preparation

Ask your instructor for help with defining the NAFO groups that will be used on your assigned cluster system.

You will create a single NAFO group on each cluster host that is named `nafo0` that contains one or two network adapters.

Ask your instructor for help with selecting the IP address and adapters for use during this exercise. Record them below.

Node	**Logical Host Name**	**IP Address**	**NAFO Group Number**	**Adapters**
Node 1	`nfshost`		`0`	
Node 2	`webhost`		`0`	

Note – During this exercise, when you see italicized names such as *`IPaddress`*, *`enclosure_name`*, *`node1`*, or *`clustername`* imbedded in a command string, substitute the names appropriate for your cluster.

Task – Verifying EEPROM Status

Perform the following steps to verify the EEPROM `local-mac-address?` variable is set to `false` on both nodes.

1. Type the `eeprom` command on each node and verify the setting of the `local-mac-address?` variable.

   ```
   # eeprom | grep mac
   local-mac-address?=false
   ```

2. If necessary, change the `local-mac-address?` value, and reboot the node.

   ```
   # eeprom local-mac-address?=false
   # init 6
   ```

Note – When the nodes are running fully configured data services, you should not simply shut them down. You should first identify data services running on the node and use the `scswitch` command to migrate them to a backup node.

Task – Creating a NAFO Group

Perform the following steps on each node to create a new NAFO group.

1. Verify that NAFO groups do not exist on each cluster host.

   ```
       # pnmstat -l
   ```

2. On Node 1, create a single NAFO group, numbered 1.

   ```
       # pnmset
   ```

Note – If at all possible, each group should consist of two interfaces, one of which can be the primary node interface.

3. On Node 2, create a single NAFO group, numbered 2.
4. Verify that the status of each new NAFO group is OK on all nodes.

   ```
       # pnmstat -l
   ```

Exercise Summary

Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

❑ List the main features of the Sun Cluster PNM software

❑ Explain the basic PNM fault monitoring mechanism

❑ Describe the three NAFO group states

❑ Configure a NAFO group

Think Beyond

Are there other system components that would benefit from the approach taken to network adapters by PNM?

What are the advantages and disadvantages of automatic adapter failover?

Module 10

Resource Groups

Objectives

Upon completion of this module, you should be able to:

- Describe the primary purpose of resource groups
- List the components of a resource group
- Describe the resource group configuration process
- List the primary functions of the `scrgadm` command
- Explain the difference between standard and extended resource type properties

Relevance

Discussion – The following questions are relevant to understanding the content of this module:

- What is the purpose of a resource group?
- What needs to be defined for a resource group?
- What are the restrictions on resource groups?

Additional Resources

Additional resources – The following references can provide additional information on the topics described in this module:

- *Sun Cluster 3.0 Installation Guide*, part number 806-1419
- *Sun Cluster 3.0 Hardware Guide*, part number 806-1420
- *Sun Cluster 3.0 Data Services Installation and Configuration Guide*, part number 806-1421
- *Sun Cluster 3.0 Data Service Developers Guide*, part number 806-1422
- *Sun Cluster 3.0 System Administration Guide*, part number 806-1423
- *Sun Cluster 3.0 Concepts*, part number 806-1424
- *Sun Cluster 3.0 Error Message Guide*, part number 806-1426
- *Sun Cluster 3.0 Release Notes*, part number 806-1428

Resource Group Manager

The RGM controls data services (applications) as *resources*, which are managed by *resource type* implementations. These implementations are either supplied by Sun or created by a developer with a generic data service template, the Data Service Development Library application program interface (API), or the Sun Cluster Resource Management API. The cluster administrator creates and manages resources in containers called *resource groups*, which form the basic unit of failover and switchover. The RGM stops and starts resource groups on selected nodes in response to cluster membership changes.

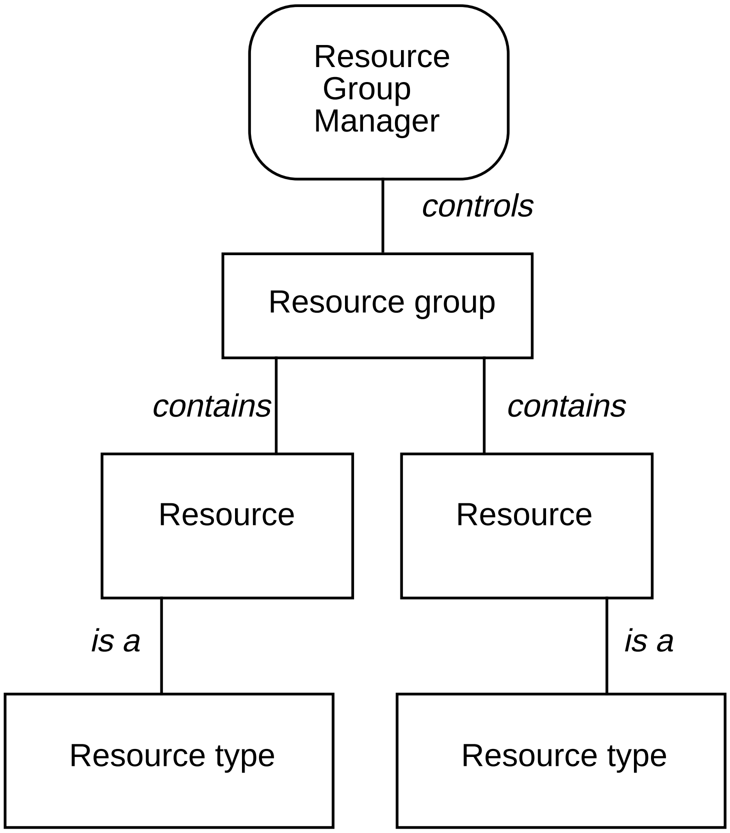


Figure 10-1 Resource Group Management

Resource Types

When you configure a data service and resource groups, you must furnish information about resource types. There are two resource types:

- Data service resource types
- Preregistered resource types (used by most data services)

Data Service Resource Types

The following table lists the resource types associated with each Sun Cluster 3.0 data service.

Table 10-1 Data Service Resource Types

Data Service	**Resource Type**
Sun Cluster HA for Oracle	`SUNW.oracle_server` `SUNW.oracle_listener`
Sun Cluster HA for iPlanet	`SUNW.iws`
Sun Cluster HA for Netscape Directory Server	`SUNW.nsldap`
Sun Cluster HA for Apache	`SUNW.apache`
Sun Cluster HA for DNS	`SUNW.dns`
Sun Cluster HA for NFS	`SUNW.nfs`

Note – You must register a data service resource type once from any cluster node after the data service software is installed.

Preregistered Resource Types

The following resource types are common to many data services.

- `SUNW.HAStorage`
- `SUNW.LogicalHostname` (used by failover data services)
- `SUNW.SharedAddress` (use by scalable data services)

Note – If you accidentally remove a preregistered resource type, it can be re-registered like any resource type.

Failover Data Service Resources

If the node on which the data service is running (the primary node) fails, the service is migrated to another working node without user intervention. Failover services use a *failover resource group*, which is a container for application instance resources and network resources (*logical host names*). Logical host names are IP addresses that can be configured up on one node, and, later, automatically configured down on the original node and configured up on another node.

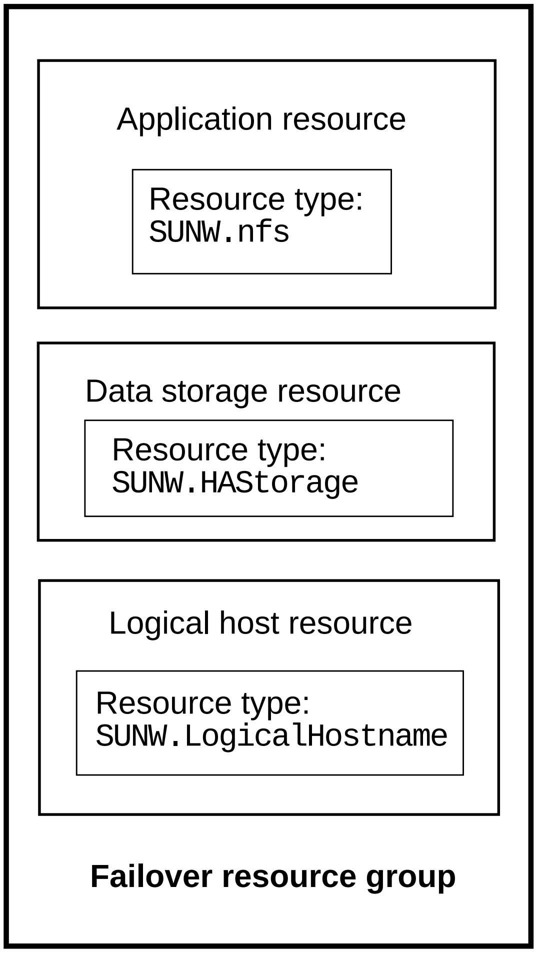


Figure 10-2 Failover Data Service

For failover data services, application instances run only on a single node. If the fault monitor detects an error, it either attempts to restart the instance on the same node, or starts the instance on another node (failover), depending on how the data service has been configured.

Scalable Data Service Resources

The scalable data service has the potential for active instances on multiple nodes. Scalable services use a *scalable resource group* to contain the application resources and a failover resource group to contain the network resources (*shared addresses*) on which the scalable service depends. The scalable resource group can be online on multiple nodes, so multiple instances of the service can be running at once. The failover resource group that hosts the shared address is online on only one node at a time. All nodes hosting a scalable service use the same shared address to host the service.

Service requests come into the cluster through a single network interface (the *global interface* or GIF) and are distributed to the nodes based on one of several predefined algorithms set by the *load-balancing policy*. For scalable services, application instances run on several nodes simultaneously. If the node that hosts the global interface fails, the global interface fails over to another node.

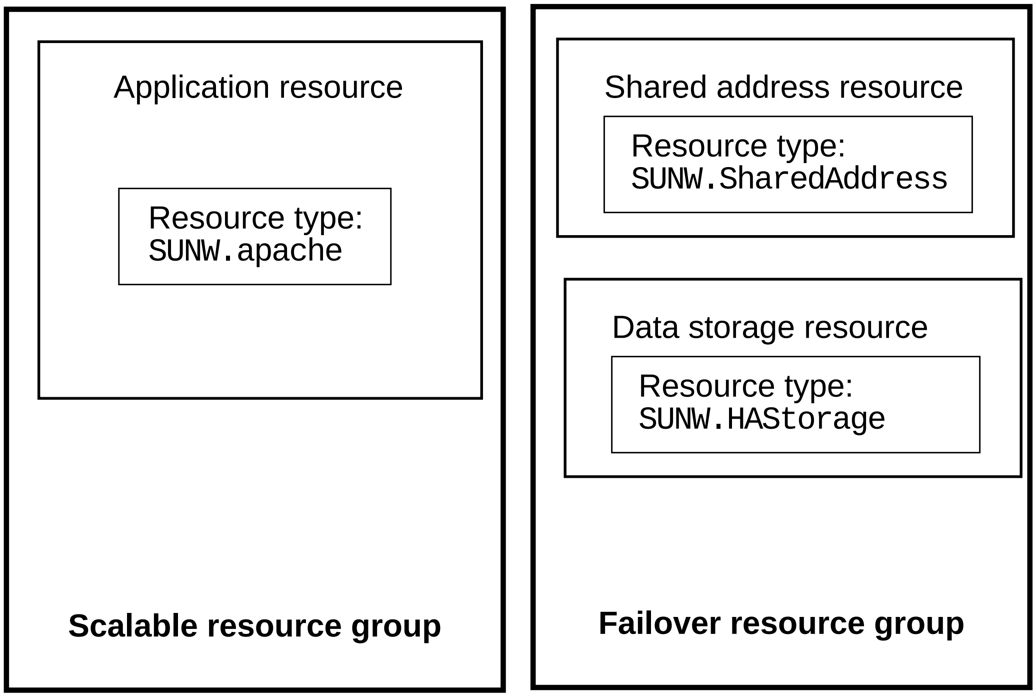


Figure 10-3 Scalable Data Service

Sun Cluster Resource Groups

The primary purpose of the Sun Cluster environment is to provide a platform on which to run data services. The data services can be of the failover or scalable types. Regardless of the data service type, they all need resources to provide effective service to clients.

Each data service configuration requires the following resource information:

- The resource group name (with a node list)
- The logical host name (client access path)
- A NAFO group name (not required)
- A data storage resource
- A data service resource
- Resource properties (mostly defaults)

Configuring a Resource Group

The process of configuring a data service mostly involves configuring a resource group and adding resources to it that are appropriate for the data service. The general process is as follows.

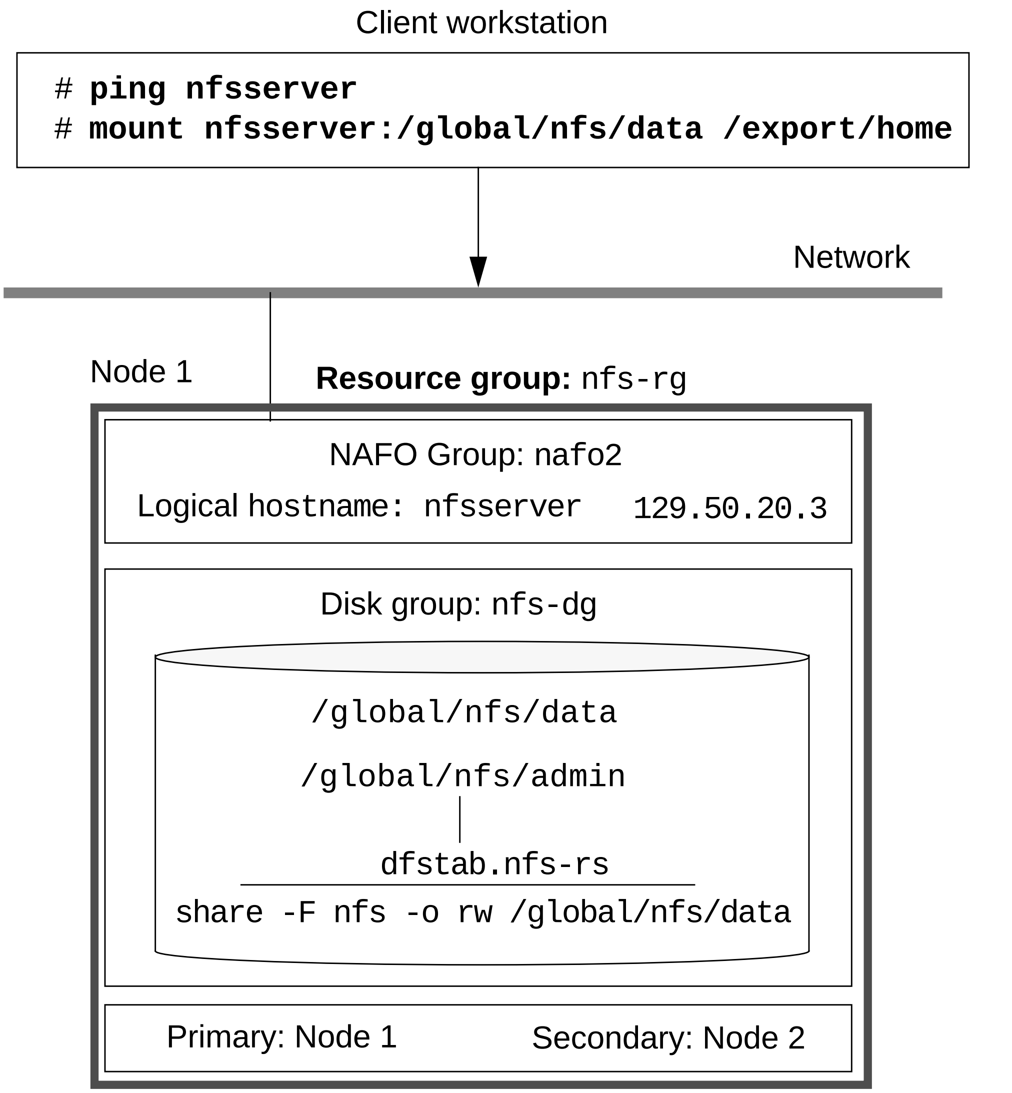
1. Install the data service software.
2. Register the data service (resource type).
3. Create a blank resource group and associate a list of nodes.
4. Associate resources with the new resource group:
   a. Add a logical host name.
   b. Add paths to data resources.
   c. Add the data service resource.
5. Enable the resource group.

Note – Most resource-related properties default to acceptable values.

Resource Group Components

The Sun Cluster HA for NFS data service is widely used and is an example of resources that are necessary to configure a data service.

Figure 10-4 shows the general types of resources that are required for the Sun Cluster HA for NFS data service to function.



```
# ping nfsserver
# mount nfsserver:/global/nfs/data /export/home
```

```
share -F nfs -o rw /global/nfs/data
```

Figure 10-4 Resource Group Elements

Note – The Sun Cluster HA for NFS data service requires a small additional file system for administrative use (`/global/nfs/admin` in Figure 10-4).

Resource Group Administration

Use the `scrgadm` command to administer resource group. Use the `scrgadm` command to *add*, *change*, or *remove* the following:

- Resource types
- Resource groups
- Resources
- Logical host names (failover data service)
- Shared addresses (scalable data service)

The online manual pages for the `scrgadm` command are difficult to understand without practical examples. The general `scrgadm` command process follows.

1. Register resource types needed for the data service (done once after the appropriate data service software is installed).

```
# scrgadm -a -t SUNW.nfs
```

2. Create (add) a blank resource group and give it an arbitrary name.

```
# scrgadm -a -g nfs-rg
```

3. Add the logical host name resource to the new group

```
# scrgadm -a -L -g nfs-rg -l nfs-server
```

4. Add the storage resource (`SUNW.HAStorage`) to the new group.

```
# scrgadm -a -j has-res -g nfs-rg \
-t SUNW.HAStorage \
-x ServicePaths=/global/nfs/data \
-x AffinityOn=True
```

5. Add the data service resource (`SUNW.nfs`) to the new group.

```
# scrgadm -a -j nfs-res -g nfs-rg \
-t SUNW.nfs -y Resource_dependencies=has-res
```

6. Enable the resource group and all of its resources.

```
# scswitch -Z -g nfs-rg
```

Resource Properties

Resource types, resource groups, and resources each have multiple properties associated with them. Some of the properties can be modified, many are fixed. Figure 10-5 shows the relationship of the various resource properties.

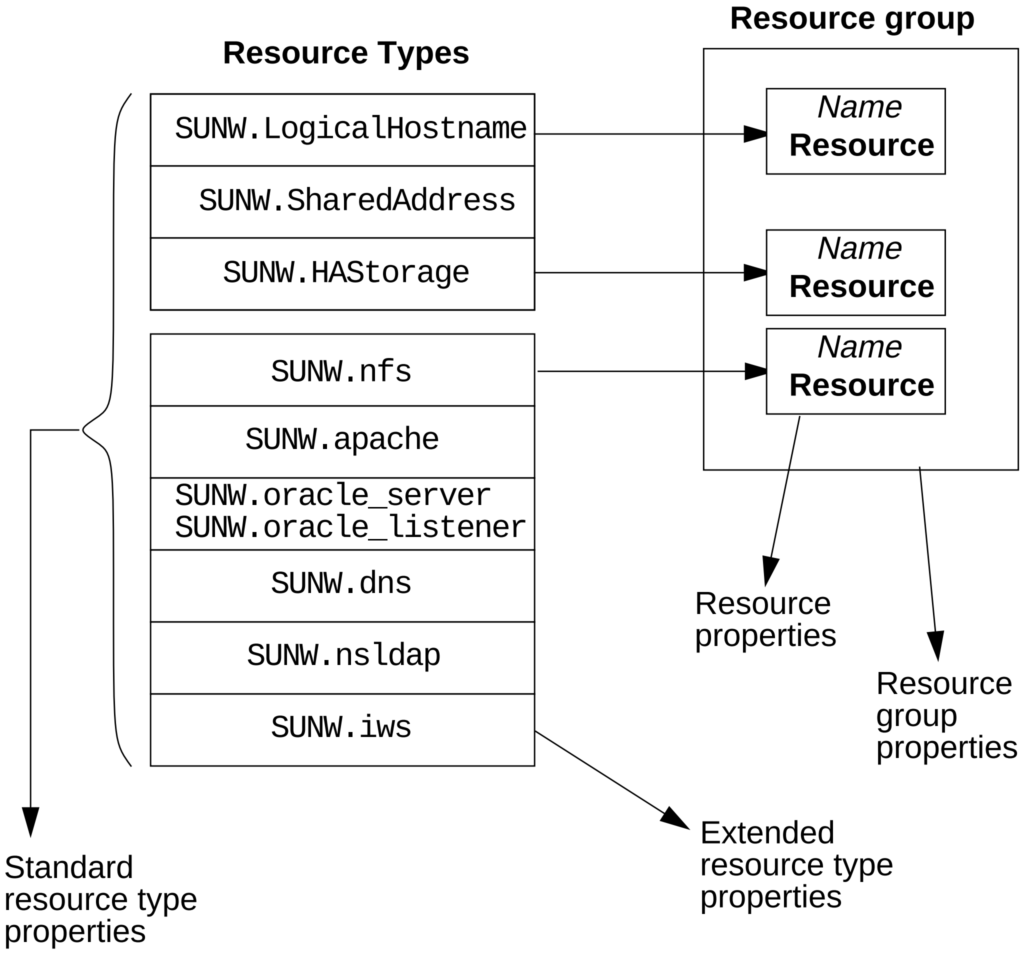


Figure 10-5 Resource Properties

When configuring resource groups, some of the properties must be modified. Most of the properties can be left at their default values.

Standard Resource Type Properties

When registering a resource type, only the Type property is required. The Type property is the name of the resource type. An example follows.

```
# scrgadm -a -t SUNW.nfs
```

Extended Resource Type Properties

Resource types also have *extended* properties that are unique to each type. Some resource types require extended resources. For instance, the SUNW.HAStorage resource type manages global devices associated with a resource group. You must define the path to the global data storage so that it can be managed and ensure that the storage and data service software are both resident on the same node. The SUNW.HAStorage resource type has two extended properties, ServicePaths and AffinityOn, that manage this. An example of their usage follows.

```
# scrgadm -a -j has-res -g nfs-rg \
-t SUNW.HAStorage -x ServicePaths=/global/nfs/data \
-x AffinityOn=True
```

Resource Properties

When you define a resource and add it to an existing resource group, you give the resource an arbitrary and unique name. A resource type is part of the resource definition.

You can apply standard properties to the resource. Standard resource properties apply to any resource type. The only standard resource properties that are required are Resource_name and Type. An example follows.

```
# scrgadm -a -j has-res -g nfs-rg \
-t SUNW.HAStorage
```

Resource Group Properties

When creating (adding) a new resource group, only the resource group RG_name property is required. The RG_name property is the name of the resource group. An example follows.

```
# scrgadm -a -g my-rg
```

The Nodelist property defaults to all cluster nodes, but it is recommended that you supply a list nodes that can bring the resource group online. For a failover resource group, the list is in order of preference (primary/secondary) and should match the node list you supplied when you registered the disk device group that will be used by this resource group. An example follows.

```
# scrgadm -a -g web-rg -h pnode1, pnode2
```

The Nodelist property is not needed with a resource group intended for a scalable application. Scalable applications are usually intended to run on more that one node. The Maximum_primaries and Desired_primaries properties are usually explicitly set when creating a new scalable resource group. An example follows.

```
# scrgadm -a -g web-rg -y Maximum_primaries=2 \
-y Desired_primaries=2
```

A scalable resource group is usually dependent on a different resource group for proper operation. The RG_dependencies property is usually set when creating a scalable resource group. An example follows.

```
# scrgadm -a -g web-rg -y Maximum_primaries=2 \
-y Desired_primaries=2 -y RG_dependencies=sa-rg
```

Note – Configuring standard and extended properties for all the components of a data service is extremely complex. The *Sun Cluster 3.0 Data Services Installation and Configuration Guide* summarizes the extended properties for each resource type. There is also an appendix that lists all of the standard resource properties.

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

❑ Describe the primary purpose of resource groups

❑ List the components of a resource group

❑ Describe the resource group configuration process

❑ List the primary functions of the `scrgadm` command

❑ Explain the difference between standard and extended resource type properties

Think Beyond

If the concept of a logical host did not exist, what would that imply for failover?

What complexities does having multiple backup hosts for a single logical host add to the high-availability environment?

Module 11

Data Services Configuration

Objectives

Upon completion of this module, you should be able to:

- Describe the function of Sun Cluster data services
- Distinguish between highly available and scalable data services
- Describe the operation of data service fault monitors
- Configure the Sun Cluster HA for NFS failover data service
- Configure the Sun Cluster HA for Apache scalable data service
- Switch resource groups between nodes
- Monitor resource groups

Relevance

Discussion – The following questions are relevant to your learning the material presented in this material:

- Why is NFS configured as a failover data service instead of a scalable data service?
- Why might you choose to make the Apache Web Server a failover service instead of a scalable service?

Additional Resources

Additional resources – The following references can provide additional information on the topics described in this module:

- *Sun Cluster 3.0 Data Service Developers Guide*, part number 806-1422
- *Sun Cluster 3.0 System Administration Guide*, part number 806-1423
- *Sun Cluster 3.0 Concepts*, part number 806-1424
- *Sun Cluster 3.0 Release Notes*, part number 806-1428

Sun Cluster Data Service Methods

Each Sun Cluster data service agent consist of specialized software that performs application-specific tasks such as starting, stopping, and monitoring a specific application in the cluster environment. The data service agent components are referred to as *methods*.

Data Service Methods

Each Sun Cluster data service agent supplies a set of data service methods. These methods run under the control of the RGM, which uses them to start, stop, and monitor the application on the cluster nodes. These methods, along with the cluster framework software and multihost disks, enable applications to become highly available data services. As highly available data services, they can prevent significant application interruptions after any single failure within the cluster. The failure could be to a node, an interface component, or to the application itself.

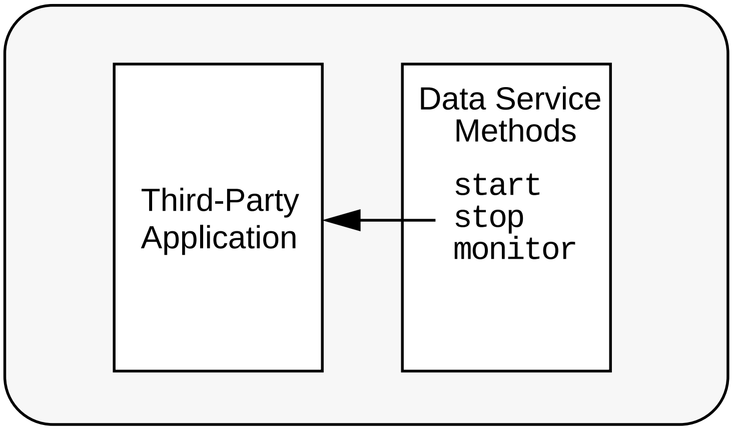


Figure 11-1 Sun Cluster Data Service Methods

Data Service Fault Monitors

Each Sun Cluster data service supplies fault monitor methods that periodically probe the data service to determine its health. A fault monitor verifies that the application daemons are running and that clients are being served. Based on the information returned by probes, predefined actions, such as restarting daemons or causing a failover, can be initiated.

The methods used to verify that the application is running or to restart the application depend upon the application, and they are defined when the data service is developed.

Sun Cluster HA for NFS Methods

The Sun Cluster HA for NFS data service package installs a file, `SUNW.nfs`, that equates generic functions (methods) with specific programs.

```
PRENET_START          =       nfs_prenet_start;
START                 =       nfs_svc_start;
STOP                  =       nfs_svc_stop;
POSTNET_STOP          =       nfs_postnet_stop;
VALIDATE              =       nfs_validate;
UPDATE                =       nfs_update;
MONITOR_START         =       nfs_monitor_start;
MONITOR_STOP          =       nfs_monitor_stop;
MONITOR_CHECK         =       nfs_monitor_check;
```

Each method program file performs tasks that are appropriate for the data service. For instance, the *START* method, when called by the resource group manager, runs the compiled program, `nfs_svc_start`. The `nfs_svc_start` program performs basic NFS startup tasks such as verifying NFS-related daemons are running, restarting some daemons to initiate NFS client lock recovery, and exporting shared file systems.

Most of the basic functions, such as *START*, *STOP*, *MONITOR_START*, and *MONITOR_STOP* are common to all data services. The specific files for each method, however, are different for each data service.

The fault monitoring methods usually start one or more related daemons.

Disk Device Group Considerations

One of Sun Cluster's major improvements over previous releases is the removal of the requirement that a data service must be physically attached to the physical storage storing its data. This means that in a three-or-more node cluster, the data service's resource groups might reside on a node that does not have physical access to the disk device group. File access is done over the interconnect between the data service's node and the node directly connected to the disk device group.

While this provides the benefit of faster failover, it could provide performance degradation for disk-intensive services, such as Oracle or NFS.

In addition, during the cluster boot or failover, a data service might attempt to start before global devices and cluster file systems are online. When this happens, manual intervention is required to reset the state of the resource groups.

To alleviate these problems, the Sun Cluster software includes a `SUNW.HAStorage` resource type.

`SUNW.HAStorage` Resource Type

The resource type `SUNW.HAStorage` serves the following purposes:

- It coordinates the boot order by monitoring the global devices and cluster file systems and causing the `START` methods of the other resources in the same group with the `SUNW.HAStorage` resource to wait until the disk device resources become available.
- With the resource property `AffinityOn` set to `True`, it enforces colocation of resource groups and disk device groups on the same node, thus enhancing the performance of disk-intensive data services.

SUNW.HAStorage Guidelines

On a two-node cluster, configuring SUNW.HASstorage is optional. To avoid additional administrative tasks, however, you could set up SUNW.HAStorage for all resource groups with data service resources that depend on global devices or cluster file systems.

You might also want to set the resource property AffinityOn to true for disk-intensive applications if performance is a concern. Be aware, however, that this increases failover times because the application does not start up on the new node until the device service has been switched over and is online on the new node.

In all cases, consult the documentation for the data service for specific recommendations.

Overview of Data Service Installation

Data service installation consists of the following steps:

- Preparing for the data service installation
- Installing and configuring the application software
- Installing the Sun Cluster data service software packages
- Registering and configuring the data service

The following sections provide an overview of the process. The actual installation and configuration of a failover and scalable data service is discussed later in the module.

Preparing for Data Service Installation

Proper planning will ensure that the data service installation proceeds smoothly. Before proceeding with the installation, be sure to:

- Determine the location of the application binaries
- Verify the contents of the `nsswitch.conf` file
- Plan the cluster file system configuration

Determining the Location of the Application Binaries

You can install the application software and configuration files on the local disks of each cluster node or on the cluster file system.

The advantage to placing the software and configuration files on the individual cluster nodes is that if you want to upgrade the application software later, you can do so without shutting down the cluster.

The disadvantage is that you then have several copies of the software and configuration files to maintain and administer.

If you can spare the downtime for upgrades, install the application on the cluster file system. If it is critical that the application remain up, install it on the local disks of each cluster node. In some cases, the data service documentation might contain recommendations for placement of the binaries and configuration files. Be sure to adhere to any guidelines listed in the documentation.

Verifying the Contents of the `nsswitch.conf` File

The `nsswitch.conf` file is the configuration file for name service lookups. This file determines which databases within the Solaris environment to use for name service lookups and in what order to consult the databases.

For some data services, you must change the "group" line in the file so that the entry is `cluster files`. To determine whether you need to change the "group" line, consult the documentation for the data service you are configuring.

Planning the Cluster File System Configuration

Depending on the data service, you might need to configure the cluster file system to meet Sun Cluster requirements. To determine whether any special considerations apply, consult the documentation for the data service you are configuring.

Installing and Configuring the Application Software

Installation and configuration of the application software is specific to the application itself. In some cases, additional software must be purchased.

Installing the Sun Cluster Data Service Software Packages

Sun Cluster data service software packages are included with the Sun Cluster software package on Sun Cluster Agent CD. Installation is specific to the data service.

Registering and Configuring a Data Service

The steps to register and configure a data service depend upon the particular data service and whether you will configure it as a scalable or failover service.

Sun Cluster's release notes include an appendix with charts for planning resources and resource groups. These charts are also included as an appendix in your course book. Using these charts as a guide might help the installation to proceed more smoothly.

Installing Sun Cluster HA for NFS

The Sun Cluster HA for NFS data service is an example of a failover data service. One node of the cluster serves as the NFS server. If it fails, the service fails over to one of the other nodes in the cluster.

This section details the steps for installing and configuring Sun Cluster HA for NFS on Sun Cluster servers and the steps for adding the service to a system that is already running Sun Cluster.

Preparing for Installation

It is important to perform preinstallation planning before installing any of the data services.

Determining the Location of the Application Binaries

The NFS server is part of the Solaris Operating Environment and is already installed locally on each node.

Verifying the Contents of the nsswitch.conf File

Ensure that the hosts line in /etc/nsswitch.conf is configured as follows:

```
hosts:      cluster files nis
```

This prevents timing-related failures because of name service lookup.

Planning the Cluster File System Configuration

Identify the global file system you created earlier to house the NFS data. This directory should already be mounted on your cluster nodes.

If you are not sure whether the directory is mounted, run the following command to verify:

```
# df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/md/dsk/d30      1984564  821360 1103668    43%    /
/proc                      0       0       0     0%    /proc
fd                         0       0       0     0%    /dev/fd
mnttab                     0       0       0     0%    /etc/mnttab
swap                 5503960     184 5503776     1%    /var/run
swap                 5503808      32 5503776     1%    /tmp
/dev/md/dsk/d70        95702    3502   82630     5%
/global/.devices/node@2
/dev/md/dsk/d40        95702    3502   82630     5%
/global/.devices/node@1
/dev/md/webds/dsk/d100
                     70011836   65819 69245899     1%    /global/web
/dev/md/nfsds/dsk/d100
                     70011836   65822 69245896     1%    /global/nfs
```

Installing and Configuring the Application Software

The NFS server is part of the Solaris Operating Environment and is already installed locally on each node.

Installing the Sun Cluster Data Service Software Packages

If the data service packages were installed as part of your initial Sun Cluster installation, use the following command to verify the installation:

```
# pkginfo -l SUNWscnfs
   PKGINST:  SUNWscnfs
      NAME:  Sun Cluster NFS Server Component
  CATEGORY:  application
      ARCH:  sparc
   VERSION:  3.0.0,REV=2000.10.01.01.00
   BASEDIR:  /opt
    VENDOR:  Sun Microsystems, Inc.
      DESC:  Sun Cluster nfs server data service
    PSTAMP:  octavia20001001013148
  INSTDATE:  Dec 05 2000 13:48
   HOTLINE:  Please contact your local service provider
    STATUS:  completely installed
     FILES:     18 installed pathnames
                 3 directories
                15 executables
              2596 blocks used (approx)
```

Registering and Configuring the Data Service

The following steps complete the installation and configuration process.

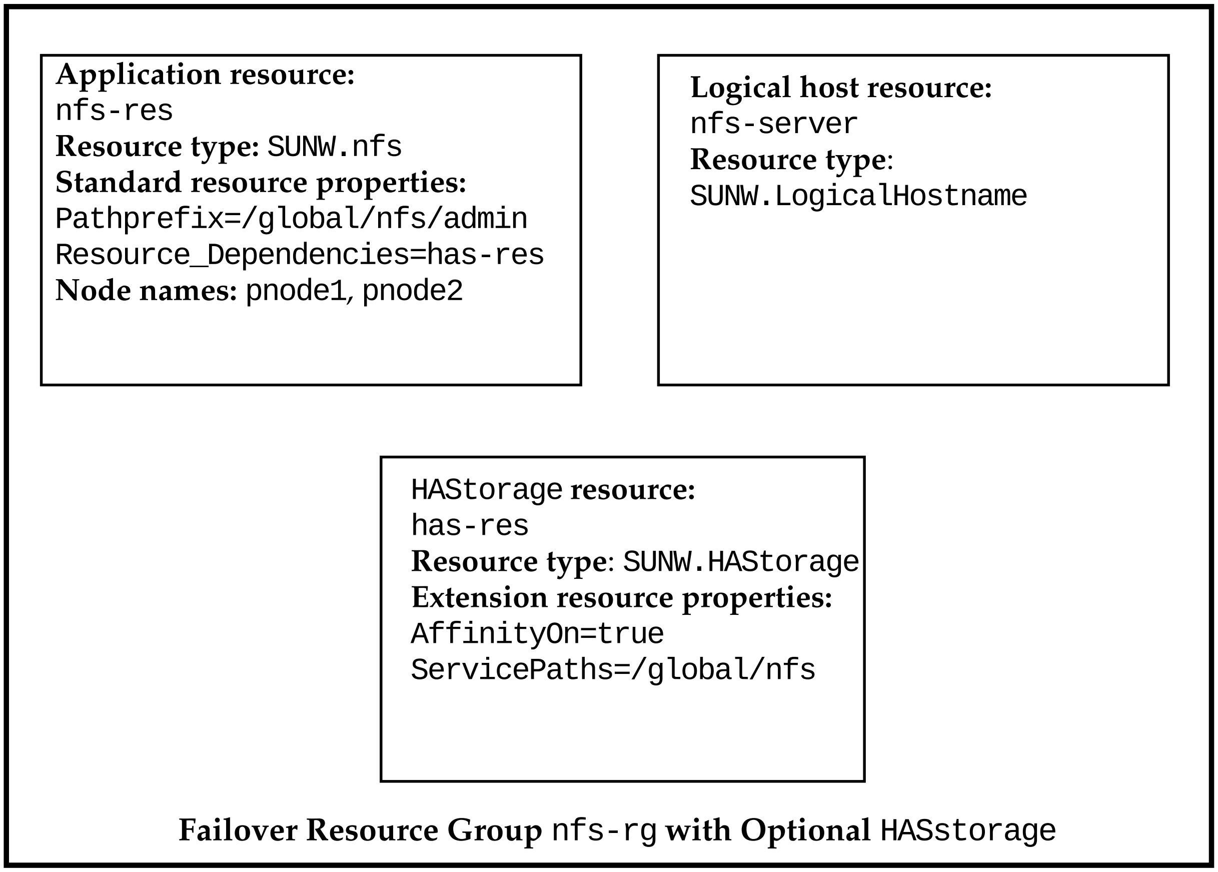
1. Become superuser on a node in the cluster.
2. Verify that all nodes in the cluster are up and functional:

```
# scstat
```

3. Create a plan for the resources that will be in the failover resource group.

Planning the Resources and Resource Group

Because this is a failover data service, plan for one resource group for the application and logical host resource. Configuring `HAStorage` is optional but highly recommended for NFS.



Application resource:
`nfs-res`
Resource type: `SUNW.nfs`
Standard resource properties:
`Pathprefix=/global/nfs/admin`
`Resource_Dependencies=has-res`
Node names: pnode1, pnode2

Logical host resource:
`nfs-server`
Resource type:
`SUNW.LogicalHostname`

`HAStorage` **resource:**
`has-res`
Resource type: `SUNW.HAStorage`
Extension resource properties:
`AffinityOn=true`
`ServicePaths=/global/nfs`

Failover Resource Group `nfs-rg` **with Optional** `HASstorage`

Figure 11-2 NFS Resource Group Plan

NFS Resource `Pathprefix` Extension Property

The NFS data service supports a large set of standard and extension resource properties, but only one resource property must be specified at installation. This property, the `Pathprefix` property, specifies a directory path to hold NFS state information and `dfstab` files for the NFS resource.

The NFS resource contains the full list of standard and extension properties available for the NFS data service. In general, these properties contain appropriate defaults and do not need modification.

`HAStorage` Resource `ServicePaths` Extension Property

Service paths can contain global device names, paths to global devices, or the global mount point.

4. Verify that the logical host name has been added to your name service database.

Note – To avoid failures because of name service lookup, verify that the logical host name is present in the server's and client's `/etc/hosts` file.

5. Create the directory specified in `Pathprefix`.

```
# cd /global/nfs
# mkdir admin
```

6. From any node of the cluster, create the administrative directory below the `Pathprefix` directory for the NFS configuration files.

```
# cd admin
# mkdir SUNW.nfs
```

7. Create a `dfstab.`*`resource-name`* file in the newly created `SUNW.nfs` directory. Set up the share options for each path you have created to be shared. The format of this file is exactly the same as the format used in `/etc/dfs/dfstab`:

```
# cd SUNW.nfs
# vi dfstab.nfs-res

share -F nfs -o rw -d”Home Dirs” /global/nfs/data
```

The `share -o rw` command grants write access to all clients, including the host names used by Sun Cluster, and enables Sun Cluster HA for NFS fault monitoring to operate most efficiently.

You can specify a client list or net group for security purposes. Be sure to include all cluster nodes and logical hosts so that the cluster fault monitoring can do a thorough job.

8. Create the directory specified in the `dfstab.nfs-res file.`

   ```
   # cd /global/nfs
   # mkdir /global/nfs/data
   ```

9. Register the NFS and `HAStorage` resource types.

   ```
   # scrgadm -a -t SUNW.nfs
   # scrgadm -a -t SUNW.HAStorage
   ```

Note – The `scrgadm` does not produce any output if the command completes without errors.

10. Create the failover resource group.

    ```
    # scrgadm -a -g nfs-rg -h pnode1,pnode2 \
    -y Pathprefix=/global/nfs/admin
    ```

11. Add the logical host name resource to the resource group.

    ```
    # scrgadm -a -L -g nfs-rg -l nfs-server
    ```

12. Create the `SUNW.HAStorage` resource.

```
# scrgadm -a -j has-res -g nfs-rg -t SUNW.HAStorage \
-x ServicePaths=/global/nfs -x AffinityOn=True
```

13. Create the NFS resource.

    ```
    # scrgadm -a -j nfs-res -g nfs-rg -t SUNW.nfs -y
    Resource_dependencies=has-res
    ```

14. Enable the resources and the resource monitors, manage the resource group, and switch the resource group into the online state.

    ```
    # scswitch -Z -g nfs-rg
    ```

15. Verify that the data service is online.

    ```
    # scstat -g
    ```

Note – Use the `scstat -g` command to monitor the status of the cluster's resources.

Testing NFS Failover

Verify that the NFS data service is working properly by switching the NFS failover resource group to the other node.

1. Use the output from the `scstat -g` command to determine which node is currently hosting the NFS server. In the following example, pnode2 is the current host.

```
-- Resource Groups and Resources --

            Group Name     Resources
            ----------     ---------
Resources:  nfs-rg         nfs-server nfs-res has-res


-- Resource Groups --

            Group Name     Node Name      State
            ----------     ---------      -----
Group:      nfs-rg         pnode1         Offline
Group:      nfs-rg         pnode2         Online


-- Resources --

            Resource Name Node Name  State    Status Message
            ------------- ---------  -----    --------------
Resource:   nfs-server     pnode1     Offline Offline
Resource:   nfs-server     pnode2     Online  Online -
LogicalHostname online.

Resource:   nfs-res        pnode1     Offline Offline
Resource:   nfs-res        pnode2     Online  Online -
Service is online.

Resource:   has-res        pnode1     Offline Offline
Resource:   has-res        pnode2     Online  Online
```

2. Use the scswitch command to switch the NFS resource group to pnode1.

```
# scswitch -z -h pnode1 -g nfs-rg
```

Note – As with the scrgadm command, the scswitch command only generates output if there is an error.

3. Run scstat -g again to verify that the switch was successful.

```
-- Resource Groups and Resources --

            Group Name    Resources
            ----------    ---------
Resources:  nfs-rg        nfs-server nfs-res has-res


-- Resource Groups --

            Group Name    Node Name     State
            ----------    ---------     -----
Group:      nfs-rg        pnode1        Online
Group:      nfs-rg        pnode2        Offline


-- Resources --

            Resource Name Node Name  State   Status Message
            ------------- ---------  -----   -----
Resource:   nfs-server    pnode1     Offline Online -
LogicalHostname online
Resource:   nfs-server    pnode2     Online  Offline -
LogicalHostname off.

Resource:   nfs-res       pnode1     Offline Online -
Successfully started NFS service
Resource:   nfs-res       pnode2     Online  Offline -
Completed successfully.

Resource:   has-res       pnode1     Offline Online
Resource:   has-res       pnode2     Online  Offline
```

Installing Sun Cluster Scalable Service for Apache

Sun Cluster HA for Apache is an example of a data service that can be configured as either a failover or scalable service. In this section, you learn how to configure Sun Cluster HA for Apache as a scalable service.

As a scalable service, Apache Web Server runs on all nodes of the cluster. If one of the nodes fail, the service continues to operate on the remaining nodes of the cluster.

Preparing for Installation

It is important to perform preinstallation planning before installing any of the data services.

Determining the Location of the Application Binaries

The Apache software is included in the Solaris 8 Operating Environment CD. If you installed the Entire Distribution of the Solaris 8 Operating Environment, the Apache software is already installed locally on each node.

Verify that the Apache packages are installed on your cluster by issuing the following command on all cluster nodes:

```
# pkginfo | grep Apache

system      SUNWapchd      Apache Web Server Documentation
system      SUNWapchr      Apache Web Server (root)
system      SUNWapchu      Apache Web Server (usr)
```

If the packages are not installed, you still have the option to place the binaries on a global file system. However, for performance purposes, it is recommended that you place the software on each cluster node.

Verifying the Contents of the `nsswitch.conf` File

Ensure that the `hosts` line in the `/etc/nsswitch.conf` file is configured as follows:

```
hosts:       cluster files nis
```

This prevents timing-related failures because of name service lookup.

Planning the Cluster File System Configuration

Identify the global file system you created earlier to house the Web server data. This directory should already be mounted on your cluster nodes.

If you are not sure whether the directory is mounted, run the following command to verify that the directory is mounted:

```
              # df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/md/dsk/d30      1984564  821360 1103668    43%    /
/proc                      0       0       0     0%    /proc
fd                         0       0       0     0%    /dev/fd
mnttab                     0       0       0     0%    /etc/mnttab
swap                 5503960     184 5503776     1%    /var/run
swap                 5503808      32 5503776     1%    /tmp
/dev/md/dsk/d70        95702    3502   82630     5%
/global/.devices/node@2
/dev/md/dsk/d40        95702    3502   82630     5%
/global/.devices/node@1
/dev/md/webds/dsk/d100
                     70011836   65819 69245899     1%    /global/web
/dev/md/nfsds/dsk/d100
                     70011836   65822 69245896     1%    /global/nfs
```

Installing and Configuring the Application Software

You must perform the following procedures on all nodes of the cluster. Use the cluster console to ensure that the steps are performed on each node.

If the Apache Web Server software is not installed, you can install it from the Solaris 8 Software CD 2 of 2.

```
# pkgadd -d /cdrom/cdrom0/Solaris_8/Product SUNWapchr SUNWapchu SUNWapchd
...
Installing Apache Web Server (root) as SUNWapchr
...
[ verifying class initd ]
/etc/rc0.d/K16apache linked pathname
/etc/rc1.d/K16apache linked pathname
/etc/rc2.d/K16apache linked pathname
/etc/rc3.d/S50apache linked pathname
/etc/rcS.d/K16apache linked pathname
...
```

1. Disable the `START` and `STOP` run control scripts that were just installed as part of the `SUNWapchr` package.

   This step is necessary because Sun Cluster HA for Apache starts and stops the Apache application once you have configured the data service. Perform the following three steps:

   a. List the Apache run control scripts.

```
# ls -1 /etc/rc?.d/*apache
/etc/rc0.d/K16apache
/etc/rc1.d/K16apache
/etc/rc2.d/K16apache
/etc/rc3.d/S50apache
/etc/rcS.d/K16apache
```

   b. Rename the Apache run control scripts.

```
# mv /etc/rc0.d/K16apache /etc/rc0.d/k16apache
# mv /etc/rc1.d/K16apache /etc/rc1.d/k16apache
# mv /etc/rc2.d/K16apache /etc/rc2.d/k16apache
# mv /etc/rc3.d/S50apache /etc/rc3.d/s50apache
# mv /etc/rcS.d/K16apache /etc/rcS.d/k16apache
```

c. Verify that all the Apache-related scripts have been renamed.

```
# ls -1 /etc/rc?.d/*apache
/etc/rc0.d/k16apache
/etc/rc1.d/k16apache
/etc/rc2.d/k16apache
/etc/rc3.d/s50apache
/etc/rcS.d/k16apache
```

2. Configure the Apache Web Server `/etc/apache/httpd.conf` configuration file.

```
# cp /etc/apache/httpd.conf-example /etc/apache/httpd.conf
# vi /etc/apache/httpd.conf
```

a. Locate the following line in the file:

```
#ServerName new.host.name
```

b. Remove the # from `ServerName` and change `new.host.name` to the name you will use for the shared address.

```
ServerName web-server
```

c. Locate the following lines:

```
DocumentRoot "/var/apache/htdocs"
<Directory "/var/apache/htdocs">
ScriptAlias /cgi-bin/ "/var/apache/cgi-bin/"
<Directory "/var/apache/cgi-bin">
```

d. Change the entries to point to the global directories:

```
DocumentRoot "/global/web/htdocs"
<Directory "/global/web/htdocs">
ScriptAlias /cgi-bin/ "/global/web/cgi-bin/"
<Directory "/global/web/cgi-bin">
```

e. Save the file and exit.

3. Verify that the logical host name has been added to your name service database.

Note – To avoid any failures because of name service lookup, also verify that the shared address name is present in the server's and client's `/etc/hosts` file.

4. *On one node of the cluster only,* copy the default `htdocs` and `cgi-bin` directories to the global location as shown.

```
# cp -rp /var/apache/htdocs /global/web
# cp -rp /var/apache/cgi-bin /global/web
```

Testing the Application Software Installation

Test the server on each node before configuring the data service resources.

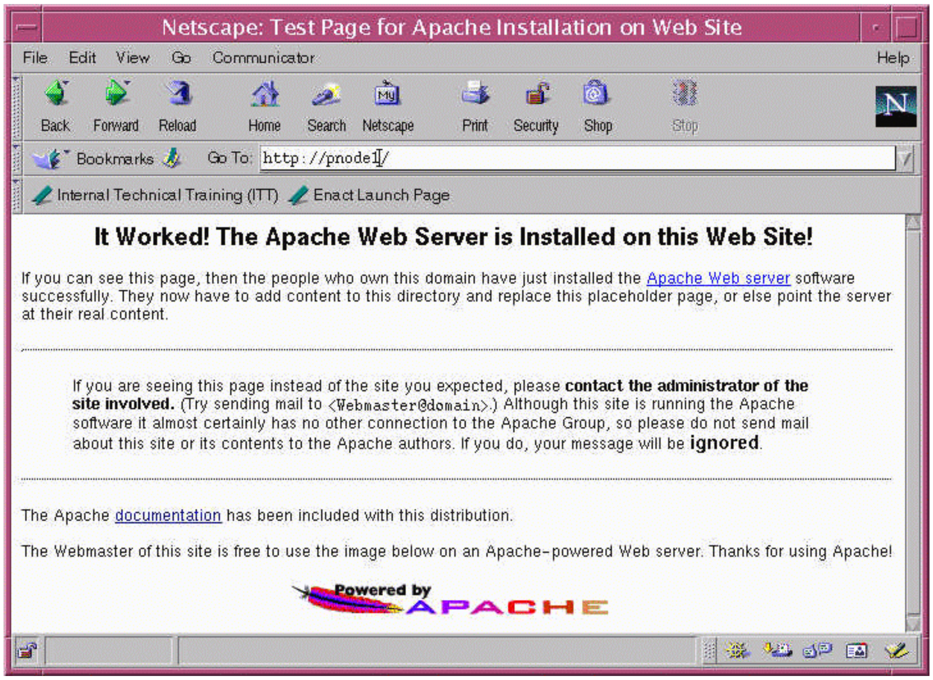
1. Start the server.

```
# /usr/apache/bin/apachectl start
```

2. Verify that the server is running.

```
# ps -ef | grep httpd
nobody   490   488  0 15:36:27 ?        0:00 /usr/apache/bin/httpd -f
/etc/apache/httpd.conf
    root   488     1  0 15:36:26 ?        0:00 /usr/apache/bin/httpd -f
/etc/apache/httpd.conf
  nobody   489   488  0 15:36:27 ?        0:00 /usr/apache/bin/httpd -f
/etc/apache/httpd.conf
  nobody   491   488  0 15:36:27 ?        0:00 /usr/apache/bin/httpd -f
/etc/apache/httpd.conf
  nobody   492   488  0 15:36:27 ?        0:00 /usr/apache/bin/httpd -f
/etc/apache/httpd.conf
  nobody   493   488  0 15:36:27 ?        0:00 /usr/apache/bin/httpd -f
/etc/apache/httpd.conf
```

3. Connect to the server from the Web browser on your console server. Use `http://`*`nodename`* where *`nodename`* is the name of one of your cluster nodes.



It Worked! The Apache Web Server is Installed on this Web Site!

If you can see this page, then the people who own this domain have just installed the Apache Web server software successfully. They now have to add content to this directory and replace this placeholder page, or else point the server at their real content.

> If you are seeing this page instead of the site you expected, please **contact the administrator of the site involved.** (Try sending mail to <Webmaster@domain>.) Although this site is running the Apache software it almost certainly has no other connection to the Apache Group, so please do not send mail about this site or its contents to the Apache authors. If you do, your message will be **ignored**.

The Apache documentation has been included with this distribution.

The Webmaster of this site is free to use the image below on an Apache-powered Web server. Thanks for using Apache!

Figure 11-3 Apache Server Test Page

4. Stop the Apache Web Server.

```
# /usr/apache/bin/apachectl stop
```

5. Verify that the server has stopped.

```
# ps -ef | grep httpd
root   8394  8393  0 17:11:14 pts/6    0:00 grep httpd
```

Installing the Sun Cluster Data Service Software Packages

The data service packages were installed as part of your initial Sun Cluster installation. Use the following command to verify the installation:

```
# pkginfo -l SUNWscapc
   PKGINST:  SUNWscapc
      NAME:  Sun Cluster Apache Web Server Component
  CATEGORY:  application
      ARCH:  sparc
   VERSION:  3.0.0,REV=2000.10.01.01.00
   BASEDIR:  /opt
    VENDOR:  Sun Microsystems, Inc.
      DESC:  Sun Cluster Apache web server data service
    PSTAMP:  octavia20001001013147
  INSTDATE:  Dec 05 2000 13:48
   HOTLINE:  Please contact your local service provider
    STATUS:  completely installed
     FILES:     13 installed pathnames
                 3 directories
                10 executables
              1756 blocks used (approx)
```

Registering and Configuring the Data Service

The following steps complete the installation and configuration process.

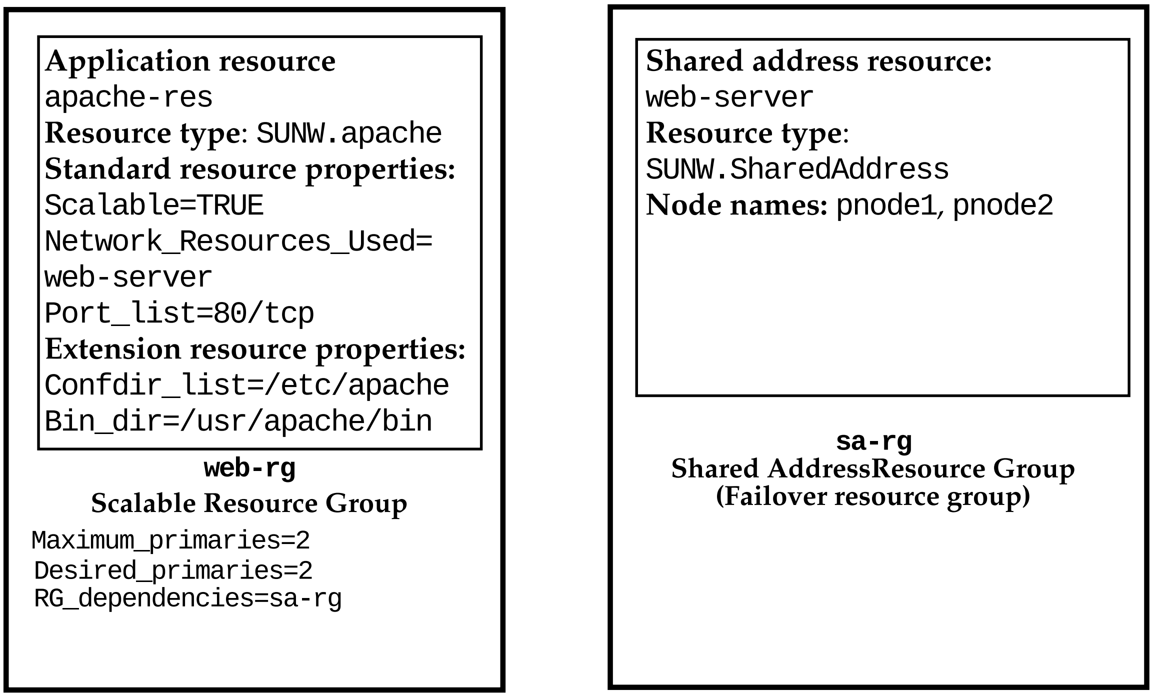
1. Become superuser on a node in the cluster.
2. Verify that all nodes in the cluster are up and functional:

```
# scstat
```

3. Create a plan for the resources that will be in the failover resource group.

Planning the Resources and Resource Group

Since this is a scalable data service, plan for one resource group for the application and one resource group for the shared address. Configuring HAStorage is optional but highly recommended for scalable services. It is not included in the following examples for simplicity. However, the procedure for adding HAStorage is identical to the example for the NFS failover data service.



Application resource
apache-res
Resource type: SUNW.apache
Standard resource properties:
Scalable=TRUE
Network_Resources_Used=
web-server
Port_list=80/tcp
Extension resource properties:
Confdir_list=/etc/apache
Bin_dir=/usr/apache/bin

web-rg
Scalable Resource Group
Maximum_primaries=2
Desired_primaries=2
RG_dependencies=sa-rg

Shared address resource:
web-server
Resource type:
SUNW.SharedAddress
Node names: pnode1, pnode2

sa-rg
Shared AddressResource Group
(Failover resource group)

Figure 11-4 Apache Web Server Sample Resource Plan

Scalable Resource Group Properties

As with resources, resource groups have standard properties that in most cases default to reasonable values. However, scalable resource groups require that the following resource group properties be set.

`-y Maximum_primaries=`*m*
Specifies the maximum number of active primary nodes allowed for this resource group. If you do not assign a value to this property, the default is 1.

`-y Desired_primaries=`*n*
Specifies the desired number of active primary nodes allowed for this resource group. If you do not assign a value to this property, the default is 1.

`-y RG_dependencies=`*resource-group-name*
Identifies the resource group that contains the shared address resource on which the resource group being created depends.

Scalable Resource Properties

The following lists the required Apache application standard and extension resource properties. The `Port_list` property is not technically required if you use the default ports, but it is required if you use more than one port or a non-standard port.

`-y Network_resources_used=`*network-resource*`, …`
Specifies a comma-separated list of network resource names that identify the shared addresses used by the data service.

`-y Port_list=port-`*number/protocol*`, …`
Specifies a comma-separated list of port numbers and protocol to be used. Defaults to `80/tcp`.

`-y Scalable=`
Specifies a required parameter for scalable services. It must be set to `True`.

`-x Confdir_list=config-`*directory*,…
Specifies a comma-separated list of the locations of the Apache configuration files. This is a required extension property.

`-x Bin_dir=`*bin-directory*
Specifies the location where the Apache binaries are installed. This is a required extension property.

The *Data Services Installation and Configuration Guide* contains the full list of standard and extension properties available for the Apache data service. In general, these properties contain appropriate defaults and do not need modification.

4. Register the resource type for the Apache data service.

```
# scrgadm -a -t SUNW.apache
```

5. Create a failover resource group to hold the shared network address.

```
# scrgadm -a -g sa-rg -h pnode1,pnode2
```

6. Add a network resource to the failover resource group.

```
# scrgadm -a -S -g sa-rg -l web-server
```

7. Create a scalable resource group to run on all nodes of the cluster.

```
# scrgadm -a -g web-rg -y Maximum_primaries=2 \
-y Desired_primaries=2 -y RG_dependencies=sa-rg
```

8. Create an application resource in the scalable resource group.

```
# scrgadm -a -j apache-res -g web-rg \
-t SUNW.apache -x Confdir_list=/etc/apache -x Bin_dir=/usr/apache/bin \
-y Scalable=TRUE -y Network_Resources_Used=web-server
```

9. Bring the failover resource group online.

```
# scswitch -Z -g sa-rg
```

10. Bring the scalable resource group online.

```
# scswitch -Z -g web-rg
```

11. Verify that the data service is online.

```
# scstat -g
```

12. Connect to the server from the Web browser on your console server using `http://web-server`. The test page should be identical to the test you ran to test the application installation.

Advanced Resource Commands

The following examples demonstrate how to use the `scrgadm` command to perform advanced operations on resource groups, resources, and data service fault monitors.

Advanced Resource Group Operations

Use the following commands to:

- Shut down a resource group

  ```
  # scrgadm -F -g nfs-rg
  ```

- Turn on a resource group

  ```
  # scrgadm -Z -g nfs-rg
  ```

- Restart a resource group

  ```
  # scrgadm -R -h node,node -g nfsrg
  ```

- Evacuate all resources and resource groups from a node

  ```
  # scrgadm -S -h node
  ```

Advanced Resource Operations

Use the following commands to:

- Disable a resource and its fault monitor

  ```
  # scrgadm -n -j nfs-res
  ```

- Enable a resource and its fault monitor

  ```
  # scrgadm -e -j nfs-res
  ```

Advanced Fault Monitor Operations

Use the following commands to:

- Disable the fault monitor for a resource

```
# scrgadm -n -M -j nfs-res
```

- Enable a resource fault monitor

```
# scrgadm -e -M -j nfs-res
```

Note – You should not manually kill any resource group operations that are underway. Operations such as `scswitch` must be allowed to complete.

Exercise: Installing and Configuring Sun Cluster HA for NFS

In this exercise, you complete the following tasks:

- Prepare for Sun Cluster HA for NFS data service registration and configuration
- Register and configure the Sun Cluster HA for NFS data service
- Verify that the Sun Cluster HA for NFS data service is registered and functional
- Verify that the Sun Cluster HA for NFS file system is mounted and exported
- Verify that clients can access Sun Cluster HA for NFS file systems
- Switch the Sun Cluster HA for NFS data services from one server to another

Tasks

The following tasks are explained in this section:

- Preparing for Sun Cluster HA for NFS registration and configuration
- Registering and configuring the Sun Cluster HA for NFS data service
- Verifying access by NFS clients
- Observing Sun Cluster HA for NFS failover behavior

Note – During this exercise, when you see italicized names such as *`IPaddress`*, *`enclosure_name`*, *`node1`*, or *`clustername`* imbedded in a command string, substitute the names appropriate for your cluster.

Task – Preparing for Sun Cluster HA for NFS Data Service Configuration

In earlier exercises, you created the global file system for NFS. Confirm that this file system is available and ready to configure for Sun Cluster HA for NFS.

1. Log in to your console server and log in as root from the Cluster Console.
2. Verify that your cluster is active.

   ```
   # scstat -p
   ```

3. Verify that the global file system is mounted and ready for use.

   ```
   # df -k
   ```

4. Verify the `hosts` line in the `/etc/nsswitch.conf` file. If necessary, correct it to read:

   ```
   hosts: cluster files nis
   ```

5. Install the Sun Cluster HA for NFS data service software package by running `scinstall` on each node. Use option 4.

   ```
   # scinstall
   ```

6. Add an entry to the `/etc/hosts` file on each cluster node and the administrative workstation for the logical host name resource *`clustername`*`-nfs`. Substitute the IP address supplied by your instructor.

   ```
   clustername-nfs IP_address
   ```

Perform the remaining steps on just one node of the cluster.

7. Create the administrative directory that will contain the `dfstab.nfs-res` file for the NFS resource.

   ```
   # cd /global/nfs
   # mkdir admin
   # cd admin
   # mkdir SUNW.nfs
   ```

8. Create the dfstab.nfs-res file in the /global/nfs/admin/SUNW.nfs directory. Add the entry to share /global/nfs/data.

```
# cd SUNW.nfs
# vi dfstab.nfs-res
```

```
share -F nfs -o rw -d”Home Dirs” /global/nfs/data
```

9. Create the directory specified in the dfstab.nfs-res file.

```
# cd /global/nfs
# mkdir /global/nfs/data
# chmod 777 /global/nfs/data
```

Note – You are changing the mode of the home directory only for the purposes of this lab. In practice, you would be more specific about the share options in the dfstab.nfs-res file.

Task – Registering and Configuring the Sun Cluster HA for NFS Data Service

Perform the following steps to register the Sun Cluster HA for NFS data service.

1. Register the NFS and HAStorage resource types.

```
# scrgadm -a -t SUNW.nfs
# scrgadm -a -t SUNW.HAStorage
```

Note – The `scrgadm` command only produces output to the screen if there are errors executing the commands. If the command executes and returns, that indicates successful creation.

2. Create the failover resource group.

```
# scrgadm -a -g nfs-rg -h node1,node2 \
-y Pathprefix=/global/nfs/admin
```

3. Add the logical host name resource to the resource group.

```
# scrgadm -a -L -g nfs-rg -l clustername-nfs
```

4. Create the `SUNW.HAStorage` resource.

```
# scrgadm -a -j has-res -g nfs-rg -t SUNW.HAStorage \
-x ServicePaths=/global/nfs -x AffinityOn=True
```

5. Create the `SUNW.nfs` resource.

```
# scrgadm -a -j nfs-res -g nfs-rg -t SUNW.nfs \
-y Resource_dependencies=has-res
```

6. Enable the resources and the resource monitors, manage the resource group, and switch the resource group into the online state.

```
# scswitch -Z -g nfs-rg
```

7. Verify that the data service is online.

```
# scstat -g
```

Task – Verifying Access by NFS Clients

Verify that NFS clients can access the Sun Cluster HA for NFS file system.

1. On the administration workstation, verify that you can access the cluster file system.

   ```
   # ls -l /net/clustername-nfs/global/nfs/data
   ```

2. On the administration workstation, copy the `Scripts/test.nfs` file into the root directory.

When this script is running, it creates and writes to an NFS-mounted file system. It also displays the time to standard output (`stdout`). This script helps to time how long the NFS data service is interrupted during switchovers and takeovers.

Task – Observing Sun Cluster HA for NFS Failover Behavior

Now that the Sun Cluster HA for NFS environment is working properly, test its high-availability operation.

1. On the administration workstation, start the `test.nfs` script.
2. On one node of the cluster, determine the name of the node currently hosting the Sun Cluster HA for NFS service.
3. On one node of the cluster, use the `scswitch` command to transfer control of the NFS service from one HA server to the other.

   ```
   # scswitch -z -h dest-node -g nfs-rg
   ```

Substitute the name of your offline node for *`dest-node`*.

4. Observe the messages displayed by the `test.nfs` script.
5. How long was the Sun Cluster HA for NFS data service interrupted during the switchover from one physical host to another?

   ____

6. Use the `mount` and `share` commands on both nodes to verify which file systems they are now mounting and exporting.

   ____

   ____

   ____

7. Use the `ifconfig` command on both nodes to observe the multiple IP addresses (physical and logical) configured on the same physical network interface.

   ```
   # ifconfig -a
   ```

Exercise: Installing and Configuring Sun Cluster Scalable Service for Apache

In this exercise, you complete the following tasks:

- Prepare for Sun Cluster HA for Apache data service registration and configuration
- Register and configure the Sun Cluster HA for Apache data service
- Verify that the Sun Cluster HA for Apache data service is registered and functional
- Verify that clients can access the Apache Web Server
- Verify the functionality of the scalable service

Tasks

The following tasks are explained in this section:

- Preparing for Sun Cluster HA for Apache registration and configuration
- Registering and configuring the Sun Cluster HA for Apache data service
- Verifying Apache Web Server access and scalable functionality

Note – During this exercise, when you see italicized names such as *`IPaddress`*, *`enclosure_name`*, *`node1`*, or *`clustername`* imbedded in a command string, substitute the names appropriate for your cluster.

Task – Preparing for HA-Apache Data Service Configuration

On each node of the cluster, perform the following steps:

1. Install the Sun Cluster Apache data service software package by running `scinstall` on each node. Use option 4.

```
# scinstall
```

2. Disable the `START` and `STOP` run control scripts that were just installed as part of the `SUNWapchr` package.

   This step is necessary because Sun Cluster HA for Apache starts and stops the Apache application once you have configured the data service. Perform the following three steps:

   a. List the Apache run control scripts.

```
# ls -1 /etc/rc?.d/*apache
/etc/rc0.d/K16apache
/etc/rc1.d/K16apache
/etc/rc2.d/K16apache
/etc/rc3.d/S50apache
/etc/rcS.d/K16apache
```

   b. Rename the Apache run control scripts.

```
# mv /etc/rc0.d/K16apache /etc/rc0.d/k16apache
# mv /etc/rc1.d/K16apache /etc/rc1.d/k16apache
# mv /etc/rc2.d/K16apache /etc/rc2.d/k16apache
# mv /etc/rc3.d/S50apache /etc/rc3.d/s50apache
# mv /etc/rcS.d/K16apache /etc/rcS.d/k16apache
```

   c. Verify that all the Apache-related scripts have been renamed.

```
# ls -1 /etc/rc?.d/*apache
/etc/rc0.d/k16apache
/etc/rc1.d/k16apache
/etc/rc2.d/k16apache
/etc/rc3.d/s50apache
/etc/rcS.d/k16apache
```

3. Create an entry in `/etc/hosts` for the shared address you will be configuring with the Apache Web server. In addition, create the entry on the administration workstation. Substitute the IP address supplied by your instructor.

   ```
   clustername-web IP_address
   ```

4. Copy the sample `/etc/apache/httpd.conf-example` to `/etc/apache/httpd.conf.`

```
# cp /etc/apache/httpd.conf-example /etc/apache/httpd.conf
```

5. Edit the `/etc/apache/httpd.conf` file and change the following entries as shown.

   From:

   ```
   #ServerName new.host.name
   DocumentRoot "/var/apache/htdocs"
   <Directory "/var/apache/htdocs">
   ScriptAlias /cgi-bin/ "/var/apache/cgi-bin/"
   <Directory "/var/apache/cgi-bin">
   ```

   To:

   ```
   ServerName clustername-web     (Uncomment the line)
   DocumentRoot “/global/web/htdocs”
   <Directory “/global/web/htdocs”>
   ScriptAlias /cgi-bin/ “/global/web/cgi-bin”
   <Directory "/global/web/cgi-bin">
   ```

On one node of the cluster, perform the following steps:

6. Create directories for the HTML and CGI files.

   ```
   # mkdir /global/web/htdocs
   # mkdir /global/web/cgi-bin
   ```

7. Copy the sample HTML documents to the `htdocs` directory.

   ```
   # cp -rp /var/apache/htdocs /global/web
   # cp -rp /var/apache/cgi-bin /global/web
   ```

8. Copy the file called "test-cluster.cgi" from the classroom server to /global/web/cgi-bin. You use this file to test the scalable service. Make sure that test-cluster.cgi is executable by all users.

```
# chmod 755 /global/web/cgi-bin/test-cluster.cgi
```

Task – Registering and Configuring the Sun Cluster HA for Apache Data Service

1. Register the resource type required for the Apache data service.

   ```
   # scrgadm -a -t SUNW.apache
   ```

2. Create a resource group for the shared address resource. Use the appropriate node names for the -h argument.

   ```
   # scrgadm -a -g sa-rg -h node1,node2
   ```

3. Add the network resource to the resource group.

   ```
   # scrgadm -a -S -g sa-rg -l clustername-web
   ```

4. Create a scalable resource group to run on all nodes of the cluster.

```
# scrgadm -a -g web-rg -y Maximum_primaries=2 \
-y Desired_primaries=2 -y RG_dependencies=sa-rg
```

5. Create an application resource in the scalable resource group.

```
# scrgadm -a -j apache-res -g web-rg \
-t SUNW.apache -x Confdir_list=/etc/apache -x Bin_dir=/usr/apache/bin \
-y Scalable=TRUE -y Network_Resources_Used=clustername-web
```

6. Bring the failover resource group online.

```
# scswitch -Z -g sa-rg
```

7. Bring the scalable resource group online.

```
# scswitch -Z -g web-rg
```

8. Verify that the data service is online.

```
# scstat -g
```

Task – Verifying Apache Web Server Access and Scalable Functionality

1. Connect to the Web server using the browser on the administrator workstation using `http://`*`clustername`*`-web/cgi-bin/test-cluster.cgi`.

2. Repeatedly press the refresh button on the browser. The `test-cluster.cgi script` displays the actual node name that is servicing the request. It may take several iterations before the packet is distributed to a new node.

Exercise Summary

- Experiences
- Interpretations
- Conclusions
- Applications

Check Your Progress

Before continuing on to the next module, check that you are able to accomplish or answer the following:

❑ Describe the function of Sun Cluster data services

❑ Distinguish between highly available and scalable data services

❑ Describe the operation of data service fault monitors

❑ Configure the Sun Cluster HA for NFS failover data service

❑ Configure the Apache Web Server scalable data service

❑ Switch resource groups between nodes

❑ Monitor resource groups

Think Beyond

How difficult is it to configure additional data services while the cluster is in operation?

Module 12

Sun Cluster Administration Workshop

Objectives

Upon completion of this module, you should be able to:

- Install and configure the Sun Cluster host software
- Install and configure Veritas Volume Manager
- Create NAFO groups
- Install, configure, and test the Sun Cluster HA for NFS failover data service
- Install, configure, and test the Sun Cluster HA for Apache scalable data service

Relevance

Discussion – The following questions are relevant to understanding the content of this module:

- What is the best way to become comfortable with new knowledge?

Additional Resources

Additional resources – The following references can provide additional information on the topics described in this module:

- *Sun Cluster 3.0 Installation Guide*, part number 806-1419
- *Sun Cluster 3.0 Hardware Guide*, part number 806-1420
- *Sun Cluster 3.0 Data Services Installation and Configuration Guide*, part number 806-1421
- *Sun Cluster 3.0 Data Service Developers Guide*, part number 806-1422
- *Sun Cluster 3.0 System Administration Guide*, part number 806-1423
- *Sun Cluster 3.0 Concepts*, part number 806-1424
- *Sun Cluster 3.0 Error Message Guide*, part number 806-1426
- *Sun Cluster 3.0 Release Notes*, part number 806-1428

Sun Cluster Administration Workshop Introduction

This workshop provides practice using skills learned from this course and its prerequisites. It is designed to take an entire day to complete. To assure adequate time, it is best to begin the software installation steps (steps 1 and 2) at the end of the day preceding the workshop.

Note – It is less important to complete all the steps in the workshop than it is to understand the steps you do complete. Accordingly, work at a pace that promotes comprehension for your team, and complete as many steps as time permits.

This workshop also requires students to work in teams. Each team of two people works on a single cluster system.

As you work though the steps, take time to understand the solution to each step before proceeding to the next. The goal of this workshop is to promote understanding of administration concepts through practice.

The Configuration Steps section and the Configuration Solutions section are complementary resources. The Configuration Steps describe what to accomplish on which systems, but these steps do not tell you how. The Configuration Solutions section offers general advice on how to accomplish the required tasks.

Consider dividing responsibility for the information these sections contain among members of your team.

Note – During this exercise, when you see italicized names such as *`IPaddress`*, *`enclosure_name`*, *`node1`*, or *`clustername`* imbedded in a command string, substitute the names appropriate for your cluster.

System Preparation

Your lab must have a JumpStart server available to re-install the Solaris 8 10/00 Operating Environment on your cluster nodes.

Note – Leave your administration console systems as they are.

Configuration Steps

Perform the following steps to complete the Sun Cluster administration workshop:

1. Halt both of your cluster nodes and perform a JumpStart boot on them.
2. After the cluster nodes complete their JumpStart, verify that their boot disks have the following configuration:

Slice	Use	Size
0	`/`	remainder
1	`swap`	Twice memory
2	`backup`	Entire disk
4	`/global`	100 Mbytes
7	`unassigned`	10 Mbytes

3. Setup the `root` environment on both cluster hosts including:
   a. The `/.profile` files
   b. The `/.rhosts` files
   c. The `/etc/default/login` files
   d. The `/etc/hosts` files
4. Install the Sun Cluster software on Node 1 of your cluster.
5. Install the Sun Cluster software on Node 2 of your cluster.
6. Select a suitable quorum disk on Node 1.
7. On Node 1, use the `scsetup` utility to configure a quorum device and reset the `installmode` flag.
8. Configure the `/etc/inet/ntp.conf` file on both nodes.
9. Verify the general status and operation of your cluster.
10. Disable the Dynamic Multipathing feature on both nodes before installing the Veritas Volume Manager.

11. Install the Veritas Volume Manager software on both nodes.
12. Install any Veritas Volume Manager patches.
13. Verify that the `vxio` driver major numbers are the same on all nodes.
14. Encapsulate the boot disk on both nodes. This is a complex procedure. Follow the detailed instructions in the Configuration Solutions section.
15. Select two disks in each storage array for use in mirrored volumes. Record the logical paths to the disks (`c3t4d0`).

	Array A	Array B
`nfsdg` **disks:**	*`disk01`*: ________	*`disk02`*: ________
`webdg` **disks:**	*`disk03`*: ________	*`disk04`*: ________

16. Configure demonstration disk groups and volumes for use later.
    a. On Node 1, create the `nfsdg` disk group with two disks.
    b. On Node 1, create a 500-Mbyte mirrored volume in the `nfsdg` disk group that is named `vol-01`.
    c. On Node 2, create the `webdg` disk group with two disks.
    d. On Node 2, create a 500-Mbyte mirrored volume in the `webdg` disk group that is named `vol-01`.
17. Register the demonstration disk groups.
    a. On Node 1, register the `nfsdg` and `webdg` disk groups. Reverse the node list for the `webdg` disk group.
18. Create and mount the `/global/nfs` file system on Node 1.
    a. On Node 1, initialize the `/global/nfs` file system.
    b. On both nodes add `/etc/vfstab` mount entries for the `/global/nfs` file system.
    c. On Node 1, mount the `/global/nfs` file system.

19. Create and mount the `/global/web` file system on Node 2.

    a. On Node 2, initialize the `/global/web` file system.

    b. On both nodes add `/etc/vfstab` mount entries for the `/global/web` file system.

    c. On Node 2, mount the `/global/web` file system.

20. Practice migrating the disk device groups between nodes.

21. Verify the EEPROM `local-mac-address?` variable is set to `false` on both nodes.

22. Create a NAFO group on each node.

23. Install the Sun Cluster HA for NFS data service software on *both* nodes.

24. Verify the `hosts` line in the `/etc/nsswitch.conf` file.

25. Resolve the Sun Cluster HA for NFS logical host name in the `/etc/hosts` files on both cluster nodes.

26. Create the Sun Cluster HA for NFS administrative directory structure and `dfstab.nfs-res` file on Node 1.

27. Create the `/global/nfs/data` directory.

28. From Node 1, create and activate the Sun Cluster HA for NFS data service.

    a. Register the NFS and `HAStorage` resource types.

    b. Create the failover resource group.

    c. Add the logical host name resource to the resource group.

    d. Create the `SUNW.HAStorage` resource.

    e. Create the NFS resource.

    f. Enable the resources and the resource monitors, manage the resource group, and switch the resource group into the online state.

    g. Verify that the data service is online.

29. Verify that clients can access the `/global/nfs/data` file system.
30. Test the `/global/nfs/data` file system while migrating the NFS data service between nodes.
31. Install the Sun Cluster HA for Apache data service software on *both* nodes.
32. Disable the standard Apache run control scripts.
33. Resolve the Sun Cluster HA for Apache logical host name in the `/etc/hosts` files on both cluster nodes.
34. Configure the Apache `httpd.conf` file, and the `htdocs` and `cgi-bin` directories.
    a. Create the `httpd.conf` file.
    b. Edit the `httpd.conf` file and change path names.
    c. Create global `htdocs` and `cgi-bin` directories.
    d. Copy sample files into the `htdocs` and `cgi-bin` directories.
    e. Copy the `test-cluster.cgi` file into the `cgi-bin` directory.
35. From Node 1, create and activate the Sun Cluster HA for Apache data service.
    a. Register the resource type required for the Apache data service.
    b. Create a resource group for the shared address resource. Use the appropriate node names for the `-h` argument.
    c. Add the network resource to the resource group.
    d. Create a scalable resource group to run on all nodes of the cluster.
    e. Create an application resource in the scalable resource group.
    f. Bring the failover resource group online.
    g. Bring the scalable resource group online.
    h. Verify that the data service is online.
36. Test the Apache installation by using a Web browser on the administration workstation and connecting to: `http://`*`clustername`*`-web/cgi-bin/test-cluster.cgi`.

Configuration Solutions

The following section provides configuration advice for completing the workshop. The information in this section is intended to support the tasks described in the Configuration Steps section of the workshop.

Note – The step numbers that follow match those in the Configuration Steps section.

1. Installing the Solaris Operating Environment both nodes might take as long as 60 minutes, depending on network traffic in the lab.

   It is a good idea to start the JumpStart operation late Thursday afternoon so the systems are ready Friday morning.

   Make sure the JumpStart operation has started to load software packages before leaving.

   Initiate the JumpStart operation on each node using:

   ```
   ok boot net - install
   ```

2. Use the `prtconf` command to asses memory size and the `format` utility to examine the boot disk partitions.

   Slice 2 (backup) must always remain the entire disk. Never alter slice 2.

   The `/globaldevices` partition must be at least 100 Mbytes in size.

   The small slice 7 partition for Solstice DiskSuite state databases (replicas) is not needed unless you are planning to install Solstice DiskSuite instead of Veritas Volume Manager.

   For this exercise, the swap partition size is not particularly important. In a production environment, it could be critical.

3. Setting up the environment early saves a lot of frustration looking for files and manual pages.

   The `/.profile` files should contain the following variables:

   ```
   PATH=$PATH:/usr/cluster/bin:/etc/vx/bin

   MANPATH=$MANPATH:/usr/cluster/man:/usr/share/man:/opt/V
   RTSvxvm/man:/opt/VRTSvmsa/man

   TERM=vt220
   export PATH MANPATH TERM
   ```

On both nodes, create a `.rhosts` file in the root directory. Edit the file and add a single line with a plus (+) sign.

On both cluster nodes, edit the `/etc/default/login` file and comment out the `CONSOLE=/dev/console` line.

Edit the `/etc/hosts` file on the administrative workstation and all cluster nodes and add the IP addresses and host names of the administrative workstation and cluster nodes.

Note – The `.rhosts` and `/etc/default/login` file modifications used here can be a security risk in some environments. They are used here to simplify some of the lab exercises.

4. Run `scinstall` from the `SunCluster_3.0/Tools` directory.
   - Use option 1, *Establish a new cluster.*
   - Do not use DES authentication.
   - Accept default transport addresses.
   - Use the default `/globaldevices` file system (it must already exist).
   - Accept the automatic reboot as there are no Sun Cluster patches.
5. Use the same process as the first node except for:
   - Use option 2, *Add this machine as a node in an established cluster.*
6. Use the `scdidadm -l` command on Node 1 to assist in selecting a quorum disk.

   Do not use a local disk such as the system boot disk.

   You can use the quorum disk for other purposes such as data storage.

   The number of quorum disks usually equals the number of cluster nodes minus 1 (N-1).
7. The `scsetup` utility behaves differently the first time it is run on a newly installed cluster.

   The quorum device path must be in a Disk ID device format (`d4`).

8. You must remove all private node name entries that do not apply to your cluster. For a two-node cluster remove the following:

```
peer clusternode3-priv
peer clusternode4-priv
peer clusternode5-priv
peer clusternode6-priv
peer clusternode7-priv
peer clusternode8-priv
```

9. Use the following commands to verify general cluster status and operation:

   - `scstat -q`
   - `scdidadm -L`
   - `scconf -p`
   - `sccheck`
   - `scshutdown -y -g 15`

10. Creating the links as shown, does not actually disable DMP but instead, prevents it from being enabled during the Veritas Volume Manager installation.

```
    # mkdir /dev/vx
    # ln -s /dev/dsk /dev/vx/dmp
    # ln -s /dev/rdsk /dev/vx/rdmp
```

11. Do not install all of the packages; they are not needed. Use the following command:

```
# pkgadd -d . VRTSvmdev VRTSvmman VRTSvxvm
```

12. As of January 2001, the only Veritas Volume Manager patch for the Solaris 8 10/00 release is 110259-01.

    This patches the Veritas `vxio` and `vxdmp` drivers.

13. Use the following command sequence to verify the vxio driver major numbers on all nodes:

    ```
    # grep vxio /etc/name_to_major
    ```

    If the numbers are not the same, edit the /etc/name_to_major file on one of the nodes and modify the major number so they match.

    Before you modify the vxio major number on a node, you must first make sure the new number is not already in use on that node. If it is, you have to select a unique and unused number for both nodes.

    Normally, you would reboot your systems after change major numbers but that is done in the next step of this procedure.

14. Perform the following steps on both nodes to encapsulate the system boot disks, preserve the /global file system, and reminor the rootdg disk group numbers.

    a. Run vxinstall on *both* nodes to encapsulate their boot disks.

       1. Select Custom Installation (**2**).
       2. Answer yes (**y**) to Encapsulate Boot Disk.
       3. Enter a unique disk name on each node (rootdisk1, rootdisk2)
       4. Select Leave these disk alone (4) until you get to the final summary of your choices.
       5. Reply yes (**y**) to the boot disk encapsulation.
       6. Reply no (**n**) to Shutdown and reboot now.

    b. On Node 1, change the /global mount entry in vfstab to use the logical device paths that were used in the original /globaldevices mount entry.

       After modification, the change on Node 1 should look similar to the following:

```
/dev/dsk/c0t0d0s4 /dev/rdsk/c0t0d0s4 /global/.devices/node@1 ufs 2 no global
```

c. On Node 2, change the `/global` mount entry in `vfstab` to use the logical device paths that were used in the original `/globaldevices` mount entry.

The changes on Node 2 should look similar to the following:

```
/dev/dsk/c0t0d0s4 /dev/rdsk/c0t0d0s4 /global/.devices/node@2 ufs 2 no global
```

d. Shut down the cluster with **`scshutdown -y -g 15`**.

e. Boot Node 1 in non-cluster mode (`ok` **`boot -x`**).

Note – The Veritas Volume Manager software initiates a reboot after it finishes encapsulating the boot disk.

f. Boot Node 2 in non-cluster mode. During the automatic reboot, bypass any `/global` `fsck` errors by pressing Control-D.

One node will successfully mount its `/global` file system.

g. Unmount the one successful `/global` file system. This should be on Node 1.

```
# umount /global/.devices/node@1
```

h. Check the `rootdg` minor numbers on each node and compare them.

```
# ls -l /dev/vx/dsk/rootdg
total 0
brw------- 1 root root 171,5 Dec 21 16:47 rootdisk24vol
brw------- 1 root root 171,6 Dec 21 16:47 rootdisk27vol
brw------- 1 root root 171,0 Dec 21 16:47 rootvol
brw------- 1 root root 171,7 Dec 21 16:47 swapvol
```

i. Reminor the `rootdg` disk group on Node 1.

```
# vxdg reminor rootdg 100
```

j. Reminor the `rootdg` disk group on Node 2.

```
# vxdg reminor rootdg 200
```

k. Verify the root disk volume minor numbers are unique on each node.

```
# ls -l /dev/vx/dsk/rootdg
total 0
brw------- 1 root root 55,100 Apr 4 10:48 rootdiska3vol
brw------- 1 root root 55,101 Apr 4 10:48 rootdiska7vol
brw------- 1 root root 55,  0 Mar 30 16:37 rootvol
brw------- 1 root root 55,  7 Mar 30 16:37 swapvol
```

Note – The `rootvol` and `swapvol` minor numbers are automatically renumbered after a reboot.

l. Shut down the cluster and reboot each node in cluster mode. During the reboot. the following error message is displayed:

```
VxVM starting special volumes ( swapvol )...
/dev/vx/dsk/swapvol: No such device or address
```

Caution – If your boot disk had a separated `/var/` or `/usr` partition before encapsulation, you must perform additional steps before rebooting. Consult Appendix B of the *Sun Cluster 3.0 Installation Guide* for additional information.

15. Make sure the disks for each disk group are in different storage arrays. Mirroring across storage arrays is general cluster requirement.

16. Use the following command summary as a guideline for creating the demonstration disk groups and volumes.

```
# vxdiskadd disk01 disk02   (nfsdg)
# vxassist -g nfsdg make vol-01 500m layout=mirror
# vxdiskadd disk03 disk04   (webdg)
# vxassist -g webdg make vol-01 500m layout=mirror
```

17. Use the following command summary as a guideline to register the new disk groups so they become disk device groups.

```
# scconf -a -D type=vxvm,name=nfsdg,nodelist=node1:node2
# scconf -a -D type=vxvm,name=webdg,nodelist=node2:node1
```

18. Use the following command summary as a guideline to create and mount the `/global/nfs` file system on Node 1.

```
# newfs /dev/vx/rdsk/nfsdg/vol-01
# mkdir /global/nfs    (on both nodes)
# vi /etc/vfstab       (on both nodes)
/dev/vx/dsk/nfsdg/vol-01 /dev/vx/rdsk/nfsdg/vol-01 \
/global/nfs  ufs 2 yes global,logging
# mount /global/nfs
```

19. Use the following command summary as a guideline to create and mount the `/global/web` file system on Node 2.

```
# newfs /dev/vx/rdsk/webdg/vol-01
# mkdir /global/web    (on both nodes)
# vi /etc/vfstab       (on both nodes)
/dev/vx/dsk/webdg/vol-01 /dev/vx/rdsk/webdg/vol-01 \
/global/web ufs 2 yes global,logging
# mount /global/web
```

20. Use the `scswitch` command as shown to migrate the disk device groups between nodes. You can run the commands from either node.

```
# scswitch -z -D nfsdg -h node2
# scswitch -z -D webdg -h node1
# scswitch -z -D nfsdg -h node1
# scswitch -z -D webdg -h node2
```

21. Use the following command summary to verify the EEPROM `local-mac-address?` variable is set to `false` on both nodes and, if necessary, change the value and reboot the nodes.

```
# eeprom | grep mac
# eeprom local-mac-address?=false
# init 6
```

22. Use the following example to create a NAFO group on each node.

```
# pnmset
In the following, you will be prompted to do configuration
for network adapter failover
Do you want to continue ... [y/n]: y
How many NAFO groups to configure [1]: 1
Enter NAFO group number [0]: 2
Enter space-separated list of adapters in nafo2: qfe0 qfe1
Checking configuration of nafo2:
Testing active adapter qfe0...
Testing adapter qfe1...
NAFO configuration completed
```

23. Install the Sun Cluster HA for NFS data service software on *both* nodes by running the `scinstall` utility. Use option 4.

    The `scinstall` utility is on both node in the `/usr/cluster/bin` directory.

24. Verify the `hosts` line in the `/etc/nsswitch.conf` file. If necessary, correct it to read:

```
hosts: cluster files nis
```

25. Add an entry to the `/etc/hosts` file on each cluster node and the administrative workstation for the logical host name resource *clustername*-`nfs`. Substitute the IP address supplied by your instructor.

    clustername-nfs *IP_address*

26. Use the following command sequence to create the administrative directory and the `dfstab.nfs-res` file for the NFS resource.

```
# cd /global/nfs
# mkdir admin
# cd admin
# mkdir SUNW.nfs
# cd SUNW.nfs
# vi dfstab.nfs-res
share -F nfs -o rw -d ”Home Dirs” /global/nfs/data
```

27. Use the following commands to create the /global/nfs/data directory.

```
# cd /global/nfs
# mkdir /global/nfs/data
# chmod 777 /global/nfs/data
```

28. On Node 1, use the following command sequence to activate the Sun Cluster HA for NFS data service.

    a. Register the NFS and HAStorage resource types.

```
# scrgadm -a -t SUNW.nfs
# scrgadm -a -t SUNW.HAStorage
```

    b. Create the failover resource group.

```
# scrgadm -a -g nfs-rg -h node1,node2 \
-y Pathprefix=/global/nfs/admin
```

    c. Add the logical host name resource to the resource group.

```
# scrgadm -a -L -g nfs-rg -l clustername-nfs
```

    d. Create the SUNW.HAStorage resource.

```
# scrgadm -a -j has-res -g nfs-rg -t SUNW.HAStorage
\
-x ServicePaths=/global/nfs -x AffinityOn=True
```

    e. Create the SUNW.nfs resource.

```
# scrgadm -a -j nfs-res -g nfs-rg -t SUNW.nfs \
-y Resource_dependencies=has-res
```

    f. Enable the resources and the resource monitors, manage the resource group, and switch the resource group into the online state.

```
# scswitch -Z -g nfs-rg
```

    g. Verify that the data service is online.

```
# scstat -g
```

29. Use the following command from the administration workstation to verify client access to the /global/nfs/data file system.

    ```
    # ls -l /net/clustername-nfs/global/nfs/data
    ```

30. Run the Scripts/test.nfs file on the administration workstation from the root directory while switching the NFS data service between nodes.

    ```
    # scswitch -z -h node2 -g nfs-rg
    # scswitch -z -h node1 -g nfs-rg
    ```

31. Install the Sun Cluster HA for Apache data service software on *both* nodes by running the scinstall utility. Use option 4.

    The scinstall utility is on both node in the /usr/cluster/bin directory.

32. Use the following command sequence to disable the standard Apache run control scripts. Apache startup must be controlled by the data service software and not the system startup scripts.

    a. List the Apache run control scripts.

    ```
    # ls -1 /etc/rc?.d/*apache
    /etc/rc0.d/K16apache
    /etc/rc1.d/K16apache
    /etc/rc2.d/K16apache
    /etc/rc3.d/S50apache
    /etc/rcS.d/K16apache
    ```

    b. Rename the Apache run control scripts.

    ```
    # mv /etc/rc0.d/K16apache /etc/rc0.d/k16apache
    # mv /etc/rc1.d/K16apache /etc/rc1.d/k16apache
    # mv /etc/rc2.d/K16apache /etc/rc2.d/k16apache
    # mv /etc/rc3.d/S50apache /etc/rc3.d/s50apache
    # mv /etc/rcS.d/K16apache /etc/rcS.d/k16apache
    ```

    c. Verify that all the Apache-related scripts have been renamed.

    ```
    # ls -1 /etc/rc?.d/*apache
    /etc/rc0.d/k16apache
    /etc/rc1.d/k16apache
    /etc/rc2.d/k16apache
    /etc/rc3.d/s50apache
    /etc/rcS.d/k16apache
    ```

33. Create an entry in /etc/hosts for the shared address you will be configuring with the Apache Web server. In addition, create the entry on the administration workstation. Substitute the IP address supplied by your instructor.

    ```
    clustername-web IP_address
    ```

34. Use the following command sequence to configure the Apache application files.

    ```
    # cp /etc/apache/httpd.conf-example \
    /etc/apache/httpd.conf

    # vi /etc/apache/httpd.conf
    ```

    Change from:

    ```
    #ServerName new.host.name
    DocumentRoot "/var/apache/htdocs"
    <Directory "/var/apache/htdocs">
    ScriptAlias /cgi-bin/ "/var/apache/cgi-bin/"
    <Directory "/var/apache/cgi-bin">
    ```

    To:

    ```
    ServerName clustername-web     (Uncomment the line)
    DocumentRoot “/global/web/htdocs”
    <Directory “/global/web/htdocs”>
    ScriptAlias /cgi-bin/ “/global/web/cgi-bin”
    <Directory "/global/web/cgi-bin">
    ```

    ```
    # mkdir /global/web/htdocs
    # mkdir /global/web/cgi-bin

    # cp ./test-cluster.cgi /global/web/cgi-bin
    # chmod 755 /global/web/cgi-bin/test-cluster.cgi
    ```

35. From Node 1, create and activate the Sun Cluster HA for Apache data service.

    a. Register the resource type required for the Apache data service.

```
# scrgadm -a -t SUNW.apache
```

    b. Create a resource group for the shared address resource. Use the appropriate node names for the `-h` argument.

```
# scrgadm -a -g sa-rg -h node1,node2
```

    c. Add the network resource to the resource group.

```
# scrgadm -a -S -g sa-rg -l clustername-web
```

    d. Create a scalable resource group to run on all nodes of the cluster.

```
# scrgadm -a -g web-rg -y Maximum_primaries=2\
-y Desired_primaries=2 -y RG_dependencies=sa-rg
```

    e. Create an application resource in the scalable resource group.

```
# scrgadm -a -j apache-res -g web-rg \
-t SUNW.apache -x Confdir_list=/etc/apache \
-x Bin_dir=/usr/apache/bin \
-y Scalable=TRUE \
-y Network_Resources_Used=clustername-web
```

    f. Bring the failover resource group online.

```
# scswitch -Z -g sa-rg
```

    g. Bring the scalable resource group online.

```
# scswitch -Z -g web-rg
```

    h. Verify that the data service is online.

```
# scstat -g
```

36. Perform the following steps to test the Apache installation:

    a. Connect to the Web server using the browser on the administrator workstation using `http://`*`clustername`*`-web/cgi-bin/test-cluster.cgi.`

    b. Repeatedly hit the refresh button on the browser. The `test-cluster.cgi script` displays the actual node name which is servicing the request. It may take several iterations before the packet is distributed to a new node.

Exercise Summary

Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercises.

- Experiences
- Interpretations
- Conclusions
- Applications

Check Your Progress

Check that you are able to accomplish or answer the following:

- ❑ Install and configure the Sun Cluster host software
- ❑ Install and configure Veritas Volume Manager
- ❑ Create NAFO groups
- ❑ Install, configure, and test the Sun Cluster HA for NFS failover data service
- ❑ Install, configure, and test the Sun Cluster HA for Apache scalable data service

Think Beyond

How useful might a *cookbook* document be to you in your workplace?

Appendix A

Cluster Configuration Forms

This appendix contains forms that can be used to record cluster configuration information. The following types of worksheets are in this appendix:

- Cluster and Node Names Worksheet
- Cluster Interconnect Worksheet
- Public Networks Worksheet
- Local Devices Worksheet
- Local File System Layout Worksheet
- Disk Device Group Configurations Worksheet
- Volume Manager Configurations Worksheet
- Metadevices Worksheet
- Failover Resource Types Worksheet
- Failover Resource Groups Worksheet
- Network Resource Worksheet
- HA Storage Application Resources Worksheet
- NFS Application Resources Worksheet
- Scalable Resource Types Worksheet
- Scalable Resource Groups Worksheet
- Shared Address Resource Worksheet

Cluster and Node Names Worksheet

Cluster name ______________________________________

Private network IP address ____________________________ (default: `172.16.0.0`)

Private network mask ________________________________ (default: `255.255.0.0`)

Nodes

Node name ____

Private hostname ______________________________________

Node name ____

Private hostname ______________________________________

Node name ____

Private hostname ______________________________________

Node name ____

Private hostname ______________________________________

Node name ____

Private hostname ______________________________________

Node name ____

Private hostname ______________________________________

Node name ____

Private hostname ______________________________________

Node name ____

Private hostname ______________________________________

Cluster Interconnect Worksheet

Adapters **Cabling** **Junctions**

Draw lines between cable endpoints

Node name ___________________

Adapter Name	Transport Type

Node name ___________________

Adapter Name	Transport Type

Node name ___________________

Adapter Name	Transport Type

Node name ___________________

Adapter Name	Transport Type

Junction name ___________________

Junction type ____________________

Port Number	Description (optional)

Junction name ___________________

Junction type ____________________

Port Number	Description (optional)

Public Networks Worksheet

Node name ____________________________

Primary hostname ________________________

Network name ___________________________

Adapter names __________________________

NAFO group number: nafo____

Secondary hostname______________________

Network name ___________________________

Adapter names __________________________

NAFO group number : nafo____

Secondary hostname______________________

Network name ___________________________

Adapter names __________________________

NAFO group number : nafo____

Secondary hostname______________________

Network name ___________________________

Adapter names __________________________

NAFO group number : nafo____

Node name ____________________________

Primary hostname ________________________

Network name ___________________________

Adapter names __________________________

NAFO group number: nafo____

Secondary hostname______________________

Network name ___________________________

Adapter names __________________________

NAFO group number: nafo____

Secondary hostname______________________

Network name ___________________________

Adapter names __________________________

NAFO group number: nafo____

Secondary hostname______________________

Network name ___________________________

Adapter names __________________________

NAFO group number: nafo____

Local Devices Worksheet

Node name ______________________________

Local disks

Disk name ___________________ Size _______ Disk name ___________________ Size _______

Disk name ___________________ Size _______ Disk name ___________________ Size _______

Disk name ___________________ Size _______ Disk name ___________________ Size _______

Disk name ___________________ Size _______ Disk name ___________________ Size _______

Other local devices

Device type _________ Name _____________ Device type _________ Name _____________

Device type _________ Name _____________ Device type _________ Name _____________

Node name ______________________________

Local disks

Disk name ___________________ Size _______ Disk name ___________________ Size _______

Disk name ___________________ Size _______ Disk name ___________________ Size _______

Disk name ___________________ Size _______ Disk name ___________________ Size _______

Disk name ___________________ Size _______ Disk name ___________________ Size _______

Other local devices

Device type _________ Name _____________ Device type _________ Name _____________

Device type _________ Name _____________ Device type _________ Name _____________

Local File System Layout Worksheet

Node name ______________________________

Mirrored root

Volume Name	**Component**	**Component**	**File System**	**Size**
			`/`	
			`/usr`	
			`/var`	
			`/opt`	
			`swap`	
			`/globaldevices`	

Non-mirrored root

Device Name	**File System**	**Size**
	`/`	
	`/usr`	
	`/var`	
	`/opt`	
	`swap`	
	`/globaldevices`	

Disk Device Group Configurations Worksheet

Volume manager: ___________________________

Disk group/diskset name ______________________

Node names (1)______________ (2)______________ (3)______________ (4)______________

(5)______________ (6)______________ (7)______________ (8)______________

Ordered priority? ❏ Yes ❏ No

Maximum number of secondaries ________

Failback? ❏ Yes ❏ No

Disk group/diskset name ______________________

Node names (1)______________ (2)______________ (3)______________ (4)______________

(5)______________ (6)______________ (7)______________ (8)______________

Ordered priority? ❏ Yes ❏ No

Maximum number of secondaries ________

Failback? ❏ Yes ❏ No

Disk group/diskset name ______________________

Node names (1)______________ (2)______________ (3)______________ (4)______________

(5)______________ (6)______________ (7)______________ (8)______________

Ordered priority? ❏ Yes ❏ No

Maximum number of secondaries ________

Failback? ❏ Yes ❏ No

Volume Manager Configurations Worksheet

Volume manager: ___________________________

Name	Type	Component	Component

Metadevices Worksheet

		Metamirrors		Submirrors			Physical Device	
File System	**Metatrans**	**(Data)**	**(Log)**	**(Data)**	**(Log)**	**Hot Spare Pool**	**(Data)**	**(Log)**

Failover Resource Types Worksheet

Indicate the nodes on which the resource type will run
(other than logical host or shared address).

Resource type name ______________________________

Node names ________________ ________________ ________________ ________________
________________ ________________ ________________ ________________

Resource type name ______________________________

Node names ________________ ________________ ________________ ________________
________________ ________________ ________________ ________________

Failover Resource Groups Worksheet

Resource group name ______________________

(*Must be unique within the cluster.)*

Function of this resource group **Contains the NFS resources**

Failback? ❏ Yes ☒ No

(*Will this resource group switch back to the primary node, after the primary node has failed and been restored?)*

Node names (1)_____________ (2)_____________ (3)_____________ (4)_____________
(ordered list)
(5)_____________ (6)_____________ (7)_____________ (8)_____________

(*Indicate the cluster nodes that may host this resource group. The first node in this list should be the primary, with others being the secondaries. The order of the secondaries will indicate preference for becoming primaries.)*

Disk device groups upon which this resource group depends ______________________________

(*If the resources in this resource group need to create files for administrative purposes, include the subdirectory they should use.)*

/global/nfs/admin

Network Resource Worksheet

Resource name ______________________________

Resource group name ______________________________

Resource type:

☒ Logical hostname ❑ Shared address ❑ Data service/other

Hostnames used ______________________________

____

Network name ____________________

Adapter or NAFO group:

Node name	Adapter/NAFO group name

Resource type name ______________________

Dependencies ______________________

Extension properties:

Name	Value

HA Storage Application Resources Worksheet

Resource name ____________________________________

Resource group name ________________________________

Resource type:

❑ Logical hostname ❑ Shared address ☒ Data service/other

Hostnames used ____________________________

____

Network name _________________

Adapter or NAFO group:

Node name	Adapter/NAFO group name

Resource type name SUNW.HAStorage

Dependencies _________________________

Extension properties:

Name	Value

NFS Application Resources Worksheet

Resource name ____________________________________

Resource group name ________________________________

Resource type:

❑ Logical hostname ❑ Shared address ☒ Data service/other

Hostnames used ____________________________

____

Network name _________________

Adapter or NAFO group:

Node name	**Adapter/NAFO group name**

Resource type name SUNW.NFS

Dependencies ________________________

Extension properties:

Name	**Value**

Scalable Resource Types Worksheet

Indicate the nodes on which the resource type will run
(other than logical host or shared address)

Resource type name SUNW.apache

Node names ________ ________ ________ ________
________ ________ ________ ________

Resource type name ________________

Node names ________ ________ ________ ________
________ ________ ________ ________

Scalable Resource Groups Worksheet

Resource group name web-rg

(*Must be unique within the cluster.)*

Function of this resource group Contains the web server resources

Maximum number of primaries ________

Desired number of primaries ________

Failback? ❏ Yes ☒ No

(*Will this resource group switch back to the primary node, after the primary node has failed*

Node names (1)______________ (2)______________ (3)______________ (4)______________
(ordered list)
(5)______________ (6)______________ (7)______________ (8)______________

(*Indicate the cluster nodes that may host this resource group. The first node in this list should be the primary, with others being the secondaries. The order of the secondaries will indicate preference for becoming primaries.)*

Dependencies sa-rg

(*Does this resource depend upon another resource group.)*

Resource group name sa-rg

(*Must be unique within the cluster.)*

Function of this resource group Contains the shared address resources

Maximum number of primaries 1

Desired number of primaries 1

Failback? ❏ Yes ☒ No

(*Will this resource group switch back to the primary node, after the primary node has failed*

Node names (1)______________ (2)______________ (3)______________ (4)______________
(ordered list)
(5)______________ (6)______________ (7)______________ (8)______________

(*Indicate the cluster nodes that may host this resource group. The first node in this list should be the primary, with others being the secondaries. The order of the secondaries will indicate preference for becoming primaries.)*

Dependencies ______________________

Shared Address Resource Worksheet

Resource name ____________________________________

Resource group name ____sa-rg________________________

Resource type:

□ Logical hostname ☒ Shared address □ Data service/other

Hostnames used ____________________________

____

Network name _________________

Adapter or NAFO group:

Node name	Adapter/NAFO group name

Resource type name ____________________

Dependencies ___________________________

Extension properties:

Name	Value

Scalable Application Resource Worksheet

Resource name apache-res

Resource group name web-rg

Resource type:

❑ Logical hostname ❑ Shared address ☒ Data service/other

Hostnames used ______________________________

____

Network name _________________

Adapter or NAFO group:

Node name	Adapter/NAFO group name

Resource type name SUNW.apache

Dependencies ___________________________

Extension properties:

Name	Value
-x Confdir_list	/etc/apache
-x Bin_dir	/usr/apache/bin
-y Scalable	TRUE
-y Network_Resources_Used	web-server

Appendix B

Configuring Multi-Initiator SCSI

This appendix contains information that can be used to configure multi-initiator SCSI storage devices including the Sun StorEdge MultiPack and SunStorEdge D1000 storage arrays.

Multi-Initiator Overview

This section applies only to SCSI storage devices and not to Fibre Channel storage used for the multihost disks.

In a standalone server, the server node controls the SCSI bus activities using the SCSI host adapter circuit connecting this server to a particular SCSI bus. This SCSI host adapter circuit is referred to as the *SCSI initiator*. This circuit initiates all bus activities for this SCSI bus. The default SCSI address of SCSI host adapters in Sun systems is 7.

Cluster configurations share storage between multiple server nodes. When the cluster storage consists of singled-ended or differential SCSI devices, the configuration is referred to as multi-initiator SCSI. As this terminology implies, more than one SCSI initiator exists on the SCSI bus.

The SCSI specification requires that each device on a SCSI bus has a unique SCSI address. (The host adapter is also a device on the SCSI bus.) The default hardware configuration in a multi-initiator environment results in a conflict because all SCSI host adapters default to 7.

To resolve this conflict, on each SCSI bus, leave one of the SCSI host adapters with the SCSI address of 7, and set the other host adapters to unused SCSI addresses. Proper planning dictates that these “unused” SCSI addresses include both currently and eventually unused addresses. An example of addresses unused in the future is the addition of storage by installing new drives into empty drive slots. In most configurations, the available SCSI address for a second host adapter is 6.

You can change the selected SCSI addresses for these host adapters by setting the `scsi-initiator-id` Open Boot PROM (OBP) property. You can set this property globally for a node or on a per-host-adapter basis. Instructions for setting a unique `scsi-initiator-id` for each SCSI host adapter are included in the chapter for each disk enclosure in the *Sun Cluster 3.0 Hardware Guide*.

Installing a StorEdge MultiPack

This section provides the procedure for an initial installation of a StorEdge MultiPack enclosure.

Use this procedure to install a StorEdge MultiPack enclosure in a cluster prior to installing the Solaris Operating Environment and Sun Cluster software. Perform this procedure in conjunction with the procedures in *Sun Cluster 3.0 Installation Guide* and your server hardware manual.

1. Ensure that each device in the SCSI chain has a unique SCSI address.

The default SCSI address for host adapters is 7. Reserve SCSI address 7 for one host adapter in the SCSI chain. This procedure refers to the host adapter you choose for SCSI address 7 as the host adapter on the `second` node. To avoid conflicts, in Step 7 you will change the `scsi-initiator-id` of the remaining host adapter in the SCSI chain to an available SCSI address. This procedure refers to the host adapter with an available SCSI address as the host adapter on the `first` node. Depending on the device and configuration settings of the device, either SCSI address 6 or 8 is usually available.

Caution – Even though a slot in the enclosure might not be in use, you should avoid setting the `scsi-initiator-id` for the first node to the SCSI address for that disk slot. This precaution minimizes future complications if you install additional disk drives.

For more information, see the *OpenBoot 3.x Command Reference Manual* and the labels inside the storage device.

2. Install the host adapters in the nodes that will be connected to the enclosure.

For the procedure on installing host adapters, see the documentation that shipped with your nodes.

3. Connect the cables to the enclosure, as shown in Figure B-1.

Make sure that the *entire* SCSI bus length to each enclosure is less than 6 meters. This measurement includes the cables to both nodes, as well as the bus length internal to each enclosure, node, and host adapter. Refer to the documentation that shipped with the enclosure for other restrictions regarding SCSI operation.

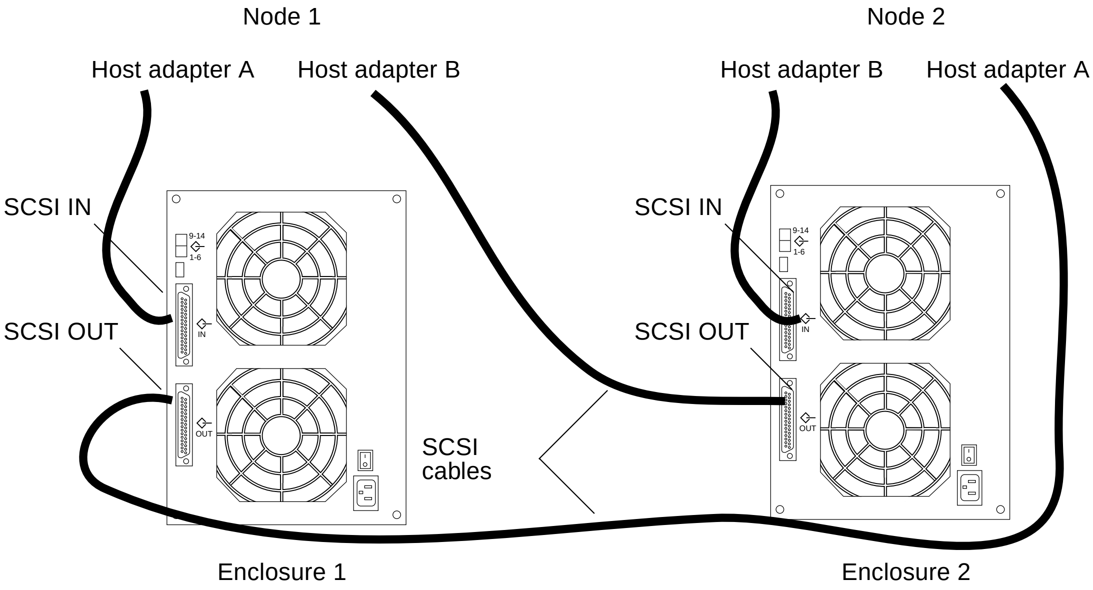


Figure B-1 Example of a StorEdge MultiPack Enclosure Mirrored Pair

4. Connect the AC power cord for each enclosure of the pair to a different power source.
5. Without allowing the node to boot, power on the first node. If necessary, abort the system to continue with OpenBoot PROM Monitor tasks.
6. Find the paths to the host adapters.

```
ok show-disks
```

Identify and record the two controllers that will be connected to the storage devices, and record these paths. Use this information to change the SCSI addresses of these controllers in the `nvramrc` script. Do not include the `/sd` directories in the device paths.

7. Edit the nvramrc script to set the scsi-initiator-id for the host adapter on the first node.

For a list of nvramrc editor and nvedit keystroke commands, see the "NVRAMRC Editor and NVEDIT Keystroke Commands" on page B-11.

The following example sets the scsi-initiator-id to 6. The OpenBoot PROM Monitor prints the line numbers (0:, 1:, and so on).

```
nvedit
0: probe-all
1: cd /sbus@1f,0/
2: 6 encode-int " scsi-initiator-id" property
3: device-end
4: cd /sbus@1f,0/SUNW,fas@2,8800000
5: 6 encode-int " scsi-initiator-id" property
6: device-end
7: install-console
8: banner <Control C>
ok
```

Caution – Insert exactly one space after the first double quote and before scsi-initiator-id.

8. Store the changes.

The changes you make through the nvedit command are done on a temporary copy of the nvramrc script. You can continue to edit this copy without risk. After you complete your edits, save the changes. If you are not sure about the changes, discard them.

- To discard the changes, type:

```
ok nvquit
ok
```

- To store the changes, type:

```
ok nvstore
ok
```

9. Verify the contents of the nvramrc script you created in Step 7.

```
ok printenv nvramrc
nvramrc = probe-all
cd /sbus@1f,0/
6 encode-int " scsi-initiator-id" property
device-end
cd /sbus@1f,0/SUNW,fas@2,8800000
6 encode-int " scsi-initiator-id" property
device-end
install-console
banner
ok
```

10. Instruct the OpenBoot PROM Monitor to use the nvramrc script.

```
ok setenv use-nvramrc? true
use-nvramrc? = true
ok
```

11. Without allowing the node to boot, power on the second node. If necessary, abort the system to continue with OpenBoot PROM Monitor tasks.

12. Verify that the scsi-initiator-id for the host adapter on the second node is set to 7.

```
ok cd /sbus@1f,0/SUNW,fas@2,8800000
ok .properties
scsi-initiator-id        00000007
...
```

13. Continue with the Solaris Operating Environment, Sun Cluster software, and volume management software installation tasks.

For software installation procedures, see *Sun Cluster 3.0 Installation Guide.*

Installing a StorEdge D1000 Disk Array

This section provides the procedure for an initial installation of a StorEdge D1000 disk array.

Use this procedure to install a StorEdge D1000 disk array in a cluster prior to installing the Solaris Operating Environment and Sun Cluster software. Perform this procedure in conjunction with the procedures in the *Sun Cluster 3.0 Installation Guide* and your server hardware manual.

1. Ensure that each device in the SCSI chain has a unique SCSI address.

The default SCSI address for host adapters is 7. Reserve SCSI address 7 for one host adapter in the SCSI chain. This procedure refers to the host adapter you choose for SCSI address 7 as the host adapter on the `second` node. To avoid conflicts, in Step 7 you change the `scsi-initiator-id` of the remaining host adapter in the SCSI chain to an available SCSI address. This procedure refers to the host adapter with an available SCSI address as the host adapter on the `first` node. SCSI address 6 is usually available.

Caution – Even though a slot in the enclosure might not be in use, you should avoid setting the `scsi-initiator-id` for the first node to the SCSI address for that disk slot. This precaution minimizes future complications if you install additional disk drives.

For more information, see the *OpenBoot 3.x Command Reference Manual* and the labels inside the storage device.

2. Install the host adapters in the node that will be connected to the disk array.

For the procedure on installing host adapters, see the documentation that shipped with your nodes.

3. Connect the cables to the disk arrays, as shown in Figure B-2.

Make sure that the *entire* bus length connected to each disk array is less than 25 meters. This measurement includes the cables to both nodes, as well as the bus length internal to each disk array, node, and the host adapter.

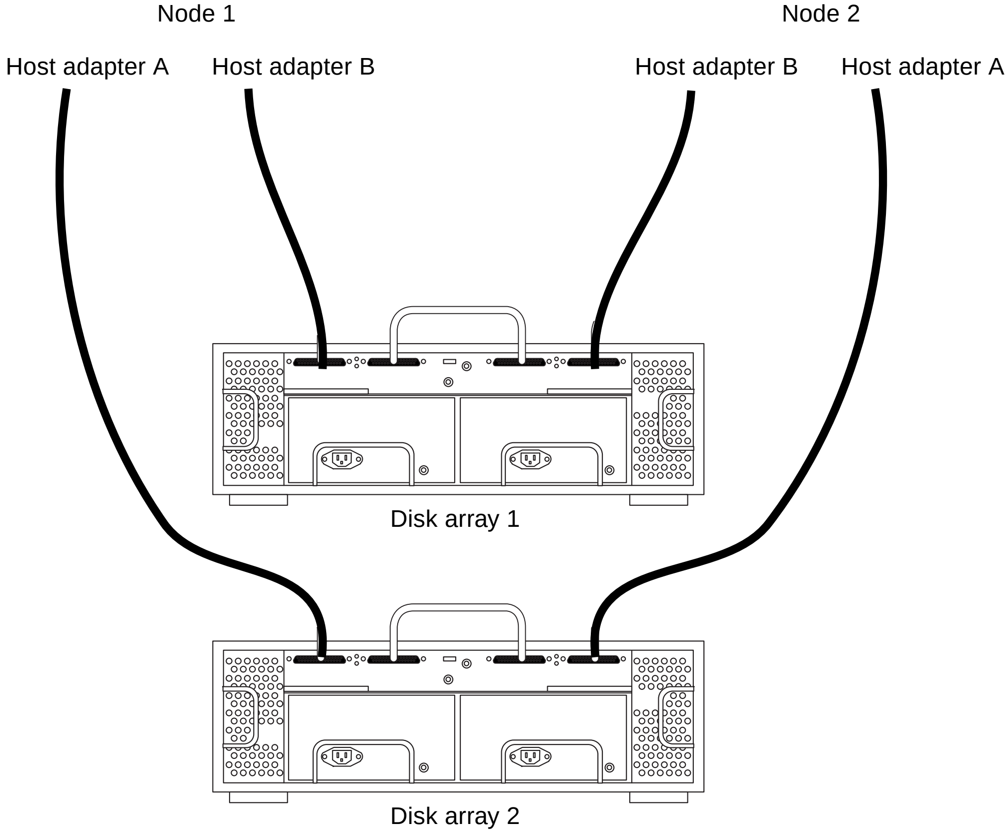


Figure B-2 Example of a StorEdge D1000 Disk Array Mirrored Pair

4. Connect the AC power cord for each disk array of the pair to a different power source.
5. Power on the first node and the disk arrays.
6. Find the paths to the host adapters.

```
ok show-disks
```

Identify and record the two controllers that will be connected to the storage devices and record these paths. Use this information to change the SCSI addresses of these controllers in the `nvramrc` script. Do not include the `/sd` directories in the device paths.

7. Edit the `nvramrc` script to change the `scsi-initiator-id` for the host adapter on the first node.

For a list of `nvramrc` editor and `nvedit` keystroke commands, see the "NVRAMRC Editor and NVEDIT Keystroke Commands" on page B-11.

The following example sets the `scsi-initiator-id` to 6. The OpenBoot PROM Monitor prints the line numbers (`0:`, `1:`, and so on).

```
nvedit
0: probe-all
1: cd /sbus@1f,0/QLGC,isp@3,10000
2: 6 encode-int " scsi-initiator-id" property
3: device-end
4: cd /sbus@1f,0/
5: 6 encode-int " scsi-initiator-id" property
6: device-end
7: install-console
8: banner [Control C]
ok
```

Caution – Insert exactly one space after the first double quote and before `scsi-initiator-id`.

8. Store or discard the changes.

The edits are done on a temporary copy of the `nvramrc` script. You can continue to edit this copy without risk. After you complete your edits, save the changes. If you are not sure about the changes, discard them.

- To store the changes, type:

```
ok nvstore
ok
```

- To discard the changes, type:

```
ok nvquit
ok
```

9. Verify the contents of the `nvramrc` script you created in Step 7.

```
ok printenv nvramrc
nvramrc = probe-all
cd /sbus@1f,0/QLGC,isp@3,10000
6 encode-int " scsi-initiator-id" property
device-end
cd /sbus@1f,0/
6 encode-int " scsi-initiator-id" property
device-end
install-console
banner
ok
```

10. Instruct the OpenBoot PROM Monitor to use the `nvramrc` script.

```
ok setenv use-nvramrc? true
use-nvramrc? = true
ok
```

11. Without allowing the node to boot, power on the second node. If necessary, abort the system to continue with OpenBoot PROM Monitor tasks.

12. Verify that the `scsi-initiator-id` for each host adapter on the second node is set to 7.

```
ok cd /sbus@1f,0/QLGC,isp@3,10000
ok .properties
scsi-initiator-id       00000007
differential
isp-fcode               1.21 95/05/18
device_type             scsi
...
```

13. Continue with the Solaris Operating Environment, Sun Cluster software, and volume management software installation tasks.

For software installation procedures, see the *Sun Cluster 3.0 Installation Guide*.

NVRAMRC Editor and NVEDIT Keystroke Commands

The OBP Monitor builds its own device tree based on the devices attached to the system when the boot sequence is invoked. The OBP Monitor has a set of default aliases for the commonly occurring devices in the system.

An `nvramrc` script contains a series of OBP commands that are executed during the boot sequence. The procedures in this guide assume that this script is empty. If your `nvramrc` script contains data, add the entries to the end of the script. To edit an `nvramrc` script or merge new lines in an `nvramrc` script, you must use `nvedit` editor and `nvedit` keystroke commands.

Table B-1 and Table B-2 list useful `nvedit` editor and `nvedit` keystroke commands. For an entire list of `nvedit` editor and `nvedit` keystroke commands, see the *OpenBoot 3.x Command Reference Manual*.

Table B-1 NVEDIT Editor Commands

Command	Description
`nvedit`	Enter the `nvramc` editor. If the data remains in the temporary buffer from a previous `nvedit` session, resume editing previous contents. Otherwise, read the contents of `nvramrc` into the temporary buffer and begin editing it. This command works on a buffer, and you can save the contents of this buffer by using the `nvstore` command.
`nvstore`	Copy the contents of the temporary buffer to `nvramrc` and discard the contents of the temporary buffer.
`nvquit`	Discard the contents of the temporary buffer, without writing it to `nvramrc`. Prompt for confirmation.
`nvrecover`	Attempts to recover the content of the `nvramrc` if the content was lost as a result of the execution of set-defaults, then enters the `nvramrc` editors as with `nvedit`. This command fails if `nvedit` is executed between the time the content of `nvramrc` was lost and the time the content of the `nvramrc` was executed.
`nvrun`	Executes the contents of the temporary buffer.

Table B-2 NVEDIT Keystroke Commands

Keystroke	Description
`^A`	Move to the beginning of the line.
`^B`	Move backward one character.
`^C`	Exit the script editor.
`^F`	Move forward one character.
`^K`	Delete until end of line.
`^L`	List all lines.
`^N`	Move to the next line of the nvramrc editing buffer.
`^O`	Insert a new line at the cursor position and stay on the current line.
`^P`	Move to the previous line of the `nvramrc` editing buffer.
`^R`	Replace the current line.
`Delete`	Delete previous character.
`Return`	Insert a new line at the cursor position and advance to the next line.

Appendix C

Sun Cluster Administrative Command Summary

This appendix contains brief overviews of the Sun Cluster 3.0 command formats and examples.

Command Overview

Sun Cluster 3.0 comes with a number of commands and utilities that you can use to administer the cluster. Refer the man pages for more detailed information on each of these commands and utilities.

The SC 3.0 administrative commands and utilities are:

- `scinstall` – Installs cluster software and initializes cluster nodes
- `scconf` – Updates the Sun Cluster software configuration
- `scsetup` – Interactive Sun Cluster configuration tool
- `sccheck` – Checks and validates Sun Cluster configuration
- `scstat` – Displays the current status of the Cluster
- `scgdevs` – Administers the global device namespace
- `scdidadm` – Disk ID configuration and administration utility
- `scshutdown` – Utility to shutdown a cluster
- `scrgadm` – Manages registration and configuration of resource types, resources, and resource groups
- `scswitch` – Performs ownership or state changes of Sun Cluster resource groups and disk device groups
- `pnmset` – Sets up and updates the configuration for PNM
- `pnmstat` – Reports status for NAFO groups managed by PNM
- `pnmptor`, `pnmrtop` – Maps pseudo adapter to real adapter name (`pnmptor`) or real adapter to pseudo adapter name (`pnmrtop`) in NAFO groups
- `ccp` – Cluster control panel (administrative console)
- `cconsole`, `ctelnet`, `crlogin` – Multi-window, multi-machine remote administration console tools

The scinstall Utility

Used to install Sun Cluster software and initialize new cluster nodes. When run with no arguments, scinstall(1M) runs in an interactive fashion, presenting the user with menus and prompts to perform its tasks. scinstall can also be run in a non-interactive mode by supplying the proper command line arguments.

The scinstall utility is located in the SunCluster_3_0/Tools directory of the Sun Cluster 3.0 CDROM, or in /usr/cluster/bin on a node where Sun Cluster has already been installed.

All forms of scinstall affect only the node it is run on.

Command Formats

- To run scinstall interactively:

  ```
  scinstall
  ```

- To install Sun Cluster software and/or initialize a node as a new Sun Cluster member:

  ```
  scinstall -i  [-k] [-d <cdimage_dir>] [-s <srvc>,...]
                [-N <clusternode>

                   [-C <clustername>]
                   [-T <authentication_options>]
                   [-G {<special> | <filesystem>} ]
                   [-A <adapter_options>]
                   [-B <junction_options>]
                   [-m <cable_options>]
                   [-w [<netaddr_options>]
                ]
  ```

- To upgrade a Sun Cluster node:

  ```
  scinstall -u  [-d <cdimage_dir>] [-s <srvc>,...]
                [-N <clusternode>
                   [-C <clustername>]
                   [-G{<special> | <filesystem>]
                   [-T authenticaion_options]
                ]
  ```

- To setup a Sun Cluster install server (copies the CD-ROM image to an install directory):

  ```
  scinstall -a <install_dir> [-d <cdimage_dir>]
  ```

- To establish the given nodename as an installation client of the install server:

  ```
  scinstall -c <jumpstart_dir> -h <nodename>
                [-d <cdimage_dir>] [-s <srvc>,...]
                [-N <clusternode>
                   [-C <clustername>]
                   [-G {<special> | <filesystem>}]
                   [-T <authentication_options>]
                   [-A <adapter_options>]
                   [-B <junction_options>]
                   [-m <cable_options>]
                    [-w <netaddr_options>]
  ```

- To print the release and package version information:

  ```
  scinstall -p [-v]
  ```

The scconf Command

Use the scconf(1M) command to manage the cluster software configuration. Use it to add new items to the configuration, change the properties of already configured items and remove items from the configuration. You can run the scconf command from any node that is a member of the cluster (and is usually only run on one node). Items that scconf manages include:

- Quorum options
- Disk device groups (SDS disksets, VxVM disk groups or global raw devices)
- The name of the cluster
- Adding or removing cluster nodes
- Cluster transport adapters, junctions and cables
- Private host names for the nodes (host name used over the cluster transport)
- Node authentication options

When used with the -p option, scconf prints the current cluster configuration.

The scconf command is located in the /usr/cluster/bin directory.

Command Formats

To add or initialize a new item to the software configuration (for example, a new node, transport adapter, junction or cable, quorum device, device group or authentication option):

```
scconf -a [-Hv] [-h <node_options>]
          [-A <adapter_options>]
          [-B <junction_options>]
          [-m <cable_options>]
          [-p <privatehostname_options>]
           [-q <quorum_options>] [-D <devicegroup_options>]
           [-T <authentication_options>]
```

- To change the options for an existing item in the software configuration (that is, the cluster name, a transport adapter, junction or cable, the private hostnames, quorum devices, device groups options and authentication options):

```
scconf -c [-Hv] [-c <cluster_options][-A <adapter_options>]
          [-B <junction_options>] [-m <cable_options>]
          [-P <privatehostname_options>]
          [-q <quorum_options>]
          [-D <devicegroup_options>]
          [-T <authentication_options>]
```

- To remove an item from the software configuration (that is, a node, adapter, junction, cable, quorum device, device group or authentication):

```
scconf -r [-Hv] [-h <node_options>] [-A <adapter_options>]
          [-B <junction_options>]
          [-m <cable_options>] [-q <quorum_options>]
          [-D <devicegroup_options>]
          [-T <authentication_options>]
```

- To print out the current configuration:

```
scconf -p [-Hv}
```

- To print help information about the command options:

```
scconf [-H]
```

Each form of the command accepts a -H option. If present, this option causes scconf to print help information (specific to the form of the command used) and ignore any other options given.

Command Example

Adds a new adapter, hme3, on node venus

```
# scconf -a -A trtype=dlpi,name=hme3,node=venus
```

The scsetup Utility

The scsetup utility is an interactive, menu-driven utility that can perform most of the postinstallation cluster configuration tasks that are handled by scconf. scsetup. It should be run immediately after the cluster software has been installed and all nodes have joined the cluster (scsetup automatically detects the new installation and prompt for the proper quorum configuration information).

You can run the scsetup utility from any node of the cluster.

Command Example

```
# scsetup

 *** Main Menu ***

  Please select from one of the following options:

    1) Quorum
    2) Cluster interconnect
    3) Private hostnames
    4) Device groups
    5) New nodes
    6) Other cluster properties

    ?) Help with menu options
    e) Exit

  Option:
```

The sccheck Utility

The sccheck utility, when run on a node of the cluster (can be run on any node currently in the cluster) checks the validity of the cluster configuration. It checks to make sure that the basic configuration of the cluster is correct and consistent across all nodes.

Command Format

Options can be given to sccheck to invoke a brief check, print verbose messages, suppress warning messages or to perform the check on only certain nodes of the cluster.

```
sccheck [-bvW] [-h <hostlist>
   -b : perform a brief check
   -v : verbose mode
   -W : disable warnings
   -h : Run check on specific hosts
```

Command Example

```
# sccheck -v
vfstab-check: CHECKED - Check for node id
vfstab-check: CHECKED - Check for node id
vfstab-check: CHECKED - Check for
/global/.devices/node@<id>
vfstab-check: CHECKED - Check for mount point
vfstab-check: CHECKED - Check for identical global entries
vfstab-check: CHECKED - Check for option ‘syncdir’
vfstab-check: CHECKED - Check for physical connectivity
vfstab-check: CHECKED - Check for option ‘logging’ for raw
device
vfstab-check: CHECKED - vfstab check completed.
```

The scstat Command

The scstat command prints the current status of various cluster components. You can use it to display the following information:

- The cluster name
- List of cluster members
- Status of each cluster member
- Status of resource groups and resources
- Status of every path in the cluster interconnect
- Status of every disk device group
- Status of every quorum device

Command Format

```
scstat -[-DWgnpq] [-h node]
       -D - Disk group status
       -W - interconnect status,
       -g - resource group status,
       -n node status,
       -p - all components status,
       -q - quorum device status
```

Command Example

```
# scstat -g
Resource Group
  Resource Group Name:                    netscape-rg
  Status
    Node Name:                            venus
    Resource Group State:                 Online

    Node Name:                            mars
    Resource Group State:                 Offline

  Resource
    Resource Name:                        netscape-server
    Status
      Node Name:                          venus
    Resource Monitor Status/Message:    Online -
     SharedAddress online
    Resource State:                     Online


    Node Name:                          mars
    Resource Monitor Status/Message:    Offline -
     SharedAddress offline
    Resource State:                     Offline

Resource Group Name:                    netscape-rg-2
Status
  Node Name:                            venus
  Resource Group State:                 Online

  Node Name:                            mars
  Resource Group State:                 Online

Resource
  Resource Name:                        netscape-res
  Status
    Node Name:                          venus
    Resource Monitor Status/Message:    Online -
       Successfully started Netscape Web Server
       for resource <netscape-res>.
    Resource State:                     Online

    Node Name:                          mars
    Resource Monitor Status/Message:    Online -
       Successfully started Netscape Web Server
       for resource <netscape-res>.
    Resource State:                     Online
```

The scgdevs Utility

Use the scgdevs utility to manage the global devices namespace. The global devices namespace is mounted under /global and consists of a set of symbolic links to physical device files.

By calling scgdevs, an administrator can attach new global devices (such as a tape drive, CD-ROM drive, or disk drive) to the global devices namespace without requiring a system reboot. The drvconfig, disks, tapes, or devlinks commands must be run prior to running scgdevs. Also run devfsadm before running scgdevs.

This command should be run on the node (the node must be a current cluster member) where the new device is being installed.

Command Example

```
# drvconfig
# disks
# devfsadm
# scgdevs
Configuring DID devices
Configuring the /dev/global directory (global devices)
obtaining access to all attached disks
reservation program successfully exiting
```

The scdidadm Command

Use the scdidadm command to administer the DID pseudo device driver. It can create driver configuration files, modify entries in the configuration file, load the current configuration files into the kernel and list the mapping between DID devices and the physical devices.

Run the scdidadm command during cluster startup to initialize the DID driver. It is also used by the scgdevs command to update the DID driver. The primary use of the scdidadm command by administrator is to list the current DID device mappings.

Command Formats

The scdidadm command runs from any node of the cluster.

- To perform a consistency check against the kernel representation of the devices and the physical devices:

  ```
  scdidadm -c
  ```

- To remove all DID references to underlying devices that have been detached from the current node (use after running the normal Solaris device commands to remove references to non-existent devices):

  ```
  scdidadm -C
  ```

- To print out the DID device mappings:

  ```
  scdidadm -l | -L [-h] [-o <fmt>,...] [<path> |
  <DID_instance>]
     fmt can be instance, path, fullpath, host, name,
  fullname, diskid or asciidiskid
  ```

- To reconfigure the DID database to add any new devices (this is performed by scgdevs):

  ```
  scdidadm -r
  ```

- To replace a disk device in the DID database:

  ```
  scdidadm -R <path> | <DID_instance>
  ```

- To run scgdevs on each member of the cluster:

  ```
  scdidadm -S
  ```

- To initialize and load the DID configuration into the kernel:

  `scdidadm -ui`

- To print the version number of this program:

  `scdidadm -v`

Command Example

```
# scdidadm -hlo instance,host,path,name
Instance Host       Physical Path       Pseudo Path
1        venus      /dev/rdsk/c0t0d0    d1
2        venus      /dev/rdsk/c1t2d0    d2
3        venus      /dev/rdsk/c1t3d0    d3
4        venus      /dev/rdsk/c1t4d0    d4
5        venus      /dev/rdsk/c1t5d0    d5
6        venus      /dev/rdsk/c2t2d0    d6
7        venus      /dev/rdsk/c2t3d0    d7
8        venus      /dev/rdsk/c2t4d0    d8
9        venus      /dev/rdsk/c2t5d0    d9
```

The scswitch Command

Use the scswitch command to perform the following tasks:

- Switch resource groups or disk device groups to new primary nodes:
  `scswitch -z ...`
- Bring resource groups or disk device groups online or offline:
  `scswitch -z ...` or `scswitch -m ...`
- Restart a resource group on a node:
  `scswitch -R ...`
- Enable or disable resources and resource monitors:
  `scswitch -e|-n ...`
- Switch resource groups to or from an "unmanaged" state:
  `scswitch -o|-u ...`
- Clear error flags on resources (`scswitch -c ...`)
- Bring resource group offline on all nodes:
  `scswitch -F -g ....`
- Enable all resources, make resource group managed, and bring resource group online on default master(s):
  `scswitch -Z -g [optional]...`

The scswitch command runs on any node of the cluster.

Command Formats

To switch the primary for a resource group (or bring the resource group online if it is not online on any node):

```
scswitch -z -g <resource_grp>[,<resource_grp>...] -h
<node>[,<node>...]
```

To switch the primary for a disk device group (or bring the disk device group online if it is currently offline):

```
scswitch -z -D <device_group_name>[,<device_group_name>...]
-h <node>[,<node>...]
```

To place a resource group offline:

```
scswitch -z -g <resource_grp>[,<resource_grp>...] -h ""
```

To place a disk device group offline (places the disk device group into “maintenance mode”):

```
scswitch -m -D <device_group_name>[,<device_group_name>...]
```

To restart a resource group on a node:

```
scswitch -R -g <resource_group>[,<resource_group>...] -h
<node>[,<node>...]
```

To enable a resource or resource monitor:

```
scswitch -e [-M] -j <resource>[,<resource>...]
```

To disable a resource or resource monitor:

```
scswitch -n [-M] -j <resource>[,<resource>...]
```

To make a resource group “managed” (that is, bring the resource group under cluster control):

```
scswitch -o -g <resource_grp>[,<resource_grp>...]
```

To make a resource group “unmanaged” (that is, take the resource group away from cluster control):

```
scswitch -u -g <resource_grp>[,<resource_grp>...]
```

To clear a resource’s error flags:

```
scswitch -c -h <node>[,<node>...] -j
<resource>[,<resource>...] -f <flag_name>
```

(flag_name can be: BOOT_FAILED, UPDATE_FAILED, INIT_FAILED, FINI_FAILED, or STOP_FAILED)

Before clearing a STOP_FAILED flag, make sure that the data service is actually down.

Note – Only STOP_FAILED is currently implemented.

The scshutdown Command

Use the scshutdown command to shut down the entire cluster. It runs from any active cluster node.

When shutting down the cluster, scshutdown performs the following tasks:

1. Places all of the functioning resource groups on the cluster into an offline state. If any of the transitions fail, scshutdown aborts.
2. Unmounts all of the cluster file systems. If any of the unmounts fail, scshutdown aborts.
3. Shuts down all of the active device services. If any of the transitions fail, scshutdown aborts.
4. Runs /usr/sbin/init 0 on all nodes.

Command Format

- To shutdown all nodes in the cluster:

```
scshutdown [-g <grace_period>] [-y] [<message>]
```

Command Example

Use the following command to shut down a cluster in 60 seconds and issue a warning message.

```
# scshutdown -y -g 60 “Log Off Now”
```

The scrgadm Command

Use the scrgadm command for the following tasks:

- Add, change, or remove resource types
- Create or change the properties of or remove resource groups
- Add or change the properties of or remove resources within resource groups, including logical host name or shared address resources
- Print the properties of resource groups and their resources

The scrgadm command runs on any node that is a member of the cluster.

Command Formats

- To register a resource type:

```
scrgadm -a -t <resource_type_name> [-h
<RT_installed_node_list]
   [-f <registration_file_path>]
```

- To deregister a resource type:

```
scrgadm -r -t <resource_type_name>
```

- To create a new resource group:

```
scrgadm -a -g <RG_name> [-h <nodelist>] [-y
<property=value> [...]]
```

  Use -y Maximum_primaries=n to create a scalable resource group.

- To add a logical host name or shared address resource to a resource group:

```
scrgadm -a -g <RG_name> -l <hostnamelist> [-n
<netiflist>]
```

- To add a resource to a resource group:

```
scrgadm -a -j <resource_name> -t <resource_type_name> -
g <RG_name>
[-y <property=value> [...]] [-x
<extension_property=value> [...]]
```

- To change the properties of a resource group:

  ```
  scrgadm -c -g <RG_name> -y <property=value> [-y
  <property=value>]
  ```

- To change the properties of a resource:

  ```
  scrgadm -c -j <resource_name> [-y <property=value>
  [...]]
     [-x <extension_property=value> [...]]
  ```

- To remove a resource from a resource group:

  ```
  scrgadm -r -j <resource_name>
  ```

  A resource must be disabled (using `scswitch -n`) before it can be removed.

- To remove a resource group:

  ```
  scrgadm -r [-L|-S] -g <RG_name>
  ```

Before removing a resource group, perform the following steps:

1. Place the resource group offline:

   ```
   scswitch -z -g <RG_name> -h ““
   ```

2. Disable all resources:

   ```
   scswitch -n -j <resource_name>
   ```

3. Remove all resources:

   ```
   scrgadm -r -j <resource_name>
   ```

4. Make the resource group unmanaged:

   ```
   scswitch -u -g <RG_name>
   ```

- To print out the resource types, resource groups and resources (and their properties) in the cluster:

  ```
  scrgadm -p[v[v]]
  ```

  The additional `-v` flags provide more verbose output.

The pnmset Utility

Use the pnmset utility to configure NAFO groups on a node. It can be used to:

- Create, change, or remove a NAFO group
- Migrate IP addresses from the active adapter to a configured standby adapter
- Print out the current NAFO group configuration

You can run the pnmset utility interactively or, if backup groups have already been configured, non-interactively.

The pnmset utility only affects the node it is run on. It must be run separately on each node of the cluster.

Command Formats

- To create NAFO groups (initial setup):

```
pnmset [-f <filename>] [-n[-t]] [-v]

Where:
-f <filename> indicates a filename to save or read the
configuration to/from (see pnmconfig(4)). Default is
/etc/cluster/pnmconfig.

-n Do not run interactively, instead read configuration
file for NAFO group information (see pnmconfig(4) for
file format)

-t Do not run a test of the interfaces before
configuring the NAFO groups (only valid with the -n
option)

-v Do not start or restart the pnmd daemon, verify
only. Any new groups will not be active until the
daemon is restarted.
```

- To reconfigure PNM (after the PNM service has already been started):

  ```
  pnmset -c <NAFO_group> -o <subcommand> [<subcommand
  args> ...]

  Subcommands:

  create [<adp1> <adp2> ...] - creates a new NAFO group

  delete - deletes the NAFO group

  add <adp> - adds the specified adapter to the NAFO
  group

  remove <adp> - removes the specified adapter from the
  NAFO group

  switch <adp> - moves the IP addresses from the current
  live adapter to the specified adapter
  ```

- To print out the current NAFO group configuration:

  ```
  pnmset -p
  ```

The pnmstat Command

The pnmstat command reports the current status of the NAFO groups configured on a node. It reports the following information:

- Status of the NAFO groups:
  a. OK - The NAFO group(s) are working
  b. DOUBT - The NAFO group(s) are currently in a transition state. PNM has not determined if the group is healthy or down.
  c. DOWN - The NAFO group is down, no adapters in the group are capable of hosting the configured IP addresses.
- Seconds since last failover
- Currently active adapter

If run without any arguments, pnmstat displays the overall status of PNM on the node. When run with a specific NAFO group (-c) or with the -l option it displays all three statistics.

The pnmstat command reports only on the node it is run on unless the -h option is given, in which case it reports the NAFO group status on the specified host.

Command Formats

- To report the general status of PNM on a node:

  ```
  pnmstat
  ```

- To report the status of all the NAFO groups on a node:

  ```
  pnmstat -l
  ```

- To report the status of a particular NAFO group on a node:

  ```
  pnmstat -c <NAFO_group>
  ```

- To report the status of the NAFO groups on another node:

  ```
  pnmstat -h <host> [-s] [-c <NAFO_group>] [-l]
  ```

  The -s option indicates that the cluster interconnect should be used instead of the public network.

Command Examples

```
# pnmstat
OK

# pnmstat -c nafo0
OK
NEVER
hme0

# pnmstat -l
group   adapters        status  fo_time act_adp
nafo0   hme0            OK      NEVER   hme0

# pnmstat -h venus -c nafo0
OK
NEVER
hme0
```

The pnmptor and pnmrtop Commands

The pnmrtop and pnmptor commands map NAFO group names (pseudo adapter names) to actual network adapter names and vice versa.

These commands only report on NAFO groups that are configured on the local node.

Command Formats and Examples

- To convert a NAFO group name to the name of the currently active adapter:

  ```
  pnmptor <NAFO_group>
  ```

  ```
  # pnmptor nafo0
  hme0
  ```

- To display the NAFO group for a specified network adapter:

  ```
  pnmptor <adp>
  ```

  ```
  # pnmrtop hme0
  nafo0
  ```

Appendix D

Sun Cluster Node Replacement

This appendix describes the process used to replace a node that has suffered a catastrophic failure.

Node Replacement Overview

If a node in the cluster completely fails and requires replacement, it first must be removed from the cluster, and then re-added to the cluster as a new node. You can also use this procedure to reestablish a node that had a complete failure of its boot disk and no mirror or backup was available. The procedure for replacing a failed node is:

1. Remove the failed node from the cluster framework.

   This involves determining and removing all references to the node in the various cluster subsystems (device group access, quorum devices, resource groups, and so on.)

2. Physically replace the failed node.

   Verify that the new node has the proper hardware installed to serve as a replacement for the failed node. This includes the transport adapters, public network interfaces, and storage adapters.

3. Add the new node to the existing cluster.

   This involves configuring the existing cluster to accept the new node into the cluster and performing a standard Sun Cluster installation on the “new” node. After installation, the resource group, quorum, and device group relationships can be reestablished.

Replacement Preparation

To simulate a complete node failure, shut down one of your nodes and initiate a JumpStart operation to reload the Solaris 8 10/00 Operating Environment.

```
ok boot net - install
```

While the JumpStart operation proceeds, you can start the replacement procedure on the remaining active cluster node.

Logically Removing a Failed Node

The removal process is a set of steps in which you remove all references to a node from the existing cluster framework. This allows the cluster to later accept the replacement node as a “new” node, making it easier to keep the framework consistent across all the nodes of the cluster.

Perform the following steps to remove a failed node from the cluster framework.

1. Place the node into a maintenance state

   Use the `scconf` command to place the node into a maintenance state. This removes it from the quorum vote count total, which helps minimize possible quorum loss problems while you work on replacing the node.

   ```
   # scconf -c -q node=node_name,maintstate
   ```

2. Remove the node from all resource groups.

   a. Determine the node ID of the node being removed:

   ```
   # scconf -pv | grep "Node ID"
     (venus) Node ID:                1
     (saturn) Node ID:               2
   ```

   b. Determine the set of resource groups that reference the node ID for the node being removed. The node ID is referred to in the `NetIfList` property of the network resource in the resource group.

```
# scrgadm -pvv | grep -i netiflist | grep "property value"
```

```
(nfs-rg:nfs-server:NetIfList) Res property value: nafo0@1
nafo0@2
```

c. Update the `NetIfList` property for the resources that refer to the node being removed. Change the `netiflist` property to leave off the node being removed.

```
# scrgadm -c -j nfs-server -x netiflist=nafo0@1
```

d. Determine the set of resource groups that refer to the node in its `nodelist` property:

```
# scrgadm -pvv | grep "Res Group Nodelist"
(nfs-rg) Res Group Nodelist:       venus saturn
```

e. Delete the node being removed from the `nodelist` property of the resource groups:

```
# scrgadm -c -g nfs-rg -y nodelist=venus
```

3. Remove the node from all VxVM device group node lists.

   a. Determine the VxVM device groups and node lists.

```
# scconf -p | grep “Device group”
```

   b. Use the `scconf` command to remove the failed node from the device group node lists.

```
# scconf -r -D name=nfs_dg_1,nodelist=saturn
```

4. Remove the failed node from all SDS diskset node lists.

   a. Determine the SDS diskset node lists.

```
# metaset
```

   b. Use the `metaset` command to remove the failed node from the diskset node lists.

```
# metaset -s nfs_ds_1 -f -d -h saturn
```

5. Reset the `localonly` flag for the failed node's boot disk DID instance.

```
# scconf -pvv | grep Local_Disk
(dsk/d10) Device group type:  Local_Disk
(dsk/d1) Device group type:   Local_Disk

# scdidadm -L d10 d1
10   saturn:/dev/rdsk/c0t0d0  /dev/did/rdsk/d10
1    venus:/dev/rdsk/c0t0d0   /dev/did/rdsk/d1

# scconf -c -D name=dsk/d10,localonly=false
```

6. Remove the node from all raw disk devices.

```
# scconf -pvv | grep saturn | grep Device
(dsk/d12)   Device group node list:    saturn
(dsk/d11)   Device group node list:    venus, saturn
(dsk/d10)   Device group node list:    venus, saturn
(dsk/d9)    Device group node list:    venus, saturn

# scconf -r -D name=dsk/d12,nodelist=saturn
# scconf -r -D name=dsk/d11,nodelist=saturn
# scconf -r -D name=dsk/d10,nodelist=saturn
# scconf -r -D name=dsk/d9,nodelist=saturn
```

7. Remove all transport connections to the node.

   a. Use `scsetup` or `scconf` to remove all transport cables that have an endpoint on the node being removed. You can use the `scconf -pvv` command to show which transport cables have endpoints on the node being removed.

```
# scconf -pvv | grep -i "transport cable"
  Transport cable:   saturn:hme0@0   venus:hme1@0  Enabled
  Transport cable:   saturn:hme1@0   venus:hme2@0  Enabled
# scconf -r -m endpoint=saturn:hme0
# scconf -r -m endpoint=saturn:hme1
```

b. Use the scsetup or scconf commands to remove all transport adapters in the node being removed. You can use the scconf -pvv command to show which transport adapters are configured for the node.

```
# scconf -pvv | grep -i "transport adapters"
  (venus) Node transport adapters:   hme1 hme2
  (saturn) Node transport adapters:  hme0 hme1
# scconf -r -A name=hme0,node=saturn
# scconf -r -A name=hme1,node=saturn
```

8. Remove all quorum devices to which the node is connected.

   All quorum devices to which the node being removed is connected must be removed from the cluster configuration. Use the scconf command to remove the quorum device.

   If the quorum device being removed is the last quorum device in a two-node cluster, the cluster must first be placed back into installmode prior to removing the device. This can be done by using scconf.

```
# scconf -pvv | grep -i quorum | grep saturn
  (saturn) Node quorum vote count:           0
  (d2) Quorum device hosts (enabled):        venus saturn
# scconf -r -q globaldev=d2
scconf:  Make sure you are not attempting to remove the
last required quorum device.
scconf:  Failed to remove quorum device (d2) - quorum could
be compromised.
# scconf -c -q installmode
# scconf -r -q globaldev=d2
```

9. Remove the node from the cluster framework

   Finally, remove the node itself from the cluster configuration using the scconf command. At this point, all traces of the node have been removed from the cluster.

```
# scconf -r -h saturn
```

Physically Replacing a Failed Node

Now that the failed node has been completely removed from the cluster framework, the node can be physically replaced with a new server system.

1. Remove the failed system, taking care to properly label any network, storage, and power cables so they can be easily identified and correctly plugged into the replacement system.
2. Verify that the replacement system has the proper network interfaces, storage interfaces, memory, and so on, to adequately serve as a replacement node in the cluster.
3. Connect the cluster interconnect cables, public network cables and storage cables to the replacement system.
4. Connect the power cable to the new system.
5. Power up the replacement system.

Logically Adding a Replacement Node

Add the replacement node to the existing cluster. As far as the existing cluster is concerned, this is to be treated as a brand new node.

1. On one of the existing nodes of the cluster, configure the cluster to accept a new member.

   ```
   # scconf -s -T node=saturn
   ```

2. Install the Solaris 8 Operating Environment on the new node and install the Sun Cluster software. Use option 2, *Add this machine as a node in an established cluster.*

   During the installation, the proper raw device groups should be automatically built along with the cluster interconnect components for the node (adapters and cables).

3. After the new node has joined the cluster, configure any quorum devices required. If the cluster is still in install mode, make sure to reset the quorum configuration using scconf or scsetup.

   ```
   # scconf -a -q globaldev=d2
   # scconf -c -q reset
   ```

4. Add the node to any volume management groups it should be a part of.

   For VxVM: Use scsetup or scconf to add the node into any VxVM disk groups the node should be a part of.

   ```
   # scconf -a -D name=nfs_dg_1,nodelist=saturn
   ```

   For SDS: Use the metaset command to add the host to any disksets it needs to be a member of. *This needs to be done from a node other than the one you are replacing*.

   ```
   # metaset -s nfs_ds_1 -a -h saturn
   ```

5. Create NAFO groups for the node's public network interfaces. Try to mirror the configuration of the failed node.

   ```
   # pnmset -c nafo0 -o create hme0 qfe0
   # pnmset -c nafo1 -o create hme4 qfe1
   ```

6. Add the node to any resource groups that might be mastered by the new node.

   Refer to the list of resource groups from which you removed the node from the `nodelist` parameter. Reconstruct the `nodelist` parameter for these resource groups, adding the node name for the "new" node.

   Also, add the node's NAFO group back into each of the resources from which it was deleted when the node was removed. When rebuilding the `netiflist` for each resource, be sure to use the format of `nafo`*`<NAFO Instance>`*`@`*`<NodeName or NodeID>`*`.` Make sure to assign the proper NAFO group based on the IP address of the resource.

```
# scrgadm -c -g nfs-rg -y nodelist=venus,saturn
# scrgadm -c -j nfs-server -x netiflist=nafo0@venus,nafo0@saturn
```

7. At this point, the replacement node should be fully operational. You can use `scswitch` to switchover any device or resource groups to help balance the cluster.

   ```
   # scswitch -z -D nfs_ds_1 -h saturn
   # scswitch -z -g nfs-rg -h saturn
   ```

Appendix E

Sun Cluster HA for Oracle Installation

This appendix describes the basic process of installing and configuring a highly-available Oracle database.

Installation Process

Following is a summary of a Sun Cluster HA for Oracle data service installation using Oracle 8.1.6. The installation is intended for a two-node cluster, a single pair configuration.

1. Install of SunCluster 3.0 software on both nodes.
2. Install the Sun Cluster HA for Oracle software on both nodes.
3. Install Oracle 8.1.6 in the following configuration:

    a. The Oracle binaries are local to each node under `/oracle`.

    a. The database is in Veritas file system volumes that are globally mounted under `/global/oracle`.

4. Edit the `listener.ora` and `tnsnames.ora` to point to the logical host name (`ford-ora`) you are using for this data service.
5. Register the `SUNW.oracle_server` and `SUNW.oracle_listener` resource types.

```
# scrgadm -a -t SUNW.oracle_server

Jan 12 12:26:14 mustang Cluster.CCR: resource type
SUNW.oracle_server added.

# scrgadm -a -t SUNW.oracle_listener

Jan 12 12:26:41 mustang Cluster.CCR: resource type
SUNW.oracle_listener added.
```

6. Create a blank resource group, oracle-rg with a node list.

```
# scrgadm -a -g oracle-rg -h mustang,cobra

Jan 12 12:28:11 mustang Cluster.CCR: resource group
oracle-rg added.
```

7. Try to register the SUNW.HAStorage resource type. The registration fails because it is a preregistered resource type.

```
# scrgadm -a -t SUNW.HAStorage

SUNW.HAStorage: resource type exists; can’t create
```

8. Add the logical host name resource along with the appropriate NAFO group to use for each node in the node list.

```
# scrgadm -a -L -g oracle-rg -l ford-ora \
-n nafo2@mustang,nafo1@cobra

Jan 12 12:33:46 mustang Cluster.RGM.rgmd: launching
method <hafoip_validate> for resource <ford-ora>,
resource group <oracle-rg>, timeout <300> seconds

Jan 12 12:33:46 mustang Cluster.RGM.rgmd: method
<hafoip_validate> completed successfully for resource
<ford-ora>, resource group <oracle-rg>

Jan 12 12:33:47 mustang Cluster.CCR: resource ford-ora
added.
```

9. Add the SUNW.HAstorage resource type. Specify the Oracle data path and enable affinity so the data storage must follow the data service if just the data service fails over to another node.

```
# scrgadm -a -j hastorage-res -g oracle-rg \
-t SUNW.HAStorage \
-x ServicePaths=/global/oracle \
-x AffinityOn=TRUE

Jan 12 12:36:16 mustang Cluster.RGM.rgmd: launching
method <hastorage_validate> for resource <hastorage-
res>, resource group <oracle-rg>,timeout <300> seconds

Jan 12 12:36:17 mustang Cluster.RGM.rgmd: method
<hastorage_validate> completed successfully for
resource <hastorage-res>, resource group <oracle-rg>

Jan 12 12:36:17 mustang Cluster.CCR: resource
hastorage-res added.
```

10. Create local Oracle alert logs on both nodes.

```
# mkdir /var/oracle (on both nodes)
# touch /var/oracle/alert.log (on both nodes)
```

11. Add the SUNW.oracle_server resource and set standard and extended resource properties.

```
# scrgadm -a -j ora_server_1 -t SUNW.oracle_server \
-g oracle-rg \
-x CONNECT_STRING=scott/tiger \
-y Resource_dependencies=hastorage-res \
-x ORACLE_SID=test \
-x ORACLE_HOME=/oracle \
-x Alert_log_file=/var/oracle/alert.log

Jan 12 12:50:33 mustang Cluster.RGM.rgmd: launching
method <bin/oracle_server_validate> for resource
<ora_server_1>, resource group <oracle-rg>, timeout
<120> seconds

Jan 12 12:50:34 mustang Cluster.RGM.rgmd: method
<bin/oracle_server_validate> completed successfully for
resource <ora_server_1>, resource group <oracle-rg>

Jan 12 12:50:34 mustang Cluster.CCR: resource
ora_server_1 added.
```

12. Add the SUNW.oracle_listener resource and set standard and extended resource properties.

    ```
    # scrgadm -a -j ora_listener_1 \
    -t SUNW.oracle_listener -g oracle-rg \
    -y Resource_dependencies=ora_server_1 \
    -x ORACLE_HOME=/oracle \
    -x LISTENER_NAME=listener \

    Jan 12 12:56:59 mustang Cluster.RGM.rgmd: launching
    method <bin/oracle_listener_validate> for resource
    <ora_listener_1>, resource group <oracle-rg>, timeout
    <60> seconds

    Jan 12 12:56:59 mustang Cluster.RGM.rgmd: method
    <bin/oracle_listener_validate> completed successfully
    for resource <ora_listener_1>, resource group <oracle-
    rg>

    Jan 12 12:57:00 mustang Cluster.CCR: resource
    ora_listener_1 added.
    ```

13. Bring the resource group online.

    ```
    # scswitch -Z -g oracle-rg
    ```

Glossary

A

active server

A node in the Sun Cluster configuration that is providing highly available data services.

administration console

A workstation that is outside the cluster that is used to run cluster administrative software.

API

application program interface

ATM

asynchronous transfer mode

B

backup group

Used by NAFO. A set of network adapters on the same subnet. Adapters within a set provide backup for each other.

C

CCR

(Cluster Configuration Repository) A highly-available, replicated database that can be used to store data for HA data services and other Sun Cluster configuration needs.

Cluster

Two to four nodes configured together to run either parallel database software or highly available data services.

cluster interconnect

The private network interface between cluster nodes.

cluster node

A physical machine that is part of a Sun cluster. Also referred to as a cluster host or cluster server.

cluster quorum

The set of cluster nodes that can participate in the cluster membership.

cluster reconfiguration

An ordered multistep process that is invoked whenever there is a significant change in cluster state. During cluster reconfiguration, the Sun Cluster software coordinates all of the physical hosts that are up and communicating. Those hosts agree on which logical host(s) should be mastered by which physical hosts.

cluster pair topology

Two pairs of Sun Cluster nodes operating under a single cluster administrative framework.

CMM

(Cluster Membership Monitor) The software that maintains a consistent cluster membership roster to avoid database corruption and subsequent transmission of corrupted or inconsistent data to clients. When nodes join or leave the cluster, thus changing the membership, CMM processes on the nodes coordinate global reconfiguration of various system services.

concatenation

A Veritas volume or a Solstice DiskSuite metadevice created by sequentially mapping data storage space to a logical virtual device. Two or more physical components can be concatenated. The data accessed sequentially rather than interlaced (as with stripes).

D

data service

A network service that implements read-write access to disk-based data from clients on a network. NFS is an example of a data service. The data service may be composed of multiple processes that work together.

DES

Data Encryption Standard

DID

Disk ID

disk group

A well defined group of multhost disks that move as a unit between two servers in an HA configuration. This can be either a Solstice DiskSuite diskset or a Veritas Volume Manager disk group.

diskset

See disk group.

DiskSuite state database

A replicated database that is used to store the configuration of metadevices and the state of these metadevices.

DLM

(distributed lock management) Locking software used in a shared disk OPS environment. The DLM enables Oracle processes running on different nodes to synchronize database access. The DLM is designed for high availability; if a process or node crashes, the remaining nodes do not have to be shut down and restarted. A quick reconfiguration of the DLM is performed to recover from such a failure.

DLPI

Data Link Provider Interface

DNS

domain name system

DR

dynamic reconfiguration

DRL

dirty region log

E

EEPROM

electrically erasable programmable read-only memory

F

fault detection

Sun Cluster programs that detect two types of failures. The first type includes low-level failures such as system panics and hardware faults (that is, failures that cause the entire server to be inoperable). These failures can be detected quickly. The second type of failures are related to data service. These types of failures take longer to detect.

fault monitor

A fault daemon and the programs used to probe various parts of data services.

FC-AL

Fibre Channel Arbitrated Loop

FCOM

Fiber Channel Optical Module

FDDI

Fiber Distributed Data Interface

FF

failfast

Fibre Channel connections

Fibre connections connect the nodes with the SPARCstorage Arrays.

G

GBIC

gigabit interface converters

golden mediator

In Solstice DiskSuite configurations, the in-core state of a mediator host set if specific conditions were met when the mediator data was last updated. The state permits take operations to proceed even when a quorum of mediator hosts is not available.

GUI

graphical user interface

H

HA

high availability

HA administrative file system

A special file system created on each logical host when Sun Cluster is first installed. It is used by Sun Cluster and by layered data services to store copies of their administrative data.

HA-NFS

Highly Availability - Network File System

heartbeat

A periodic message sent between the several membership monitors to each other. Lack of a heartbeat after a specified interval and number of retries may trigger a takeover.

HFS

High Sierra file system.

highly available data service

A data service that appears to remain continuously available, despite single-point failures of server hardware or software components.

host

A physical machine that can be part of a Sun cluster. In Sun Cluster documentation, host is synonymous with node.

hot standby server

In an N+1 configuration, the node that is connected to all multihost disks in the cluster. The hot standby is also the administrative node. If one or more active nodes fail, the data services move from the failed node to the hot standby. However, there is no requirement that the +1 node cannot run data services in normal operation.

HTML

Hypertext Markup Language

HTTP

Hypertext Transfer Protocol

I

IDLM

(integrated distributed lock manager) A data access coordination scheme used by newer versions of the Oracle Parallel Server database. Also see Distributed Lock Manager.

I/O

input/output

IP

Internet Protocol

L

LAN

local area network

LANE

local area network emulation

LDAP

Lightweight Directory Access Protocol

local disks

Disks attached to a HA server but not included in a diskset. The local disks contain the Solaris distribution and the Sun Cluster and volume management software packages. Local disks must not contain data exported by the Sun Cluster data service.

logical host

A set of resources that moves as a unit between HA servers. In the current product, the resources include a collection of network host names and their associated IP addresses plus a group of disks (a diskset). Each logical host is mastered by one physical host at a time.

logical host name

The name assigned to one of the logical network interfaces. A logical host name is used by clients on the network to refer to the location of data and data services. The logical host name is the name for a path to the logical host. Because a host may be on multiple networks, there may be multiple logical host names for a single logical host.

logical network interface

In the Internet architecture, a host may have one or more IP addresses. HA configures additional logical network interfaces to establish a mapping between several logical network interfaces and a single physical network interface. This allows a single physical network interface to respond to multiple logical network interfaces. This also enables the IP address to move from one HA server to the other in the event of a takeover or `haswitch(1M)`, without requiring additional hardware interfaces.

M

MAC Address

(media access control address) The worldwide unique identifying address assigned to every Ethernet interface card.

master

The server with exclusive read and write access to a diskset. The current master host for the diskset runs the data service and has the logical IP addresses mapped to its Ethernet address.

mediator

In a dual-string configuration, provides a “third vote” in determining whether access to the metadevice state database replicas can be granted or must be denied. Used only when exactly half of the metadevice state database replicas are accessible.

mediator host

A host that is acting in the capacity of a “third vote” by running the `rpc.metamed(1M)` daemon and that has been added to a diskset.

mediator quorum

The condition achieved when half + 1 of the mediator hosts are accessible.

membership monitor

A process running on all HA servers that monitors the servers. The membership monitor sends and receives heartbeats to its sibling hosts. The monitor can initiate a takeover if the heartbeat stops. It also keeps track of which servers are active.

metadevice

A group of components accessed as a single logical device by concatenating, striping, mirroring, or logging the physical devices. Metadevices are sometimes called pseudo devices.

metadevice state database

Information kept in nonvolatile storage (on disk) for preserving the state and configuration of metadevices.

metadevice state database replica

A copy of the state database. Keeping multiple copies of the state database protects against the loss of state and configuration information. This information is critical to all metadevice operations.

MI

multi-initiator

mirroring

Replicating all writes made to a single logical device (the mirror) to multiple devices (the submirrors), while distributing read operations. This provides data redundancy in the event of a failure.

multihomed host

A host that is on more than one public network.

multihost disk

A disk configured for potential accessibility from multiple servers. Sun Cluster software enables data on a multihost disk to be exported to network clients via a highly available data service.

N

NAFO

network adapter failover

NFS

network file system

NIS

Network Information Service

N-to-N topology

All nodes are directly connected to a set of shared disks.

N+1 topology

Some number (N) active servers and one (+1) hot-standby server. The active servers provide on-going data services and the hot-standby server takes over data service processing if one or more of the active servers fail.

node

A physical machine that can be part of a Sun cluster. In Sun Cluster documentation, it is synonymous with host or node.

nodelock

The mechanism used in greater than two-node clusters using Cluster Volume Manager or Veritas Volume Manager to failure fence failed nodes.

NTP

Network Time Protocol

NVRAM

nonvolatile random access memory

O

OBP

OpenBoot PROM

OGMS

Oracle Group Membership Services

OLAP
online Analytical Processing

OLTP
online transaction processing

OPS
Oracle Parallel Server

P

parallel database
A single database image that can be accessed concurrently through multiple hosts by multiple users.

partial failover
Failing over a subset of logical hosts mastered by a single physical host.

PCI
Peripheral Component Interconnect

PDB
parallel database

PGA
program global area

PMON
process monitor

PNM
public network monitoring

potential master
Any physical host that is capable of mastering a particular logical host.

PROM
programmable read-only memory

primary logical host name
The name by which a logical host is known on the primary public network.

primary physical host name
The name by which a physical host is known on the primary public network.

primary public network
A name used to identify the first public network.

private links

The private network between nodes used to send and receive heartbeats between members of a server set.

Q

quorum device

In SSVM or CVM configurations, the system votes by majority quorum to prevent network partitioning. Since it is impossible for two nodes to vote by majority quorum, a quorum device is included in the voting. This device could be either a controller or a disk.

R

RAID

redundant arrays of independent disks

RDBMS

relational database management system

replica

See metadevice state database replica.

replica quorum

A Solstice DiskSuite concept; the condition achieved when HALF + 1 of the metadevice state database replicas are accessible.

ring topology

One primary and one backup server is specified for each set of data services.

RP

remote probes

ROM

read-only memory

RPC

remote procedure call

S

SAP

service access point

SCI

(Scalable Coherent Interface) A high speed interconnect used as a private network interface.

SCSI

Small Computer Systems Interface

scalable topology

See N-to-N topology.

secondary logical host name

The name by which a logical host is known on a secondary public network.

secondary physical host name

The name by which a physical host is known on a secondary public network.

secondary public network

A name used to identify the second or subsequent public networks.

server

A physical machine that can be part of a Sun cluster. In Sun Cluster documentation, it is synonymous with host or node.

SGA

system global area

Sibling host

One of the physical servers in a symmetric HA configuration.

SIMM

single inline memory module

SNMP

Simple Network Management Protocol

SOC

serial optical channel

Solstice DiskSuite

A software product that provides data reliability through disk striping, concatenation, mirroring, UFS logging, dynamic growth of metadevices and file systems, and metadevice state database replicas.

SPARC

Scalable Processor Architecture

SSP

system service processor

stripe

Similar to concatenation, except the addressing of the component blocks is non-overlapped and interlaced on the slices (partitions), rather than placed sequentially. Striping is used to gain performance. By striping data across disks on separate controllers, multiple controllers can access data simultaneously.

submirror

A metadevice that is part of a mirror. See also mirroring.

Sun Cluster

Software and hardware that enables several machines to act as read-write data servers while acting as backups for each other.

Switch Management Agent

The software component that manages sessions for the SCI and Ethernet links and switches.

switchover

The coordinated moving of a logical host from one operational HA server to the other. A switchover is initiated by an administrator using the `haswitch(1M)` command.

symmetric configuration

A two-node configuration where one server operates as the hot-standby server for the other.

T

takeover

The automatic moving of a logical host from one HA server to another after a failure has been detected. The HA server that has the failure is forced to give up mastery of the logical host.

TC

(terminal concentrator) A device used to enable an administrative workstation to securely communicate with all nodes in the Sun Cluster.

TCP

Transmission Control Protocol

TCP/IP

Transmission Control Protocol/Internet Protocol

TPE

twisted-pair Ethernet

trans device

In Solstice DiskSuite configurations, a pseudo device responsible for managing the contents of a UFS log.

U

UDLM

UNIX distributed lock manager

UDP

User Data Protocol

UFS

UNIX file system

UFS logging

Recording UFS updates to a log (the logging device) before the updates are applied to the UFS (the master device).

UFS logging device

In Solstice DiskSuite configurations, the component of a transdevice that contains the UFS log.

UFS master device

In Solstice DiskSuite configurations, the component of a transdevice that contains the UFS file system.

UPS

Uninterruptable Power Supply

V

VMSA

Volume Manager Storage Administrator. A Veritas graphical storage administration application.

VxFS

Veritas file system