Sun Cluster 3.0 Concepts

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto,, CA 94303-4900
U.S.A. 650-960-1300

Part Number 806-1424
November 2000, Revision A

Please Recycle

Contents

Preface

Sun™ Cluster 3.0 Concepts contains conceptual and reference information for the Sun Cluster software.

This document is intended for experienced system administrators with extensive knowledge of Sun software and hardware. Do not use this document as a planning or presales guide. You should have already determined your system requirements and purchased the appropriate equipment and software before reading this document.

To understand the concepts described in this book, you should have knowledge of the Solaris™ operating environment and expertise with the volume manager software used with Sun Cluster.

Typographic Conventions

Typeface or Symbol	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`
`AaBbCc123`	What you type, when contrasted with on-screen computer output	`% su` `Password:`

Typeface or Symbol	Meaning	Examples
AaBbCc123	Book titles, new words or terms, words to be emphasized	Read Chapter 6 in the *User's Guide.* These are called *class* options. You *must* be superuser to do this.
	Command-line variable; replace with a real name or value	To delete a file, type `rm` *filename.*

Shell Prompts

Shell	Prompt
C shell	*machine_name*%
C shell superuser	*machine_name*#
Bourne shell and Korn shell	$
Bourne shell and Korn shell superuser	#

Related Documentation

Subject	Title	Part Number
Installation	*Sun Cluster 3.0 Installation Guide*	806-1419
Hardware	*Sun Cluster 3.0 Hardware Guide*	806-1420
Data Services	*Sun Cluster 3.0 Data Services Installation and Configuration Guide*	806-1421
API Development	*Sun Cluster 3.0 Data Services Developers' Guide*	806-1422

Subject	Title	Part Number
Administration	*Sun Cluster 3.0 System Administration Guide*	806-1423
Error Messages and Problem Resolution	*Sun Cluster 3.0 Error Messages Manual*	806-1426
Release Notes	*Sun Cluster 3.0 Release Notes*	806-1428

Ordering Sun Documentation

Fatbrain.com, an Internet professional bookstore, stocks select product documentation from Sun Microsystems, Inc. For a list of documents and how to order them, visit the Sun Documentation Center on Fatbrain.com at:

`http://www1.fatbrain.com/documentation/sun`

Accessing Sun Documentation Online

The `docs.sun.com`SM web site enables you to access Sun technical documentation on the Web. You can browse the `docs.sun.com` archive or search for a specific book title or subject at:

`http://docs.sun.com`

Getting Help

If you have problems installing or using Sun Cluster, contact your service provider and provide the following information:

- Your name and email address (if available)
- Your company name, address, and phone number
- The model and serial numbers of your systems
- The release number of the operating environment (for example, Solaris 8)

- The release number of Sun Cluster (for example, Sun Cluster 3.0)

Use the following commands to gather information about each node on your system for your service provider:.

Command	Function
`prtconf -v`	Displays the size of the system memory and reports information about peripheral devices
`psrinfo -v`	Displays information about processors
`showrev --p`	Reports which patches are installed
`prtdiag -v`	Displays system diagnostic information
`scinstall -pv`	Displays Sun Cluster release and package version information

Also have available the contents of the `/var/adm/messages` file.

CHAPTER 1

Introduction and Overview

Sun Cluster 3.0 Concepts provides the conceptual information needed by the primary audience for Sun Cluster documentation. This audience includes:

- Service providers who install and service cluster hardware
- System administrators who install, configure, and administer Sun Cluster software
- Application developers who develop data services for applications not currently included with the Sun Cluster product

This book works with the rest of the Sun Cluster documentation set to provide a complete view of Sun Cluster.

This chapter:

- Provides an introduction and high-level overview of Sun Cluster
- Describes the several viewpoints of the Sun Cluster audience
- Identifies key concepts you need to understand before working with Sun Cluster
- Maps key concepts to the Sun Cluster documentation that includes procedures and related information
- Maps cluster-related tasks to the documentation containing procedures used to accomplish those tasks

Introduction to Sun Cluster

Sun Cluster extends the Solaris™ operating environment into a cluster operating system. A cluster is a collection of loosely coupled computing nodes that provides a single client view of network services or applications, including databases, web services, and file services.

Each cluster node is a standalone server that runs its own processes. These processes can communicate with one another to form what looks like (to a network client) a single system that cooperatively provides applications, system resources, and data to users.

A cluster offers several advantages over traditional single server systems. These advantages include support for highly available and scalable applications, capacity for modular growth, and low entry price compared to traditional hardware fault-tolerant systems.

The goals of Sun Cluster are:

- Reduce or eliminate system downtime because of software or hardware failure
- Ensure availability of data and applications to end users, regardless of the kind of failure that would normally take down a single server system
- Increase application throughput by enabling services to scale to additional processors by adding nodes to the cluster
- Provide enhanced availability of the system by enabling you to perform maintenance without shutting down the entire cluster

High Availability in Sun Cluster

Sun Cluster is designed as a *highly available* (HA) system, that is, a system that provides near continuous access to data and applications.

By contrast, *fault-tolerant* hardware systems provide constant access to data and applications, but at a higher cost because of specialized hardware. Additionally, fault-tolerant systems usually do not account for software failures.

Sun Cluster achieves high availability through a combination of hardware and software. Redundant cluster interconnects, storage, and public networks protect against single points of failure. The cluster software continuously monitors the health of member nodes and prevents failing nodes from participating in the cluster to protect against data corruption. Also, the cluster monitors applications and their dependent system resources, and fails over or restarts applications in case of failures.

Refer to “High Availability FAQ” on page 61 for questions and answers on high availability.

Failover and Scalability in Sun Cluster

Sun Cluster enables you to implement applications on either a *failover* or *scalable* basis. Failover and scalable applications can also run on the same cluster concurrently. In general, a failover application provides high availability (redundancy), whereas a scalable application provides high availability along with increased performance. A single cluster can support both failover and scalable applications.

Failover

Failover is the process by which the cluster automatically relocates an application from a failed primary node to a designated secondary node. With failover, Sun Cluster provides high availability.

When a failover occurs, clients might see a brief interruption in service and might need to reconnect after the failover has finished. However, clients are not aware of the physical server from which they are provided the application and data.

Scalability

While failover is concerned with redundancy, scalability provides constant response time or throughput without regard to load. A scalable application leverages the multiple nodes in a cluster to concurrently run an application, thus providing increased performance. In a scalable configuration, each node in the cluster can provide data and process client requests.

Refer to “Data Services” on page 48 for more specific information on failover and scalable services.

Three Views of Sun Cluster

This section describes three different viewpoints on Sun Cluster and the key concepts and documentation relevant to each viewpoint. These viewpoints come from:

- Hardware installation and service personnel
- System administrators
- Application programmers. Sun Cluster provides a set of highly available data services. These services are applications such as Oracle, Apache Web Server, and DNS that have been configured to become highly available data services running on a cluster. Other applications can be made into highly available data services using the Sun Cluster API. Application programmers can write shell scripts or C programs that use the API.

Hardware Installation and Service Viewpoint

To hardware service people, Sun Cluster looks like a collection of off-the-shelf hardware that includes servers, networks, and storage. These components are all cabled together so that every component has a backup and no single point of failure exists.

Key Concepts – Hardware

Hardware service people need to understand the following cluster concepts.

- Cluster hardware configurations and cabling
- Installing and servicing (adding, removing, replacing):
  - Network interface components (adapters, junctions, cables)
  - Disk interface cards
  - Disk arrays
  - Disk drives
  - The administrative console and the console access device
- Setting up the administrative console and console access device

Suggested Hardware Conceptual References

The following sections contain material relevant to the preceding key concepts:

- "Cluster Nodes" on page 20
- "Multihost Disks" on page 22
- "Local Disks" on page 23
- "Cluster Interconnect" on page 24
- "Public Network Interfaces" on page 25
- "Client Systems" on page 25
- "Administrative Console" on page 25
- "Console Access Devices" on page 26
- "Clustered Pairs Topology" on page 27
- "N+1 (Star) Topology" on page 28

Relevant Sun Cluster Documentation

The following Sun Cluster document includes procedures and information associated with hardware service concepts:

- *Sun Cluster 3.0 Hardware Guide*

System Administrator Viewpoint

To the system administrator, Sun Cluster looks like a set of servers (nodes) cabled together, sharing storage devices. The system administrator sees:

- Specialized cluster software integrated with Solaris software to monitor the connectivity between cluster nodes
- Specialized software to monitor the health of user application programs running on the cluster nodes
- Volume management software to set up and administer disks
- Specialized cluster software to enable all nodes to access all storage devices, even those not directly connected to disks
- Specialized cluster software to enable files to appear on every node as though they were locally attached to that node

Key Concepts – System Administration

System administrators need to understand the following concepts and processes:

- The interaction between the hardware and software components
- The general flow of how to install and configure the cluster including:
  - Installing the Solaris operating environment
  - Installing and configuring Sun Cluster
  - Installing and configuring a volume manager
  - Installing and configuring application software to be cluster ready
  - Installing and configuring Sun Cluster data service software
- Cluster administrative procedures for adding, removing, replacing, and servicing cluster hardware and software components
- Configuration modifications to improve performance

Suggested System Administrator Conceptual References

The following sections contain material relevant to the preceding key concepts:

- “Administrative Interfaces” on page 33
- “High-Availability Framework” on page 34
- “Global Devices” on page 37
- “Disk Device Groups” on page 38
- “Global Namespace” on page 40
- “Cluster File Systems” on page 41
- “Quorum and Quorum Devices” on page 43
- “Volume Managers” on page 48
- “Data Services” on page 48

- "Resources and Resource Types" on page 56
- "Public Network Management (PNM) and Network Adapter Failover (NAFO)" on page 57
- Chapter 4

Relevant Sun Cluster Documentation – System Administrator

The following Sun Cluster documents include procedures and information associated with the system administration concepts:

- *Sun Cluster 3.0 Installation Guide*
- *Sun Cluster 3.0 System Administration Guide*
- *Sun Cluster 3.0 Error Messages Manual*

Application Programmer Viewpoint

Sun Cluster provides several highly available data services for such applications as Oracle, NFS, DNS, iPlanet Web Server, Apache Web Server, and Netscape Directory Server.If a site has to make another application run on a cluster, it can use the Sun Cluster Application Programming Interface (API) and the Data Service Development Library API (DSDL API) to develop the necessary data service software that enables its application to run as a highly available data service on the cluster.

Key Concepts – Application Programmer

Application programmers need to understand the following:

- The characteristics of their application to determine whether it can be made to run as a highly available or scalable data service.
- The Sun Cluster API, DSDL API, and the "generic" data service. Programmers need to determine which tool is most suitable for them to use to write programs or scripts to configure their application for the cluster environment.

Suggested Application Programmer Conceptual References

The following sections contain material relevant to the preceding key concepts:

- "Data Services" on page 48
- "Resources and Resource Types" on page 56
- Chapter 4

Relevant Sun Cluster Documentation – Application Programmer

The following Sun Cluster documents include procedures and information associated with the application programmer concepts:

- *Sun Cluster 3.0 Data Services Developers' Guide*
- *Sun Cluster 3.0 Data Services Installation and Configuration Guide*

Sun Cluster Tasks

All concepts map into tasks and all tasks require some conceptual background. The following table provides a high-level view of the tasks and the documentation that describes task steps. The concepts sections in this book describe how the concepts map to these tasks.

TABLE 1–1 Task Map: Mapping User Tasks to Documentation

To Do This Task...	Use This Documentation...
Install cluster hardware	*Sun Cluster 3.0 Hardware Guide*
Install Solaris software on the cluster	*Sun Cluster 3.0 Installation Guide*
Install Sun™ Management Center software	*Sun Cluster 3.0 Installation Guide*
Install and configure Sun Cluster software	*Sun Cluster 3.0 Installation Guide*
Install and configure volume management software	*Sun Cluster 3.0 Installation Guide* Your volume management documentation
Install and configure Sun Cluster data services	*Sun Cluster 3.0 Data Services Installation and Configuration Guide*
Service cluster hardware	*Sun Cluster 3.0 Hardware Guide*
Administer Sun Cluster software	*Sun Cluster 3.0 System Administration Guide*

TABLE 1–1 Task Map: Mapping User Tasks to Documentation *(continued)*

To Do This Task...	Use This Documentation...
Administer volume management software	*Sun Cluster 3.0 System Administration Guide* and your volume management documentation
Administer application software	Your application documentation
Problem identification and suggested user actions	*Sun Cluster 3.0 Error Messages Manual*
Create a new data service	*Sun Cluster 3.0 Data Services Developers' Guide*

CHAPTER 2

Key Concepts – Hardware Service Providers

This chapter describes the key concepts related to the hardware components of a Sun Cluster configuration.

Sun Cluster Hardware Components

This information is directed primarily toward hardware service providers. These concepts can help service providers understand the relationships between the hardware components before they install, configure, or service cluster hardware. Cluster system administrators might also find this information useful as background to installing, configuring, and administering cluster software.

A cluster is composed of several hardware components including:

- Cluster nodes with local disks (unshared)
- Multihost storage (disks shared between nodes)
- Removable media (tapes and CD-ROM)
- Cluster interconnect
- Public network interfaces
- Client systems
- Administrative console
- Console access devices

Sun Cluster enables you to combine these components into a variety of configurations, described in “Sun Cluster Topologies” on page 27.

The following figure shows a sample cluster configuration.

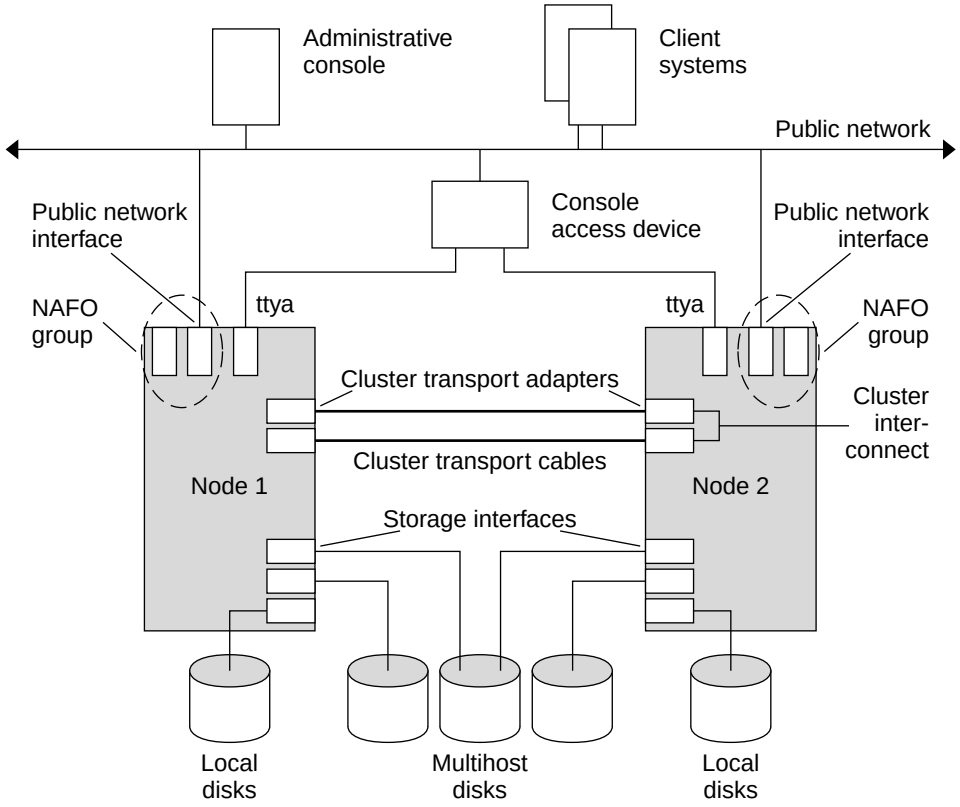

Figure 2–1 Sample Two-Node Cluster Configuration

Cluster Nodes

A cluster node is a machine running both the Solaris operating environment and Sun Cluster software, and is either a current member of the cluster (a *cluster member*), or a potential member. The Sun Cluster software enables you to have from two to eight nodes in a cluster. See “Sun Cluster Topologies” on page 27 for the supported node configurations.

Cluster nodes are generally attached to one or more multihost disks. One scalable services configuration allows nodes to service requests without being directly attached to multihost disks. The nodes not attached to multihost disks use the cluster file system to access the multihost disks.

In parallel database configurations, nodes share concurrent access to all the disks. See “Multihost Disks” on page 22 and Chapter 3 for more information on parallel database configurations.

All nodes in the cluster are grouped under a common name—the cluster name—which is used for accessing and managing the cluster.

Public network adapters attach nodes to the public networks, providing client access to the cluster.

Cluster members communicate with the other nodes in the cluster through one or more physically independent networks referred to as *private networks*. The set of private networks in the cluster is referred to as the *cluster interconnect*.

Every node in the cluster is aware when another node joins or leaves the cluster. Additionally, every node in the cluster is aware of the resources that are running locally as well as the resources that are running on the other cluster nodes.

Configure cluster members with resources (applications, disk storage, and so forth) in such a fashion as to provide failover and/or scalable capabilities.

Make sure that the nodes in the same cluster are of similar processing, memory, and I/O capability to enable failover to occur without significant degradation in performance. Because of the possibility of failover, ensure that every node has enough excess capacity to take on the workload of all nodes for which they are a backup or secondary.

Each node boots its own individual root (/) file system.

Software Components for Cluster Members

To function as a cluster member, the following software must be installed:

- Solaris operating environment
- Sun Cluster
- Volume management (Solstice DiskSuite™ or VERITAS Volume Manager)
- Data service application

One exception is in an Oracle Parallel Server(OPS) configuration that uses hardware redundant array of independent disks (RAID). This configuration does not require a software volume manager such as Solstice DiskSuite or VERITAS Volume Manager to manage the Oracle data.

See the *Sun Cluster 3.0 Installation Guide* for information on how to install the Solaris operating environment, Sun Cluster, and volume management software. See the *Sun Cluster 3.0 Data Services Installation and Configuration Guide* for information on how to install and configure data services.

See Chapter 3 for conceptual information on the preceding software components.

The following figures provides a high-level view of the software components that work together to create the Sun Cluster software environment.

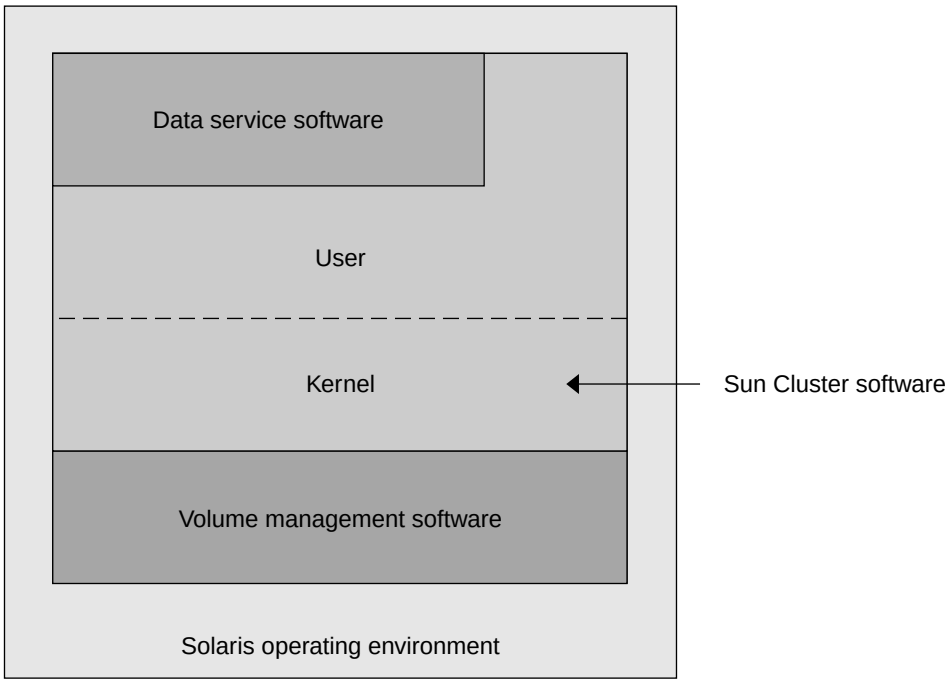

Figure 2–2 High-Level Relationship of Sun Cluster Software Components

See Chapter 4 for questions and answers about cluster members.

Multihost Disks

Sun Cluster requires multihost disk storage: disks that can be connected to more than one node at a time. In the Sun Cluster environment, multihost storage makes disk devices highly available. Disk devices resident on the multihost storage can tolerate single node failures.

The multihost disks store application data and can also store data service binaries and configuration files.

Multihost disks are either accessed globally through a primary node that “masters” the disks, or by direct concurrent access through local paths. The only application that uses direct concurrent access currently is OPS.

Multihost disks protects against node failures. If client requests are accessing the data through one node and it fails, the requests are switched over to use another node that has a direct connection to the same disks.

A volume manager provides for mirrored or RAID-5 configurations for data redundancy of the multihost disks. Currently, Sun Cluster supports Solstice DiskSuite and VERITAS Volume Manager as volume managers, and the RDAC RAID-5 hardware controller in the Sun StorEdge™ A3x00 storage unit.

Combining multihost disks with disk mirroring and striping protects against both node failure and individual disk failure.

See Chapter 4 for questions and answers about multihost storage.

Multi-Initiator SCSI

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 by way of 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 have 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*.

Local Disks

Local disks are the disks that are only connected to a single node. They are, therefore, not protected against node failure (not highly available). However, all disks, including local disks, are included in the global namespace and are configured as *global devices*. Therefore, the disks themselves are visible from all cluster nodes. You can make the file systems on these disks available to other nodes by putting them under a global mount point. If the node that currently has one of these global file systems mounted fails, all nodes lose access to that file system. Using a volume manager enables you

to mirror these disks so that a disk failure cannot cause these file systems to become inaccessible, but volume managers do not protect against node failure.

Removable Media

Removable media such as tape drives and CD-ROM drives are supported in a cluster. In general, you install, configure, and service these devices in the same way as in a non-clustered environment. These devices are configured as global devices in Sun Cluster, so each device can be accessed from any node in the cluster. Refer to the *Sun Cluster 3.0 Hardware Guide* for information on installing and configuring removable media.

Cluster Interconnect

The *cluster interconnect* is the physical configuration of devices used to transfer cluster-private communications and data service communications between cluster nodes. Because the interconnect is used extensively for cluster-private communications, it can limit performance.

Only cluster nodes can be connected to the private interconnect. The Sun Cluster security model assumes that only cluster nodes have physical access to the private interconnect.

All nodes must be connected by the cluster interconnect through at least two redundant *private networks*, or paths, to avoid a single point of failure. You can have several private networks (two to six) between any two nodes. The cluster interconnect consists of three hardware components: adapters, junctions, and cables. Each private network is configured so that it shares no common hardware component with any other private network.

The following list describes each of these hardware components.

- Adapters – The physical network cards that reside in each cluster node. Their names are derived from the name of the product, for example, qfe for Quad FastEthernet. Some adapters have only one physical network connection, but others, like the qfe card, have multiple physical connections. Some also contain both network interfaces and storage interfaces.

  A network card with multiple interfaces could become a single point of failure if the entire card fails. For maximum availability, plan your cluster so that the only path between two nodes does not depend on a single network card.

- Junctions – The switches that reside outside of the cluster nodes. They perform pass-through and switching functions to enable you to connect more than two nodes together. In a two-node cluster, you do not need junctions because the nodes can be directly connected to each other through redundant physical cables

connected to redundant adapters on each node. Greater than two-node configurations generally require junctions.

- Cables – The physical connections that go either between two network adapters or between an adapter and a junction.

See Chapter 4 for questions and answers about the cluster interconnect.

Public Network Interfaces

Clients connect to the cluster through the public network interfaces. Each network adapter card can connect to one or more public networks, depending on whether the card has multiple hardware interfaces. You can set up nodes to include multiple public network interface cards configured so that one card is active and others operate as backups. A subsystem of the Sun Cluster software called “Public Network Management” (PNM) monitors the active interface. If the active adapter fails, Network Adapter Failover (NAFO) software is called to fail over the interface to one of the backup adapters.

No special hardware considerations relate to clustering for the public network interfaces.

See Chapter 4 for questions and answers about public networks.

Client Systems

Client systems include workstations or other servers that access the cluster over the public network. Client-side programs use data or other services provided by server-side applications running on the cluster.

Client systems are not highly available. Data and applications on the cluster are highly available.

See Chapter 4 for questions and answers about client systems.

Administrative Console

You can use a dedicated SPARCstation™ system, known as the *administrative console*, to administer the active cluster. Usually, you install and run administrative tool software, such as the Cluster Control Panel (CCP) and the Sun Cluster module for the Sun Management Center product, on the administrative console. Using `cconsole` under the CCP enables you to connect to more than one node console at a time. For more information on using the CCP, see the *Sun Cluster 3.0 System Administration Guide*.

The administrative console is not a cluster node. You use the administrative console for remote access to the cluster nodes, either over the public network, or optionally through a network-based terminal concentrator. If your cluster consists of the Sun™ Enterprise E10000 platform, you must have the ability to log in from the administrative console to the System Service Processor (SSP) and connect by using the `netcon(1M)` command.

Typically, you configure nodes without monitors. Then, you access the node's console through a `telnet` session from the administrative console, which is connected to a terminal concentrator, and from the terminal concentrator to the node's serial port. (In the case of a Sun Enterprise E10000 server, you connect from the System Service Processor.) See "Console Access Devices" on page 26 for more information.

Sun Cluster does not require a dedicated administrative console, but using one provides these benefits:

- Enables centralized cluster management by grouping console and management tools on the same machine
- Provides potentially quicker problem resolution by your hardware service provider

See Chapter 4 for questions and answers about the administrative console.

Console Access Devices

You must have console access to all cluster nodes. To gain console access, use the terminal concentrator purchased with your cluster hardware, the System Service Processor (SSP) on Sun Enterprise E10000 server servers, or another device that can access `ttya` on each node.

Only one supported terminal concentrator is available from Sun. Use of the supported Sun terminal concentrator is optional. The terminal concentrator enables access to `ttya` on each node by using a TCP/IP network. The result is console-level access for each node from a remote workstation anywhere on the network.

The System Service Processor (SSP) provides console access for Sun Enterprise E10000 servers. The SSP is a SPARCstation system on an Ethernet network that is configured to support the Sun Enterprise E10000 server. The SSP is the administrative console for the Sun Enterprise E10000 server. Using the Sun Enterprise E10000 Network Console feature, any workstation in the network can open a host console session.

Other console access methods include other terminal concentrators, `tip(1)` serial port access from another node and dumb terminals. You can use Sun™ keyboards and monitors, or other serial port devices if your hardware service provider supports them.

See Chapter 4 for questions and answers about console devices.

Sun Cluster Topologies

A topology is the connection scheme that connects the cluster nodes to the storage platforms used in the cluster.

Sun Cluster supports the following topologies:

- Clustered pairs
- N+1 (star)

The following sections describe each topology.

Clustered Pairs Topology

A clustered pairs topology is two or more pairs of nodes operating under a single cluster administrative framework. 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. You might use this topology to run a parallel database application on one pair and a highly available application on another pair. Using the cluster file system, you could also have a two-pair configuration where more than two nodes run a scalable service or parallel database even though all of the nodes are not directly connected to the disks that store the application data.

The following figure illustrates a clustered pair configuration.

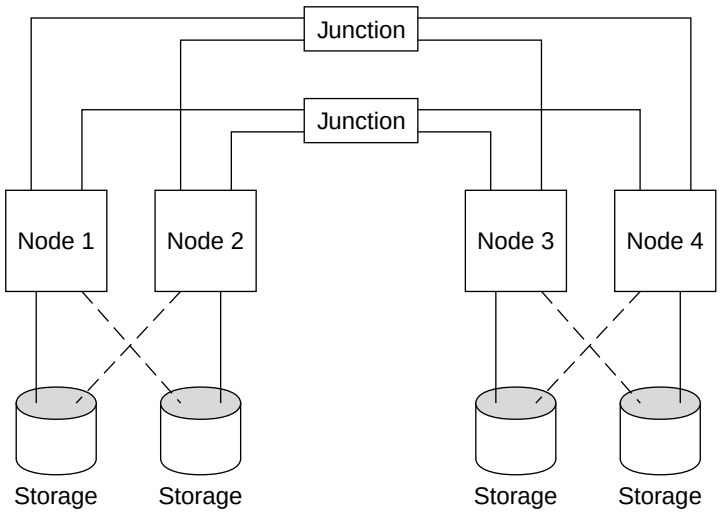

Figure 2–3 Clustered Pairs Topology

Pair+M Topology

The pair+M topology includes a pair of nodes directly connected to shared storage and an additional set of nodes that use the cluster interconnect to access shared storage—they have no direct connection themselves. All nodes in this configuration are still configured with volume managers.

The following figure illustrates a pair+M topology where two of the four nodes (Node 3 and Node 4) use the cluster interconnect to access the storage. This configuration can be expanded to include additional nodes that do not have direct access to the shared storage.

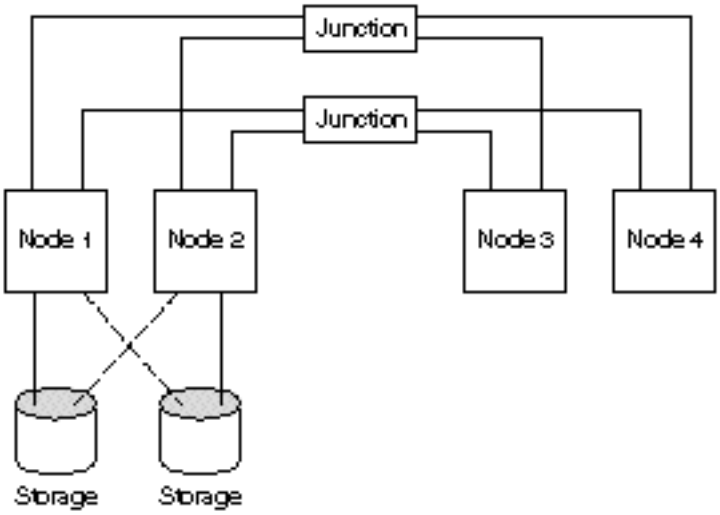

Figure 2–4 Pair+M Topology

N+1 (Star) Topology

An N+1 topology includes some number of primary nodes and one secondary node. You do not have to configure the primary nodes and secondary node identically. The primary nodes actively provide application services. The secondary node need not be idle while waiting for a primary to fail.

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

If a failure occurs on a primary, Sun Cluster fails over the resources to the secondary, where the resources function until they are switched back (either automatically or manually) to the primary.

The secondary must always have enough excess CPU capacity to handle the load if one of the primaries fails.

The following figure illustrates an N+1 configuration.

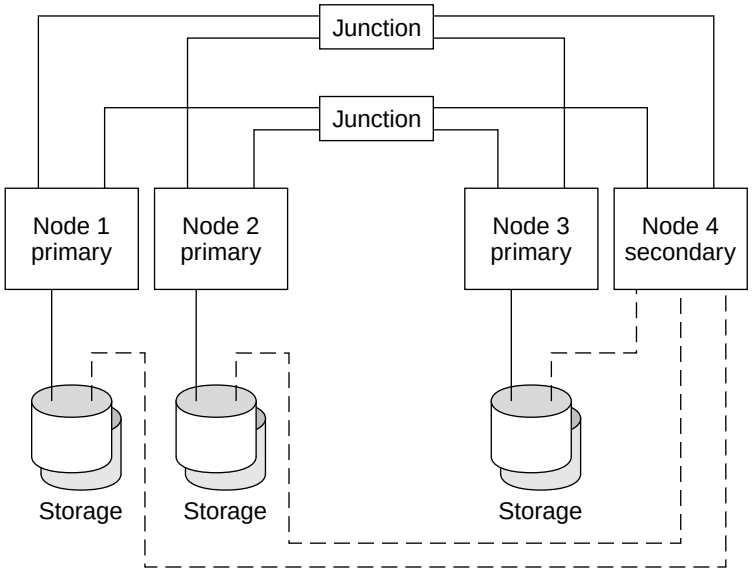

Figure 2–5 N+1 Topology

CHAPTER 3

Key Concepts – Administration and Application Development

This chapter describes the key concepts related to the software components of a Sun Cluster configuration. The topics covered include:

Cluster Administration and Application Development

This information is directed primarily toward system administrators and application developers using the Sun Cluster API and SDK. Cluster system administrators can use this information as background to installing, configuring, and administering cluster software. Application developers can use the information to understand the cluster environment in which they will be working.

The following figure shows a high-level view of how the cluster administration concepts map to the cluster architecture.

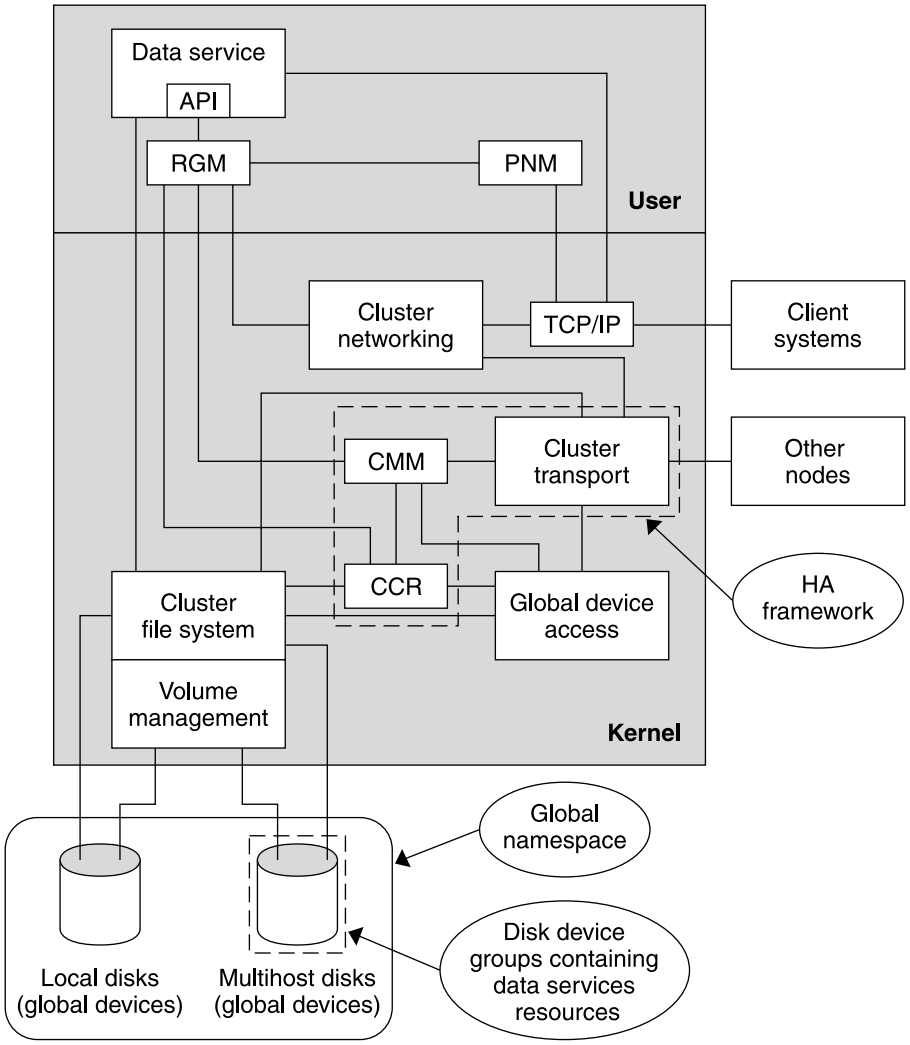

Figure 3–1 Sun Cluster Software Architecture

Administrative Interfaces

You can choose from several user interfaces to install, configure, and administer Sun Cluster and Sun Cluster data services. You can accomplish system administration tasks through the documented command-line interface. On top of the command-line interface are some utilities to simplify selected configuration tasks. Sun Cluster also has a module that runs as part of Sun Management Center that provides a GUI to certain cluster tasks. Refer to the introductory chapter in the *Sun Cluster 3.0 System Administration Guide* for complete descriptions of the administrative interfaces.

Cluster Time

Time between all nodes in a cluster must be synchronized. Whether you synchronize the cluster nodes with any outside time source is not important to cluster operation. Sun Cluster employs the Network Time Protocol (NTP) to synchronize the clocks between nodes.

In general, a change in the system clock of a fraction of a second causes no problems. However, if you run `date(1)`, `rdate(1M)`, or `xntpdate(1M)` (interactively, or within `cron` scripts) on an active cluster, you can force a time change much larger than a fraction of a second to synchronize the system clock to the time source. This forced change might cause problems with file modification timestamps or confuse the NTP service.

When you install the Solaris operating environment on each cluster node, you have an opportunity to change the default time and date setting for the node. In general, you can accept the factory default.

When you install Sun Cluster using `scinstall(1M)`, one step in the process is to configure NTP for the cluster. Sun Cluster supplies a template file, `ntp.cluster` (see `/etc/inet/ntp.cluster` on an installed cluster node), that establishes a peer relationship between all cluster nodes, with one node being the “preferred” node. Nodes are identified by their private hostnames and time synchronization occurs across the cluster interconnect. The instructions for how to configure the cluster for NTP are included in the *Sun Cluster 3.0 Installation Guide.*

Alternately, you can set up one or more NTP servers outside the cluster and change the `ntp.conf` file to reflect that configuration.

In normal operation, you should never need to adjust the time on the cluster. However, if the time was set incorrectly when you installed the Solaris operating environment and you want to change it, the procedure for doing so is included in the *Sun Cluster 3.0 System Administration Guide.*

High-Availability Framework

Sun Cluster makes all components on the “path” between users and data highly available, including network interfaces, the applications themselves, the file system, and the multihost disks. In general, a cluster component is highly available if it survives any single (software or hardware) failure in the system.

The following table shows the kinds of Sun Cluster component failures (both hardware and software) and the kinds of recovery built into the high-availability framework.

TABLE 3–1 Levels of Sun Cluster Failure Detection and Recovery

Failed Cluster Resource	Software Recovery	Hardware Recovery
Data service	HA API, HA framework	N/A
Public network adapter	Network Adapter Failover (NAFO)	Multiple public network adapter cards
Cluster file system	Primary and secondary replicas	Multihost disks
Mirrored multihost disk	Volume management (Solstice DiskSuite and VERITAS Volume Manager)	Hardware RAID-5 (for example, Sun StorEdge A3x00)
Global device	Primary and secondary replicas	Multiple paths to the device, cluster transport junctions
Private network	HA transport software	Multiple private hardware-independent networks
Node	CMM, failfast driver	Multiple nodes

The Sun Cluster high-availability framework detects a node failure quickly and creates a new equivalent server for the framework resources on a remaining node in the cluster. At no time are all framework resources unavailable. Framework resources unaffected by a crashed node are fully available during recovery. Furthermore, framework resources of the failed node become available as soon as they are recovered. A recovered framework resource does not have to wait for all other framework resources to complete their recovery.

Most highly available framework resources are recovered transparently to the applications (data services) using the resource. The semantics of framework resource access are fully preserved across node failure. The applications simply cannot tell that the framework resource server has been moved to another node. Failure of a single node is completely transparent to programs on remaining nodes using the files, devices, and disk volumes attached to this node, as long as an alternative hardware path exists to the disks from another node. An example is the use of multihost disks that have ports to multiple nodes.

Cluster Membership Monitor

The Cluster Membership Monitor (CMM) is a distributed set of agents, one per cluster member. The agents exchange messages over the cluster interconnect to:

- Enforce a consistent membership view on all nodes (quorum)
- Drive synchronized reconfiguration in response to membership changes, using registered callbacks
- Handle cluster partitioning (split brain, amnesia)
- Ensure full connectivity among all cluster members

Unlike previous Sun Cluster releases, CMM runs entirely in the kernel.

Cluster Membership

The main function of the CMM is to establish cluster-wide agreement on the set of nodes that participates in the cluster at any given time. Sun Cluster refers to this constraint as *cluster membership.*

To determine cluster membership, and ultimately, ensure data integrity, the CMM:

- Accounts for a change in cluster membership, such as a node joining or leaving the cluster
- Ensures that a “bad” node leaves the cluster
- Ensures that a “bad” node stays out of the cluster until it is repaired
- Prevents the cluster from partitioning itself into subsets of nodes

See “Quorum and Quorum Devices” on page 43 for more information on how the cluster protects itself from partitioning into multiple separate clusters.

Cluster Membership Monitor Reconfiguration

To ensure that data is kept safe from corruption, all nodes must reach a consistent agreement on the cluster membership. When necessary, the CMM coordinates a cluster reconfiguration of cluster services (applications) in response to a failure.

The CMM receives information about connectivity to other nodes from the cluster transport layer. The CMM uses the cluster interconnect to exchange state information during a reconfiguration.

After detecting a change in cluster membership, the CMM performs a synchronized configuration of the cluster, where cluster resources might be redistributed based on the new membership of the cluster.

Cluster Configuration Repository (CCR)

The Cluster Configuration Repository (CCR) is a private, cluster-wide database for storing information pertaining to the configuration and state of the cluster. The CCR is a distributed database. Each node maintains a complete copy of the database. The

CCR ensures that all nodes have a consistent view of the cluster “world.” To avoid corrupting data, each node needs to know the current state of the cluster resources.

The CCR is implemented in the kernel as a highly available service.

The CCR uses a two-phase commit algorithm for updates: An update must complete successfully on all cluster members or the update is rolled back. The CCR uses the cluster interconnect to apply the distributed updates.

Caution - Although the CCR is made up of text files, never edit the CCR files manually. Each file contains a checksum record to ensure consistency. Manually updating CCR files can cause a node or the entire cluster to stop functioning.

The CCR relies on the CMM to guarantee that a cluster is running only when quorum is established. The CCR is responsible for verifying data consistency across the cluster, performing recovery as necessary, and facilitating updates to the data.

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. This means that CD-ROM and tape devices are not currently highly available devices. The local disks on each server are also not multiported, and thus are not highly available devices.

The cluster automatically assigns unique IDs to each disk, CD-ROM, and tape device in the cluster. This assignment allows consistent access to each device from any node in the cluster. The global device namespace is held in the `/dev/global` directory. See “Global Namespace” on page 40 for more information.

Multiported global devices provide more than one path to a device. In the case of multihost disks, because the disks are part of a disk device group hosted by more than one node, the multihost disks are made highly available.

Device ID (DID)

Sun Cluster manages global devices through a construct known as the device ID (DID) pseudo driver. This driver is used to automatically assign unique IDs to every device in the cluster, including multihost disks, tape drives, and CD-ROMs.

The device ID (DID) pseudo driver is an integral part of the global device access feature of the cluster. The DID driver probes all nodes of the cluster and builds a list of unique disk devices, assigning each a unique major and minor number that is

consistent on all nodes of the cluster. Access to the global devices is performed utilizing the unique device ID assigned by the DID driver instead of the traditional Solaris device IDs, such as `c0t0d0` for a disk.

This approach ensures that any application utilizing the disk devices (such as a volume manager or applications using raw devices) can use a consistent path to access the device. This consistency is especially important for multihost disks, because the local major and minor numbers for each device can vary from node to node, thus changing the Solaris device naming conventions as well. For example, node1 might see a multihost disk as `c1t2d0`, and node2 might see the same disk completely differently, as `c3t2d0`. The DID driver would assign a global name, such as d10, that the nodes would use instead, giving each node a consistent mapping to the multihost disk.

You update and administer Device IDs through `scdidadm(1M)` and `scgdevs(1M)`. See the respective man pages for more information.

Disk Device Groups

In Sun Cluster, all multihost disks must be under control of the Sun Cluster framework. You first create volume manager disk groups—either Solstice DiskSuite disksets or VERITAS Volume Manager disk groups—on the multihost disks. Then, you register the volume manager disk groups as Sun Cluster *disk device groups.* A disk device group is a type of global device. In addition, Sun Cluster registers every individual disk as a disk device group.

Note - Disk device groups are independent of resource groups. One node can master a resource group (representing a group of data service processes) while another can master the disk group(s) being accessed by the data services. However, the best practice is to keep the disk device group that stores a particular application's data and the resource group that contains the application's resources (the application daemon) on the same node. Refer to the overview chapter in the *Sun Cluster 3.0 Data Services Installation and Configuration Guide* for more information about the association between disk device groups and resource groups.

With a disk device group, the volume manager disk group becomes "global" because it provides multipath support to the underlying disks. Each cluster node physically attached to the multihost disks provides a path to the disk device group.

Note - A global device is highly available if it is part of a device group that is hosted by more than one cluster node.

Disk Device Failover

Because a disk enclosure is connected to more than one node, all disk device groups in that enclosure are accessible through an alternate path if the node currently mastering the device group fails. The failure of the node mastering the device group does not affect access to the device group except for the time it takes to perform the recovery and consistency checks. During this time, all requests are blocked (transparently to the application) until the system makes the device group available.

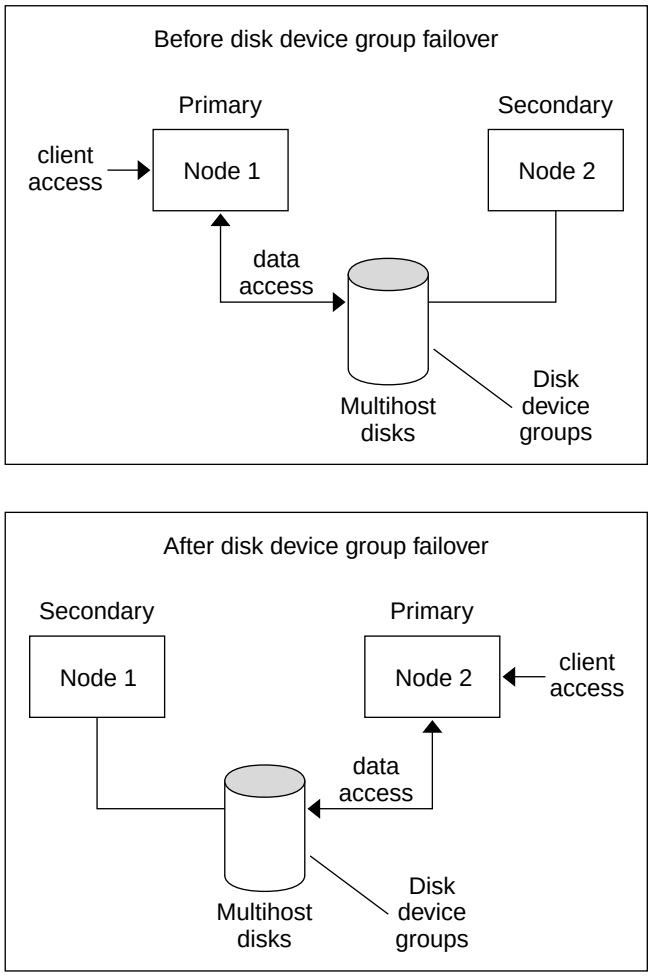

Figure 3–2 Disk Device Group Failover

Global Namespace

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 namespaces. 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.

Normally, the volume manager namespaces reside in the `/dev/md/`*diskset*`/dsk` (and `rdsk`) directories, for Solstice DiskSuite; and in the `/dev/vx/dsk/`*disk-group* and `/dev/vx/rdsk/`*disk-group* directories, for VxVM. These namespaces consist of directories for each Solstice DiskSuite diskset and each VxVM disk group imported throughout the cluster, respectively. Each of these directories houses a device node for each metadevice or volume in that diskset or disk group.

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

The advantages of the global namespace include:

- Each node remains fairly independent, with little change in the device administration model.
- Devices can be selectively made global.
- Third-party link generators continue to work.
- Given a local device name, an easy mapping is provided to obtain its global name.

Local and Global Namespaces Example

The following table shows the mappings between the local and global namespaces for a multihost disk, `c0t0d0s0`.

TABLE 3–2 Local and Global Namespaces Mappings

Component/ Path	Local Node Namespace	Global Namespace
Solaris logical name	`/dev/dsk/c0t0d0s0`	`/global/.devices/node@`*ID*`/dev/dsk/c0t0d0s0`
DID name	`/dev/did/dsk/d0s0`	`/global/.devices/node@`*ID*`/dev/did/dsk/d0s0`

TABLE 3–2 Local and Global Namespaces Mappings *(continued)*

Component/ Path	Local Node Namespace	Global Namespace
Solstice DiskSuite	`/dev/md/`*diskset*`/dsk/d0`	`/global/.devices/node@`*ID*`/dev/md/`*diskset*`/dsk/d0`
VERITAS Volume Manager	`/dev/vx/dsk/`*disk-group*`/v0`	`/global/.devices/node@`*ID*`/dev/vx/dsk/`*disk-group*`/v0`

The global namespace is automatically generated on installation and updated with every reconfiguration reboot. You can also generate the global namespace by running the `scgdevs(1M)` command.

Cluster File Systems

A cluster file system is a proxy between the kernel on one node and the underlying file system and volume manager running on a node that has a physical connection to the disk(s).

Cluster file systems are dependent on global devices (disks, tapes, CD-ROMs) with physical connections to one or more nodes. The global devices can be accessed from any node in the cluster through the same file name (for example, `/dev/global/`) whether or not that node has a physical connection to the storage device. You can use a global device the same as a regular device, that is, you can create a file system on it using `newfs` and/or `mkfs`.

You can mount a file system on a global device globally with `mount -g` or locally with `mount`.Programs can access a file in a cluster file system from any node in the cluster through the same file name (for example, `/global/foo`).A cluster file system is mounted on all cluster members. You cannot mount a cluster file system on a subset of cluster members.

Using Cluster File Systems

In Sun Cluster, all multihost disks are configured as disk device groups, which can be Solstice DiskSuite disksets, VxVM disk groups, or individual disks not under control of a software-based volume manager. Also, local disks are configured as disk device groups: a path leads to each local disk from each node. This setup does not mean the data on a disk is necessarily available from all nodes. The data only becomes available to all nodes if the file systems on the disks are mounted globally as a cluster file system.

A local file system that is made into a cluster file system only has a single connection to the disk storage. If the node with the physical connection to the disk storage fails, the other nodes no longer have access to the cluster file system. You can have local file systems on a single node that are not accessible directly from other nodes.

HA data services are set up so that the data for the service is stored on disk device groups in cluster file systems. This setup has several advantages. First, the data is highly available; that is, because the disks are multihosted, if the path from the node that currently is the primary fails, access is switched to another node that has direct access to the same disks. Second, because the data is on a cluster file system, it can be viewed from any cluster node directly—you do not have to log onto the node that currently masters the disk device group to view the data.

Proxy File System (PXFS)

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 system 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 extensive caching and provides zero-copy bulk I/O movement to move large data objects efficiently.
- 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.
- PXFS is independent of underlying file system and volume management software. PXFS makes any supported on-disk file system global.
- PXFS is built on top of the existing Solaris file system at the vnode interface. This interface enables PXFS to be implemented without extensive kernel modifications.

PXFS is not a distinct file system type. That is, clients see the underlying file system (for example, UFS).

Cluster File System Independence

The cluster file system is independent of the underlying file system and volume manager. Currently, you can build cluster file systems on UFS using either Solstice DiskSuite or VERITAS Volume Manager.

As with normal file systems, you can mount cluster file systems in two ways:

- **Manually**—Use the `mount` command and the `-g` option to mount the cluster file system from the command line, for example:

```
# mount -g /dev/global/dsk/d0s0 /global/oracle/data
```

- **Automatically**—Create an entry in the `/etc/vfstab` file with a `global` mount option to mount the cluster file system at boot. You then create a mount point under the `/global` directory on all nodes. The directory `/global` is a recommended location, not a requirement. Here's a sample line for a cluster file system from an `/etc/vfstab` file:

```
/dev/md/oracle/dsk/d1 /dev/md/oracle/rdsk/d1 /global/oracle/
data ufs 2 yes global,logging
```

Note - While Sun Cluster does not impose a naming policy for cluster file systems, you can ease administration by creating a mount point for all cluster file systems under the same directory, such as `/global/`*disk-device-group*. See *Sun Cluster 3.0 Installation Guide* and *Sun Cluster 3.0 System Administration Guide* for more information.

The Syncdir Mount Option

The `syncdir` mount option can be used for cluster file systems. However, there is a significant performance improvement if you do not specify `syncdir`. If you specify `syncdir`, the writes are guaranteed to be POSIX compliant. If you do not, you will have the same behavior that is seen with UFS file systems. For example, under some cases, without `syncdir`, you would not discover an out of space condition until you close a file. With `syncdir` (and POSIX behavior), the out of space condition would have been discovered during the write operation. The cases in which you could have problems if you do not specify `syncdir` are rare, so we recommend that you do not specify it and receive the performance benefit.

See "File Systems FAQ" on page 62 for frequently asked questions about global devices and cluster file systems.

Quorum and Quorum Devices

Because cluster nodes share data and resources, the cluster must take steps to maintain data and resource integrity. When a node does not meet the cluster rules for membership, the cluster must disallow the node from participating in the cluster.

In Sun Cluster, the mechanism that determines node participation in the cluster is known as a *quorum*. Sun Cluster uses a majority voting algorithm to implement quorum. Both cluster nodes and *quorum devices*, which are disks that are shared

between two or more nodes, vote to form quorum. A quorum device can contain user data.

The quorum algorithm operates dynamically: as cluster events trigger its calculations, the results of calculations can change over the lifetime of a cluster.Quorum protects against two potential cluster problems—split brain and amnesia—both of which can cause inconsistent data to be made available to clients. The following table describes these two problems and how quorum solves them.

TABLE 3–3 Cluster Quorum, and Split-Brain and Amnesia Problems

Problem	Description	Quorum's Solution
Split brain	Occurs when the cluster interconnect between nodes is lost and the cluster becomes partitioned into sub-clusters, each of which believes that it is the only partition	Allows only the partition (sub-cluster) with a majority of votes to run as the cluster (where at most one partition can exist with such a majority)
Amnesia	Occurs when the cluster restarts after a shutdown with cluster data older than at the time of the shutdown	Guarantees that when a cluster is booted, it has at least one node that was a member of the most recent cluster membership (and thus has the latest configuration data)

Quorum Vote Counts

Both cluster nodes and quorum devices (disks that are shared between two or more nodes) vote to form quorum. By default, cluster nodes acquire a quorum vote count of one when they boot and become cluster members. Nodes can also have a vote count of zero, for example, when the node is being installed, or when an administrator has placed a node into maintenance state.

Quorum devices acquire quorum vote counts based on the number of node connections to the device. When a quorum device is set up, it acquires a maximum vote count of *N*-1 where *N* is the number of nodes with non zero vote counts that have ports to the quorum device. For example, a quorum device connected to two nodes with non zero vote counts has a quorum count of one (two minus one).

You configure quorum devices during the cluster installation, or later by using the procedures described in the *Sun Cluster 3.0 System Administration Guide.*

Note - A quorum device contributes to the vote count only if at least one of the nodes to which it is currently attached is a cluster member. Also, during cluster boot, a quorum device contributes to the count only if at least one of the nodes to which it is currently attached is booting and was a member of the most recently booted cluster when it was shut down.

Quorum Configurations

Quorum configurations depend on the number of nodes in the cluster:

- **Two-Node Clusters –** Two quorum votes are required for a two-node cluster to form. These two votes can come from the two cluster nodes, or from just one node and a quorum device. Nevertheless, a quorum device must be configured in a two-node cluster to ensure that a single node can continue if the other node fails.
- **More Than Two-Node Clusters** – You should specify a quorum device between every pair of nodes that shares access to a disk storage enclosure. For example, suppose you have a three-node cluster similar to the one shown in Figure 3–3. In this figure, nodeA and nodeB share access to the same disk enclosure and nodeB and nodeC share access to another disk enclosure. There would be a total of five quorum votes, three from the nodes and two from the quorum devices shared between the nodes. A cluster needs a majority of the quorum votes, three, to form.

  Specifying a quorum device between every pair of nodes that shares access to a disk storage enclosure is not required or enforced by Sun Cluster. However, it can provide needed quorum votes for the case where an N+1 configuration degenerates into a two-node cluster and then the node with access to both disk enclosures also fails. If you configured quorum devices between all pairs, the remaining node could still operate as a cluster.

  See Figure 3–3 for examples of these configurations.

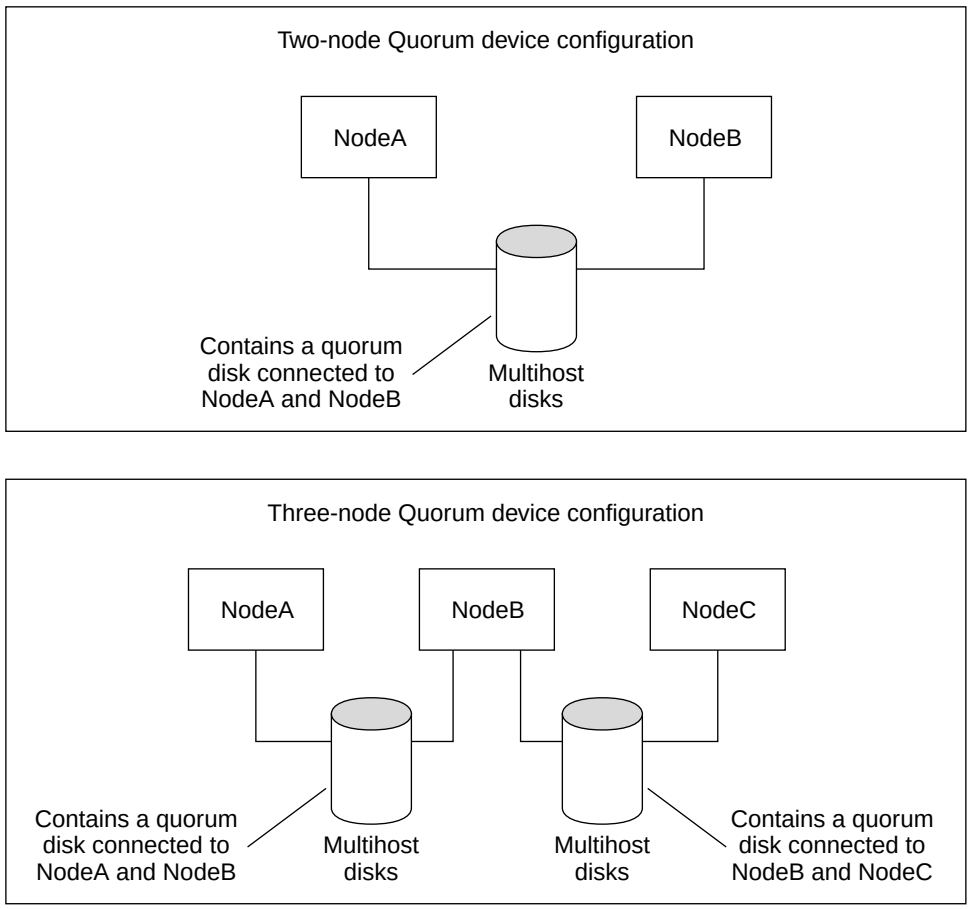

Figure 3–3 Quorum Device Configuration Examples

Quorum Guidelines

Use the following guidelines when setting up quorum devices:

- Establish a quorum device between all nodes that are attached to the same shared disk storage enclosure. Add one disk within the shared enclosure as a quorum device to ensure that if any node fails, the other nodes can maintain quorum and master the disk device groups on the shared enclosure.
- You must connect the quorum device to at least two nodes.
- A quorum device can be any SCSI-2 or SCSI-3 disk used as a dual-ported quorum device. Disks connected to more than two nodes must support SCSI-3 Persistent Group Reservation (PGR) regardless of whether the disk is used as a quorum device. See the chapter on planning in the *Sun Cluster 3.0 Installation Guide* for more information.

- You can use a disk that contains user data as a quorum device.

Tip - Configure more than one quorum device between sets of nodes. Use disks from different enclosures, and configure an odd number of quorum devices between each set of nodes. This protects against individual quorum device failures.

Failure Fencing

A major issue for clusters is a failure that causes the cluster to become partitioned (called *split brain*). When this happens, not all nodes can communicate, so individual nodes or subsets of nodes might try to form individual or subset clusters. Each subset or partition might believe it has sole access and ownership to the multihost disks. Multiple nodes attempting to write to the disks can result in data corruption.

Failure fencing limits node access to multihost disks by physically preventing access to the disks. When a node leaves the cluster (it either fails or becomes partitioned), failure fencing ensures that the node can no longer access the disks. Only current member nodes have access to the disks, resulting in data integrity.

Disk device services provide failover capability for services that make use of multihost disks. When a cluster member currently serving as the primary (owner) of the disk device group fails or becomes unreachable, a new primary is chosen, enabling access to the disk device group to continue with only minor interruption. During this process, the old primary must give up access to the devices before the new primary can be started. However, when a member drops out of the cluster and becomes unreachable, the cluster cannot inform that node to release the devices for which it was the primary. Thus, you need a means to enable surviving members to take control of and access global devices from failed members.

Sun Cluster uses SCSI disk reservations to implement failure fencing. Using SCSI reservations, failed nodes are “fenced” away from the multihost disks, preventing them from accessing those disks.

SCSI-2 disk reservations support a form of reservations, which either grants access to all nodes attached to the disk (when no reservation is in place) or restricts access to a single node (the node that holds the reservation).

When a cluster member detects that another node is no longer communicating over the cluster interconnect, it initiates a failure fencing procedure to prevent the other node from accessing shared disks. When this failure fencing occurs, it is normal to have the fenced node panic with a “reservation conflict” messages on its console.

The reservation conflict occurs because after a node has been detected to no longer be a cluster member, a SCSI reservation is put on all of the disks that are shared between this node and other nodes. The fenced node might not be aware that it is being fenced and if it tries to access one of the shared disks, it detects the reservation and panics.

Volume Managers

Sun Cluster uses volume management software to increase the availability of data by using mirrors and hot spare disks, and to handle disk failures and replacements.

Sun Cluster does not have its own internal volume manager component, but relies on the following volume managers:

- Solstice DiskSuite
- VERITAS Volume Manager

Volume management software in the cluster provides support for:

- Failover handling of node failures
- Multipath support from different nodes
- Remote transparent access to disk device groups

When setting up a volume manager in Sun Cluster, you configure multihost disks as Sun Cluster disk devices, a wrapper for a volume manager disk group. The device can be either a Solstice DiskSuite diskset or a VxVM disk group.

You must configure disk groups used for data services for mirroring to make the disks highly available within the cluster.

You can use metadevices or plexes either as a raw device (database application) or to hold UFS file systems.

Volume management objects—metadevices and volumes—come under the control of the cluster, thus becoming disk device groups. For example, in Solstice DiskSuite, when you create a diskset in the cluster (by using the `metaset(1M)` command), a corresponding disk device group of the same name is created. Then, as you create metadevices in that diskset, they become global devices. Thus, a diskset is a collection of disk devices (DID devices) and hosts to which all devices in the set are ported. All disksets in a cluster need to be created with more than one host in the set to achieve HA. A similar situation occurs when you use VERITAS Volume Manager. The details of setting up each volume manager are included in the appendixes of the *Sun Cluster 3.0 Installation Guide*.

An important consideration when planning your disksets or disk groups is to understand how their associated disk device groups are associated with the application resources (data) within the cluster. Refer to the *Sun Cluster 3.0 Installation Guide* and the *Sun Cluster 3.0 Data Services Installation and Configuration Guide* for discussions of these issues.

Data Services

The term *data service* is used to describe a third-party application such as Apache Web Server that has been configured to run on a cluster rather than on a single

server. A data service includes the application software and Sun Cluster software that starts, stops, and monitors the application.

Sun Cluster supplies data service methods that are used to control and monitor the application within the cluster. These methods run under the control of the Resource Group Manager (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.

The RGM also manages resources in the cluster, including instances of an application and network resources (logical hostnames and shared addresses).

Sun Cluster also supplies an API and data service development tools to enable application programmers to develop the data service methods needed to make other applications run as highly available data services with Sun Cluster.

Resource Group Manager (RGM)

Sun Cluster provides an environment for making applications highly available or scalable. The RGM acts on *resources*, which are logical components that can be:

- Brought online and taken offline (switched)
- Managed by the RGM framework
- Hosted on a single node (failover mode) or multiple nodes (scalable mode)

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 API (DSDL API), or the Sun Cluster Resource Management API (RMAPI). 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.

Failover Data Services

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 utilize a *failover resource group*, which is a container for application instance resources and network resources (*logical hostnames*). Logical hostnames 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.

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 to start the instance on another node (failover), depending on how the data service has been configured.

Scalable Data Services

The scalable data service has the potential for active instances on multiple nodes. Scalable services utilize 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*. The cluster can use the load-balancing policy to balance the service load between several nodes. Note that there can be multiple GIFs on different nodes hosting other shared addresses.

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. If an application instance running fails, the instance attempts to restart on the same node.

If an application instance cannot be restarted on the same node, and another unused node is configured to run the service, the service fails over to the unused node. Otherwise, it continues to run on the remaining nodes, possibly causing a degradation of service throughput.

Note - TCP state for each application instance is kept on the node with the instance, not on the GIF node. Therefore, failure of the GIF node does not affect the connection.

Figure 3–4 shows an example of failover and a scalable resource group and the dependencies that exist between them for scalable services. This example shows three resource groups. The failover resource group contains application resources for highly available DNS, and network resources used by both highly available DNS and highly available Apache Web Server. The scalable resource groups contain only application instances of the Apache Web Server. Note that resource group dependencies exist between the scalable and failover resource groups (solid lines) and that all of the Apache application resources are dependent on the network resource `schost-2`, which is a shared address (dashed lines).

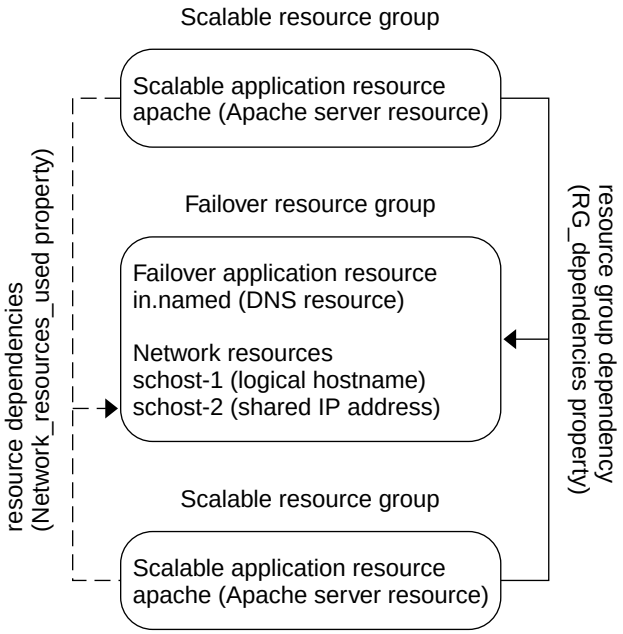

Figure 3–4 Failover and Scalable Resource Group Example

Scalable Service Architecture

The primary goal of cluster networking is to provide scalability for data services. Scalability means that as the load offered to a service increases, a data service can maintain a constant response time in the face of this increased workload as new nodes are added to the cluster and new server instances are run. We call such a service a scalable data service. A good example of a scalable data service is a web service. Typically, a scalable data service is composed of several instances, each of which runs on different nodes of the cluster. Together these instances behave as a single service from the standpoint of a remote client of that service and implement the functionality of the service.We might, for example, have a scalable web service made up of several `httpd` daemons running on different nodes. Any `httpd` daemon may serve a client request. The daemon that serves the request depends on a *load-balancing policy*. The reply to the client appears to come from the service, not the particular daemon that serviced the request, thus preserving the single service appearance.

A scalable service is composed of:

- Networking infrastructure support for scalable services
- Load balancing
- HA support for networking and data services (using the Resource Group Manager)

The following figure depicts the scalable service architecture.

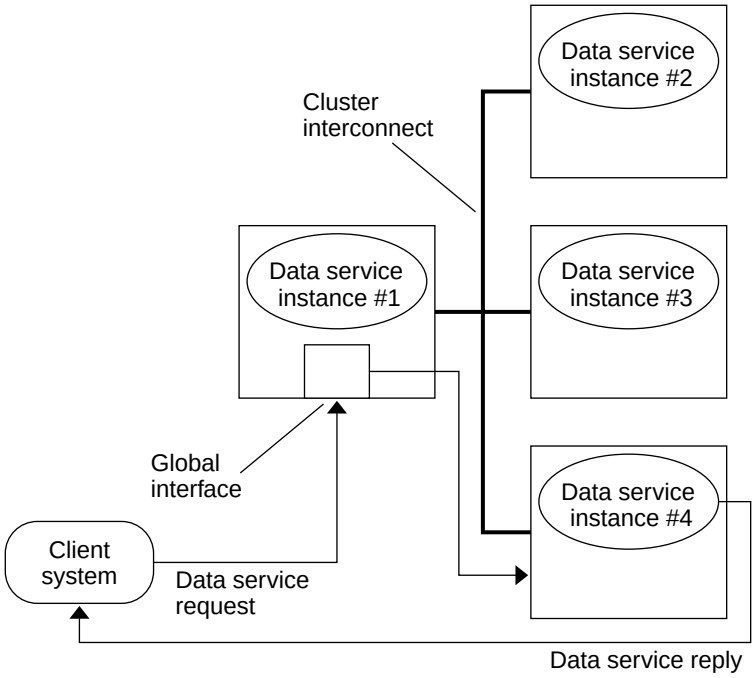

Figure 3–5 Scalable Service Architecture

The nodes that are not hosting the global interface (proxy nodes) have the shared address hosted on their loopback interfaces. Packets coming into the GIF are distributed to other cluster nodes based on configurable load-balancing policies. The possible load-balancing policies are described next.

Load-Balancing Policies

Load balancing improves performance of the scalable service, both in response time and in throughput.

There are two classes of scalable data services: *pure* and *sticky*. A pure service is one where any instance of it can respond to client requests. A sticky service is one where a client sends requests to the same instance. Those requests are not redirected to other instances.

A pure service uses a weighted load-balancing policy. Under this load-balancing policy, client requests are by default uniformly distributed over the server instances in the cluster. For example, in a three-node cluster, let us suppose that each node has the weight of 1. Each node will service 1/3 of the requests from any client on behalf of that service. Weights can be changed at any time by the administrator through the `scrgadm(1M)` command interface.

A sticky service has two flavors, *ordinary sticky* and *wildcard sticky*. Sticky services allow concurrent application-level sessions over multiple TCP connections to share in-state memory (application session state).

Ordinary sticky services permit a client to share state between multiple concurrent TCP connections. The client is said to be “sticky” with respect to that server instance listening on a single port. The client is guaranteed that all of his requests go to the same server instance, provided that instance remains up and accessible and the load balancing policy is not changed while the service is online.

For example, a web browser on the client connects to a shared IP address on port 80 using three different TCP connections, but the connections are exchanging cached session information between them at the service.

A generalization of a sticky policy extends to multiple scalable services exchanging session information behind the scenes at the same instance. When these services exchange session information behind the scenes at the same instance, the client is said to be “sticky” with respect to multiple server instances on the same node listening on different ports.

For example, a customer on an e-commerce site fills his shopping cart with items using ordinary HTTP on port 80, but switches to SSL on port 443 to send secure data in order to pay by credit card for the items in the cart.

Wildcard sticky services use dynamically assigned port numbers, but still expect client requests to go to the same node. The client is “sticky wildcard” over ports with respect to the same IP address.

A good example of this policy is passive mode FTP. A client connects to an FTP server on port 21 and is then informed by the server to connect back to a listener port server in the dynamic port range. All requests for this IP address are forwarded to the same node that the server informed the client through the control information.

Note that for each of these sticky policies the weighted load-balancing policy is in effect by default, thus, a client’s initial request is directed to the instance dictated by the load balancer. After the client has established an affinity for the node where the instance is running, then future requests are directed to that instance as long as the node is accessible and the load balancing policy is not changed.

Additional details of the specific load balancing policies are discussed below.

- Weighted. The load is distributed among various nodes according to specified weight values. This policy is set using the `LB_WEIGHTED` value for the `Load_balancing_weights` property. If a weight for a node is not explicitly set, the weight for that node defaults to one.

  Note that this policy is not round robin. A round-robin policy would always cause each request from a client to go to a different node: the first request to node 1, the second request to node 2, and so on. The weighted policy guarantees that a certain percentage of the traffic from clients is directed to a particular node. This policy does not address individual requests.

- Sticky. In this policy, the set of ports is known at the time the application resources are configured. This policy is set using the `LB_STICKY` value for the `Load_balancing_policy` resource property.
- Sticky-wildcard. This policy is a superset of the ordinary “sticky” policy. For a scalable service identified by the IP address, ports are assigned by the server (and are not known in advance). The ports might change. This policy is set using the `LB_STICKY_WILD` value for the `Load_balancing_policy` resource property.

Failback Settings

Resource groups fail over from one node to another. You can specify that, in the event that a resource group fails over to another node, after the node it was previously running on returns to the cluster, it will “fail back” to the original node. This option is set using the `Failback` resource group property setting.

In certain instances, for example if the original node hosting the resource group is failing and rebooting repeatedly, setting failback might result in reduced availability for the resource group.

Data Services Fault Monitors

Each Sun Cluster data service supplies a fault monitor that periodically probes the data service to determine its health. A fault monitor verifies that the application daemon(s) 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.

Developing New Data Services

Sun supplies software that enables you to make various applications operate as highly available data services within a cluster. If the application that you want to run as a highly available data service is not one that is currently offered by Sun, you can use an API or the DSDL API to take your application and configure it to run as a highly available data service. There are two flavors of data services, failover and scalable. There is a set of criteria for determining whether your application can use one of these data service flavors. The specific criteria is described in the Sun Cluster documents that describe the APIs you can use for your application.

Here, we present some guidelines to help you understand whether your service can take advantage of the scalable data services architecture. Review the section, “Scalable Data Services” on page 50 for more general information on scalable services.

New services that satisfy the following guidelines may make use of scalable services. If an existing service doesn’t follow these guidelines exactly, portions may need to be rewritten so that the service complies with the guidelines.

A scalable data service has the following characteristics. First, such a service is composed of one or more server *instances*. Each instance runs on a different node of the cluster. Two or more instances of the same service cannot run on the same node.

Second, if the service provides an external logical data store, then concurrent access to this store from multiple server instances must be synchronized to avoid losing updates or reading data as it's being changed. Note that we say "external" to distinguish the store from in-memory state, and "logical" because the store appears as a single entity, although it may itself be replicated. Furthermore, this logical data store has the property that whenever any server instance updates the store, that update is immediately seen by other instances.

Sun Cluster provides such an external storage through its cluster file system and its global raw partitions. As an example, suppose a service writes new data to an external log file or modifies existing data in place. When multiple instances of this service run, each has access to this external log, and each may simultaneously access this log. Each instance must synchronize its access to this log, or else the instances interfere with each other. The service could use ordinary Solaris file locking via `fcntl(2)` and `lockf(3C)` to achieve the desired synchronization.

Another example of such a store is a backend database such as highly available Oracle or Oracle Parallel Server. Note that such a back-end database server provides built-in synchronization using database query or update transactions, and so multiple server instances need not implement their own synchronization.

An example of a service that is not a scalable service in its current incarnation is Sun's IMAP server. The service updates a store, but that store is private and when multiple IMAP instances write to this store, they overwrite each other because the updates are not synchronized. The IMAP server must be rewritten to synchronize concurrent access.

Finally, note that instances may have private data that's disjoint from the data of other instances. In such a case, the service need not concern itself with synchronizing concurrent access because the data is private, and only that instance can manipulate it. In this case, you must be careful not to store this private data under the cluster file system because it has the potential to become globally accessible.

Data Service API and Data Service Development Library API

Sun Cluster provides the following to make applications highly available:

- Data services supplied as part of Sun Cluster
- A data service API
- A data service development library API
- A "generic" data service

The *Sun Cluster 3.0 Data Services Installation and Configuration Guide* describes how to install and configure the data services supplied with Sun Cluster. The *Sun*

Cluster 3.0 Data Services Developers' Guide describes how to instrument other applications to be highly available under the Sun Cluster framework.

The Sun Cluster API and Data Service Development Library API enable application programmers to develop fault monitors and scripts that start and stop data services instances. With these tools, an application can be instrumented to be a failover or a scalable data service. In addition, Sun Cluster provides a "generic" data service that can be used to quickly generate an application's required start and stop methods to make it run as a highly available data service.

Resources and Resource Types

Data services utilize several types of *resources*: applications such as Apache Web Server or iPlanet Web Server utilize network addresses (logical hostnames and shared addresses) upon which the applications depend. Application and network resources form a basic unit that is managed by the RGM.

A resource is an instantiation of a *resource type* that is defined cluster wide. There are several resource types defined.

Data services are resource types. For example, Sun Cluster HA for Oracle is the resource type `SUNW.oracle` and Sun Cluster HA for Apache is the resource type `SUNW.apache`.

Network resources are either `SUNW.LogiclaHostname` or `SUNW.SharedAddress` resource types. These two resource types are pre-registered by the Sun Cluster product.

The `SUNW.HAStorage` resource type is used to synchronize the startup of resources and disk device groups upon which the resources depend. It ensures that before a data service starts, the paths to cluster file system mount points, global devices, and device group names are available.

RGM-managed resources are placed into groups, called *resource groups*, so that they can be managed as a unit. A resource group is migrated as a unit if a failover or switchover is initiated on the resource group.

Note - When you bring a resource group containing application resources online, the application is started. The data service start method waits until the application is up and running before exiting successfully. The determination of when the application is up and running is accomplished the same way the data service fault monitor determines that a data service is serving clients. Refer to the *Sun Cluster 3.0 Data Services Installation and Configuration Guide* for more information on this process.

Resource and Resource Group Properties

You can configure property values for resources and resource groups for your Sun Cluster data services. A set of standard properties are common to all data services and a set of extension properties are specific to each data service. Some standard and extension properties are configured with default settings so that you do not have to modify them. Others need to be set as part of the process of creating and configuring resources. The documentation for each data service specifies what properties are used by the resource type, and how they should be configured.

The standard properties are used to configure resource and resource group properties that are usually independent of any particular data service. The set of standard properties is described in an appendix to the *Sun Cluster 3.0 Data Services Installation and Configuration Guide.*

The extension properties provide information such as the location of application binaries and configuration files. You modify extension properties as you configure your data services. The set of extension properties is described in the individual chapter for the data service in the *Sun Cluster 3.0 Data Services Installation and Configuration Guide.*

Public Network Management (PNM) and Network Adapter Failover (NAFO)

Clients make data requests to the cluster through the public network. Each cluster node is connected to at least one public network through a public network adapter.

Sun Cluster Public Network Management (PNM) software provides the basic mechanism for monitoring public network adapters and failing over IP addresses from one adapter to another when a fault is detected. Each cluster node has its own PNM configuration, which can be different from that on other cluster nodes.

Public network adapters are organized into *Network Adapter Failover groups* (NAFO groups). Each NAFO group has one or more public network adapters. While only one adapter can be active at any time for a given NAFO group, more adapters in the same group serve as backup adapters that are used during adapter failover in the case that a fault is detected by the PNM daemon on the active adapter. A failover causes the IP addresses associated with the active adapter to be moved to the backup adapter, thereby maintaining public network connectivity for the node. Because the failover happens at the adapter interface level, higher-level connections such as TCP are not affected, except for a brief transient delay during the failover.

Note - Because of the congestion recovery characteristics of TCP, TCP endpoints can suffer further delay after a successful failover as some segments could be lost during the failover, activating the congestion control mechanism in TCP.

NAFO groups provide the building blocks for logical hostname and shared address resources. The `scrgadm(1M)` command automatically creates NAFO groups for you if necessary. You can also create NAFO groups independently of logical hostname and shared address resources to monitor public network connectivity of cluster nodes. The same NAFO group on a node can host any number of logical hostname or shared address resources. For more information on logical hostname and shared address resources, see the *Sun Cluster 3.0 Data Services Installation and Configuration Guide*.

Note - The design of the NAFO mechanism is meant to detect and mask adapter failures. The design is not intended to recover from an administrator using `ifconfig(1M)` to remove one of the logical (or shared) IP addresses. The Sun Cluster design views the logical and shared IP addresses as resources managed by the RGM. The correct way for an administrator to add or remove an IP address is to use `scrgadm(1M)` to modify the resource group containing the resource.

PNM Fault Detection and Failover Process

PNM checks the packet counters of an active adapter regularly, assuming that the packet counters of a healthy adapter will change because of normal network traffic through the adapter. If the packet counters do not change for some time, PNM goes into a `ping` sequence, which forces traffic through the active adapter. PNM checks for any change in the packet counters at the end of each sequence, and declares the adapter faulty if the packet counters remain unchanged after the ping sequence is repeated a couple of times. These events trigger a failover to a backup adapter, as long as one is available.

Both input and output packet counters are monitored by PNM so that when either or both remain unchanged for some time, the ping sequence is initiated.

The ping sequence consists of a ping of the `ALL_ROUTER` multicast address (224.0.0.2), the `ALL_HOST` multicast address (224.0.0.1), and the local subnet broadcast address.

Pings are structured in a least-costly-first manner, so that a more costly ping is not run if a less costly one has succeeded. Also, pings are used only as a means to generate traffic on the adapter. Their exit statuses do not contribute to the decision of whether an adapter is functioning or faulty.

Four tunable parameters are in this algorithm: `inactive_time`, `ping_timeout`, `repeat_test`, and `slow_network`. These parameters provide an adjustable trade-off between speed and correctness of fault detection. Refer to the procedure for changing public network parameters in the *Sun Cluster 3.0 System Administration Guide* for details on the parameters and how to change them.

After a fault is detected on a NAFO group's active adapter, if a backup adapter is not available, the group is declared DOWN, while testing of all its backup adapters continues. Otherwise, if a backup adapter is available, a failover occurs to the backup

adapter. Logical addresses and their associated flags are “transferred” to the backup adapter while the faulty active adapter is brought down and unplumbed.

When the failover of IP addresses completes successfully, gratuitous ARP broadcasts are sent. The connectivity to remote clients is therefore maintained.

CHAPTER 4

Frequently Asked Questions

This chapter includes answers to the most frequently asked questions about Sun Cluster. The questions are organized by topic.

High Availability FAQ

- **What exactly is a highly available system?**

  Sun Cluster defines high availability (HA) as the ability of a cluster to keep an application up and running, even though a failure has occurred that would normally make a server system unavailable.

- **What is the process by which the cluster provides high availability?**

  Through a process known as failover, the cluster framework provides a highly available environment. Failover is a series of steps performed by the cluster to migrate an application from a failing node to another operational node in the cluster.

- **What is the difference between an HA and scalable service?**

  An HA service means that an application runs on only one primary node in the cluster at a time. Other nodes might run other applications, but each application runs on only a single node. If a primary node fails, the applications running on the failed node fail over to another node and continue running.

  A scalable service spreads an application across multiple nodes to create a single, logical service. Scalable services leverage the number of nodes and processors in the entire cluster on which they run. One node receives all application requests and dispatches them to multiple nodes on which the application server is running. If this node fails (it is called the Global Interface Node or GIF), the global interface fails over to a surviving node. If any of the nodes on which the application is

running fails, the application continues to run on the other nodes with some performance degradation until the failed node returns to the cluster.

File Systems FAQ

- **Can I run one or more of the cluster nodes as highly available NFS server(s) with other cluster nodes as clients?**

  No. Issues exist with local locking interferes having the ability to kill and restart `lockd` (which occurs during NFS failover). Between the kill and restart, a blocked local process can be granted the lock, which prevents the client system that owns the lock from reclaiming it after failover.

- **Can I use a cluster file system for applications that are not under Resource Group Manager control?**

  Yes. However, without RGM control, the applications cannot survive the failure of the node on which they are running.

- **Must all cluster file systems have a mount point under the /global/*device-group* directory?**

  No. However, placing cluster file systems under the same mount point, such as `/global/`*device-group*, enables better organization and management of these file systems.

- **What are the differences between using the cluster file system and exporting NFS file systems?**

  There are several differences:

  1. The cluster file system supports global devices. NFS does not support remote access to devices.
  2. The cluster file system has a global namespace. Only one mount command is required. With NFS, you must mount the file system on each node.
  3. The cluster file system caches files in more cases than does NFS. For example when a file is being accessed from multiple nodes for read, write, file locks, async I/O.
  4. The cluster file system supports seamless failover if one server fails. NFS supports multiple servers, but failover is only possible for read-only file systems.
  5. The cluster file system is built to exploit future fast cluster interconnects that provide remote DMA and zero-copy functions.
  6. If you change the attributes on a file (using `chmod(1M)`, for example) in a cluster file system, the change is reflected immediately on all nodes. With an exported NFS file system, this can take much longer.

Volume Management FAQ

- **Do I need to mirror all disk devices?**

  For a disk device to be considered highly available, it must be mirrored, or use RAID-5 hardware. All data services should use either highly available disk devices, or cluster file systems mounted on highly available disk devices. Such configurations can tolerate single disk failures.

Data Services FAQ

- **What Sun Cluster data services are available?**

  The list of supported data services is included in the *Sun Cluster 3.0 Release Notes*.

- **What application versions are supported by Sun Cluster data services?**

  The list of supported application versions is included in the *Sun Cluster 3.0 Release Notes.*

- **Can I write my own data service?**

  Yes. See the *Sun Cluster 3.0 Data Services Developers' Guide* and the Data Service Enabling Technologies documentation provided with the Data Service Development Library API for more information.

- **When creating network resources, should I specify numeric IP addresses or hostnames?**

  The preferred method for specifying network resources is to use the UNIX hostname rather than the numeric IP address.

- **When creating network resources, what is the difference between using a logical hostname (a LogicalHostname resource) or a shared address (a SharedAddress resource)?**

  Wherever the documentation calls for the use of a `LogicalHostname` resource in a `Failover` mode resource group, a `SharedAddress` resource or `LogicalHostname` resource may be used interchangeably. The use of a `SharedAddress` resource incurs some additional overhead because the cluster networking software is configured for a `SharedAddress` but not for a `LogicalHostname`.

  The advantage to using a `SharedAddress` is the case where you are configuring both scalable and failover data services, and want clients to be able to access both services using the same hostname. In this case, the `SharedAddress` resource(s)

along with the failover application resource are contained in one resource group, while the scalable service resource is contained in a separate resource group and configured to use the `SharedAddress`. Both the scalable and failover services may then use the same set of hostnames/addresses which are configured in the `SharedAddress` resource.

Public Network FAQ

- **What public network adapters does Sun Cluster support?**

  Currently, Sun Cluster supports Ethernet (10/100BASE-T and 1000BASE-SX Gb) public network adapters. Because new interfaces might be supported in the future, check with your Sun sales representative for the most current information.

- **What is the role of the MAC address in failover?**

  When a failover occurs, new Address Resolution Protocol (ARP) packets are generated and broadcast to the world. These ARP packets contain the new MAC address (of the new physical adapter to which the node failed over) and the old IP address. When another machine on the network receives one of these packets, it flushes the old MAC-IP mapping from its ARP cache and uses the new one.

- **Is Sun Cluster supported to set local-mac-address?=true in the OpenBoot PROM for a host adapter?**

  No, this variable is not supported.

Cluster Members FAQ

- **Do all cluster members need to have the same root password?**

  You are not required to have the same root password on each cluster member. However, you can simplify administration of the cluster by using the same root password on all nodes.

- **Is the order in which nodes are booted significant?**

  In most cases, no. However, the boot order is important to prevent amnesia (refer to “Quorum and Quorum Devices” on page 43 for details on amnesia). For example, if node two was the owner of the quorum device and node one is down, and then you bring node two down, you must bring up node two before bringing back node one. This prevents you from accidentally bringing up a node with out of date cluster configuration information.

- **Do I need to mirror local disks in a cluster node?**

Yes. Though this mirroring is not a requirement, mirroring the cluster node's disks precludes against a non-mirrored disk failure taking down the node. The downside to mirroring a cluster node's local disks is more system administration overhead.

- **What are the cluster member backup issues?**

  You can use several backup methods for a cluster. One method is to have a node as the backup node with a tape drive/library attached. Then use the cluster file system to back up the data. Do not connect this node to the shared disks.

  See the *Sun Cluster 3.0 System Administration Guide* for additional information on backup and restore procedures.

Cluster Storage FAQ

- **What makes multihost storage highly available?**

  Multihost storage is highly available because it can survive the loss of a single disk due to mirroring (or due to hardware-based RAID-5 controllers). Because a multihost storage device has more than one host connection, it can also withstand the loss of a single node to which it is connected.

- **What multihost storage configurations are supported?**

  Currently, greater than two-node connectivity is not supported. All multihosted disks within a single enclosure must connect to the same two nodes. Refer to "Sun Cluster Topologies" on page 27 for more information.

- **Can I use disks configured for SCSI-3 PGR as global devices?**

  Currently, SCSI-3 PGR is not supported in Sun Cluster. Only SCSI-2 semantics are supported for global disk devices. Since SCSI-3 disks are not supported, you must use the `–R` option to `scdidadm(1M)` to set the correct SCSI semantics for any SCSI-3 disks that you want to use as global devices in a cluster.

Cluster Interconnect FAQ

- **What cluster interconnects does Sun Cluster support?**

  Currently Sun Cluster supports Ethernet (100BASE-T Fast Ethernet and 1000BASE-SX Gb) cluster interconnects. Support is also planned for Scalable Coherent Interface (SCI).

Client Systems FAQ

- **Do I need to consider any special client needs or restrictions for use with a cluster?**

  Client systems connect to the cluster as they would any other server. In some instances, depending on the data service application, you might need to install client-side software or perform other configuration changes so that the client can connect to the data service application. See individual chapters in *Sun Cluster 3.0 Data Services Installation and Configuration Guide* for more information on client-side configuration requirements.

Administrative Console FAQ

- **Does Sun Cluster require an administrative console?**

  Yes.

- **Does the administrative console have to be dedicated to the cluster, or can it be used for other tasks?**

  Sun Cluster does not require a dedicated administrative console, but using one provides these benefits:

  - Enables centralized cluster management by grouping console and management tools on the same machine
  - Provides potentially quicker problem resolution by your hardware service provider

- **Does the administrative console need to be located "close" to the cluster itself, for example, in the same room?**

  Check with your hardware service provider. The provider might require that the console be located in close proximity to the cluster itself. No technical reason exists for the console to be located in the same room.

- **Can an administrative console serve more than one cluster, as long as any distance requirements are also first met?**

  Yes. You can control multiple clusters from a single administrative console. You can also share a single terminal concentrator between clusters.

Terminal Concentrator and System Service Processor FAQ

- **Does Sun Cluster require a terminal concentrator?**

  Sun Cluster 3.0 does not require a terminal concentrator to run. Unlike the Sun Cluster 2.2 product, which required a terminal concentrator for failure fencing, Sun Cluster 3.0 does not depend on the terminal concentrator.

- **I see that most Sun Cluster servers use a terminal concentrator, but the E10000 does not. Why is that?**

  The terminal concentrator is effectively a serial-to-Ethernet converter for most servers. Its console port is a serial port. The Sun Enterprise E10000 server doesn't have a serial console. The System Service Processor (SSP) is the console, either through an Ethernet or `jtag` port. For the Sun Enterprise E10000 server, you always use the SSP for consoles.

- **What are the benefits of using a terminal concentrator?**

  Using a terminal concentrator provides console-level access to each node from a remote workstation anywhere on the network, including when the node is at the OpenBoot PROM (OBP).

- **If I use a terminal concentrator not supported by Sun, what do I need to know to qualify the one that I want to use?**

  The main difference between the terminal concentrator supported by Sun and other console devices is that the Sun terminal concentrator has special firmware that prevents the terminal concentrator from sending a break to the console when it boots. Note that if you have a console device that can send a break, or a signal that might be interpreted as a break to the console, it shuts down the node.

- **Can I free a locked port on the terminal concentrator supported by Sun without rebooting it?**

  Yes. Note the port number that needs to be reset and do the following:

```
telnet tc
Enter Annex port name or number: cli
annex: su -
annex# admin
admin : reset port_number
admin : quit
annex# hangup
#
```

  Refer to the *Sun Cluster 3.0 System Administration Guide* for more information about configuring and administering the terminal concentrator supported by Sun.

- **What if the terminal concentrator itself fails? Must I have another one standing by?**

  No. You do not lose any cluster availability if the terminal concentrator fails. You do lose the ability to connect to the node consoles until the concentrator is back in service.

- **If I do use a terminal concentrator, what about security?**

  Generally, the terminal concentrator is attached to a small network used by system administrators, not a network that is used for other client access. You can control security by limiting access to that particular network.

Glossary

This glossary of terms is used in the Sun Cluster 3.0 documentation.

A

administrative console A workstation that is used to run cluster administrative software.

amnesia A condition in which a cluster restarts after a shutdown with stale cluster configuration data (CCR). For example, on a two-node cluster with only node 1 operational, if a cluster configuration change occurs on node 1, node 2's CCR becomes stale. If the cluster is shut down then restarted on node 2, an amnesia condition results because of node 2's stale CCR.

automatic failback A process of returning a resource group or device group to its primary node after the primary node has failed and later is restarted as a cluster member.

B

backup group See "Network Adapter Failover group."

C

checkpoint The notification sent by a primary node to a secondary node to keep the software state synchronized between them. See also "primary" and "secondary."

cluster Two or more interconnected nodes or domains that share a cluster file system and are configured together to run failover, parallel, or scalable resources.

Cluster Configuration Repository (CCR) — A highly available, replicated data store that is used by Sun Cluster software to persistently store cluster configuration information.

cluster file system — A cluster service that provides cluster-wide, highly available access to existing local file systems.

cluster interconnect — The hardware networking infrastructure that includes cables, cluster transport junctions, and cluster transport adapters. The Sun Cluster and data service software use this infrastructure for intra-cluster communication.

cluster member — An active member of the current cluster incarnation. This member is capable of sharing resources with other cluster members and providing services both to other cluster members and to clients of the cluster. See also “cluster node.”

Cluster Membership Monitor (CMM) — The software that maintains a consistent cluster membership roster. This membership information is used by the rest of the clustering software to decide where to locate highly available services. The CCM ensures that non-cluster members cannot corrupt data and transmit corrupt or inconsistent data to clients.

cluster node — A node that is configured to be a cluster member. A cluster node might or might not be a current member. See also “cluster member.”

cluster transport adapter — The network adapter that resides on a node and connects the node to the cluster interconnect. See also “cluster interconnect.”

cluster transport cables — The network connection that connects to the endpoints. A connection between cluster transport adapters and cluster transport junctions or between two cluster transport adapters. See also “cluster interconnect.”

cluster transport junction — A hardware switch that is used as part of the cluster interconnect. See also “cluster interconnect.”

collocation — The property of being on the same node. This concept is used during cluster configuration to improve performance.

D

data service — An application that has been instrumented to run as a highly available resource under control of the Resource Group Manager (RGM).

default master The default cluster member on which a failover resource type is brought online.

device group A user-defined group of device resources, such as disks, that can be mastered from different nodes in a cluster HA configuration. This group can include device resources of disks, Solstice DiskSuite disksets, and VERITAS Volume Manager disk groups.

device id A mechanism of identifying devices that are made available via Solaris. Device ids are described in the `devid_get(3DEVID)` man page.

The Sun Cluster DID driver uses device ids to determine correlation between the Solaris logical names on different cluster nodes. The DID driver probes each device for its device id. If that device id matches another device somewhere else in the cluster, both devices are given the same DID name. If the device id hasn't been seen in the cluster before, a new DID name is assigned. See also "Solaris logical name" and "DID driver."

DID driver A driver implemented by Sun Cluster used to provide a consistent device namespace across the cluster. See also "DID name."

DID name Used to identify global devices in Sun Cluster. It is a clustering identifier with a one-to-one or a one-to-many relationship with Solaris logical names. It takes the form d*X*s*Y*, where *X* is an integer and *Y* is t he slice name. See also "Solaris logical name."

disk device group See "device group."

Distributed Lock Manager (DLM) The locking software used in a shared disk Oracle Parallel Server (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.

diskset See "device group."

disk group See "device group."

E

endpoint — A physical port on a cluster transport adapter or cluster transport junction.

event — A change in the state, mastery, severity, or description of a managed object.

F

failback — See “automatic failback.”

failfast — The orderly shutdown and removal from the cluster of a faulty node before its potentially incorrect operation can prove damaging.

failover — The automatic relocation of a resource group or a device group from a current primary node to a new primary node after a failure has occurred.

failover resource — A resource, each of whose resources can correctly be mastered by only one node at a time. See also “single instance resource” and “scalable resource.”

fault monitor — A fault daemon and the programs used to probe various parts of data services and take action. See also “resource monitor.”

G

generic resource type — A template for a data service. A generic resource type can be used to make a simple application into a failover data service (stop on one node, start on another). This type does not require programming by the Sun Cluster API.

generic resource — An application daemon and its child processes put under control of the Resource Group Manager as part of a generic resource type.

global device — A device that is accessible from all cluster members, such as disk, CD-ROM, and tape.

global device namespace — A namespace that contains the logical, cluster-wide names for global devices. Local devices in the Solaris environment are defined in the `/dev/dsk`, `/dev/rdsk`, and `/dev/rmt` directories. The global device namespace defines global devices in the `/dev/global/dsk`, `/dev/global/rdsk`, and `/dev/global/rmt` directories.

global interface A global network interface that physically hosts shared addresses. See also “shared address.”

global interface node A node hosting a global interface.

global resource A highly available resource provided at the kernel level of the Sun Cluster software. Global resources can include disks (HA device groups), the cluster file system, and global networking.

H

HA data service See “data service.”

heartbeat A periodic message sent across all available cluster interconnnect transport paths. Lack of a heartbeat after a specified interval and number of retries might trigger an internal failover of transport communication to another path. Failure of all paths to a cluster member results in the CMM reevaluating the cluster quorum.

I

instance See “resource invocation.”

L

load balancing Applies only to scalable services. The process of distributing the application load across nodes in the cluster so that the client requests are serviced in a timely manner. Refer to “Scalable Data Services” on page 50 for more details.

load-balancing policy Applies only to scalable services. The preferred way in which application request load is distributed across nodes. Refer to “Scalable Data Services” on page 50 for more details.

local disk A disk that is physically private to a given cluster node.

logical host A Sun Cluster 2.0 (minimum) concept that includes an application, the disksets, or disk groups on which the application data resides, and the network addresses used to access the cluster. This concept no longer exists in Sun Cluster 3.0. Refer to “Disk Device Groups” on page 38 and “Resources and Resource Types” on page 56 for a description of how this concept is implemented in Sun Cluster 3.0.

logical hostname resource — A resource that contains a collection of logical hostnames representing network addresses. Logical hostname resources can only be mastered by one node at a time. See also “logical host.”

logical network interface — In the Internet architecture, a host can have one or more IP addresses. Sun Cluster configures additional logical network interfaces to establish a mapping between several logical network interfaces and a single physical network interface. Each logical network interface has a single IP address. This mapping enables a single physical network interface to respond to multiple IP addresses. This mapping also enables the IP address to move from one cluster member to the other in the event of a takeover or switchover without requiring additional hardware interfaces.

M

master — See “primary.”

metadevice state database replica (replica) — A database, stored on disk, that records configuration and state of all metadevices and error conditions. This information is important to the correct operation of Solstice DiskSuite disksets and it is replicated.

multihomed host — A host that is on more than one public network.

multihost disk — A disk that is physically connected to multiple nodes.

N

Network Adapter Failover (NAFO) group — A set of one or more network adapters on the same node and on the same subnet configured to back up each other in the event of an adapter failure.

network address resource — See “network resource.”

network resource — A resource that contains one or more logical hostnames or shared addresses. See also “logical hostname resource” and “shared address resource.”

node — A physical machine or domain (in the Sun Enterprise E10000 server) that can be part of a Sun cluster. Also called “host.”

non-cluster mode — The resulting state achieved by booting a cluster member with the –x boot option. In this state the node is no longer a cluster member,

but is still a cluster node. See also “cluster member” and “cluster node.”

P

parallel resource type — A resource type, such as a parallel database, that has been instrumented to run in a cluster environment so that it can be mastered by multiple (two or more) nodes simultaneously.

parallel service instance — An instance of a parallel resource type running on an individual node.

potential master — See “potential primary.”

potential primary — A cluster member that is able to master a failover resource type if the primary node fails. See also “default master.”

primary — A node on which a resource group or device group is currently online. That is, a primary is a node that is currently hosting or implementing the service associated with the resource. See also “secondary.”

primary host name — The name of a node on the primary public network. This is always the node name specified in `/etc/nodename`. See also, “secondary host name.”

private hostname — The hostname alias used to communicate with a node over the cluster interconnect.

Public Network Management (PNM) — Software that uses fault monitoring and failover to prevent loss of node availability because of single network adapter or cable failure. PNM failover uses sets of network adapters called Network Adapter Failover groups to provide redundant connections between a cluster node and the public network. The fault monitoring and failover capabilities work together to ensure availability of resources. See also “Network Adapter Failover group.”

Q

quorum device — A disk shared by two or more nodes that contributes votes used to establish a quorum for the cluster to run. The cluster can operate only when a quorum of votes is available. The quorum device is used when a cluster becomes partitioned into separate sets of nodes to establish which set of nodes constitutes the new cluster.

R

resource — An instance of a resource type. Many resources of the same type might exist, each resource having its own name and set of property values, so that many instances of the underlying application might run on the cluster.

resource group — A collection of resources that are managed by the RGM as a unit. Each resource that is to be managed by the RGM must be configured in a resource group. Typically, related and interdependent resources are grouped.

Resource Group Manager (RGM) — A software facility used to make cluster resources highly available and scalable by automatically starting and stopping these resources on selected cluster nodes. The RGM acts according to pre-configured policies, in the event of hardware or software failures or reboots.

resource group state — The state of the resource group on any given node.

resource invocation — An instance of a resource type running on a node. An abstract concept representing a resource that was started on the node.

Resource Management API (RMAPI) — The application programming interface within Sun Cluster that makes an application highly available in a cluster environment.

resource monitor — An optional part of a resource type implementation that runs periodic fault probes on resources to determine if they are running correctly and how they are performing.

resource state — The state of a Resource Group Manager resource on a given node.

resource status — The condition of the resources as reported by the fault monitor.

resource type — The unique name given to a data service, LogicalHostname, or SharedAddress cluster object. Data service resource types can either be failover types or scalable types. See also “data service,” “failover resource,” and “scalable resource.”

resource type property — A key-value pair, stored by the RGM as part of the resource type, that is used to describe and manage resources of the given type.

S

Scalable Coherent Interface (SCI) A high-speed interconnect hardware used as the cluster interconnect.

scalable resource A resource that runs on multiple nodes (an instance on each node) that uses the cluster interconnect to give the appearance of a single service to remote clients of the service.

scalable service A data service implemented that runs on multiple nodes simultaneously.

secondary A cluster member that is available to master disk device groups and resource groups in the event that the primary fails. See also “primary.”

secondary host name The name used to access a node on a secondary public network. See also “primary host name.”

shared address resource A network address that can be bound by all scalable services running on nodes within the cluster to make them scale on those nodes. A cluster can have multiple shared addresses, and a service can be bound to multiple shared addresses.

single instance resource A resource for which at most one resource may be active across the cluster.

Solaris logical name The names typically used to manage Solaris devices. For disks, these usually look something like `/dev/rdsk/c0t2d0s2`. For each one of these Solaris logical device names, there is an underlying Solaris physical device name. See also “DID name” and “Solaris physical name.”

Solaris physical name The names that is given to a device by its device driver in Solaris. This shows up on a Solaris machine as a path under the `/devices` tree. For example, a typical SCSI disk has a Solaris physical name of something like: `/devices/sbus@1f,0/SUNW,fas@e,8800000/sd@6,0:c,raw`

See also “Solaris logical name.”

Solstice DiskSuite A volume manager used by Sun Cluster. See also “volume manager.”

split brain A condition in which a cluster breaks up into multiple partitions, with each partition forming without knowledge of the existence of any other.

switchback — See “failback.”

switchover — The orderly transfer of a resource group or device group from one master (node) in a cluster to another master (or multiple masters, if resource groups are configured for multiple primaries). A switchover is initiated by an administrator by using the `scswitch(1M)` command.

System Service Processor (SSP) — In Enterprise 10000 configurations, a device, external to the cluster, used specifically to communicate with cluster members.

T

takeover — See “failover.”

terminal concentrator — In non-Enterprise 10000 configurations, a device that is external to the cluster, used specifically to communicate with cluster members.

V

VERITAS Volume Manager — A volume manager used by Sun Cluster. See also “volume manager.”

volume manager — A software product that provides data reliability through disk striping, concatenation, mirroring, and dynamic growth of metadevices or volumes.