Sun Cluster 3.0 12/01 Concepts

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

Part No. 816-2027-10
December 2001, Revision A

Contents

Preface

Sun™ Cluster 3.0 12/01 Concepts contains conceptual and reference information for the SunPlex™ system. The SunPlex system includes all hardware and software components that make up Sun's cluster solution.

This document is intended for experienced system administrators who are trained on Sun Cluster software. 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 volume manager software used with the SunPlex system.

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:`
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 12/01 Software Installation Guide*	816-2022
Hardware	*Sun Cluster 3.0 12/01 Hardware Guide*	816-2023
Data Services	*Sun Cluster 3.0 12/01 Data Services Installation and Configuration Guide*	816-2024
API Development	*Sun Cluster 3.0 12/01 Data Services Developer's Guide*	816-2025
Administration	*Sun Cluster 3.0 12/01 System Administration Guide*	816-2026
Error Messages and Problem Resolution	*Sun Cluster 3.0 12/01 Error Messages Guide*	816-2028
Release Notes	*Sun Cluster 3.0 12/01 Release Notes*	816-2029

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:

Getting Help

If you have problems installing or using the SunPlex system, 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 the Sun Cluster software (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 software release and package version information

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

CHAPTER 1

Introduction and Overview

The SunPlex system is an integrated hardware and Sun Cluster software solution that is used to create highly available and scalable services.

Sun Cluster 3.0 12/01 Concepts provides the conceptual information needed by the primary audience for SunPlex 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 failover and scalable services for applications not currently included with the Sun Cluster product

This book works with the rest of the SunPlex documentation set to provide a complete view of the SunPlex system.

This chapter

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

Introduction to the SunPlex System

The SunPlex system extends the Solaris operating environment into a cluster operating system. A cluster, or plex, 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 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 failover and scalable services, capacity for modular growth, and low entry price compared to traditional hardware fault-tolerant systems.

The goals of the SunPlex system 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 Versus Fault Tolerance

The SunPlex system 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.

The SunPlex system 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 services and their dependent system resources, and fails over or restarts services in case of failures.

Refer to “High Availability FAQ” on page 65 for questions and answers on high availability.

Failover and Scalability in the SunPlex System

The SunPlex system enables you to implement either *failover* or *scalable* services. In general, a failover service provides only high availability (redundancy), whereas a scalable service provides high availability along with increased performance. A single cluster can support both failover and scalable services.

Failover Services

Failover is the process by which the cluster automatically relocates a service from a failed primary node to a designated secondary node. With failover, Sun Cluster software 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 that provides the service.

Scalable Services

While failover is concerned with redundancy, scalability provides constant response time or throughput without regard to load. A scalable service 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 42 for more specific information on failover and scalable services.

Three Viewpoints of the SunPlex System

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

- Hardware installation and service personnel
- System administrators
- Application programmers

Hardware Installation and Service Viewpoint

To hardware service people, the SunPlex system 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 11
- "Multihost Disks" on page 13
- "Local Disks" on page 15
- "Cluster Interconnect" on page 15
- "Public Network Interfaces" on page 16

Relevant SunPlex Documentation

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

- *Sun Cluster 3.0 12/01 Hardware Guide*

System Administrator Viewpoint

To the system administrator, the SunPlex system 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 that monitors the health of user application programs running on the cluster nodes
- Volume management software that sets up and administers disks
- Specialized cluster software that enables all nodes to access all storage devices, even those not directly connected to disks
- Specialized cluster software that enables 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 software
  - 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:

Relevant SunPlex Documentation – System Administrator

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

- *Sun Cluster 3.0 12/01 Software Installation Guide*
- *Sun Cluster 3.0 12/01 System Administration Guide*
- *Sun Cluster 3.0 12/01 Error Messages Guide*

Application Programmer Viewpoint

The SunPlex system provides *data services* for such applications as Oracle, NFS, DNS, iPlanet™ Web Server, Apache Web Server, and Netscape™ Directory Server. Data services are created by configuring an off-the-shelf applications to run under control of the Sun Cluster software. The Sun Cluster software provides configuration files and management methods that start, stop, and monitor the applications.

If you need to create a new failover or scalable service, you can use the SunPlex Application Programming Interface (API) and the Data Service Enabling Technologies API (DSET API) to develop the necessary configuration files and management methods that enable its application to run as a 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 failover or scalable data service.
- The Sun Cluster API, DSET 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 42
- "Resources, Resource Groups, and Resource Types" on page 54
- "Frequently Asked Questions" on page 65

Relevant SunPlex Documentation – Application Programmer

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

- *Sun Cluster 3.0 12/01 Data Services Developer's Guide*
- *Sun Cluster 3.0 12/01 Data Services Installation and Configuration Guide*

SunPlex System Tasks

All SunPlex system 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 12/01 Hardware Guide*
Install Solaris software on the cluster	*Sun Cluster 3.0 12/01 Software Installation Guide*
Install Sun™ Management Center software	*Sun Cluster 3.0 12/01 Software Installation Guide*
Install and configure Sun Cluster software	*Sun Cluster 3.0 12/01 Software Installation Guide*
Install and configure volume management software	*Sun Cluster 3.0 12/01 Software Installation Guide* Your volume management documentation
Install and configure Sun Cluster data services	*Sun Cluster 3.0 12/01 Data Services Installation and Configuration Guide*
Service cluster hardware	*Sun Cluster 3.0 12/01 Hardware Guide*
Administer Sun Cluster software	*Sun Cluster 3.0 12/01 System Administration Guide*
Administer volume management software	*Sun Cluster 3.0 12/01 System Administration Guide* and your volume management documentation
Administer application software	Your application documentation
Problem identification and suggested user actions	*Sun Cluster 3.0 12/01 Error Messages Guide*
Create a new data service	*Sun Cluster 3.0 12/01 Data Services Developer's Guide*

CHAPTER 2

Key Concepts – Hardware Service Providers

This chapter describes the key concepts related to the hardware components of a SunPlex system configuration. The topics covered include:

The SunPlex System 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

The SunPlex system enables you to combine these components into a variety of configurations, described in "Sun Cluster Topology Examples" on page 19.

The following figure shows a sample cluster configuration.

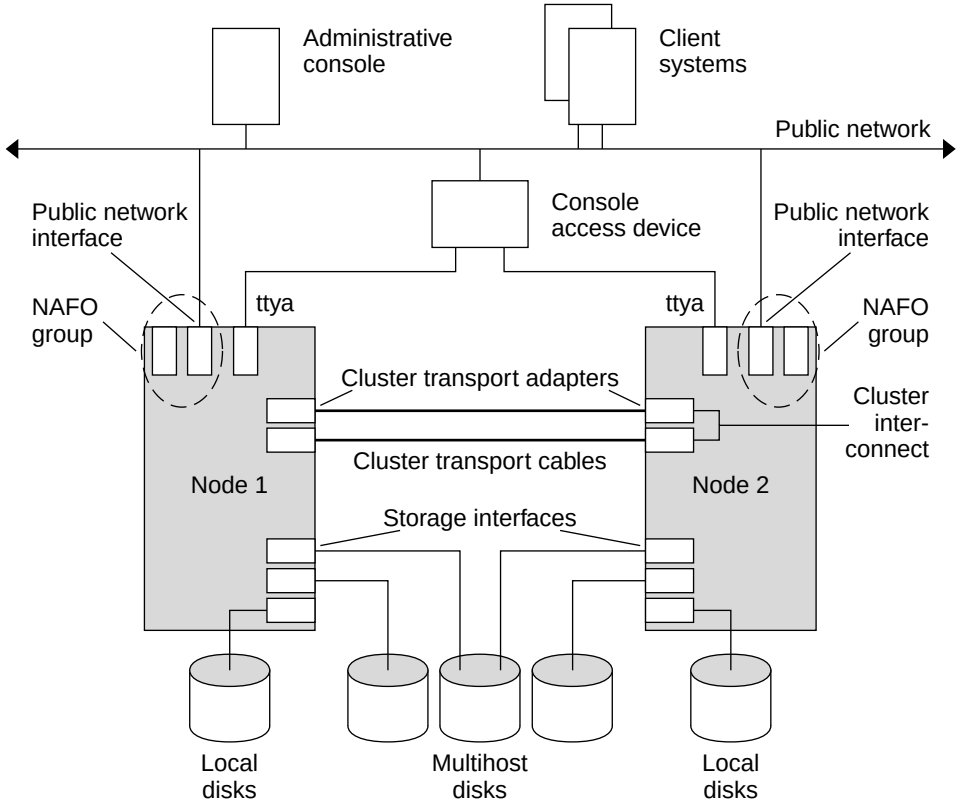

FIGURE 2-1 Sample Two-Node Cluster Configuration

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 Topology Examples" on page 19 for the supported node configurations.

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

In addition, nodes in parallel database configurations share concurrent access to all the disks. See "Multihost Disks" on page 13 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. This set of physically independent networks 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.

Nodes in the same cluster should have similar processing, memory, and I/O capability to enable failover to occur without significant degradation in performance. Because of the possibility of failover, every node must have 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 Hardware Members

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

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

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 12/01 Software Installation Guide* for information on how to install the Solaris operating environment, Sun Cluster, and volume management software.

See the *Sun Cluster 3.0 12/01 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 figure provides a high-level view of the software components that work together to create the Sun Cluster software environment.

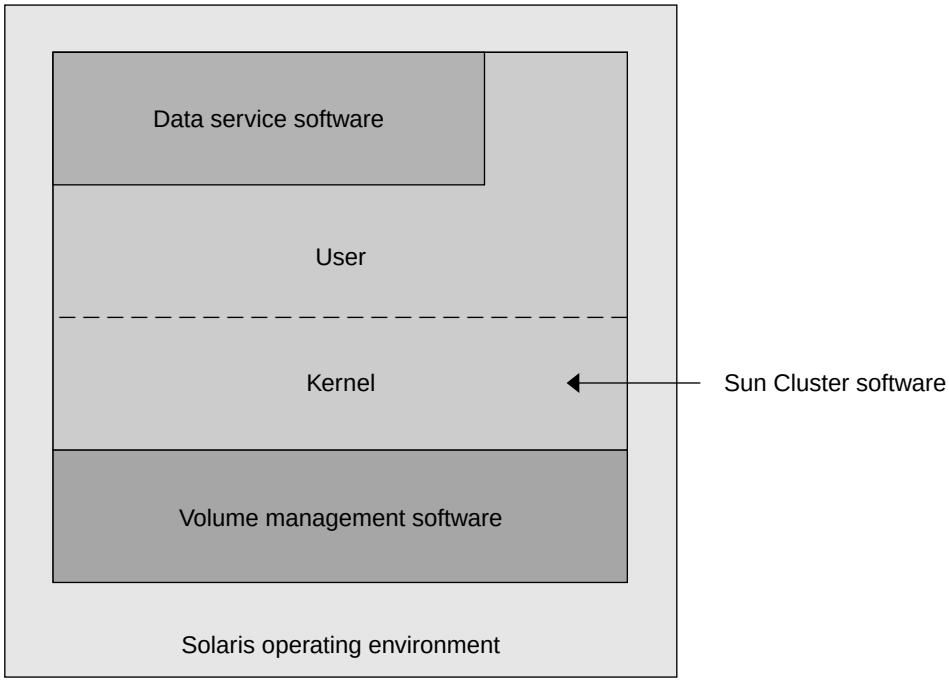

FIGURE 2-2 High-Level Relationship of Sun Cluster Software Components

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 disks highly available.

Multihost disks have the following characteristics.

- They can tolerate single node failures.
- They store application data and can also store application binaries and configuration files.
- They protect 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.

- They 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.

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, using multihost disks. When the cluster storage consists of singled-ended or differential SCSI devices, the configuration is referred to as multi-initiator SCSI. As this terminology implies, more than one SCSI initiator exists on the SCSI bus.

The SCSI specification requires that each device on a SCSI bus has a unique SCSI address. (The host adapter is also a device on the SCSI bus.) The default hardware configuration in a multi-initiator environment results in a conflict because all SCSI host adapters default to 7.

To resolve this conflict, on each SCSI bus, leave one of the SCSI host adapters with the SCSI address of 7, and set the other host adapters to unused SCSI addresses. Proper planning dictates that these "unused" SCSI addresses include both currently and eventually unused addresses. An example of addresses unused in the future is the addition of storage by installing new drives into empty drive slots. In most configurations, the available SCSI address for a second host adapter is 6.

You can change the selected SCSI addresses for these host adapters by setting the `scsi-initiator-id` Open Boot PROM (OBP) property. You can set this property globally for a node or on a per-host-adapter basis. Instructions for setting a unique `scsi-initiator-id` for each SCSI host adapter are included in the chapter for each disk enclosure in the *Sun Cluster 3.0 12/01 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 local 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 lets you mirror these disks so that a failure cannot cause these file systems to become inaccessible, but volume managers do not protect against node failure.

See the section"Global Devices" on page 30 for more information about global devices.

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 12/01 Hardware Guide* for information on installing and configuring removable media.

See the section"Global Devices" on page 30 for more information about global devices.

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 cluster interconnect. The Sun Cluster security model assumes that only cluster nodes have physical access to the cluster interconnect.

All nodes must be connected by the cluster interconnect through at least two redundant physically independent networks, or paths, to avoid a single point of failure. You can have several physically independent networks (two to six) between any two nodes. The cluster interconnect consists of three hardware components: adapters, junctions, and cables.

The following list describes each of these hardware components.

- Adapters – The network adapter cards that reside in each cluster node. Their names are constructed from a device name immediately followed by a physical-unit number, for example, qfe2. 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.

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, the system controller on Sun Fire™ servers, or another device that can access `ttya` on each node.

Only one supported terminal concentrator is available from Sun and use of the supported Sun terminal concentrator is optional. The terminal concentrator enables access to `/dev/console` 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 machine 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.

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 12/01 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 17 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.

Sun Cluster Topology Examples

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
- Pair+M
- N+1 (star)

The following sections include sample diagrams of 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 cluster interconnect and operate under Sun Cluster software control. You might use this topology to run a parallel database application on one pair and a failover or scalable 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.

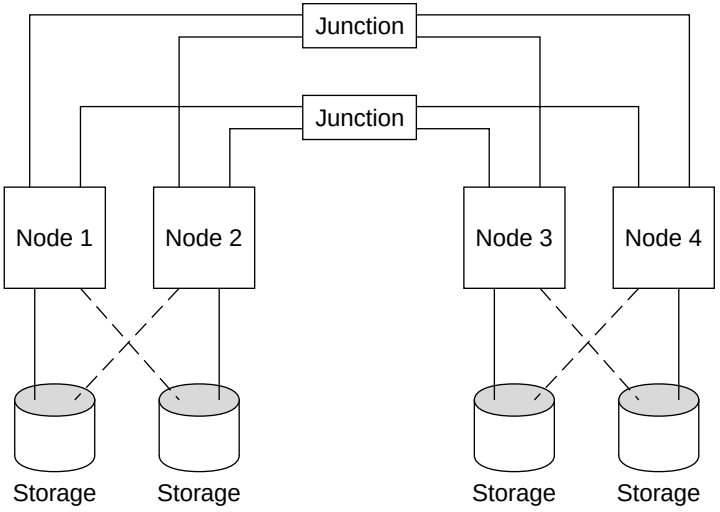

FIGURE 2-3 Clustered Pairs Topology

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.

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.

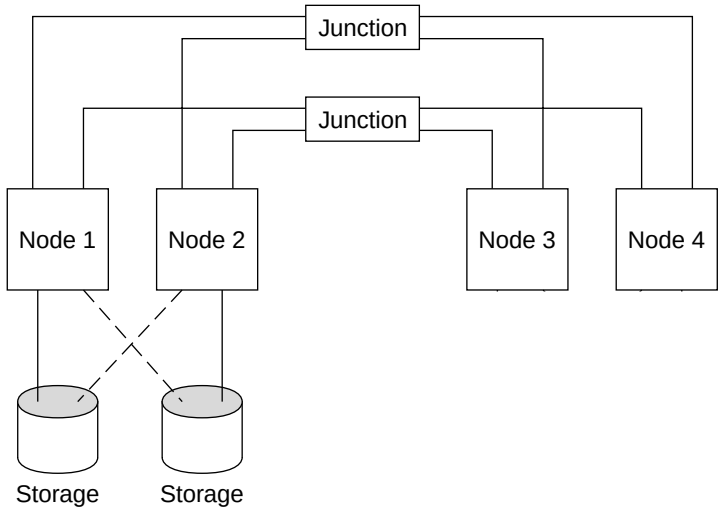

FIGURE 2-4 Pair+M Topology

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.

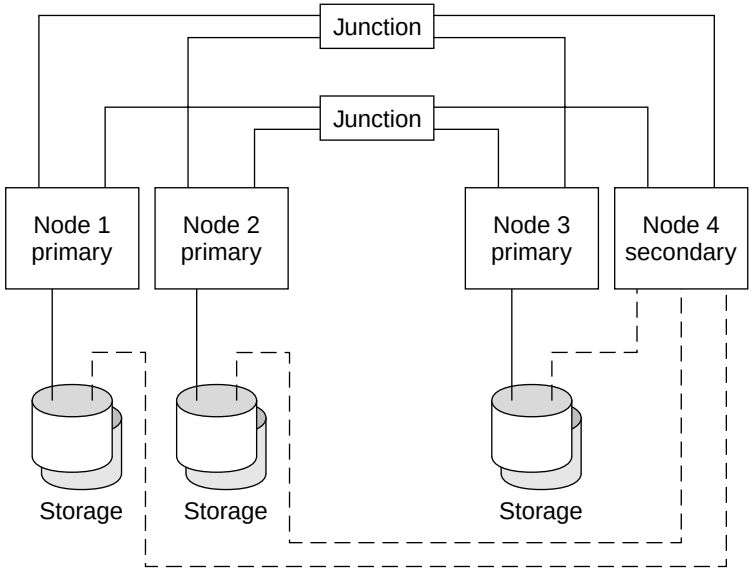

FIGURE 2-5 N+1 Topology

CHAPTER 3

Key Concepts – Administration and Application Development

This chapter describes the key concepts related to the software components of a SunPlex system. The topics covered include:

Cluster Administration and Application Development

This information is directed primarily toward system administrators and application developers using the SunPlex 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.

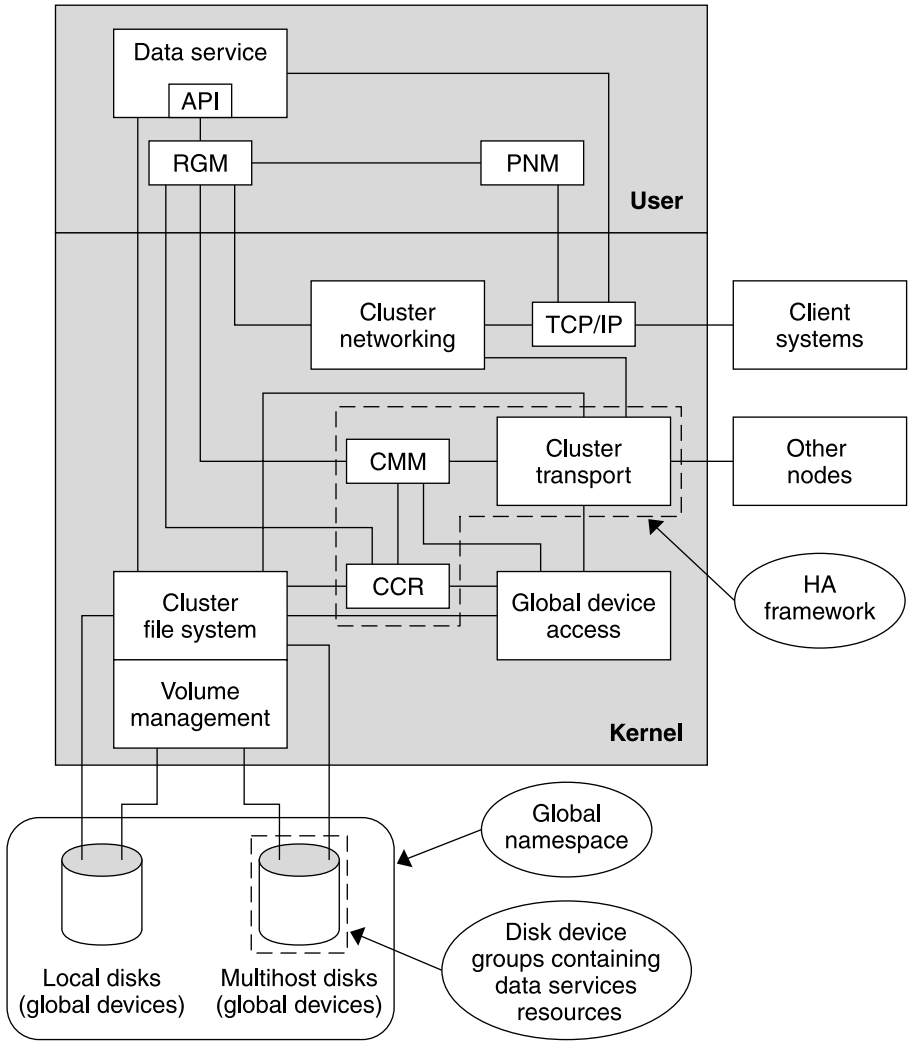

FIGURE 3-1 Sun Cluster Software Architecture

Administrative Interfaces

You can choose how you install, configure, and administer the SunPlex system from several user interfaces. 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. The SunPlex system 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 12/01 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. The SunPlex system 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 software using `scinstall(1M)`, one step in the process is to configure NTP for the cluster. Sun Cluster software 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 host names 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 12/01 Software 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 12/01 System Administration Guide*.

High-Availability Framework

The SunPlex system 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 SunPlex component failures (both hardware and software) and the kinds of recovery built into the high-availability framework.

TABLE 3-1 Levels of SunPlex Failure Detection and Recovery

Failed Cluster Component	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

Sun Cluster software's 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 software 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. This constraint is called the *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 37 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.

Failfast Mechanism

If the CMM detects a critical problem with a node, it calls upon the cluster framework to forcibly shut down (panic) the node and to remove it from the cluster membership. The mechanism by which this occurs is called *failfast*. Failfast will cause a node to shut down in two ways.

- If a node leaves the cluster and then attempts to start a new cluster without having quorum, it is "fenced" from accessing the shared disks. See "Failure Fencing" on page 40 for details on this use of failfast.
- If one or more cluster-specific daemons die (`clexecd`, `rpc.pmfd`, `rgmd`, or `rpc.ed`) the failure is detected by the CMM and the node panics.

When a panics due to the death of a cluster daemon, a message similar to the following will display on the console for that node.

```
panic[cpu0]/thread=40e60: Failfast: Aborting because "pmfd" died 35 seconds ago.
409b8 cl_runtime:__0FZsc_syslog_msg_log_no_argsPviTCPCcTB+48 (70f900, 30,
70df54, 407acc, 0)
%l0-7: 1006c80 000000a 000000a 10093bc 406d3c80 7110340 0000000 4001 fbf0
```

After the panic, the node might reboot and attempt to rejoin the cluster or stay at the OpenBoot PROM (OBP) prompt. The action taken is determined by the setting of the `auto-boot?` parameter in the OBP.

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 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 consists of text files, never edit the CCR files manually. Each file contains a checksum record to ensure consistency between nodes. 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

The SunPlex system 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, the Sun Cluster software automatically discovers another path to the device and redirects the access to that path. SunPlex 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 33 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)

The Sun Cluster software 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 accessing disks (such as a volume manager or applications using raw devices) uses a consistent path across the cluster. 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 assigns 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 the SunPlex system, all multihost disks must be under control of the Sun Cluster software. You first create volume manager disk groups—either Solstice DiskSuite disk sets or VERITAS Volume Manager disk groups—on the multihost disks. Then, you register the volume manager disk groups as *disk device groups*. A disk device group is a type of global device. In addition, the Sun Cluster software automatically creates a rawdisk device group for each disk and tape device in the cluster. However, these cluster device groups remain in an offline state until you access them as global devices.

Registration provides the SunPlex system information about which nodes have a path to what volume manager disk groups. At this point, the volume manager disk groups become globally accessible within the cluster. If more than one node can write to (master) a disk device group, the data stored in that disk device group becomes highly available. The highly available disk device group can be used to house cluster file systems.

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 12/01 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.

Disk Device Group 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.

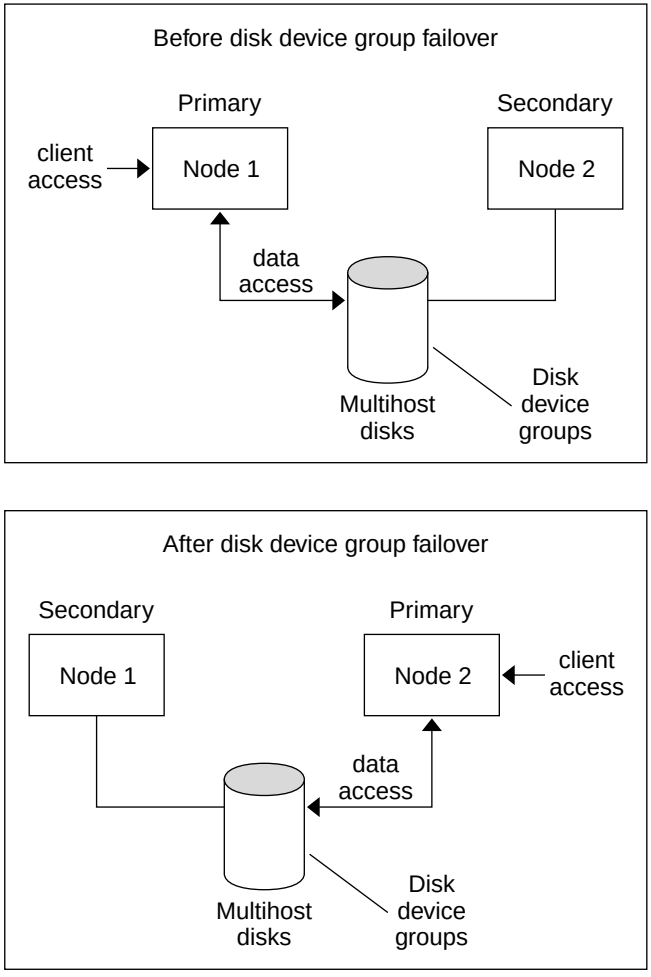

FIGURE 3-2 Disk Device Group Failover

Global Namespace

The Sun Cluster software 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 the SunPlex system, 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 software 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`
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.

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

Using Cluster File Systems

In the SunPlex system, all multihost disks are placed into disk device groups, which can be Solstice DiskSuite disksets, VxVM disk groups, or individual disks not under control of a software-based volume manager.

For a cluster file system to be highly available, the underlying disk storage must be connected to more than one node. Therefore, a local file system (a file system that is stored on a node's local disk) that is made into a cluster file system is not highly available.

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

- **Manually**—Use the `mount` command and the `-g` or `-o global` mount options 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 software 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 12/01 Software Installation Guide* and *Sun Cluster 3.0 12/01 System Administration Guide* for more information.

Cluster File System Features

The cluster file system has the following features:

- File access locations are 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.

Note – When the cluster file system reads files, it does not update the access time on those files.

- Coherency protocols are used to preserve the UNIX file access semantics even if the file is accessed concurrently from multiple nodes.
- Extensive caching is used along with zero-copy bulk I/O movement to move file data efficiently.
- The cluster file system provides highly available advisory file locking functionality using the `fcntl(2)` interfaces. Applications running on multiple cluster nodes can synchronize access to data using advisory file locking on a cluster file system file. File locks are recovered immediately from nodes that leave the cluster, and from applications that fail while holding locks.
- Continuous access to data is ensured, even when failures occur. Applications are not affected by failures as long as a path to disks is still operational. This guarantee is maintained for raw disk access and all file system operations.
- Cluster file systems are independent from the underlying file system and volume management software. Cluster file systems make any supported on-disk file system global.

The Syncdir Mount Option

The `syncdir` mount option can be used for cluster file systems that use UFS as the underlying file system. 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 NFS 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 66 for frequently asked questions about global devices and cluster file systems.

Quorum and Quorum Devices

Because cluster nodes share data and resources, it is important that a cluster never splits into separate partitions that are active at the same time. The CMM guarantees that at most one cluster is operational at any time, even if the cluster interconnect is partitioned.

There are two types of problems that arise from cluster partitions: split brain and amnesia. 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. This occurs due to communication problems between cluster nodes. Amnesia occurs when the cluster restarts after a shutdown with cluster data older than at the time of the shutdown. This can happen if multiple versions of the framework data are stored on disk and a new incarnation of the cluster is started when the latest version is not available.

Split brain and amnesia can be avoided by giving each node one vote and mandating a majority of votes for an operational cluster. A partition with the majority of votes has a *quorum* and is allowed to operate. This majority vote mechanism works fine as long as there are more than two nodes in the cluster. In a two-node cluster, a majority is two. If such a cluster becomes partitioned, an external vote is needed for either partition to gain quorum. This external vote is provided by a *quorum device*. A quorum device can be any disk that is shared between the two nodes. Disks used as quorum devices can contain user data.

TABLE 3-3 describes how Sun Cluster software uses quorum to avoid split brain and amnesia.

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

Partition Type	Quorum Solution
Split brain	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	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)

The quorum algorithm operates dynamically: as cluster events trigger its calculations, the results of calculations can change over the lifetime of a cluster.

Quorum Vote Counts

Both cluster nodes and quorum devices 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 12/01 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 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 software. 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.

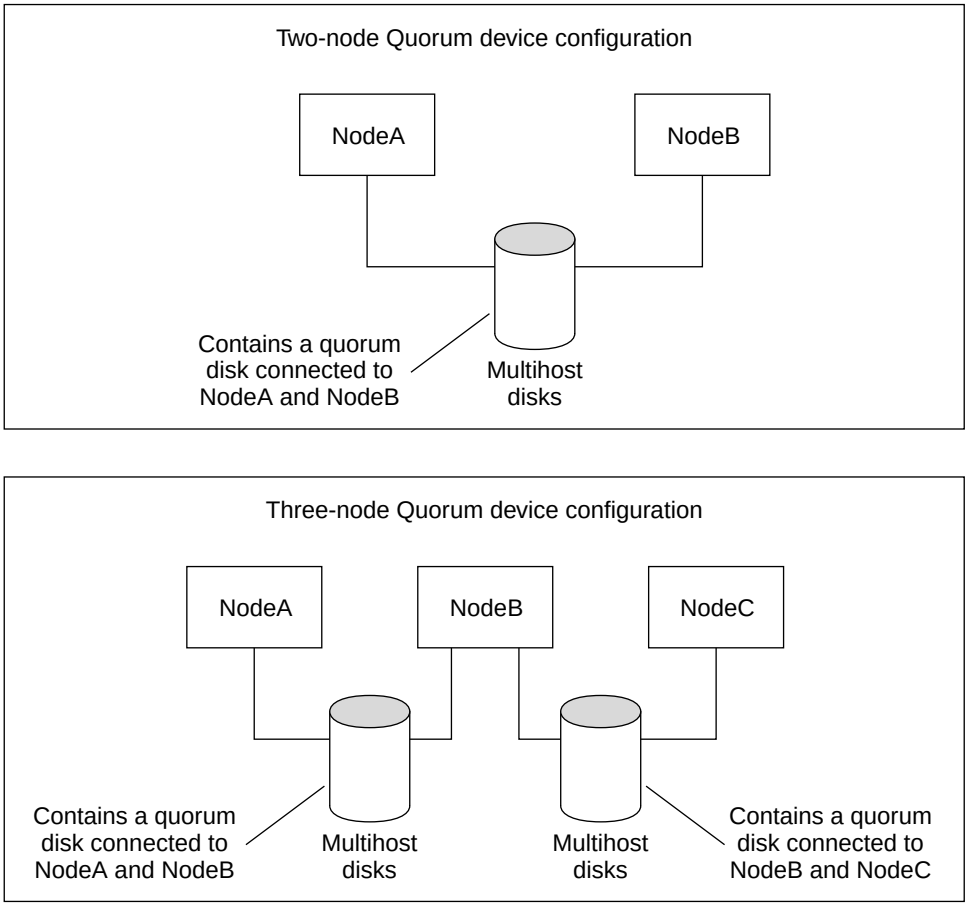

FIGURE 3-3 Quorum Device Configuration Examples

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 12/01 Software Installation Guide* for more information.
- You can use a disk that contains user data as a quorum device.

Tip – To protect against individual quorum device failures, 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.

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.

The SunPlex system 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.

Failfast Mechanism for Failure Fencing

The mechanism by which the cluster framework ensures that a failed node cannot reboot and begin writing to shared storage is called *failfast*.

Nodes that are cluster members continuously enable a specific ioctl, MHIOCENFAILFAST, for the disks to which they have access, including quorum disks. This ioctl is a directive to the disk driver, and gives a node the capability to panic itself if it cannot access the disk due to the disk being reserved by some other node.

The MHIOCENFAILFAST ioctl causes the driver to check the error return from every read and write that a node issues to the disk for the `Reservation_Conflict` error code. The ioctl periodically, in the background, issues a test operation to the disk to check for `Reservation_Conflict`. Both the foreground and background control flow paths panic if `Reservation_Conflict` is returned.

For SCSI-2 disks, reservations are not persistent—they do not survive node reboots. For SCSI-3 disks with Persistent Group Reservation (PGR), reservation information is stored on the disk and persists across node reboots. The failfast mechanism works the same regardless of whether you have SCSI-2 disks or SCSI-3 disks.

If a node loses connectivity to other nodes in the cluster, and it is not part of a partition that can achieve quorum, it is forcibly removed from the cluster by another node. Another node that is part of the partition that can achieve quorum places reservations on the shared disks and when the node that does not have quorum attempts to access the shared disks, it receives a reservation conflict and panics as a result of the failfast mechanism.

After the panic, the node might reboot and attempt to rejoin the cluster or stay at the OpenBoot PROM (OBP) prompt. The action taken is determined by the setting of the `auto-boot?` parameter in the OBP.

Volume Managers

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

The SunPlex system 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 volume management objects come under the control of the cluster, they become disk device groups. For information about volume managers, refer to your volume manager software documentation.

Note – 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 12/01 Software Installation Guide* and the *Sun Cluster 3.0 12/01 Data Services Installation and Configuration Guide* for discussions of these issues.

Data Services

The term *data service* describes a third-party application such as Oracle or iPlanet Web Server that has been configured to run on a cluster rather than on a single server. A data service consists of an application, specialized Sun Cluster configuration files, and Sun Cluster management methods that controls the following actions of the application.

- start
- stop
- monitor and take corrective measures

FIGURE 3-4 compares an application that runs on a single application server (the single-server model) to the same application running on a cluster (the clustered-server model). Note that from the user's perspective, there is no difference between the two configurations except that the clustered application might run faster and will be more highly available.

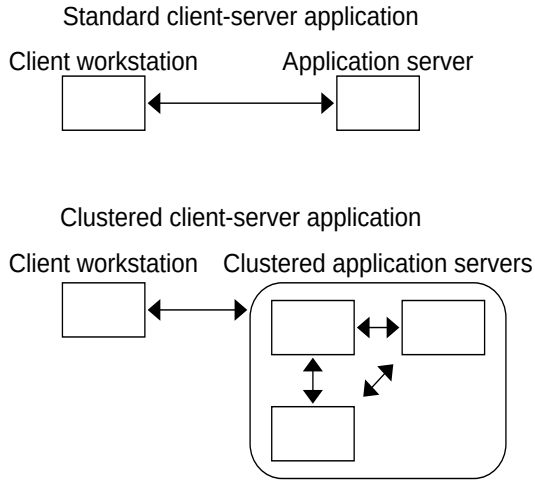

FIGURE 3-4 Standard Versus Clustered Client/Server Configuration

In the single-server model, you configure the application to access the server through a particular public network interface (a hostname). The hostname is associated with that physical server.

In the clustered-server model, the public network interface is a *logical hostname* or a *shared address*. The term *network resources* is used to refer to both logical hostnames and shared addresses.

Some data services require you to specify either logical hostnames or shared addresses as the network interfaces—they are not interchangeable. Other data services allow you to specify either logical hostnames or shared addresses. Refer to the installation and configuration for each data service for details on the type of interface you must specify.

A network resource is not associated with a specific physical server—it can migrate between physical servers.

A network resource is initially associated with one node, the *primary*. If the primary fails, the network resource, and the application resource, fails over to a different cluster node (a secondary). When the network resource fails over, after a short delay, the application resource continues to run on the secondary.

FIGURE 3-5 compares the single-server model with the clustered-server model. Note that in the clustered-server model, a network resource (logical hostname, in this example) can move between two or more of the cluster nodes. The application is configured to use this logical hostname in place of a hostname associated with a particular server.

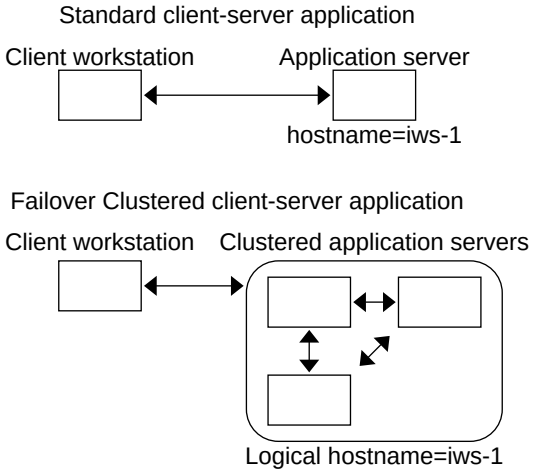

FIGURE 3-5 Fixed Hostname Versus Logical Hostname

A shared address is also initially associated with one node. This node is called the Global Interface Node (GIN). A shared address is used as the single network interface to the cluster. It is known as the *global interface*.

The difference between the logical hostname model and the scalable service model is that in the latter, each node also has the shared address actively configured up on its loopback interface. This configuration makes it possible to have multiple instances of a data service active on several nodes simultaneously. The term "scalable service" means that you can add more CPU power to the application by adding additional cluster nodes and the performance will scale.

If the GIN fails, the shared address can be brought up on another node that is also running an instance of the application (thereby making this other node the new GIN). Or, the shared address can fail over to another cluster node that was not previously running the application.

FIGURE 3-6 compares the single-server configuration with the clustered-scalable service configuration. Note that in the scalable service configuration, the shared address is present on all nodes. Similar to how a logical hostname is used for a failover data service, the application is configured to use this shared address in place of a hostname associated with a particular server.

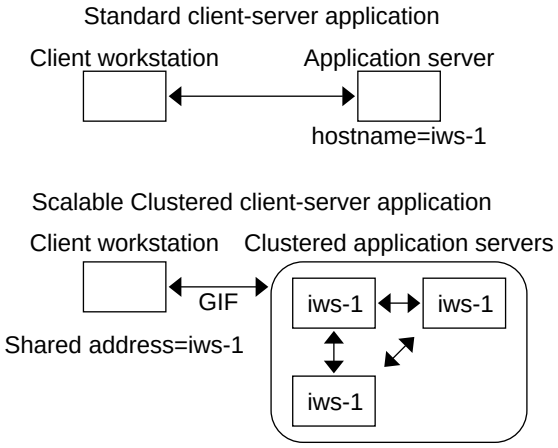

FIGURE 3-6 Fixed Hostname Versus Shared Address

Data Service Methods

The Sun Cluster software supplies a set of service management methods. 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 failover or scalable data services.

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

In addition to Sun Cluster software-supplied methods, the SunPlex system also supplies an API and several data service development tools. These tools enable application programmers to develop the data service methods needed to make other applications run as highly available data services with the Sun Cluster software.

Resource Group Manager (RGM)

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 Resource Management API (RMAPI). The cluster administrator creates and manages resources in containers called *resource groups*. The RGM stops and starts resource groups on selected nodes in response to cluster membership changes.

The RGM acts on *resources* and *resource groups.* RGM actions cause resources and resource groups to move between online and offline states. A complete description of the states and settings that can be applied to resources and resource groups is in the section "Resource and Resource Group States and Settings" on page 55.

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 use 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 use two resource groups: 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) 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 global interfaces 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 global interface node. Therefore, failure of the global interface node does not affect the connection.

FIGURE 3-7 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).

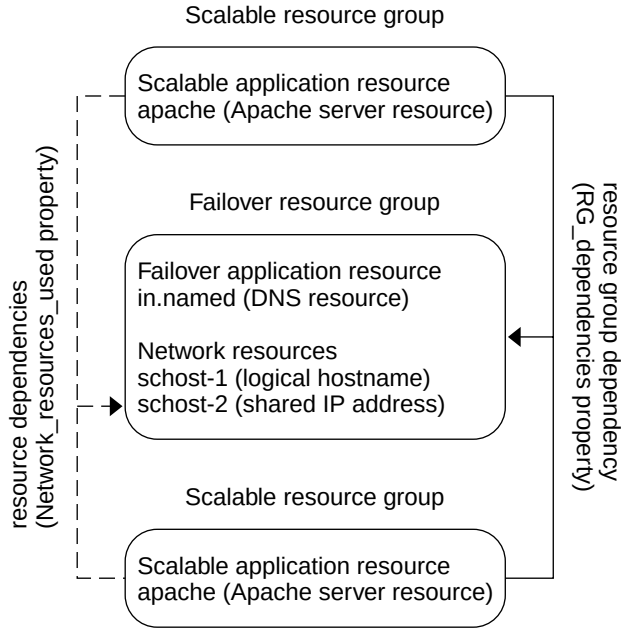

FIGURE 3-7 Failover and Scalable Resource Group Example

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
- Support for networking and data services (using the Resource Group Manager)

The following figure depicts the scalable service architecture.

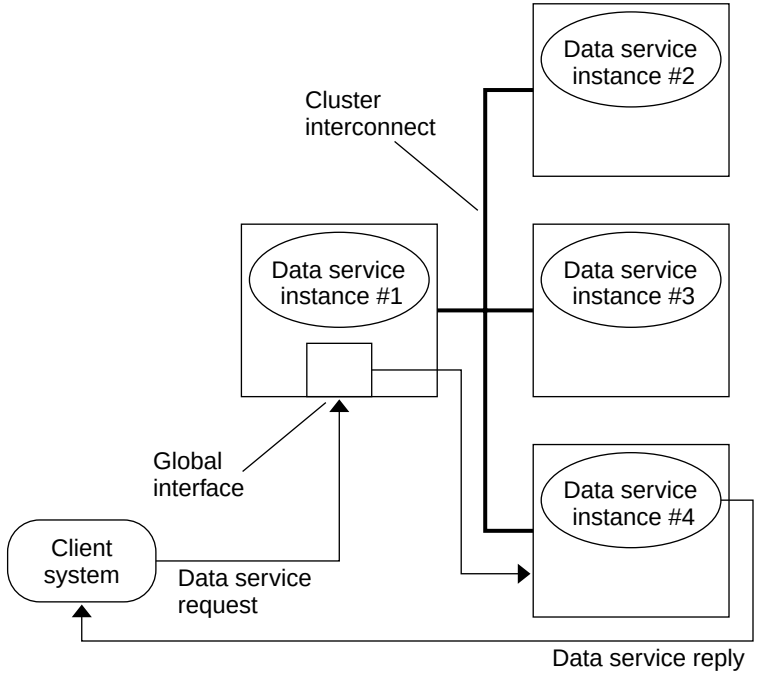

FIGURE 3-8 Scalable Service Architecture

The nodes that are not hosting the global interface (proxy nodes) have the shared address hosted on their loopback interfaces. Packets coming into the global interface 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 or through the SunPlex Manager GUI.

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. When this occurs, the original secondary becomes the new primary. The failback settings specify the actions that will take place when the original primary comes back online. The options are to have the original primary become the primary again (failback) or to allow the current primary to remain. You specify the option you want using the `Failback` resource group property setting.

In certain instances, 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 SunPlex 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 configuration files and management methods templates that enable you to make various applications operate as failover or scalable services within a cluster. If the application that you want to run as a failover or scalable service is not one that is currently offered by Sun, you can use an API or the DSET API to configure it to run as a failover or scalable service.

There is a set of criteria for determining whether an application can become a failover service. The specific criteria is described in the SunPlex 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 46 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.

The SunPlex system 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 this type of store is a backend database such as highly available Oracle or Oracle Parallel Server. Note that this type of 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

The SunPlex system provides the following to make applications highly available:

- Data services supplied as part of the SunPlex system
- A data service API
- A data service development library API
- A "generic" data service

The *Sun Cluster 3.0 12/01 Data Services Installation and Configuration Guide* describes how to install and configure the data services supplied with the SunPlex system. The *Sun Cluster 3.0 12/01 Data Services Developer's Guide* describes how to instrument other applications to be highly available under the Sun Cluster framework.

The Sun Cluster APIs 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, the SunPlex system 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 failover or scalable service.

Using the Cluster Interconnect for Data Service Traffic

A cluster must have multiple network connections between nodes, forming the cluster interconnect. The clustering software uses multiple interconnects both for high availability and to improve performance. For internal traffic (for example, file system data or scalable services data), messages are striped across all available interconnects in a round-robin fashion.

The cluster interconnect is also available to applications, for highly available communication between nodes. For example, a distributed application might have components running on different nodes that need to communicate. By using the cluster interconnect rather than the public transport, these connections can withstand the failure of an individual link.

To use the cluster interconnect for communication between nodes, an application must use the private hostnames configured when the cluster was installed. For example, if the private hostname for node 1 is `clusternode1-priv`, use that name to communicate over the cluster interconnect to node 1. TCP sockets opened using this name are routed over the cluster interconnect and can be transparently re-routed in the event of network failure.

Note that because the private hostnames can be configured during installation, the cluster interconnect can use any name chosen at that time. The actual name can be obtained from `scha_cluster_get`(3HA) with the `scha_privatelink_hostname_node` argument.

For application-level use of the cluster interconnect, a single interconnect is used between each pair of nodes, but separate interconnects are used for different node pairs, if possible. For example, consider an application running on three nodes and communicating over the cluster interconnect. Communication between nodes 1 and 2 might take place on interface `hme0`, while communication between nodes 1 and 3 might take place on interface `qfe1`. That is, application communication between any two nodes is limited to a single interconnect, while internal clustering communication is striped over all interconnects.

Note that the application shares the interconnect with internal clustering traffic, so the bandwidth available to the application depends on the bandwidth used for other clustering traffic. In the event of a failure, internal traffic can round-robin over the remaining interconnects, while application connections on a failed interconnect can switch to a working interconnect.

Two types of addresses support the cluster interconnect, and `gethostbyname`(3N) on a private hostname normally returns two IP addresses. The first address is called the *logical pairwise address*, and the second address is called the *logical pernode address*.

A separate logical pairwise address is assigned to each pair of nodes. This small logical network supports failover of connections. Each node is also assigned a fixed pernode address. That is, the logical pairwise addresses for `clusternode1-priv` are different on each node, while the logical pernode address for `clusternode1-priv` is the same on each node. A node does not have a pairwise address to itself, however, so `gethostbyname`(`clusternode1-priv`) on node 1 returns only the logical pernode address.

Note that applications accepting connections over the cluster interconnect and then verifying the IP address for security reasons must check against all IP addresses returned from `gethostbyname`, not just the first IP address.

If you need consistent IP addresses in your application at all points, configure the application to bind to the pernode address on both the client and the server side so that all connections can appear to come and go from the pernode address.

Resources, Resource Groups, 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.

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`.

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

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

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 12/01 Data Services Installation and Configuration Guide* for more information on this process.

Resource and Resource Group States and Settings

An administrator applies static settings to resources and resource groups. These settings can only be changed through administrative actions. The RGM moves resource groups between dynamic "states." These settings and states are described in the following list.

- Managed or unmanaged – These are cluster-wide settings that apply only to resource groups. Resource groups are managed by the RGM. The `scrgadm(1M)` command can be used to cause the RGM to manage or to unmanage a resource group. These settings do not change with a cluster reconfiguration.

  When a resource group is first created, it is unmanaged. It must be managed before any resources placed in the group can become active.

  In some data services, for example a scalable web server, work must be done prior to starting up network resources and after they are stopped. This work is done by initialization (INIT) and finish (FINI) data service methods. The INIT methods only run if the resource group in which the resources reside is in the managed state.

  When a resource group is moved from unmanaged to managed, any registered INIT methods for the group are run on the resources in the group.

  When a resource group is moved from managed to unmanaged, any registered FINI methods are called to perform cleanup.

  The most common use of INIT and FINI methods are for network resources for scalable services, but they can be used for any initialization or cleanup work that is not done by the application.

- Enabled or disabled – These are cluster-wide settings that apply to resources. The `scrgadm(1M)` command can be used to enable or disable a resource. These settings do not change with a cluster reconfiguration.

  The normal setting for a resource is that it is enabled and actively running in the system.

  If for some reason, you want to make the resource unavailable on all cluster nodes, you disable the resource. A disabled resource is not available for general use.

- Online or offline – These are dynamic states that apply to both resource and resource groups.

  These states change as the cluster transitions through cluster reconfiguration steps during switchover or failover. They can also be changed through administrative actions. The `scswitch(1M)` can be used to change the online or offline state of a resource or resource group.

  A failover resource or resource group can only be online on one node at any time. A scalable resource or resource group can be online on some nodes and offline on others.

  During a switchover or failover, resource groups and the resources within them are taken offline on one node and then brought online on another node.

  If a resource group is offline then all of its resources are offline. If a resource group is online, then all of its enabled resources are online.

  Resource groups can contain several resources, with dependencies between resources. These dependencies require that the resources be brought online and offline in a particular order. The methods used to bring resources online and offline might take different amounts of time for each resource. Because of resource dependencies and start and stop time differences, resources within a single resource group can have different online and offline states during a cluster reconfiguration.

Resource and Resource Group Properties

You can configure property values for resources and resource groups for your SunPlex data services. Standard properties are common to all data services. 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 which resource properties can be set and how to set them.

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 12/01 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 12/01 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 12/01 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 software 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 several times. This event 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 12/01 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.

Dynamic Reconfiguration Support

Sun Cluster 3.0 support for the dynamic reconfiguration (DR) software feature is being developed in incremental phases. This section describes concepts and considerations for Sun Cluster 3.0 12/01 support of the DR feature.

Note that all of the requirements, procedures, and restrictions that are documented for the Solaris 8 DR feature also apply to Sun Cluster DR support (except for the operating environment quiescence operation). Therefore, review the documentation for the Solaris 8 DR feature *before* using the DR feature with Sun Cluster software. You should review in particular the issues that affect non-network IO devices during a DR detach operation. The *Sun Enterprise 10000 Dynamic Reconfiguration User Guide* and the *Sun Enterprise 10000 Dynamic Reconfiguration Reference Manual* (from the *Solaris 8 on Sun Hardware* collection) are both available for download from http://docs.sun.com.

Dynamic Reconfiguration General Description

The DR feature allows operations, such as the removal of system hardware, in running systems. The DR processes are designed to ensure continuous system operation with no need to halt the system or interrupt cluster availability.

DR operates at the board level. Therefore, a DR operation affects all of the components on a board. Each board can contain multiple components, including CPUs, memory, and peripheral interfaces for disk drives, tape drives, and network connections.

Removing a board terminates the system's ability to use any of the components on the board. Before removing a board, the DR subsystem determines whether the components on the board are being used. Removing a device that is being used would result in system errors. If the DR subsystem finds that a device is in use, this subsystem rejects the DR remove-board operation. Therefore, it is always safe to issue a DR remove-board operation.

The DR add-board operation is always safe also. CPUs and memory on a newly added board are automatically brought into service by the system. However, the system administrator must manually configure the cluster in order to actively use other components that are on the newly added board.

Note – The DR subsystem has several levels. If a lower level reports an error, the upper level also reports an error. However, when the lower level reports the specific error, the upper level will report "Unknown error." System administrators should ignore the "Unknown error" reported by the upper level.

The following sections describe DR considerations for the different device types.

DR Clustering Considerations for CPU Devices

When a DR remove-board operation affects CPUs on the board, the DR subsystem allows the operation and automatically makes the node stop using these CPUs.

When a DR add-board operation affects CPUs on the added board, the DR subsystem automatically makes the node start using these CPUs.

DR Clustering Considerations for Memory

For the purposes of DR, there are two types of memory to consider. These two types differ only in usage. The actual hardware is the same for both types.

The memory used by the operating system is called the kernel memory cage. Sun Cluster software does not support remove-board operations on a board that contains the kernel memory cage and will reject any such operation. When a DR remove-board operation affects memory other than the kernel memory cage, the DR subsystem allows the operation and automatically makes the node stop using that memory.

When a DR add-board operation affects memory, the DR subsystem automatically makes the node start using the new memory.

DR Clustering Considerations for Disk and Tape Drives

DR remove operations on active drives in the primary node are *not* allowed. DR remove operations can be performed on non-active drives in the primary node and on drives in the secondary node. Cluster data access continues both before and after the DR operation.

Note – DR operations that affect the availability of quorum devices are *not* allowed. For considerations about quorum devices and the procedure for performing DR operations on them, see "DR Clustering Considerations for Quorum Devices" on page 62.

The following steps describe a brief summary of the procedure for performing a DR remove operation on a disk or tape drive. See the *Sun Cluster 3.0 U1 System Administration Guide* for detailed instructions on how to perform these actions.

1. **Determine whether the disk or tape drive is part of an active device group.**
   - If the drive is not part of an active device group, you can perform the DR remove operation on it.
   - If the DR remove-board operation would affect an active disk or tape drive, the system rejects the operation and identifies the drives that would be affected by the operation. If the drive is part of an active device group, go to Step 2.
2. **Determine whether the drive is a component of the primary node or the secondary node.**
   - If the drive is a component of the secondary node, you can perform the DR remove operation on it.
   - If the drive is a component of the primary node, you must switch the primary and secondary nodes before performing the DR remove operation on the device.

Caution – If the current primary node fails while you are performing the DR operation on a secondary node, cluster availability is impacted. The primary node has no place to fail over until a new secondary node is provided.

DR Clustering Considerations for Quorum Devices

DR remove operations *cannot* be performed on a device that is currently configured as a quorum device. If the DR remove-board operation would affect a quorum device, the system rejects the operation and identifies the quorum device that would be affected by the operation. You must disable the device as a quorum device before you can perform a DR remove operation on it.

The following steps describe a brief summary of the procedure for performing a DR remove operation on a quorum device. See the *Sun Cluster 3.0 U1 System Administration Guide* for detailed instructions on how to perform these actions.

1. **Enable a device other than the one you are performing the DR operation on to be the quorum device.**
2. **Disable the device you are performing the DR operation on as a quorum device.**
3. **Perform the DR remove operation on the device.**

DR Clustering Considerations for Private Interconnect Interfaces

DR operations cannot be performed on active private interconnect interfaces. If the DR remove-board operation would affect an active private interconnect interface, the system rejects the operation and identifies the interface that would be affected by the operation. An active interface must first be disabled before you remove it (also see the caution below). When an interface is replaced to the private interconnect, its state remains the same, avoiding any need for additional Sun Cluster reconfiguration steps.

The following steps describe a brief summary of the procedure for performing a DR remove operation on a private interconnect interface. See the *Sun Cluster 3.0 U1 System Administration Guide* for detailed instructions on how to perform these actions.

Caution – Sun Cluster requires that each cluster node has at least one functioning path to every other cluster node. Do not disable a private interconnect interface that supports the last path to any cluster node.

1. **Disable the transport cable that contains the interconnect interface upon which you are performing the DR operation.**
2. **Perform the DR remove operation on the physical private interconnect interface.**

DR Clustering Considerations for Public Network Interfaces

DR remove operations can be performed on public network interfaces that are not active. If the DR remove-board operation would affect an active public network interface, the system rejects the operation and identifies the interface that would be affected by the operation. Any active public network interface must first be removed from the status of being an active adapter instance in a network adapter fail over (NAFO) group.

Caution – If the active network adapter fails while you are performing the DR remove operation on the disabled network adapter, availability is impacted. The active adapter has no place to fail over for the duration of the DR operation.

The following steps describe a brief summary of the procedure for performing a DR remove operation on a public network interface. See the *Sun Cluster 3.0 U1 System Administration Guide* for detailed instructions on how to perform these actions.

1. **Switch the active adapter to be a backup adapter so that it can be removed from the NAFO group.**
2. **Remove the adapter from the NAFO group.**
3. **Perform the DR operation on the public network interface.**

CHAPTER 4

Frequently Asked Questions

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

High Availability FAQ

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

  The SunPlex system 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 data service resources from a failing node to another operational node in the cluster.

- **What is the difference between a failover and scalable data service?**

  There are two types of highly available data services, failover and scalable.

  A failover data service runs an application 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.

  For each application, one node hosts the physical interface to the cluster. This node is called a Global Interface Node (GIN). There can be multiple GINs in the cluster. Each GIN hosts one or more logical interfaces that can be used by scalable

services. These logical interfaces are called *global interfaces*. One GIN hosts a global interface for all requests for a particular application and dispatches them to multiple nodes on which the application server is running. If the GIN fails, 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, do not do a loopback mount.

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

  Yes. However, without RGM control, the applications need to be restarted manually after the failure of the node on which they are running.

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

  No. However, placing cluster file systems under the same mount point, such as `/global`, 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.

- **The file system /global/.devices/<node>@<node ID> appears on my cluster nodes. Can I use this file system to store data that I want to be highly available and global?**

  These file systems store the global device namespace. They are not intended for general use. While they are global, they are never accessed in a global manner--each node only accesses its own global device namespace. If a node is down, other nodes cannot access this namespace for the node that is down. These file systems are not highly available. They should not be used to store data that needs to be globally accessible or highly available.

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.

- **Can I use one volume manager for the local disks (boot disk) and a different volume manager for the multihost disks?**

  This configuration is supported with the Solstice DiskSuite software managing the local disks and VERITAS Volume Manager managing the multihost disks. No other combination is supported.

Data Services FAQ

- **What SunPlex data services are available?**

  The list of supported data services is included in the *Sun Cluster 3.0 12/01 Release Notes*.

- **What application versions are supported by SunPlex data services?**

  The list of supported application versions is included in the *Sun Cluster 3.0 12/01 Release Notes*.

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

  Yes. See the *Sun Cluster 3.0 12/01 Data Services Developer's 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)?**

  Except in the case of Sun Cluster HA for NFS, 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 the SunPlex system support?**

  Currently, the SunPlex system 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.

- **Does the SunPlex system support setting local-mac-address?=true in the OpenBoot™ PROM (OBP) for a host adapter?**

  No, this variable is not supported.

- **How much delay can I expect when NAFO performs a switchover between the active and backup adapter?**

  The delay could be several minutes. This is because when a NAFO switchover is done, it involves sending out a gratuitous ARP. However, there is no guarantee that the router between the client and the cluster will use the gratuitous ARP. So, until the ARP cache entry for this IP address on the router times out, it is possible that it could use the stale MAC address.

  A second reason for delay might be that both NAFO adapters are connected to an Ethernet switch. When a NAFO switchover is done, one of the NAFO adapters is unplumbed and the second adapter is plumbed. The Ethernet switch would now have to disable a port and enable a different port, which might take some time. Also, with Ethernet, speed negotiation takes place between the switch and the newly enabled adapter, which takes time.

  Finally, after a switchover is done, NAFO does some minimal sanity checking on the newly enabled adapter to verify that everything works fine.

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 37 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 12/01 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.

Cluster Interconnect FAQ

- **What cluster interconnects does the SunPlex system support?**

  Currently, the SunPlex system supports Ethernet (100BASE-T Fast Ethernet and 1000BASE-SX Gb) cluster interconnects.

- **What is the difference between a "cable" and a transport "path?"**

  Cluster transport cables are configured using transport adapters and switches. Cables join adapters and switches on a component-to-component basis. The cluster topology manager uses available cables to build end-to-end transport paths between nodes. A cable does not map directly to a transport path.

  Cables are statically "enabled" and "disabled" by an administrator. Cables have a "state," (enabled or disabled) but not a "status." If a cable is disabled, it is as if it were unconfigured. Cables that are disabled cannot be used as transport paths. They are not probed and therefore, it is not possible to know their status. The state of a cable can be viewed using `scconf -p`.

  Transport paths are dynamically established by the cluster topology manager. The "status" of a transport path is determined by the topology manager. A path can have a status of "online" or "offline." The status of a transport path can be viewed using `scstat(1M)`.

  Consider the following example of a two-node cluster with four cables.

```
node1:adapter0     to switch1, port0
node1:adapter1     to switch2, port0
node2:adapter0     to switch1, port1
node2:adapter1     to switch2, port1
```

  There are two possible transport paths that can be formed from these four cables.

```
node1:adapter0    to node2:adapter0
node2:adapter1    to node2:adapter1
```

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 12/01 Data Services Installation and Configuration Guide* for more information on client-side configuration requirements.

Administrative Console FAQ

- **Does the SunPlex system require an administrative console?**

  Yes.

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

  The SunPlex system 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 the SunPlex system require a terminal concentrator?**

  All software releases starting with Sun Cluster 3.0 do not require a terminal concentrator to run. Unlike the Sun Cluster 2.2 product, which required a terminal concentrator for failure fencing, later products do not depend on the terminal concentrator.

- **I see that most SunPlex 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 12/01 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 SunPlex 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 software used to provide a consistent device namespace across the cluster. See also "DID name."

DID name Used to identify global devices in a SunPlex system. 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 the 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 SunPlex 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 46 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 46 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 the

SunPlex system. Refer to “Disk Device Groups” on page 31 and “Resources, Resource Groups, and Resource Types” on page 54 for a description of how this concept is now implemented in the SunPlex system.

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 software 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 SunPlex system. 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 a SunPlex system 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 the SunPlex system. 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.

Sun Cluster (software) The software portion of the SunPlex system. (See SunPlex.)

SunPlex The integrated hardware and Sun Cluster software system that is used to create highly available and scalable services.

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 the SunPlex system. 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.