Sun Cluster Concepts Guide for Solaris OS

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819–2969–10
December 2006, Revision A

061026@15490

Contents

Preface 7

1 Introduction and Overview 11
Introduction to the Sun Cluster Environment 11
Three Views of the Sun Cluster Software 12
Hardware Installation and Service View 12
System Administrator View 13
Application Developer View 14
Sun Cluster Software Tasks 15

2 Key Concepts for Hardware Service Providers 17
Sun Cluster System Hardware and Software Components 17
Cluster Nodes 18
Software Components for Cluster Hardware Members 19
Multihost Devices 20
Multi-Initiator SCSI 20
Local Disks 21
Removable Media 21
Cluster Interconnect 22
Public Network Interfaces 22
Client Systems 23
Console Access Devices 23
Administrative Console 24
SPARC: Sun Cluster Topologies for SPARC 24
SPARC: Clustered Pair Topology for SPARC 25
SPARC: Pair+N Topology for SPARC 26
SPARC: N+1 (Star) Topology for SPARC 27
SPARC: N*N (Scalable) Topology for SPARC 28

x86: Sun Cluster Topologies for x86 29
x86: Clustered Pair Topology for x86 29

3 Key Concepts for System Administrators and Application Developers 31
Administrative Interfaces 32
Cluster Time 32
High-Availability Framework 33
Zone Membership 34
Cluster Membership Monitor 34
Failfast Mechanism 34
Cluster Configuration Repository (CCR) 35
Global Devices 35
Device IDs and DID Pseudo Driver 36
Device Groups 36
Device Group Failover 37
Multiported Device Groups 38
Global Namespace 40
Local and Global Namespaces Example 40
Cluster File Systems 41
Using Cluster File Systems 42
`HAStoragePlus` Resource Type 42
`syncdir` Mount Option 43
Disk Path Monitoring 43
DPM Overview 43
Monitoring Disk Paths 44
Quorum and Quorum Devices 46
About Quorum Vote Counts 47
About Failure Fencing 48
Failfast Mechanism for Failure Fencing 49
About Quorum Configurations 49
Adhering to Quorum Device Requirements 50
Adhering to Quorum Device Best Practices 50
Recommended Quorum Configurations 51
Atypical Quorum Configurations 54
Bad Quorum Configurations 54
Data Services 56

Data Service Methods 58
Failover Data Services 58
Scalable Data Services 59
Load-Balancing Policies 60
Failback Settings 62
Data Services Fault Monitors 62
Developing New Data Services 63
Characteristics of Scalable Services 63
Data Service API and Data Service Development Library API 64
Using the Cluster Interconnect for Data Service Traffic 64
Resources, Resource Groups, and Resource Types 65
Resource Group Manager (RGM) 66
Resource and Resource Group States and Settings 66
Resource and Resource Group Properties 68
Support for Solaris Zones on Sun Cluster Nodes 68
Support for Solaris Zones on Sun Cluster Nodes Directly Through the RGM 69
Support for Solaris Zones on Sun Cluster Nodes Through Sun Cluster HA for Solaris Containers 71
Service Management Facility 72
System Resource Usage 72
System Resource Monitoring 73
Control of CPU 73
Viewing System Resource Usage 74
Data Service Project Configuration 74
Determining Requirements for Project Configuration 76
Setting Per-Process Virtual Memory Limits 77
Failover Scenarios 78
Public Network Adapters and Internet Protocol (IP) Network Multipathing 83
SPARC: Dynamic Reconfiguration Support 85
SPARC: Dynamic Reconfiguration General Description 85
SPARC: DR Clustering Considerations for CPU Devices 85
SPARC: DR Clustering Considerations for Memory 86
SPARC: DR Clustering Considerations for Disk and Tape Drives 86
SPARC: DR Clustering Considerations for Quorum Devices 86
SPARC: DR Clustering Considerations for Cluster Interconnect Interfaces 87
SPARC: DR Clustering Considerations for Public Network Interfaces 87

4 Frequently Asked Questions 89
High Availability FAQs 89
File Systems FAQs 90
Volume Management FAQs 91
Data Services FAQs 91
Public Network FAQs 92
Cluster Member FAQs 93
Cluster Storage FAQs 94
Cluster Interconnect FAQs 94
Client Systems FAQs 95
Administrative Console FAQs 95
Terminal Concentrator and System Service Processor FAQs 96

Index 99

Preface

The *Sun Cluster Concepts Guide for Solaris OS* contains conceptual and reference information about the Sun™ Cluster product on both SPARC® and x86 based systems.

Note – This Sun Cluster release supports systems that use the SPARC and x86 families of processor architectures: UltraSPARC, SPARC64, and AMD64. In this document, the label x86 refers to systems that use the AMD64 family of processor architectures.

Who Should Use This Book

This document is intended for the following audiences:

- 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 that are not currently included with the Sun Cluster product

To understand the concepts that are described in this book, you need to be familiar with the Solaris Operating System and also have expertise with the volume manager software that you can use with the Sun Cluster product.

Before reading this document, you need to have already determined your system requirements and purchased the equipment and software that you need. The *Sun Cluster Data Services Planning and Administration Guide for Solaris OS* contains information about how to plan, install, set up, and use the Sun Cluster software.

How This Book Is Organized

The *Sun Cluster Concepts Guide for Solaris OS* contains the following chapters:

Chapter 1 provides an overview of the overall concepts that you need to know about Sun Cluster.

Chapter 2 describes the concepts with which hardware service providers need to be familiar. These concepts can help service providers understand the relationships between hardware components. These concepts can also help service providers and cluster administrators better understand how to install, configure, and administer cluster software and hardware.

Chapter 3 describes the concepts with which system administrators and developers who intend to use the Sun Cluster application programming interface (API) need to know. Developers can use this API to turn a standard user application, such as a web browser or database into a highly available data service that can run in the Sun Cluster environment.

Chapter 4 provides answers to frequently asked questions about the Sun Cluster product.

Related Documentation

Information about related Sun Cluster topics is available in the documentation that is listed in the following table. All Sun Cluster documentation is available at `http://docs.sun.com`.

Topic	Documentation
Overview	*Sun Cluster Overview for Solaris OS*
Concepts	*Sun Cluster Concepts Guide for Solaris OS*
Hardware installation and administration	*Sun Cluster 3.1 - 3.2 Hardware Administration Manual for Solaris OS*
	Individual hardware administration guides
Software installation	*Sun Cluster Software Installation Guide for Solaris OS*
Data service installation and administration	*Sun Cluster Data Services Planning and Administration Guide for Solaris OS*
	Individual data service guides
Data service development	*Sun Cluster Data Services Developer's Guide for Solaris OS*
System administration	*Sun Cluster System Administration Guide for Solaris OS*
Error messages	*Sun Cluster Error Messages Guide for Solaris OS*
Command and function references	*Sun Cluster Reference Manual for Solaris OS*

For a complete list of Sun Cluster documentation, see the release notes for your release of Sun Cluster software at `http://docs.sun.com`.

Getting Help

If you have problems installing or using the Sun Cluster software, 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 system (for example, the Solaris 10 OS)
- The release number of Sun Cluster software (for example, 3.2)

Use the following commands to gather information about your systems 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
SPARC: `prtdiag -v`	Displays system diagnostic information
`/usr/cluster/bin/clnode show-rev`	Displays Sun Cluster release and package version information

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

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.sun.com/training/`)

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`

TABLE P–1 Typographic Conventions *(Continued)*

Typeface	Meaning	Example
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX® system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

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

◆ ◆ ◆ CHAPTER 1

Introduction and Overview

The Sun Cluster product is an integrated hardware and software solution that you use to create highly available and scalable services. Sun Cluster Concepts Guide for Solaris OS provides the conceptual information that you need to gain a more complete picture of the Sun Cluster product. Use this book with the entire Sun Cluster documentation set to provide a complete view of the Sun Cluster software.

This chapter provides an overview of the general concepts that underlie the Sun Cluster product.

This chapter does the following:

- Provides an introduction and high-level overview of the Sun Cluster software
- Describes the several views of the Sun Cluster audience
- Identifies key concepts that you need to understand before you use the Sun Cluster software
- Maps key concepts to the Sun Cluster documentation that includes procedures and related information
- Maps cluster-related tasks to the documentation that contains procedures that you use to complete those tasks

This chapter contains the following sections:

- "Introduction to the Sun Cluster Environment" on page 11
- "Three Views of the Sun Cluster Software" on page 12
- "Sun Cluster Software Tasks" on page 15

Introduction to the Sun Cluster Environment

The Sun Cluster environment extends the Solaris Operating System 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 Sun Cluster software 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

For more information about fault tolerance and high availability, see "Making Applications Highly Available With Sun Cluster" in *Sun Cluster Overview for Solaris OS*.

Refer to "High Availability FAQs" on page 89 for questions and answers on high availability.

Three Views of the Sun Cluster Software

This section describes three different views of the Sun Cluster software and the key concepts and documentation relevant to each view.

These views are typical for the following professionals:

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

Hardware Installation and Service View

To hardware service professionals, the Sun Cluster software 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 professionals 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

More Hardware Conceptual Information

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

- "Cluster Nodes" on page 18
- "Multihost Devices" on page 20
- "Local Disks" on page 21
- "Cluster Interconnect" on page 22
- "Public Network Interfaces" on page 22
- "Client Systems" on page 23
- "Administrative Console" on page 24
- "Console Access Devices" on page 23
- "SPARC: Clustered Pair Topology for SPARC" on page 25
- "SPARC: N+1 (Star) Topology for SPARC" on page 27

Sun Cluster Documentation for Hardware Professionals

The *Sun Cluster 3.1 - 3.2 Hardware Administration Manual for Solaris OS* includes procedures and information that are associated with hardware service concepts.

System Administrator View

To the system administrator, the Sun Cluster software is a set of servers (nodes) that are cabled together, sharing storage devices.

The system administrator sees software that performs specific tasks:

- Specialized cluster software that is integrated with Solaris software to monitor the connectivity between cluster nodes
- Specialized software that monitors the health of user application programs that are 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 that are 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 System
  - 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

More System Administrator Conceptual Information

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

- "Administrative Interfaces" on page 32
- "Cluster Time" on page 32
- "High-Availability Framework" on page 33
- "Global Devices" on page 35
- "Device Groups" on page 36
- "Global Namespace" on page 40
- "Cluster File Systems" on page 41
- "Disk Path Monitoring" on page 43
- "About Failure Fencing" on page 48
- "Data Services" on page 56

Sun Cluster Documentation for System Administrators

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

- *Sun Cluster Software Installation Guide for Solaris OS*
- *Sun Cluster System Administration Guide for Solaris OS*
- *Sun Cluster Error Messages Guide for Solaris OS*
- *Sun Cluster 3.2 Release Notes for Solaris OS*
- *Sun Cluster 3.0-3.1 Release Notes Supplement*

Application Developer View

The Sun Cluster software provides *data services* for such applications as Oracle, NFS, DNS, Sun Java System Web Server, Apache Web Server (on SPARC based systems), and Sun Java System Directory Server. Data services are created by configuring 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 Sun Cluster 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 Development

Application developers 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. Developers 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.

More Application Developer Conceptual Information

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

- "Data Services" on page 56
- "Resources, Resource Groups, and Resource Types" on page 65
- Chapter 4

Sun Cluster Documentation for Application Developers

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

- *Sun Cluster Data Services Developer's Guide for Solaris OS*
- *Sun Cluster Data Services Planning and Administration Guide for Solaris OS*

Sun Cluster Software Tasks

All Sun Cluster software 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

Task	Instructions
Install cluster hardware	*Sun Cluster 3.1 - 3.2 Hardware Administration Manual for Solaris OS*
Install Solaris software on the cluster	*Sun Cluster Software Installation Guide for Solaris OS*

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

Task	Instructions
SPARC: Install Sun™ Management Center software	*Sun Cluster Software Installation Guide for Solaris OS*
Install and configure Sun Cluster software	*Sun Cluster Software Installation Guide for Solaris OS*
Install and configure volume management software	*Sun Cluster Software Installation Guide for Solaris OS* Your volume management documentation
Install and configure Sun Cluster data services	*Sun Cluster Data Services Planning and Administration Guide for Solaris OS*
Service cluster hardware	*Sun Cluster 3.1 - 3.2 Hardware Administration Manual for Solaris OS*
Administer Sun Cluster software	*Sun Cluster System Administration Guide for Solaris OS*
Administer volume management software	*Sun Cluster System Administration Guide for Solaris OS* and your volume management documentation
Administer application software	Your application documentation
Problem identification and suggested user actions	*Sun Cluster Error Messages Guide for Solaris OS*
Create a new data service	*Sun Cluster Data Services Developer's Guide for Solaris OS*

◆ ◆ ◆ CHAPTER 2

Key Concepts for Hardware Service Providers

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

This chapter covers the following topics:

- "Sun Cluster System Hardware and Software Components" on page 17
- "SPARC: Sun Cluster Topologies for SPARC" on page 24
- "x86: Sun Cluster Topologies for x86" on page 29

Sun Cluster System Hardware and Software Components

This information is directed primarily to 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 the following:

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

The Sun Cluster software enables you to combine these components into a variety of configurations. The following sections describe these configurations.

- "SPARC: Sun Cluster Topologies for SPARC" on page 24
- "x86: Sun Cluster Topologies for x86" on page 29

For an illustration of a sample two-node cluster configuration, see "Sun Cluster Hardware Environment" in *Sun Cluster Overview for Solaris OS*.

Cluster Nodes

A cluster node is a machine that is running both the Solaris Operating System and Sun Cluster software. A cluster node is either a current member of the cluster (a *cluster member*), or a potential member.

- SPARC: Sun Cluster software supports one to sixteen nodes in a cluster. See "SPARC: Sun Cluster Topologies for SPARC" on page 24 for the supported node configurations.
- x86: Sun Cluster software supports up to four nodes in a cluster. See "x86: Sun Cluster Topologies for x86" on page 29 for the supported node configurations.

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

In addition, nodes in parallel database configurations share concurrent access to all the disks.

- See "Multihost Devices" on page 20 for information about concurrent access to disks.
- See "SPARC: Clustered Pair Topology for SPARC" on page 25 and "x86: Clustered Pair Topology for x86" on page 29 for more information about 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 support 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, a node must have the following software installed:

- Solaris Operating System
- Sun Cluster software
- Data service application
- Volume management (Solaris Volume Manager™ or VERITAS Volume Manager)

  An exception is a configuration that uses hardware redundant array of independent disks (RAID). This configuration might not require a software volume manager such as Solaris Volume Manager or VERITAS Volume Manager.
- See the *Sun Cluster Software Installation Guide for Solaris OS* for information about how to install the Solaris Operating System, Sun Cluster, and volume management software.
- See the *Sun Cluster Data Services Planning and Administration Guide for Solaris OS* for information about how to install and configure data services.
- See Chapter 3 for conceptual information about the preceding software components.

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

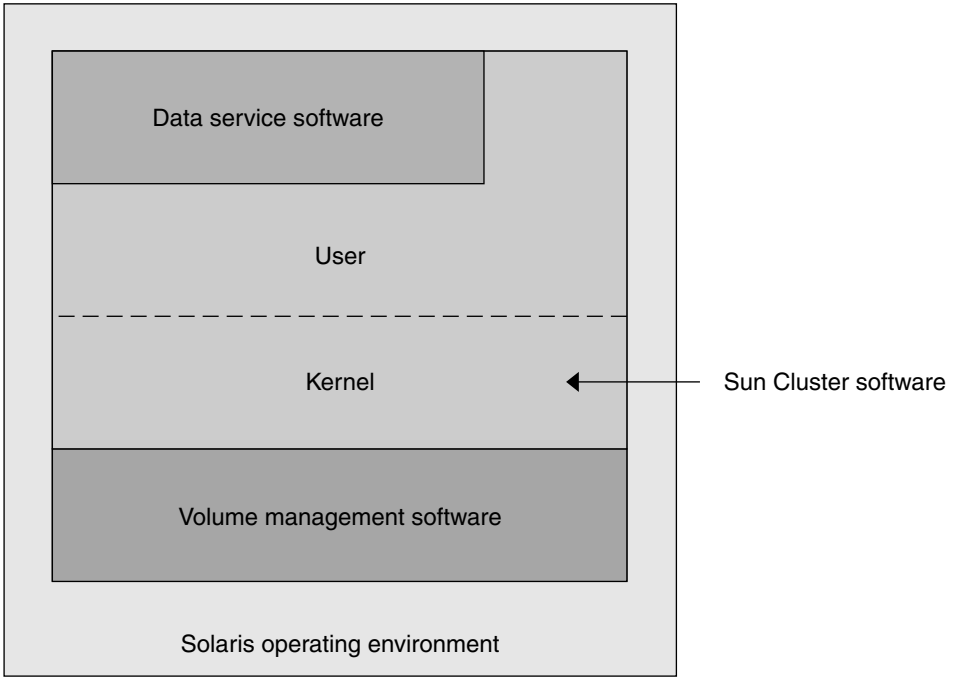


FIGURE 2–1 High-Level Relationship of Sun Cluster Software Components

See Chapter 4 for questions and answers about cluster members.

Multihost Devices

Disks that can be connected to more than one node at a time are multihost devices. In the Sun Cluster environment, multihost storage makes disks highly available. Sun Cluster software requires multihost storage for two-node clusters to establish quorum. Greater than two-node clusters do not require quorum devices. For more information about quorum, see "Quorum and Quorum Devices" on page 46.

Multihost devices have the following characteristics.

- Tolerance of single-node failures.
- Ability to store application data, application binaries, and configuration files.
- Protection against node failures. If clients request the data through one node and the node fails, the requests are switched over to use another node with a direct connection to the same disks.
- Global access through a primary node that "masters" the disks, or direct concurrent access through local paths. The only application that uses direct concurrent access currently is Oracle Real Application Clusters Guard.

A volume manager provides for mirrored or RAID-5 configurations for data redundancy of the multihost devices. Currently, Sun Cluster supports Solaris Volume Manager and VERITAS Volume Manager as volume managers, and the RDAC RAID-5 hardware controller on several hardware RAID platforms.

Combining multihost devices with disk mirroring and disk 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 devices.

In a standalone server, the server node controls the SCSI bus activities by way of the SCSI host adapter circuit that connects 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 devices. When the cluster storage consists of single-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 each device on a SCSI bus to have a unique SCSI address. (The host adapter is also a device on the SCSI bus.) The default hardware configuration in a multi-initiator environment results in a conflict because all SCSI host adapters default to 7.

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

In most configurations, the available SCSI address for a second host adapter is 6.

You can change the selected SCSI addresses for these host adapters by using one of the following tools to set the `scsi-initiator-id` property:

- `eeprom`(1M)
- The OpenBoot PROM on a SPARC based system
- The SCSI utility that you optionally run after the BIOS boots on an x86 based system

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 *Sun Cluster 3.1 - 3.2 With SCSI JBOD Storage Device Manual for Solaris OS*.

Local Disks

Local disks are the disks that are only connected to a single node. Local disks are, therefore, not protected against node failure (the are 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 placing 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 35 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 nonclustered 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 *Sun Cluster 3.1 - 3.2 Hardware Administration Manual for Solaris OS* for information about installing and configuring removable media.

See the section "Global Devices" on page 35 for more information about global devices.

Cluster Interconnect

The *cluster interconnect* is the physical configuration of devices that is 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 interface cards that are located 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 adapters also contain both network interfaces and storage interfaces.

  A network adapter with multiple interfaces could become a single point of failure if the entire adapter fails. For maximum availability, plan your cluster so that the only path between two nodes does not depend on a single network adapter.
- Junctions – The switches that are located outside of the cluster nodes. Junctions perform pass-through and switching functions to enable you to connect more than two nodes. 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 you install 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 that perform the following functions:

- Are configured so that multiple cards are active.
- Serve as failover backups for one another.

If one of the adapters fails, Internet Protocol (IP) Network Multipathing software is called to fail over the defective interface to another adapter in the group.

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 that are 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 one of the following devices:

- The terminal concentrator that you purchased with your cluster hardware
- The System Service Processor (SSP) on Sun Enterprise E10000 servers (for SPARC based clusters)
- The system controller on Sun Fire™ servers (also for SPARC based clusters)
- 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 E1000 servers. The SSP is a machine on an Ethernet network that is configured to support the Sun Enterprise E1000 server. The SSP is the administrative console for the Sun Enterprise E1000 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` 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 workstation, 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 (for use with SPARC based clusters only), 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 about to use the CCP, see the Chapter 1, "Introduction to Administering Sun Cluster," in *Sun Cluster System Administration Guide for Solaris OS*.

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 do the following:

- Log in from the administrative console to the SSP.
- Connect by using the `netcon` command.

Typically, you configure nodes without monitors. Then, you access the node's console through a `telnet` session from the administrative console. The administration console is connected to a terminal concentrator, and from the terminal concentrator to the node's serial port. In the case of a Sun Enterprise E1000 server, you connect from the System Service Processor. See "Console Access Devices" on page 23 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.

SPARC: Sun Cluster Topologies for SPARC

A topology is the connection scheme that connects the cluster nodes to the storage platforms that are used in a Sun Cluster environment. Sun Cluster software supports any topology that adheres to the following guidelines.

- A Sun Cluster environment that is composed of SPARC based systems supports a maximum of sixteen nodes in a cluster. All SPARC based topologies support up to eight nodes in a cluster. Selected SPARC based topologies support up to sixteen nodes in a cluster. Contact your Sun sales representative for more information.
- A shared storage device can connect to as many nodes as the storage device supports.
- Shared storage devices do not need to connect to all nodes of the cluster. However, these storage devices must connect to at least two nodes.

Sun Cluster software does not require you to configure a cluster by using specific topologies. The following topologies are described to provide the vocabulary to discuss a cluster's connection scheme. These topologies are typical connection schemes.

- Clustered pair
- Pair+N
- N+1 (star)
- N*N (scalable)

The following sections include sample diagrams of each topology.

SPARC: Clustered Pair Topology for SPARC

A clustered pair topology is two or more pairs of nodes that operate 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. More than two nodes can run a scalable service or parallel database, even though all the nodes are not directly connected to the disks that store the application data.

The following figure illustrates a clustered pair configuration.

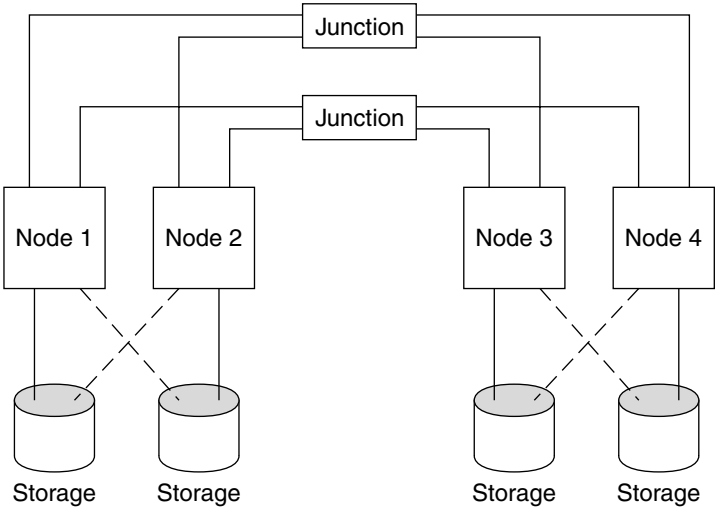


FIGURE 2–2 SPARC: Clustered Pair Topology

SPARC: Pair+N Topology for SPARC

The pair+N topology includes a pair of nodes that are directly connected to the following:

- Shared storage.
- 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+N 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.

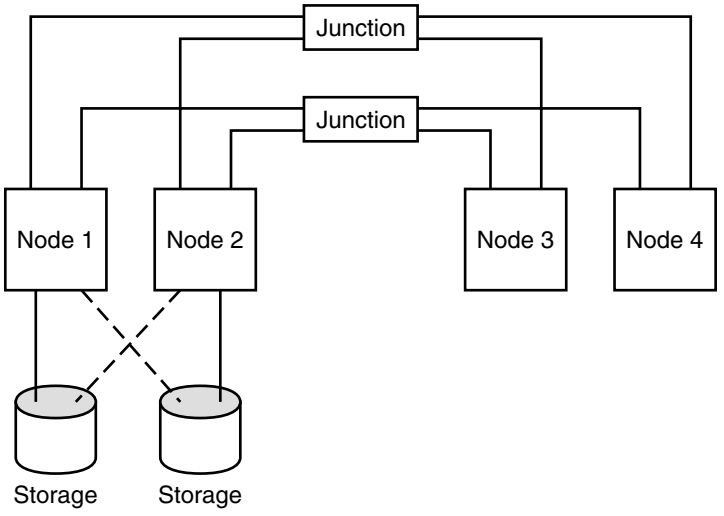


FIGURE 2–3 Pair+N Topology

SPARC: N+1 (Star) Topology for SPARC

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 node 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 node, Sun Cluster fails over the resources to the secondary node. The secondary node is where the resources function until they are switched back (either automatically or manually) to the primary node.

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

The following figure illustrates an N+1 configuration.

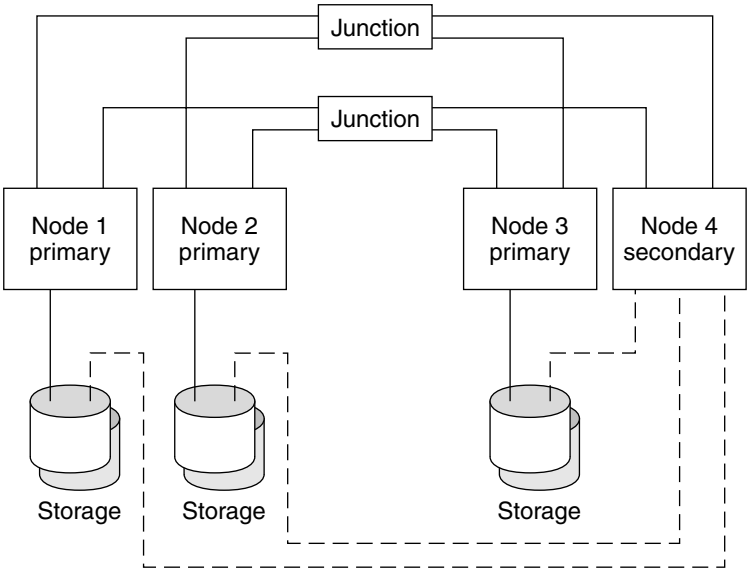


FIGURE 2–4 SPARC: N+1 Topology

SPARC: N*N (Scalable) Topology for SPARC

An N*N topology enables every shared storage device in the cluster to connect to every node in the cluster. This topology enables highly available applications to fail over from one node to another without service degradation. When failover occurs, the new node can access the storage device by using a local path instead of the private interconnect.

The following figure illustrates an N*N configuration.

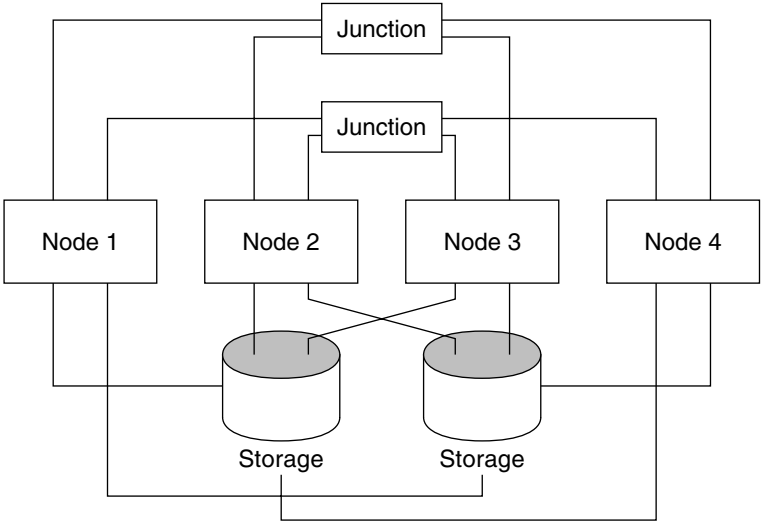


FIGURE 2–5 SPARC: N*N Topology

x86: Sun Cluster Topologies for x86

A topology is the connection scheme that connects the cluster nodes to the storage platforms that are used in the cluster. Sun Cluster supports any topology that adheres to the following guidelines.

- Sun Cluster that is composed of x86 based systems supports two nodes in a cluster.
- Shared storage devices must connect to both nodes.

Sun Cluster does not require you to configure a cluster by using specific topologies. The following clustered pair topology, which is the only topology for clusters that are composed of x86 based nodes, is described to provide the vocabulary to discuss a cluster's connection scheme. This topology is a typical connection scheme.

The following section includes a sample diagram of the topology.

x86: Clustered Pair Topology for x86

A clustered pair topology is two nodes that operate 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 or a failover or scalable application on the pair.

The following figure illustrates a clustered pair configuration.

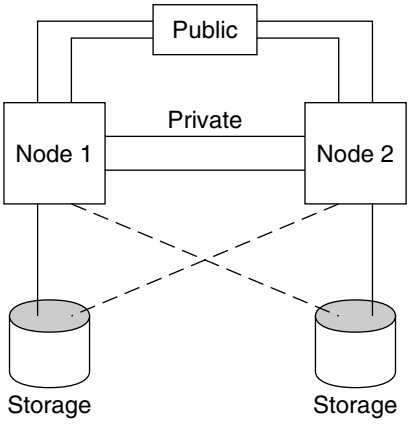


FIGURE 2–6 x86: Clustered Pair Topology

◆ ◆ ◆ CHAPTER 3

Key Concepts for System Administrators and Application Developers

This chapter describes the key concepts that are related to the software components of the Sun Cluster environment. The information in this chapter is directed primarily to system administrators and application developers who use the Sun Cluster API and SDK. Cluster administrators can use this information in preparation for installing, configuring, and administering cluster software. Application developers can use the information to understand the cluster environment in which they work.

This chapter covers the following topics:

- "Administrative Interfaces" on page 32
- "Cluster Time" on page 32
- "High-Availability Framework" on page 33
- "Global Devices" on page 35
- "Device Groups" on page 36
- "Global Namespace" on page 40
- "Cluster File Systems" on page 41
- "Disk Path Monitoring" on page 43
- "Quorum and Quorum Devices" on page 46
- "Data Services" on page 56
- "Developing New Data Services" on page 63
- "Using the Cluster Interconnect for Data Service Traffic" on page 64
- "Resources, Resource Groups, and Resource Types" on page 65
- "Support for Solaris Zones on Sun Cluster Nodes" on page 68
- "Service Management Facility" on page 72
- "System Resource Usage" on page 72
- "Data Service Project Configuration" on page 74
- "Public Network Adapters and Internet Protocol (IP) Network Multipathing" on page 83
- "SPARC: Dynamic Reconfiguration Support" on page 85

Administrative Interfaces

You can choose how you install, configure, and administer the Sun Cluster software from several user interfaces. You can accomplish system administration tasks either through the Sun Cluster Manager, formerly SunPlex™ Manager, graphical user interface (GUI), or through the command-line interface. On top of the command-line interface are some utilities, such as `scinstall` and `clsetup`, to simplify selected installation and configuration tasks. The Sun Cluster software also has a module that runs as part of Sun Management Center that provides a GUI to particular cluster tasks. This module is available for use in only SPARC based clusters. Refer to "Administration Tools" in *Sun Cluster System Administration Guide for Solaris OS* 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 Sun Cluster software 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`, `rdate`, or `xntpdate` (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 System 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 by using the `scinstall` command, 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. One node is designated the "preferred" node. Nodes are identified by their private host names and time synchronization occurs across the cluster interconnect. For instructions about how to configure the cluster for NTP, see Chapter 2, "Installing Software on the Cluster," in *Sun Cluster Software Installation Guide for Solaris OS*.

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 System and you want to change it, the procedure for doing so is included in Chapter 8, "Administering the Cluster," in *Sun Cluster System Administration Guide for Solaris OS*.

High-Availability Framework

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

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

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

Failed Cluster Component	Software Recovery	Hardware Recovery
Data service	HA API, HA framework	Not applicable
Public network adapter	Internet Protocol (IP) Network Multipathing	Multiple public network adapter cards
Cluster file system	Primary and secondary replicas	Multihost devices
Mirrored multihost device	Volume management (Solaris Volume Manager 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
Zone	HA API, HA framework	Not applicable

Sun Cluster software's high-availability framework detects a node or zone failure quickly and creates a new equivalent server for the framework resources on a remaining node or zone in the cluster. At no time are all framework resources unavailable. Framework resources that are unaffected by a crashed node or zone are fully available during recovery. Furthermore, framework resources of the failed node or zone 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) that are using the resource. The semantics of framework resource access are fully preserved across node or zone failure. The applications cannot detect that the framework resource server has been moved to another node. Failure of a single node is completely transparent to programs on remaining nodes by using the files, devices, and disk volumes that are attached to this node. This transparency exists if an alternative hardware path exists to the disks from another node. An example is the use of multihost devices that have ports to multiple nodes.

Zone Membership

Sun Cluster software also tracks zone membership by detecting when a zone boots up or halts. These changes also trigger a reconfiguration. A reconfiguration can redistribute cluster resources among the nodes and zones in the cluster.

Cluster Membership Monitor

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. In a synchronized configuration, cluster resources might be redistributed, based on the new membership of the cluster.

See "About Failure Fencing" on page 48 for more information about how the cluster protects itself from partitioning into multiple separate clusters.

Failfast Mechanism

The *failfast* mechanism detects a critical problem in either the global zone or in a non-global zone on a node. The action that Sun Cluster takes when failfast detects a problem depends on whether the problem occurs in the global zone or a non-global zone.

If the critical problem is located in the global zone, Sun Cluster forcibly shuts down the node. Sun Cluster then removes the node from cluster membership.

If the critical problem is located in a non-global zone, Sun Cluster reboots that non-global zone.

If a node loses connectivity with other nodes, the node attempts to form a cluster with the nodes with which communication is possible. If that set of nodes does not form a quorum, Sun Cluster software halts the node and "fences" the node from shared storage. See "About Failure Fencing" on page 48 for details about this use of failfast.

If one or more cluster-specific daemons die, Sun Cluster software declares that a critical problem has occurred. Sun Cluster software runs cluster-specific daemons in both the global zone and in non-global zones. If a critical problem occurs, Sun Cluster either shuts down and removes the node or reboots the non-global zone where the problem occurred.

When a cluster-specific daemon that runs in a non-global zone fails, a message similar to the following is displayed on the console.

```
cl_runtime: NOTICE: Failfast: Aborting because "pmfd" died in zone "zone4" (zone id 3)
35 seconds ago.
```

When a cluster-specific daemon that runs in the global zone fails and the node panics, a message similar to the following is displayed on the console.

```
panic[cpu1]/thread=2a10007fcc0: Failfast: Aborting because "pmfd" died in zone "global" (zone id 0)
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. Alternatively, if the cluster is composed of SPARC based systems, the node might remain at the OpenBoot™ PROM (OBP) prompt. The next action of the node is determined by the setting of the `auto-boot?` parameter. You can set `auto-boot?` with the `eeprom` command, at the OpenBoot PROM `ok` prompt. See the `eeprom`(1M) man page.

Cluster Configuration Repository (CCR)

The CCR uses a two-phase commit algorithm for updates: An update must be successfully completed 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 yourself. Each file contains a checksum record to ensure consistency between nodes. Updating CCR files yourself 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 Sun Cluster software 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. The Sun Cluster software then redirects the access to that path. Sun Cluster global devices include disks, CD-ROMs, and tapes. However, the only multiported global devices that Sun Cluster software supports are disks. Consequently, 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 enables consistent access to each device from any node in the cluster. The global device namespace is held in the `/dev/global` directory. See "Global Namespace" on page 40 for more information.

Multiported global devices provide more than one path to a device. Because multihost disks are part of a device group that is hosted by more than one node, the multihost disks are made highly available.

Device IDs and DID Pseudo Driver

The Sun Cluster software manages global devices through a construct known as the 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 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 devices, assigns each device a unique major and a minor number that are consistent on all nodes of the cluster. Access to the global devices is performed by using the unique device ID instead of the traditional Solaris device IDs, such as `c0t0d0` for a disk.

This approach ensures that any application that accesses disks (such as a volume manager or applications that use 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 identify a multihost disk as `c1t2d0`, and `Node2` might identify the same disk completely differently, as `c3t2d0`. The DID driver assigns a global name, such as `d10`, that the nodes use instead, giving each node a consistent mapping to the multihost disk.

You update and administer device IDs with the `cldevice` command. See the `cldevice`(1CL) man page.

Device Groups

In the Sun Cluster software, all multihost devices must be under control of the Sun Cluster software. You first create volume manager disk groups, either Solaris Volume Manager disk sets or VERITAS Volume Manager disk groups, on the multihost disks. Then, you register the volume manager disk groups as *device groups*. A device group is a type of global device. In addition, the Sun Cluster software automatically creates a raw 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 Sun Cluster software information about which nodes have a path to specific 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 device group, the data stored in that device group becomes highly available. The highly available device group can be used to contain cluster file systems.

Note – Device groups are independent of resource groups. One node or zone can master a resource group (representing a group of data service processes). Another node can master the disk groups that are being accessed by the data services. However, the best practice is to keep on the same node the device group that stores a particular application's data and the resource group that contains the application's resources (the application daemon). Refer to "Relationship Between Resource Groups and Device Groups" in *Sun Cluster Data Services Planning and Administration Guide for Solaris OS* for more information about the association between device groups and resource groups.

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

Device Group Failover

Because a disk enclosure is connected to more than one node, all 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 that is 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.

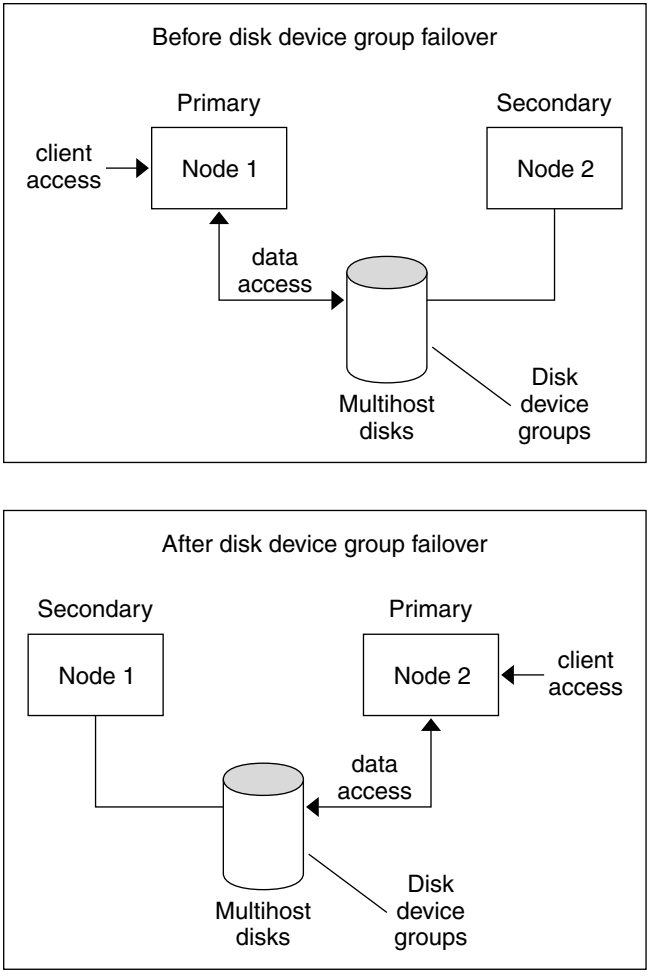


FIGURE 3–1 Device Group Before and After Failover

Multiported Device Groups

This section describes device group properties that enable you to balance performance and availability in a multiported disk configuration. Sun Cluster software provides two properties that configure a multiported disk configuration: `preferenced` and `numsecondaries`. You can control the order in which nodes attempt to assume control if a failover occurs by using the `preferenced` property. Use the `numsecondaries` property to set a desired number of secondary nodes for a device group.

A highly available service is considered down when the primary node or zone fails and when no eligible secondary nodes or zones can be promoted to primary nodes or zones. If service failover

occurs and the `preferenced` property is `true`, then the nodes or zones follow the order in the node list to select a secondary node or zone. The node list defines the order in which either nodes or zones attempt to assume primary control or transition from spare to secondary. You can dynamically change the preference of a device service by using the `clsetup` command. The preference that is associated with dependent service providers, for example a global file system, is identical to the preference of the device service.

Secondary nodes are check-pointed by the primary node during normal operation. In a multiported disk configuration, checkpointing each secondary node causes cluster performance degradation and memory overhead. Spare node support was implemented to minimize the performance degradation and memory overhead that checkpointing caused. By default, your device group has one primary and one secondary. The remaining available provider nodes become spares. If failover occurs, the secondary becomes primary and the node or highest in priority on the node list becomes secondary.

The desired number of secondary nodes can be set to any integer between one and the number of operational nonprimary provider nodes in the device group.

Note – If you are using Solaris Volume Manager, you must create the device group before you can set the `numsecondaries` property to a number other than the default.

The default desired number of secondaries for device services is one. The actual number of secondary providers that is maintained by the replica framework is the desired number, unless the number of operational nonprimary providers is less than the desired number. You must alter the `numsecondaries` property and double-check the node list if you are adding or removing nodes from your configuration. Maintaining the node list and desired number of secondaries prevents conflict between the configured number of secondaries and the actual number that is allowed by the framework.

- (Solaris Volume Manager) Use the `metaset` command for Solaris Volume Manager device groups, in conjunction with the `preferenced` and `numsecondaries` property settings, to manage the addition of nodes to and the removal of nodes from your configuration.
- (Veritas Volume Manager) Use the `cldevicegroup` command for VxVM device groups, in conjunction with the `preferenced` and `numsecondaries` property settings, to manage the addition of nodes to and the removal of nodes from your configuration.
- Refer to "Overview of the Administering Cluster File Systems" in *Sun Cluster System Administration Guide for Solaris OS* for procedural information about changing device group properties.

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 that is physically connected to multihost disks provides a path to the storage for any node in the cluster.

Normally, for Solaris Volume Manager, the volume manager namespaces are located in the `/dev/md/`*diskset*`/dsk` (and `rdsk`) directories. For Veritas VxVM, the volume manager namespaces are located in the `/dev/vx/dsk/`*disk-group* and `/dev/vx/rdsk/`*disk-group* directories. These namespaces consist of directories for each Solaris Volume Manager disk set and each VxVM disk group imported throughout the cluster, respectively. Each of these directories contains a device node for each metadevice or volume in that disk set or disk group.

In the Sun Cluster software, each device node in the local volume manager namespace is replaced by a symbolic link to a device node in the `/global/.devices/node@`*nodeID* file system. *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 the following:

- 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 Namespace Mappings

Component or Path	Local Node Namespace	Global Namespace
Solaris logical name	`/dev/dsk/c0t0d0s0`	`/global/.devices/node@`*nodeID*`/dev/dsk/c0t0d0s0`
DID name	`/dev/did/dsk/d0s0`	`/global/.devices/node@`*nodeID*`/dev/did/dsk/d0s0`
Solaris Volume Manager	`/dev/md/`*diskset*`/dsk/d0`	`/global/.devices/node@`*nodeID*`/dev/md/`*diskset*`/dsk/d0`

TABLE 3–2 Local and Global Namespace Mappings *(Continued)*

Component or Path	Local Node Namespace	Global Namespace
VERITAS Volume Manager	`/dev/vx/dsk/`*disk-group*`/v0`	`/global/.devices/node@`*nodeID*`/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 using the `cldevice` command. See the `cldevice`(1CL) man page.

Cluster File Systems

The cluster file system has the following features:

- File access locations are transparent. A process can open a file that is located anywhere in the system. 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 by using the `fcntl` command interfaces. Applications that run on multiple cluster nodes can synchronize access to data by using advisory file locking on a cluster file system. 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 if 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.

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. Clients verify the underlying file system (for example, UFS).

Using Cluster File Systems

In the Sun Cluster software, all multihost disks are placed into device groups, which can be Solaris Volume Manager disk sets, VxVM disk groups, or individual disks that are 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.

You can mount cluster file systems as you would mount file systems:

- **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:

  ```
  SPARC: # 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:

```
SPARC: /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-group*. See *Sun Cluster 3.1 9/04 Software Collection for Solaris OS (SPARC Platform Edition)* and *Sun Cluster System Administration Guide for Solaris OS* for more information.

`HAStoragePlus` Resource Type

The `HAStoragePlus` resource type is designed to make local and global file system configurations highly available. You can use the `HAStoragePlus` resource type to integrate your local or global file system into the Sun Cluster environment and make the file system highly available.

You can use the `HAStoragePlus` resource type to make a file system available to a non-global zone. To enable the `HAStoragePlus` resource type to do this, you must create a mount point in the global zone and in the non-global zone. The `HAStoragePlus` resource type makes the file system available to the non-global zone by mounting the file system in the global zone. The resource type then performs a loopback mount in the non-global zone.

Note – Local file systems include the UNIX File System (UFS), Quick File System (QFS), Veritas File System (VxFS), and Solaris ZFS (Zettabyte File System).

The HAStoragePlus resource type provides additional file system capabilities such as checks, mounts, and forced unmounts. These capabilities enable Sun Cluster to fail over local file systems. In order to fail over, the local file system must reside on global disk groups with affinity switchovers enabled.

See "Enabling Highly Available Local File Systems" in *Sun Cluster Data Services Planning and Administration Guide for Solaris OS* for information about how to use the HAStoragePlus resource type.

You can also use the HAStoragePlus resource type to synchronize the startup of resources and device groups on which the resources depend. For more information, see "Resources, Resource Groups, and Resource Types" on page 65.

syncdir Mount Option

You can use the syncdir mount option for cluster file systems that use UFS as the underlying file system. However, performance significantly improves if you do not specify syncdir. If you specify syncdir, the writes are guaranteed to be POSIX compliant. If you do not specify syncdir, you experience the same behavior as in NFS file systems. For example, without syncdir, you might 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 might have problems if you do not specify syncdir are rare.

If you are using a SPARC based cluster, VxFS does not have a mount option that is equivalent to the syncdir mount option for UFS. VxFS behavior is the same as for UFS when the syncdir mount option is not specified.

See "File Systems FAQs" on page 90 for frequently asked questions about global devices and cluster file systems.

Disk Path Monitoring

The current release of Sun Cluster software supports disk path monitoring (DPM). This section provides conceptual information about DPM, the DPM daemon, and administration tools that you use to monitor disk paths. Refer to *Sun Cluster System Administration Guide for Solaris OS* for procedural information about how to monitor, unmonitor, and check the status of disk paths.

DPM Overview

DPM improves the overall reliability of failover and switchover by monitoring secondary disk path availability. Use the cldevice command to verify the availability of the disk path that is used by a resource before the resource is switched. Options that are provided with the cldevice command enable you to monitor disk paths to a single node or to all nodes in the cluster. See the cldevice(1CL) man page for more information about command-line options.

The following table describes the default location for installation of DPM components.

Location	Component
Daemon	`/usr/cluster/lib/sc/scdpmd`
Command-line interface	`/usr/cluster/bin/cldevice`
Daemon status file (created at runtime)	`/var/run/cluster/scdpm.status`

A multithreaded DPM daemon runs on each node. The DPM daemon (`scdpmd`) is started by an `rc.d` script when a node boots. If a problem occurs, the daemon is managed by `pmfd` and restarts automatically. The following list describes how the `scdpmd` works on initial startup.

Note – At startup, the status for each disk path is initialized to `UNKNOWN`.

---

1. The DPM daemon gathers disk path and node name information from the previous status file or from the CCR database. See "Cluster Configuration Repository (CCR)" on page 35 for more information about the CCR. After a DPM daemon is started, you can force the daemon to read the list of monitored disks from a specified file name.
2. The DPM daemon initializes the communication interface to respond to requests from components that are external to the daemon, such as the command-line interface.
3. The DPM daemon pings each disk path in the monitored list every 10 minutes by using `scsi_inquiry` commands. Each entry is locked to prevent the communication interface access to the content of an entry that is being modified.
4. The DPM daemon notifies the Sun Cluster Event Framework and logs the new status of the path through the UNIX `syslogd` command. See the `syslogd`(1M) man page.

Note – All errors that are related to the daemon are reported by `pmfd`. All the functions from the API return `0` on success and `-1` for any failure.

The DPM daemon monitors the availability of the logical path that is visible through multipath drivers such as Sun StorEdge Traffic Manager, Sun StorEdge 9900 Dynamic Link Manager, and EMC PowerPath. The individual physical paths that are managed by these drivers are not monitored because the multipath driver masks individual failures from the DPM daemon.

Monitoring Disk Paths

This section describes two methods for monitoring disk paths in your cluster. The first method is provided by the `cldevice` command. Use this command to monitor, unmonitor, or display the status of disk paths in your cluster. You can also use this command to print a list of faulted disks and to monitor disk paths from a file. See the `cldevice`(1CL) man page.

The second method for monitoring disk paths in your cluster is provided by the Sun Cluster Manager, formerly SunPlex Manager, graphical user interface (GUI). Sun Cluster Manager provides a topological view of the monitored disk paths in your cluster. The view is updated every 10 minutes to provide information about the number of failed pings. Use the information that is provided by the Sun Cluster Manager GUI in conjunction with the `cldevice` command to administer disk paths. See Chapter 12, "Administering Sun Cluster With the Graphical User Interfaces," in *Sun Cluster System Administration Guide for Solaris OS* for information about Sun Cluster Manager.

Using the `cldevice` Command to Monitor and Administer Disk Paths

The `cldevice` command enables you to perform the following tasks:

- Monitor a new disk path
- Unmonitor a disk path
- Reread the configuration data from the CCR database
- Read the disks to monitor or unmonitor from a specified file
- Report the status of a disk path or all disk paths in the cluster
- Print all the disk paths that are accessible from a node

Issue the `cldevice` command with the disk path argument from any active node to perform DPM administration tasks on the cluster. The disk path argument consists of a node name and a disk name. The node name is not required. If you do not specify a node name, all nodes are affected by default. The following table describes naming conventions for the disk path.

Note – Always specify a global disk path name rather than a UNIX disk path name because a global disk path name is consistent throughout a cluster. A UNIX disk path name is not. For example, the disk path name can be `c1t0d0` on one node and `c2t0d0` on another node. To determine a global disk path name for a device that is connected to a node, use the `cldevice list` command before issuing DPM commands. See the `cldevice`(1CL) man page.

TABLE 3–3 Sample Disk Path Names

Name Type	Sample Disk Path Name	Description
Global disk path	`schost-1:/dev/did/dsk/d1`	Disk path `d1` on the `schost-1` node
	`all:d1`	Disk path `d1` on all nodes in the cluster
UNIX disk path	`schost-1:/dev/rdsk/c0t0d0s0`	Disk path `c0t0d0s0` on the `schost-1` node
	`schost-1:all`	All disk paths on the `schost-1` node
All disk paths	`all:all`	All disk paths on all nodes of the cluster

Using Sun Cluster Manager to Monitor Disk Paths

Sun Cluster Manager enables you to perform the following basic DPM administration tasks:

- Monitor a disk path
- Unmonitor a disk path
- View the status of all monitored disk paths in the cluster
- Enable or disable the automatic rebooting of a node when all monitored disk paths fail

The Sun Cluster Manager online help provides procedural information about how to administer disk paths.

Using the `clnode set` Command to Manage Disk Path Failure

You use the `clnode set` command to enable and disable the automatic rebooting of a node when all monitored disk paths fail. You can also use Sun Cluster Manager to perform these tasks.

Quorum and Quorum Devices

This section contains the following topics:

- "About Quorum Vote Counts" on page 47
- "About Failure Fencing" on page 48
- "About Quorum Configurations" on page 49
- "Adhering to Quorum Device Requirements" on page 50
- "Adhering to Quorum Device Best Practices" on page 50
- "Recommended Quorum Configurations" on page 51
- "Atypical Quorum Configurations" on page 54
- "Bad Quorum Configurations" on page 54

Note – For a list of the specific devices that Sun Cluster software supports as quorum devices, contact your Sun service provider.

Because cluster nodes share data and resources, a cluster must never split into separate partitions that are active at the same time because multiple active partitions might cause data corruption. The Cluster Membership Monitor (CMM) and quorum algorithm guarantee that at most one instance of the same cluster is operational at any time, even if the cluster interconnect is partitioned.

For an introduction to quorum and CMM, see "Cluster Membership" in *Sun Cluster Overview for Solaris OS*.

Two types of problems arise from cluster partitions:

- Split brain
- Amnesia

Split brain occurs when the cluster interconnect between nodes is lost and the cluster becomes partitioned into subclusters. Each partition "believes" that it is the only partition because the nodes in one partition cannot communicate with the node in the other partition.

Amnesia occurs when the cluster restarts after a shutdown with cluster configuration data older than at the time of the shutdown. This problem can occur when you start the cluster on a node that was not in the last functioning cluster partition.

Sun Cluster software avoids split brain and amnesia by:

- Assigning each node one vote
- Mandating a majority of votes for an operational cluster

A partition with the majority of votes gains *quorum* and is allowed to operate. This majority vote mechanism prevents split brain and amnesia when more than two nodes are configured in a cluster. However, counting node votes alone is not sufficient when more than two nodes are configured in a cluster. In a two-node cluster, a majority is two. If such a two-node cluster becomes partitioned, an external vote is needed for either partition to gain quorum. This external vote is provided by a *quorum device*.

About Quorum Vote Counts

Use the `clquorum show` command to determine the following information:

- Total configured votes
- Current present votes
- Votes required for quorum

See the `cluster`(1CL) man page.

Both nodes and quorum devices contribute votes to the cluster to form quorum.

A node contributes votes depending on the node's state:

- A node has a vote count of *one* when it boots and becomes a cluster member.
- A node has a vote count of *zero* when the node is being installed.
- A node has a vote count of *zero* when an system administrator places the node into maintenance state.

Quorum devices contribute votes that are based on the number of votes that are connected to the device. When you configure a quorum device, Sun Cluster software assigns the quorum device a vote count of *N*-1 where *N* is the number of connected votes to the quorum device. For example, a quorum device that is connected to two nodes with nonzero vote counts has a quorum count of one (two minus one).

A quorum device contributes votes if *one* of the following two conditions are true:

- At least one of the nodes to which the quorum device is currently attached is a cluster member.
- At least one of the nodes to which the quorum device is currently attached is booting, and that node was a member of the last cluster partition to own the quorum device.

You configure quorum devices during the cluster installation, or afterwards, by using the procedures that are described in Chapter 6, "Administering Quorum," in *Sun Cluster System Administration Guide for Solaris OS*.

About Failure Fencing

A major issue for clusters is a failure that causes the cluster to become partitioned (called *split brain*). When split brain occurs, 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 devices. When multiple nodes attempt to write to the disks, data corruption can occur.

Failure fencing limits node access to multihost devices by physically preventing access to the disks. Failure fencing applies only to nodes, not to zones. 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.

Device services provide failover capability for services that use multihost devices. When a cluster member that currently serves as the primary (owner) of the device group fails or becomes unreachable, a new primary is chosen. The new primary enables access to the device group to continue with only minor interruption. During this process, the old primary must forfeit 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 Sun Cluster software uses SCSI disk reservations to implement failure fencing. Using SCSI reservations, failed nodes are "fenced" away from the multihost devices, 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). Alternatively, access is restricted 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, the fenced node panics with a "reservation conflict" message on its console.

The discovery that a node is no longer a cluster member, triggers a SCSI reservation on all 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. The ioctl 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. Reservations 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, 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. 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, if the cluster is composed of SPARC based systems, stay at the OpenBoot PROM (OBP) prompt. The action that is taken is determined by the setting of the `auto-boot` parameter. You can set `auto-boot` with `eeprom`, at the OpenBoot PROM `ok` prompt in a SPARC based cluster. See the `eeprom`(1M) man page. Alternatively, you can set up this parameter with the SCSI utility that you optionally run after the BIOS boots in an x86 based cluster.

About Quorum Configurations

The following list contains facts about quorum configurations:

- Quorum devices can contain user data.
- In an N+1 configuration where *N* quorum devices are each connected to one of the `1` through *N* nodes and the `N+1` node, the cluster survives the death of either all `1` through *N* nodes or any of the *N*/**2** nodes. This availability assumes that the quorum device is functioning correctly.
- In an *N*-node configuration where a single quorum device connects to all nodes, the cluster can survive the death of any of the *N*-**1** nodes. This availability assumes that the quorum device is functioning correctly.
- In an *N*-node configuration where a single quorum device connects to all nodes, the cluster can survive the failure of the quorum device if all cluster nodes are available.

For examples of quorum configurations to avoid, see "Bad Quorum Configurations" on page 54. For examples of recommended quorum configurations, see "Recommended Quorum Configurations" on page 51.

Adhering to Quorum Device Requirements

Ensure that Sun Cluster software supports your specific device as a quorum device. If you ignore this requirement, you might compromise your cluster's availability.

Note – For a list of the specific devices that Sun Cluster software supports as quorum devices, contact your Sun service provider.

Sun Cluster software supports the following types of quorum devices:

- Multihosted shared disks that support SCSI-3 PGR reservations
- Dual-hosted shared disks that support SCSI-2 reservations
- A Network Attached Storage device from Network Appliance, Incorporated
- A quorum server process that runs on the quorum server machine

---

Note – A replicated device is not supported as a quorum device.

In a two–node configuration, you must configure at least one quorum device to ensure that a single node can continue if the other node fails. See Figure 3–2.

For examples of quorum configurations to avoid, see "Bad Quorum Configurations" on page 54. For examples of recommended quorum configurations, see "Recommended Quorum Configurations" on page 51.

Adhering to Quorum Device Best Practices

Use the following information to evaluate the best quorum configuration for your topology:

- Do you have a device that is capable of being connected to all nodes of the cluster?
  - If yes, configure that device as your one quorum device. You do *not* need to configure another quorum device because your configuration is the most optimal configuration.

    Caution – If you ignore this requirement and add another quorum device, the additional quorum device reduces your cluster's availability.

  - If no, configure your dual-ported device or devices.
- Ensure that the total number of votes contributed by quorum devices is strictly less than the total number of votes contributed by nodes. Otherwise, your nodes cannot form a cluster if all disks are unavailable, even if all nodes are functioning.

Note – In particular environments, you might desire to reduce overall cluster availability to meet your needs. In these situations, you can ignore this best practice. However, not adhering to this best practice decreases overall availability. For example, in the configuration that is outlined in "Atypical Quorum Configurations" on page 54 the cluster is less available: the quorum votes exceed the node votes. In a cluster, if access to the shared storage between `Nodes A` and `Node B` is lost, the entire cluster fails.

See "Atypical Quorum Configurations" on page 54 for the exception to this best practice.

- Specify a quorum device between every pair of nodes that shares access to a storage device. This quorum configuration speeds the failure fencing process. See "Quorum in Greater Than Two–Node Configurations" on page 52.
- In general, if the addition of a quorum device makes the total cluster vote even, the total cluster availability decreases.
- Quorum devices slightly slow reconfigurations after a node joins or a node dies. Therefore, do not add more quorum devices than are necessary.

For examples of quorum configurations to avoid, see "Bad Quorum Configurations" on page 54. For examples of recommended quorum configurations, see "Recommended Quorum Configurations" on page 51.

Recommended Quorum Configurations

This section shows examples of quorum configurations that are recommended. For examples of quorum configurations you should avoid, see "Bad Quorum Configurations" on page 54.

Quorum in Two–Node Configurations

Two quorum votes are required for a two-node cluster to form. These two votes can derive from the two cluster nodes, or from just one node and a quorum device.

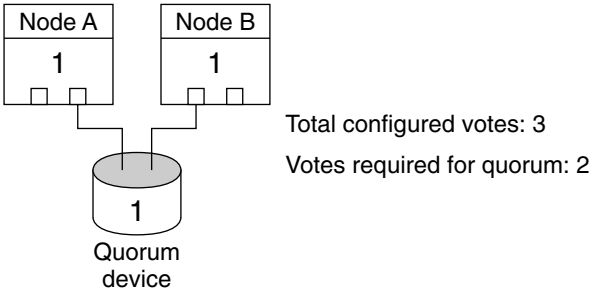
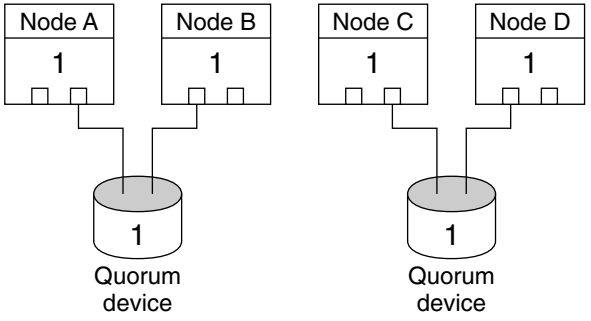


FIGURE 3–2 Two–Node Configuration

Quorum in Greater Than Two–Node Configurations

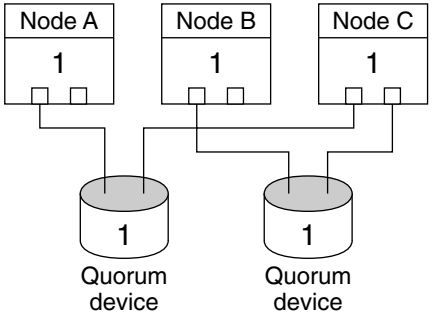
You can configure a greater than two-node cluster without a quorum device. However, if you do so, you cannot start the cluster without a majority of nodes in the cluster.



Total configured votes: 6

Votes required for quorum: 4

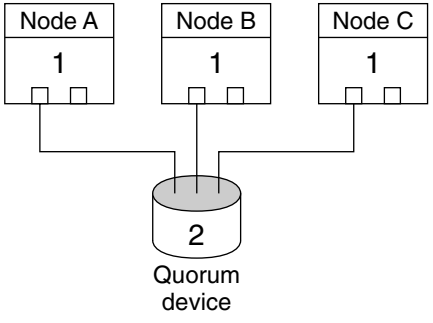
In this configuration, each pair must be available for either pair to survive.



Total configured votes: 5

Votes required for quorum: 3

In this configuration, usually applications are configured to run on Node A and Node B and use Node C as a hot spare.



Total configured votes: 5

Votes required for quorum: 3

In this configuration, the combination of any one or more nodes and the quorum device can form a cluster.

Atypical Quorum Configurations

Figure 3–3 assumes you are running mission-critical applications (Oracle database, for example) on `Node A` and `Node B`. If `Node A` and `Node B` are unavailable and cannot access shared data, you might want the entire cluster to be down. Otherwise, this configuration is suboptimal because it does not provide high availability.

For information about the best practice to which this exception relates, see "Adhering to Quorum Device Best Practices" on page 50.

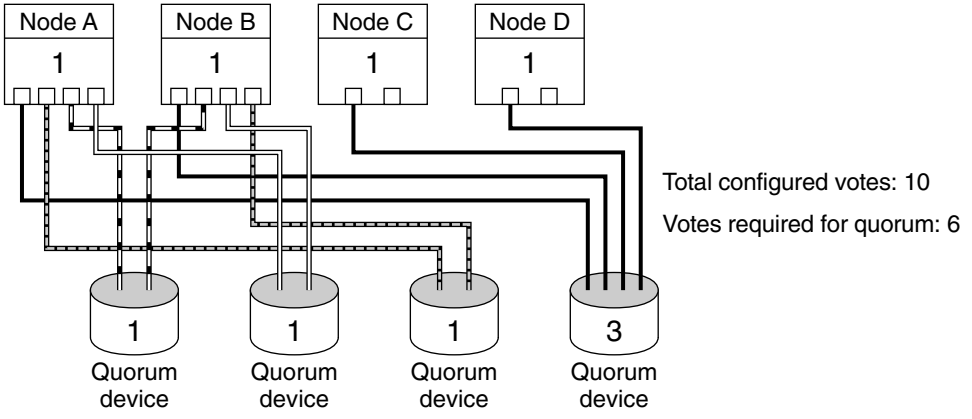
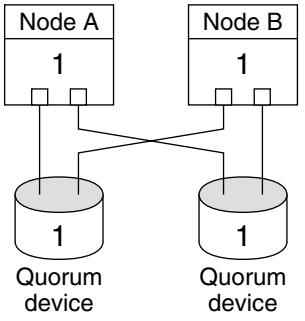


FIGURE 3–3 Atypical Configuration

Bad Quorum Configurations

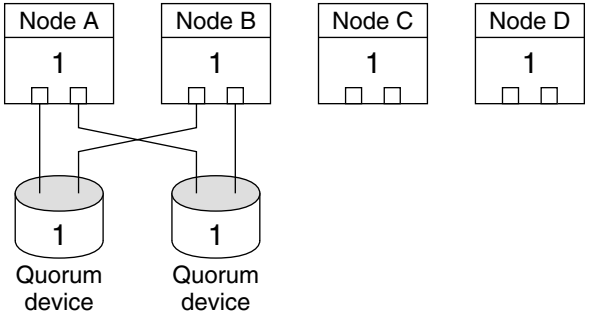
This section shows examples of quorum configurations you should avoid. For examples of recommended quorum configurations, see "Recommended Quorum Configurations" on page 51.



Total configured votes: 4

Votes required for quorum: 3

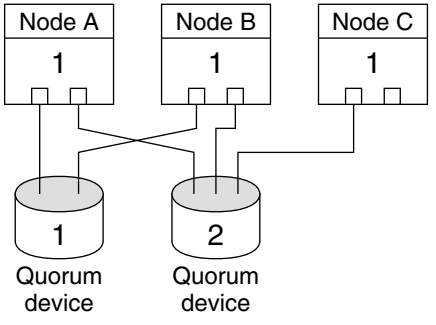
This configuration violates the best practice that quorum device votes should be strictly less than votes of nodes.



Total configured votes: 6

Votes required for quorum: 4

This configuration violates the best practice that you should not add quorum devices to make total votes even. This configuration does not add availability.



Total configured votes: 6

Votes required for quorum: 4

This configuration violates the best practice that quorum device votes should be strictly less than votes of nodes.

Data Services

The term *data service* describes an application, such as Sun Java System Web Server or Oracle, 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 control the following actions of the application.

- Start
- Stop
- Monitor and take corrective measures

For information about data service types, see "Data Services" in *Sun Cluster Overview for Solaris OS*.

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). The only difference between the two configurations is that the clustered application might run faster and is more highly available.

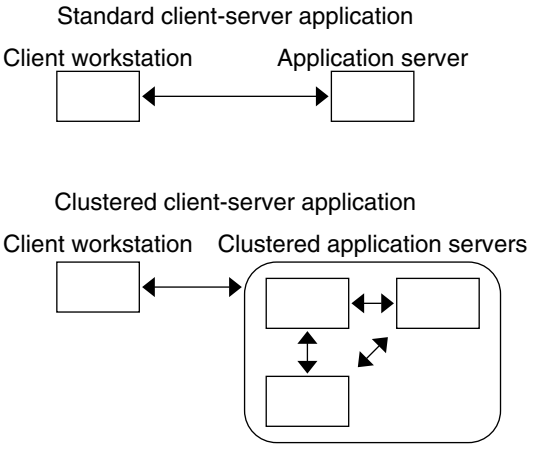


FIGURE 3–4 Standard Compared to Clustered Client-Server Configuration

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

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

Some data services require you to specify either logical host names or shared addresses as the network interfaces. Logical host names and shared addresses are not always interchangeable. Other data services allow you to specify either logical host names or shared addresses. Refer to the installation and configuration for each data service for details about the type of interface you must specify.

A network resource is not associated with a specific physical server. A network resource 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 fail 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 host name, in this example) can move between two or more of the cluster nodes. The application is configured to use this logical host name in place of a host name that is associated with a particular server.

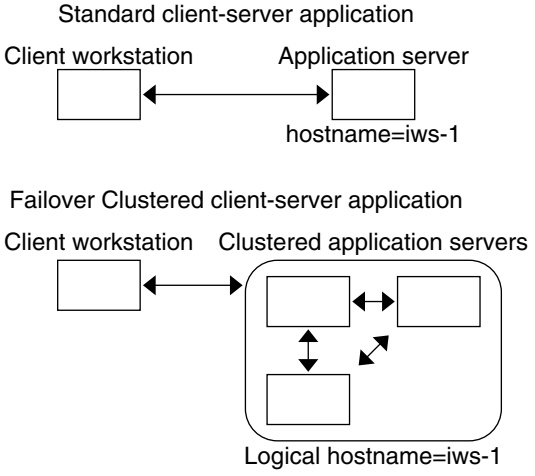


FIGURE 3–5 Fixed Host Name Compared to Logical Host Name

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

The difference between the logical host name model and the scalable service model is that in the latter, each node also has the shared address actively configured on its loopback interface. This configuration enables multiple instances of a data service to be 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 scales.

If the global interface node fails, the shared address can be started on another node that is also running an instance of the application (thereby making this other node the new global interface node). 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. The application is configured to use this shared address in place of a host name that is associated with a particular server. This scheme is similar to how a logical host name is used for a failover data service.

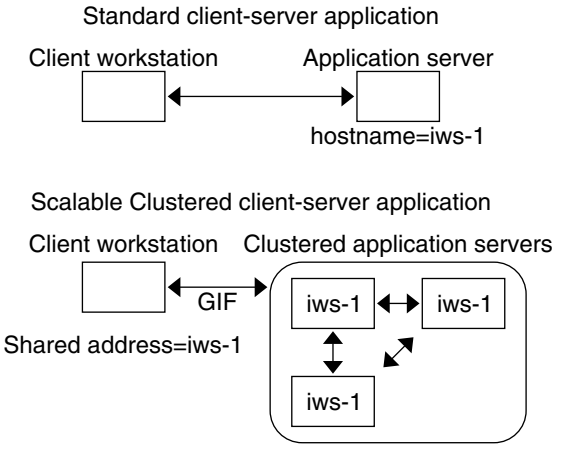


FIGURE 3–6 Fixed Host Name Compared to Shared Address

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 or in the zones. These methods, along with the cluster framework software and multihost devices, 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 host names and shared addresses).

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

Failover Data Services

If the node or zone on which the data service is running (the primary node) fails, the service is migrated to another working node or zone without user intervention. Failover services use a *failover resource group*, which is a container for application instance resources and network resources (*logical*

host names). Logical host names are IP addresses that can be configured on one node or zone, and at a later time, automatically configured down on the original node or zone and configured on another node or zone.

For failover data services, application instances run only on a single node or zone. If the fault monitor detects an error, it either attempts to restart the instance on the same node or zone, or to start the instance on another node or zone (failover). The outcome depends on how you have configured the data service.

Scalable Data Services

The scalable data service has the potential for active instances on multiple nodes or zones.

Scalable services use the following two resource groups:

- A *scalable resource group* contains the application resources.
- A failover resource group, which contains the network resources (*shared addresses*) on which the scalable service depends. A shared address is a network address. This network address can be bound by all scalable services that are running on nodes or zones within the cluster. This shared address enables these scalable services to scale on those nodes or zones. A cluster can have multiple shared addresses, and a service can be bound to multiple shared addresses.

A scalable resource group can be online on multiple nodes or zones simultaneously. As a result, multiple instances of the service can be running at once. However, a scalable resource group that uses a shared address to balance the service load between nodes can be online in only one zone per physical node. All scalable resource groups use load balancing. All nodes or zones that host a scalable service use the same shared address to host the service. The failover resource group that hosts the shared address is online on only one node or zone at a time.

Service requests enter the cluster through a single network interface (the global interface). These requests are distributed to the nodes or zones, based on one of several predefined algorithms that are set by the *load-balancing policy*. The cluster can use the load-balancing policy to balance the service load between several nodes or zones. Multiple global interfaces can exist on different nodes or zones that host other shared addresses.

For scalable services, application instances run on several nodes or zones simultaneously. If the node or zone that hosts the global interface fails, the global interface fails over to another node or zone. If an application instance that is running fails, the instance attempts to restart on the same node or zone.

If an application instance cannot be restarted on the same node or zone, and another unused node or zone is configured to run the service, the service fails over to the unused node or zone. Otherwise, the service continues to run on the remaining nodes or zones, 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 (used in SPARC-based clusters only). 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). Additionally, all the Apache application resources depend on the network resource `schost-2`, which is a shared address (dashed lines).

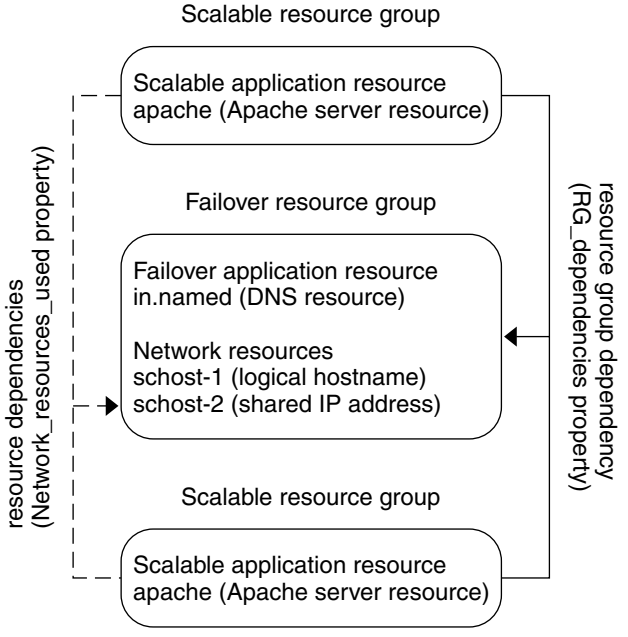


FIGURE 3–7 SPARC: Failover and Scalable Resource Group Example

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
- Sticky

A *pure* service is capable of having any of its instances respond to client requests. A *sticky* service is capable of having a client send 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. The load is distributed among various nodes according to specified weight values. For example, in a three-node cluster, suppose that each node has the weight of 1. Each node services one third of the requests from any client on behalf of that service. The cluster administrator can change weights at any time with an administrative command or with Sun Cluster Manager.

The weighted load-balancing policy is set by 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 is set to 1 by default.

The weighted policy redirects a certain percentage of the traffic from clients to a particular node. Given X=weight and A=the total weights of all active nodes, an active node can expect approximately X/A of the total new connections to be directed to the active node. However, the total number of connections must be large enough. This policy does not address individual requests.

Note that the weighted policy is not round robin. A round-robin policy would always cause each request from a client to go to a different node. For example, the first request would go to node 1, the second request would go to node 2, and so on.

A sticky service has two flavors, *ordinary sticky* and *wildcard sticky*.

Sticky services enable concurrent application-level sessions over multiple TCP connections to share in-state memory (application session state).

Ordinary sticky services enable a client to share the state between multiple concurrent TCP connections. The client is said to be “sticky” toward that server instance that is listening on a single port.

The client is guaranteed that all requests go to the same server instance, provided that the following conditions are met:

- The instance remains up and accessible.
- 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. However, the connections exchange cached session information between them at the service.

A generalization of a sticky policy extends to multiple scalable services that exchange session information in the background and at the same instance. When these services exchange session information in the background and at the same instance, the client is said to be “sticky” toward multiple server instances on the same node that is listening on different ports.

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

In the ordinary sticky policy, the set of ports is known at the time the application resources are configured. This policy is set by using the `LB_STICKY` value for the `Load_balancing_policy` resource property.

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 pots that have the same IP address.

A good example of this policy is passive mode FTP. For example, a client connects to an FTP server on port 21. The server then instructs the client 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.

The sticky-wildcard policy is a superset of the ordinary sticky policy. For a scalable service that is 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 by using the `LB_STICKY_WILD` value for the `Load_balancing_policy` resource property.

For each one of these sticky policies, the weighted load-balancing policy is in effect by default. Therefore, a client's initial request is directed to the instance that the load balancer dictates. After the client establishes an affinity for the node where the instance is running, future requests are conditionally directed to that instance. The node must be accessible and the load-balancing policy must not have changed.

Failback Settings

Resource groups fail over from one node or zone to another. When this failover occurs, the original secondary becomes the new primary. The failback settings specify the actions that occur 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 by using the `Failback` resource group property setting.

If the original node or zone that hosts the resource group fails and reboots repeatedly, setting failback might result in reduced availability for the resource group.

Data Services Fault Monitors

Each Sun Cluster data service supplies a fault monitor that periodically probes the data service to determine its health. A fault monitor verifies that the application daemon or daemons are running and that clients are being served. Based on the information that probes return, 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 Sun does not offer the application that you want to run as a failover or scalable service, you have an alternative. Use a Sun Cluster API or the DSET API to configure the application to run as a failover or scalable service. However, not all applications can become a scalable service.

Characteristics of Scalable Services

A set of criteria determines whether an application can become a scalable service. To determine if your application can become a scalable service, see "Analyzing the Application for Suitability" in *Sun Cluster Data Services Developer's Guide for Solaris OS*.

This set of criteria is summarized as follows:

- First, such a service is composed of one or more server *instances*. Each instance runs on a different node or zone. Two or more instances of the same service cannot run on the same node or zone.
- Second, if the service provides an external logical data store, you must exercise caution. 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 the use of "external" to distinguish the store from in-memory state. The term "logical" indicates that the store appears as a single entity, although it might itself be replicated. Furthermore, in this data store, when any server instance updates the data store, this update is immediately "seen" by other instances.

  The Sun Cluster software 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 instance has access to this external log, and each might 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 through `fcntl` and `lockf` to achieve the synchronization that you want.

  Another example of this type of store is a back-end database, such as highly available Oracle Real Application Clusters Guard for SPARC based clusters or Oracle. This type of back-end database server provides built-in synchronization by using database query or update transactions. Therefore, multiple server instances do not need to implement their own synchronization.

  The Sun IMAP server is an example of a service that is not a scalable service. 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 can have private data that is disjoint from the data of other instances. In such a case, the service does not need synchronized 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 this data can become globally accessible.

Data Service API and Data Service Development Library API

The Sun Cluster software provides the following to make applications highly available:

- Data services that are supplied as part of the Sun Cluster software
- A data service API
- A development library API for data services
- A "generic" data service

The *Sun Cluster Data Services Planning and Administration Guide for Solaris OS* describes how to install and configure the data services that are supplied with the Sun Cluster software. The *Sun Cluster 3.1 9/04 Software Collection for Solaris OS (SPARC Platform Edition)* describes how to instrument other applications to be highly available under the Sun Cluster framework.

The Sun Cluster APIs enable application developers to develop fault monitors and scripts that start and stop data service instances. With these tools, an application can be implemented as a failover or a scalable data service. The Sun Cluster software provides a "generic" data service. Use this generic data service to quickly generate an application's required start and stop methods and to implement the data service 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.

Sun Cluster software uses multiple interconnects to achieve the following goals:

- Ensure high availability
- Improve performance

For both internal and external traffic such as file system data or scalable services data, messages are striped across all available interconnects. The cluster interconnect is also available to applications, for highly available communication between nodes. For example, a distributed application might have components that are 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 or zones, an application must use the private host names that you configured during the Sun Cluster installation. For example, if the private host name for `node1` is `clusternode1-priv`, use this name to communicate with `node1` over the cluster interconnect. TCP sockets that are opened by using this name are routed over the cluster interconnect and can be transparently rerouted if a private network adapter fails. Application communication between any two nodes is striped over all interconnects. The traffic for a given TCP connection flows on one interconnect at any point. Different TCP connections are striped across all interconnects. Additionally, UDP traffic is always striped across all interconnects.

An application can optionally use a zone's private host name to communicate over the cluster interconnect between zones. However, you must first set each zone's private host name before the application can begin communicating. Each zone must have its own private host name to communicate. An application that is running in one zone must use the private host name in the same zone to communicate with private host names in other zones. An application in one zone cannot communicate through the private host name in another zone.

Because you can configure the private host names during your Sun Cluster installation, the cluster interconnect uses any name that you choose at that time. To determine the actual name, use the `scha_cluster_get` command with the `scha_privatelink_hostname_node` argument. See the `scha_cluster_get`(1HA) man page.

Each node or zone is also assigned a fixed per-node address. This per-node address is plumbed on the `clprivnet` driver. The IP address maps to the private host name for the node or zone: `clusternode1-priv`. See the `clprivnet`(7) man page.

If your application requires consistent IP addresses at all points, configure the application to bind to the per-node address on both the client and the server. All connections appear then to originate from and return to the per-node address.

Resources, Resource Groups, and Resource Types

Data services use several types of *resources*: applications such as Sun Java System Web Server or Apache Web Server use network addresses (logical host names and shared addresses) on 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-server` 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. Several resource types are defined.

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

The `HAStoragePlus` resource type is used to synchronize the startup of resources and device groups on which the resources depend. This resource type ensures that before a data service starts, the paths to a cluster file system's mount points, global devices, and device group names are available. For more information, see "Synchronizing the Startups Between Resource Groups and Device Groups" in *Sun Cluster Data Services Planning and Administration Guide for Solaris OS*. The `HAStoragePlus` resource type also enables local file systems to be highly available. For more information about this feature, see "`HAStoragePlus` Resource Type" on page 42.

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 that contains application resources online, the application is started. The data service start method waits until the application is 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 Data Services Planning and Administration Guide for Solaris OS* for more information about this process.

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 or in selected zones 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 located in "Resource and Resource Group States and Settings" on page 66.

Refer to "Data Service Project Configuration" on page 74 for information about how to launch Solaris projects under RGM control.

Resource and Resource Group States and Settings

A system administrator applies static settings to resources and resource groups. You can change these settings only by administrative action. The RGM moves resource groups between dynamic "states."

These settings and states are as follows:

- **Managed or unmanaged settings.** These cluster-wide settings apply only to resource groups. The RGM manages resource groups. You can use the `clresourcegroup` command to request that the RGM manage or unmanage a resource group. These resource group settings do not change when you reconfigure a cluster.

  When a resource group is first created, it is unmanaged. A resource group 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 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 are located 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 the `INIT` and `FINI` methods are for network resources for scalable services. However, a data service developer can use these methods for any initialization or cleanup work that is not performed by the application.

- **Enabled or disabled settings.** These settings apply to resources on one or more nodes or zones. A system administrator can use the `clresource` command to enable or disable a resource on one or more nodes or zones. These settings do not change when the cluster administrator reconfigures a cluster.

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

  If you want to make the resource unavailable on all cluster nodes or zones, disable the resource on all cluster nodes or zones. A disabled resource is not available for general use on the cluster nodes or zones that you specify.

- **Online or offline states.** These dynamic states apply to both resources and resource groups.

  Online and offline states change as the cluster transitions through cluster reconfiguration steps during switchover or failover. You can also change the online or offline state of a resource or a resource group by using the `clresource` and `clresourcegroup` commands.

  A failover resource or resource group can only be online on one node or in one zone at any time. A scalable resource or resource group can be online on some nodes or zones and offline on others. During a switchover or failover, resource groups and the resources within them are taken offline on one node or in one zone and then brought online on another node or zone.

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

  You can temporarily suspend the automatic recovery actions of a resource group. You might need to suspend the automatic recovery of a resource group to investigate and fix a problem in the cluster. Or, you might need to perform maintenance on resource group services.

A suspended resource group is *not* automatically restarted or failed over until you explicitly issue the command that resumes automatic recovery. Whether online or offline, suspended data services remain in their current state. You can still manually switch the resource group to a different state on specified nodes or zones. You can also still enable or disable individual resources in the resource group.

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 that are 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 Sun Cluster 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. For the set of standard properties, see Appendix B, "Standard Properties," in *Sun Cluster Data Services Planning and Administration Guide for Solaris OS*.

The RGM 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 guide for the data service.

Support for Solaris Zones on Sun Cluster Nodes

Solaris zones provide a means of creating virtualized operating system environments within an instance of the Solaris 10 OS. Solaris zones enable one or more applications to run in isolation from other activity on your system. The Solaris zones facility is described in Part II, "Zones," in *System Administration Guide: Solaris Containers-Resource Management and Solaris Zones*.

When you run Sun Cluster software on the Solaris 10 OS, you can create any number of zones on a physical node.

You can use Sun Cluster software to manage the availability and scalability of applications that are running in Solaris non-global zones on cluster nodes. Sun Cluster software provides support for applications that are running in non-global zones as follows:

- Directly through the RGM
- Through the Sun Cluster HA for Solaris Containers data service

Support for Solaris Zones on Sun Cluster Nodes Directly Through the RGM

On a cluster where the Solaris 10 OS is running, you can configure a resource group to run in the global zone or in non-global zones. The RGM manages each zone as a switchover target. If a non-global zone is specified in the node list of a resource group, the RGM brings the resource group online in the specified zone on the node.

Figure 3–8 illustrates the failover of resource groups between zones on Sun Cluster nodes in a two-node cluster. In this example, identical zones are configured on each node to simplify the administration of the cluster.

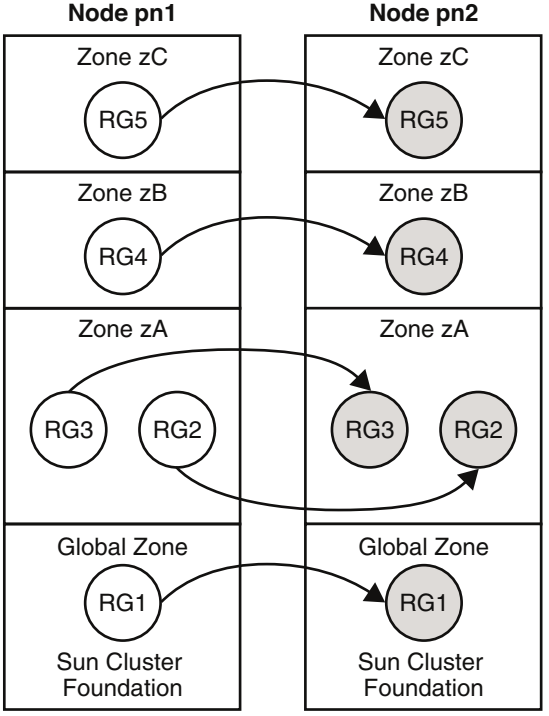


FIGURE 3–8 Failover of Resource Groups Between Zones on Sun Cluster Nodes

A failover resource group can fail over to a zone on another node or on the same node. However, if the node fails, the failing over of this resource group to a zone on the same node does not provide high availability. Nonetheless, you might find this failing over of a resource group to a zone on the same node useful in testing or prototyping.

You can configure a scalable resource group (which uses network load balancing) to run in a non-global zone as well. However, do not configure a scalable resource group to run in multiple zones on the same node.

In Sun Cluster commands, you specify a zone by appending the name of the zone to the name of the physical node, and separating them with a colon, for example:

```
phys-schost-1:zoneA
```

You can specify a zone with several Sun Cluster commands, for example:

- `clnode`(1CL)
- `clreslogicalhostname`(1CL)
- `clresource`(1CL)
- `clresourcegroup`(1CL)
- `clresourcetype`(1CL)
- `clressharedaddress`(1CL)

Criteria for Using Support for Solaris Zones Directly Through the RGM

Use support for Solaris zones directly through the RGM if any of following criteria is met:

- Your application cannot tolerate the additional failover time that is required to boot a zone.
- You require minimum downtime during maintenance.
- You require dual-partition software upgrade.
- You are configuring a data service that uses a shared address resource for network load balancing.

Requirements for Using Support for Solaris Zones Directly Through the RGM

If you plan to use support for Solaris zones directly through the RGM for an application, ensure that the following requirements are met:

- The application is supported to run in non-global zones.
- The data service for the application is supported to run in non-global zones.

If you use support for Solaris zones directly through the RGM, configure resources and resource groups as follows:

- Ensure that resource groups that are related by an affinity are configured to run in the same zone. An affinity between resource groups that are configured to run in different zones on the same node has no effect.
- Configure every application in the non-global zone as a resource.

Additional Information About Support for Solaris Zones Directly Through the RGM

For information about how to configure support for Solaris zones directly through the RGM, see the following documentation:

- "Guidelines for Non-Global Zones in a Cluster" in *Sun Cluster Software Installation Guide for Solaris OS*
- "Zone Names" in *Sun Cluster Software Installation Guide for Solaris OS*
- "Configuring a Non-Global Zone on a Cluster Node" in *Sun Cluster Software Installation Guide for Solaris OS*
- *Sun Cluster Data Services Planning and Administration Guide for Solaris OS*
- Individual data service guides

Support for Solaris Zones on Sun Cluster Nodes Through Sun Cluster HA for Solaris Containers

The Sun Cluster HA for Solaris Containers data service manages each zone as a resource that is controlled by the RGM.

Criteria for Using Sun Cluster HA for Solaris Containers

Use the Sun Cluster HA for Solaris Containers data service if any of following criteria is met:

- You require delegated root access.
- The application is not supported in a cluster.
- You require affinities between resource groups that are to run in different zones on the same node.

Requirements for Using Sun Cluster HA for Solaris Containers

If you plan to use the Sun Cluster HA for Solaris Containers data service for an application, ensure that the following requirements are met:

- The application is supported to run in non-global zones.
- The application is integrated with the Solaris OS through a script, a run-level script, or a Solaris Service Management Facility (SMF) manifest.
- The additional failover time that is required to boot a zone is acceptable.
- Some downtime during maintenance is acceptable.

Additional Information About Sun Cluster HA for Solaris Containers

For information about how to use the Sun Cluster HA for Solaris Containers data service, see *Sun Cluster Data Service for Solaris Containers Guide*.

Service Management Facility

The Solaris Service Management Facility (SMF) enables you to run and administer applications as highly available and scalable resources. Like the Resource Group Manager (RGM), the SMF provides high availability and scalability, but for the Solaris Operating System.

Sun Cluster provides three proxy resource types that you can use to enable SMF services in a cluster. These resource types, `SUNW.Proxy_SMF_failover`, `SUNW.Proxy_SMF_loadbalanced`, and `SUNW.Proxy_SMF_multimaster`, enable you to run SMF services in a failover, scalable, and multi-master configuration, respectively. The SMF manages the availability of SMF services on a single node. The SMF uses the callback method execution model to run services.

The SMF also provides a set of administrative interfaces for monitoring and controlling services. These interfaces enable you to integrate your own SMF-controlled services into Sun Cluster. This capability eliminates the need to create new callback methods, rewrite existing callback methods, or update the SMF service manifest. You can include multiple SMF resources in a resource group and you can configure dependencies and affinities between them.

The SMF is responsible for starting, stopping, and restarting these services and managing their dependencies. Sun Cluster is responsible for managing the service in the cluster and for determining the nodes on which these services are to be started.

The SMF runs as a daemon, `svc.startd`, on each cluster node. The SMF daemon automatically starts and stops resources on selected nodes according to pre-configured policies.

The services that are specified for an SMF proxy resource can reside in a global zone or in a non-global zone. However, all the services that are specified for the same SMF proxy resource must be located in the same zone. SMF proxy resources work in any zone.

System Resource Usage

System resources include aspects of CPU usage, memory usage, swap usage, and disk and network throughput. Sun Cluster enables you to monitor how much of a specific system resource is being used by an *object type*. An object type includes a node, zone, disk, network interface, or resource group. Sun Cluster also enables you to control the CPU that is available to a resource group.

Monitoring and controlling system resource usage can be part of your resource management policy. The cost and complexity of managing numerous machines encourages the consolidation of several applications on larger servers. Instead of running each workload on separate systems, with full access to each system's resources, you use resource management to segregate workloads within the system. Resource management enables you to lower overall total cost of ownership by running and controlling several applications on a single Solaris system.

Resource management ensures that your applications have the required response times. Resource management can also increase resource use. By categorizing and prioritizing usage, you can effectively use reserve capacity during off-peak periods, often eliminating the need for additional processing power. You can also ensure that resources are not wasted because of load variability.

To use the data that Sun Cluster collects about system resource usage, you must do the following:

- Analyze the data to determine what it means for your system.
- Make a decision about the action that is required to optimize your usage of hardware and software resources.
- Take action to implement your decision.

By default, system resource monitoring and control are not configured when you install Sun Cluster. For information about configuring these services, see Chapter 9, "Configuring Control of CPU Usage," in *Sun Cluster System Administration Guide for Solaris OS*.

System Resource Monitoring

By monitoring system resource usage, you can do the following:

- Collect data that reflects how a service that is using specific system resources is performing.
- Discover resource bottlenecks or overload and so preempt problems.
- More efficiently manage workloads.

Data about system resource usage can help you determine the hardware resources that are underused and the applications that use many resources. Based on this data, you can assign applications to nodes or zones that have the necessary resources and choose the node or zone to which to failover. This consolidation can help you optimize the way that you use your hardware and software resources.

Monitoring all system resources at the same time might be costly in terms of CPU. Choose the system resources that you want to monitor by prioritizing the resources that are most critical for your system.

When you enable monitoring, you choose the *telemetry attribute* that you want to monitor. A telemetry attribute is an aspect of system resources. Examples of telemetry attributes include the amount of free CPU or the percentage of blocks that are used on a device. If you monitor a telemetry attribute on an object type, Sun Cluster monitors this telemetry attribute on all objects of that type in the cluster. Sun Cluster stores a history of the system resource data that is collected for seven days.

If you consider a particular data value to be critical for a system resource, you can set a *threshold* for this value. When setting a threshold, you also choose how critical this threshold is by assigning it a severity level. If the threshold is crossed, Sun Cluster changes the severity level of the threshold to the severity level that you choose.

Control of CPU

Each application and service that is running on a cluster has specific CPU needs. Table 3–4 lists the CPU control activities that are available on different versions of the Solaris OS.

TABLE 3–4 CPU Control

Solaris Version	Zone	Control
Solaris 9 OS	Not available	Assign CPU shares
Solaris 10 OS	Global	Assign CPU shares
Solaris 10 OS	Non-global	Assign CPU shares
		Assign number of CPU
		Create dedicated processor sets

Note – If you want to apply CPU shares, you must specify the Fair Share Scheduler (FFS) as the default scheduler in the cluster.

Controlling the CPU that is assigned to a resource group in a dedicated processor set in a non-global zone offers the strictest level of control. If you reserve CPU for a resource group, this CPU is not available to other resource groups.

Viewing System Resource Usage

You can view system resource data and CPU assignments by using the command line or through Sun Cluster Manager. The system resources that you choose to monitor determine the tables and graphs that you can view.

By viewing the output of system resource usage and CPU control, you can do the following:

- Anticipate failures due to the exhaustion of system resources.
- Detect unbalanced usage of system resources.
- Validate server consolidation.
- Obtain information that enables you to improve the performance of applications.

Sun Cluster does not provide advice about the actions to take, nor does it take action for you based on the data that it collects. You must determine whether the data that you view meets your expectations for a service. You must then take action to remedy any observed performance.

Data Service Project Configuration

Data services can be configured to launch under a Solaris project name when brought online by using the RGM. The configuration associates a resource or resource group managed by the RGM with a Solaris project ID. The mapping from your resource or resource group to a project ID gives you the ability to use sophisticated controls that are available in the Solaris OS to manage workloads and consumption within your cluster.

Note – You can perform this configuration if you are using Sun Cluster on the Solaris 9 OS or on the Solaris 10 OS.

Using the Solaris management functionality in a Sun Cluster environment enables you to ensure that your most important applications are given priority when sharing a node or zone with other applications. Applications might share a node or zone if you have consolidated services or because applications have failed over. Use of the management functionality described herein might improve availability of a critical application by preventing lower-priority applications from overconsuming system supplies such as CPU time.

Note – The Solaris documentation for this feature describes CPU time, processes, tasks and similar components as "resources". Meanwhile, Sun Cluster documentation uses the term "resources" to describe entities that are under the control of the RGM. The following section uses the term "resource" to refer to Sun Cluster entities that are under the control of the RGM. The section uses the term "supplies" to refer to CPU time, processes, and tasks.

This section provides a conceptual description of configuring data services to launch processes on a specified Solaris OS `project`(4). This section also describes several failover scenarios and suggestions for planning to use the management functionality provided by the Solaris Operating System.

For detailed conceptual and procedural documentation about the management feature, refer to Chapter 1, "Network Service (Overview)," in *System Administration Guide: Network Services*.

When configuring resources and resource groups to use Solaris management functionality in a cluster, use the following high-level process:

1. Configuring applications as part of the resource.
2. Configuring resources as part of a resource group.
3. Enabling resources in the resource group.
4. Making the resource group managed.
5. Creating a Solaris project for your resource group.
6. Configuring standard properties to associate the resource group name with the project you created in step 5.
7. Bringing the resource group online.

To configure the standard `Resource_project_name` or `RG_project_name` properties to associate the Solaris project ID with the resource or resource group, use the `-p` option with the `clresource set` and the `clresourcegroup set` command. Set the property values to the resource or to the resource group. See Appendix B, "Standard Properties," in *Sun Cluster Data Services Planning and Administration Guide for Solaris OS* for property definitions. See the `r_properties`(5) and `rg_properties`(5) man pages for descriptions of properties.

The specified project name must exist in the projects database (`/etc/project`) and the root user must be configured as a member of the named project. Refer to Chapter 2, "Projects and Tasks (Overview)," in *System Administration Guide: Solaris Containers-Resource Management and Solaris Zones* for conceptual information about the project name database. Refer to `project`(4) for a description of project file syntax.

When the RGM brings resources or resource groups online, it launches the related processes under the project name.

Note – Users can associate the resource or resource group with a project at any time. However, the new project name is not effective until the resource or resource group is taken offline and brought back online by using the RGM.

Launching resources and resource groups under the project name enables you to configure the following features to manage system supplies across your cluster.

- Extended Accounting – Provides a flexible way to record consumption on a task or process basis. Extended accounting enables you to examine historical usage and make assessments of capacity requirements for future workloads.
- Controls – Provide a mechanism for constraint on system supplies. Processes, tasks, and projects can be prevented from consuming large amounts of specified system supplies.
- Fair Share Scheduling (FSS) – Provides the ability to control the allocation of available CPU time among workloads, based on their importance. Workload importance is expressed by the number of shares of CPU time that you assign to each workload. Refer to the following man pages for more information.
  - `dispadmin`(1M)
  - `priocntl`(1)
  - `ps`(1)
  - `FSS`(7)
- Pools – Provide the ability to use partitions for interactive applications according to the application's requirements. Pools can be used to partition a server that supports a number of different software applications. The use of pools results in a more predictable response for each application.

Determining Requirements for Project Configuration

Before you configure data services to use the controls provided by Solaris in a Sun Cluster environment, you must decide how to control and track resources across switchovers or failovers. Identify dependencies within your cluster before configuring a new project. For example, resources and resource groups depend on device groups.

Use the `nodelist`, `failback`, `maximum_primaries` and `desired_primaries` resource group properties that you configure with the `clresourcegroup set` command to identify node list priorities for your resource group.

- For a brief discussion of the node list dependencies between resource groups and device groups, refer to "Relationship Between Resource Groups and Device Groups" in *Sun Cluster Data Services Planning and Administration Guide for Solaris OS*.
- For detailed property descriptions, refer to `rg_properties`(5).

Use the `preferenced` property and `failback` property that you configure with the `cldevicegroup` and `clsetup` commands to determine device group node list priorities. See the `clresourcegroup`(1CL), `cldevicegroup`(1CL), and `clsetup`(1CL) man pages.

- For conceptual information about the `preferenced` property, see "Multiported Device Groups" on page 38.
- For procedural information, see "How To Change Disk Device Properties" in "Administering Device Groups" in *Sun Cluster System Administration Guide for Solaris OS*.
- For conceptual information about node configuration and the behavior of failover and scalable data services, see "Sun Cluster System Hardware and Software Components" on page 17.

If you configure all cluster nodes or zones identically, usage limits are enforced identically on primary and secondary nodes or zones. The configuration parameters of projects do not need to be identical for all applications in the configuration files on all nodes. All projects that are associated with the application must at least be accessible by the project database on all potential masters of that application. Suppose that Application 1 is mastered by *phys-schost-1* or a zone on *phys-schost-1* but could potentially be switched over or failed over to *phys-schost-2* or *phys-schost-3* or a zone on either of these nodes. The project that is associated with Application 1 must be accessible on all three nodes (*phys-schost-1*, *phys-schost-2*, and *phys-schost-3*) or zones on these nodes.

Note – Project database information can be a local `/etc/project` database file or can be stored in the NIS map or the LDAP directory service.

The Solaris Operating System enables for flexible configuration of usage parameters, and few restrictions are imposed by Sun Cluster. Configuration choices depend on the needs of the site. Consider the general guidelines in the following sections before configuring your systems.

Setting Per-Process Virtual Memory Limits

Set the `process.max-address-space` control to limit virtual memory on a per-process basis. See the `rctladm`(1M) man page for information about setting the `process.max-address-space` value.

When you use management controls with Sun Cluster software, configure memory limits appropriately to prevent unnecessary failover of applications and a "ping-pong" effect of applications. In general, observe the following guidelines.

- Do not set memory limits too low.

  When an application reaches its memory limit, it might fail over. This guideline is especially important for database applications, when reaching a virtual memory limit can have unexpected consequences.

- Do not set memory limits identically on primary and secondary nodes or zones.

  Identical limits can cause a ping-pong effect when an application reaches its memory limit and fails over to a secondary node or zone with an identical memory limit. Set the memory limit slightly higher on the secondary node or zone. The difference in memory limits helps prevent the ping-pong scenario and gives the system administrator a period of time in which to adjust the parameters as necessary.
- Do use the resource management memory limits for load balancing.

  For example, you can use memory limits to prevent an errant application from consuming excessive swap space.

Failover Scenarios

You can configure management parameters so that the allocation in the project configuration (`/etc/project`) works in normal cluster operation and in switchover or failover situations.

The following sections are example scenarios.

- The first two sections, "Two-Node Cluster With Two Applications" on page 79 and "Two-Node Cluster With Three Applications" on page 80, show failover scenarios for entire nodes.
- The section "Failover of Resource Group Only" on page 82 illustrates failover operation for an application only.

In a Sun Cluster environment, you configure an application as part of a resource. You then configure a resource as part of a resource group (RG). When a failure occurs, the resource group, along with its associated applications, fails over to another node or zone. In the following examples the resources are not shown explicitly. Assume that each resource has only one application.

Note – Failover occurs in the order in which nodes or zones are specified in the node list and set in the RGM.

The following examples have these constraints:

- Application 1 (App-1) is configured in resource group RG-1.
- Application 2 (App-2) is configured in resource group RG-2.
- Application 3 (App-3) is configured in resource group RG-3.

Although the numbers of assigned shares remain the same, the percentage of CPU time that is allocated to each application changes after failover. This percentage depends on the number of applications that are running on the node or in the zone and the number of shares that are assigned to each active application.

In these scenarios, assume the following configurations.

- All applications are configured under a common project.
- Each resource has only one application.

- The applications are the only active processes on the nodes or in the zones.
- The projects databases are configured the same on each node of the cluster or in each zone.

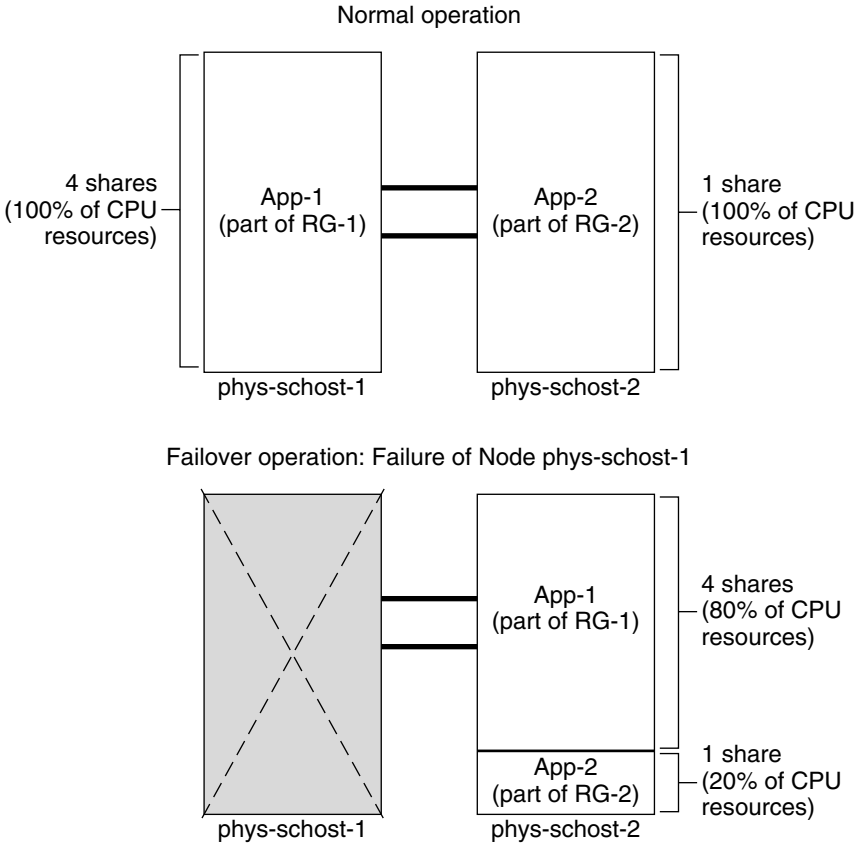
Two-Node Cluster With Two Applications

You can configure two applications on a two-node cluster to ensure that each physical host (*phys-schost-1*, *phys-schost-2*) acts as the default master for one application. Each physical host acts as the secondary node for the other physical host. All projects that are associated with Application 1 and Application 2 must be represented in the projects database files on both nodes. When the cluster is running normally, each application is running on its default master, where it is allocated all CPU time by the management facility.

After a failover or switchover occurs, both applications run on a single node where they are allocated shares as specified in the configuration file. For example, this entry in the `/etc/project` file specifies that Application 1 is allocated 4 shares and Application 2 is allocated 1 share.

```
Prj_1:100:project for App-1:root::project.cpu-shares=(privileged,4,none)
Prj_2:101:project for App-2:root::project.cpu-shares=(privileged,1,none)
```

The following diagram illustrates the normal and failover operations of this configuration. The number of shares that are assigned does not change. However, the percentage of CPU time available to each application can change. The percentage depends on the number of shares that are assigned to each process that demands CPU time.



Two-Node Cluster With Three Applications

On a two-node cluster with three applications, you can configure one physical host (*phys-schost-1*) as the default master of one application. You can configure the second physical host (*phys-schost-2*) as the default master for the remaining two applications. Assume the following example projects database file on every node. The projects database file does not change when a failover or switchover occurs.

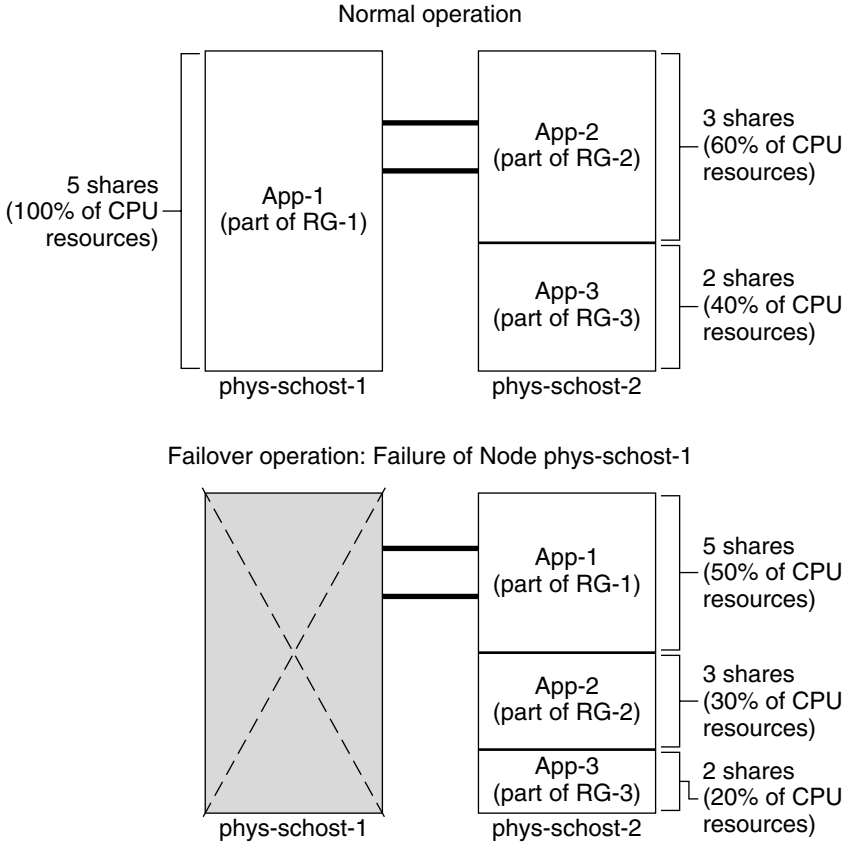
```
Prj_1:103:project for App-1:root::project.cpu-shares=(privileged,5,none)
Prj_2:104:project for App_2:root::project.cpu-shares=(privileged,3,none)
Prj_3:105:project for App_3:root::project.cpu-shares=(privileged,2,none)
```

When the cluster is running normally, Application 1 is allocated 5 shares on its default master, *phys-schost-1*. This number is equivalent to 100 percent of CPU time because it is the only application that demands CPU time on that node. Applications 2 and 3 are allocated 3 and 2 shares, respectively, on their default master, *phys-schost-2*. Application 2 would receive 60 percent of CPU time and Application 3 would receive 40 percent of CPU time during normal operation.

If a failover or switchover occurs and Application 1 is switched over to *phys-schost-2*, the shares for all three applications remain the same. However, the percentages of CPU resources are reallocated according to the projects database file.

- Application 1, with 5 shares, receives 50 percent of CPU.
- Application 2, with 3 shares, receives 30 percent of CPU.
- Application 3, with 2 shares, receives 20 percent of CPU.

The following diagram illustrates the normal operations and failover operations of this configuration.



Failover of Resource Group Only

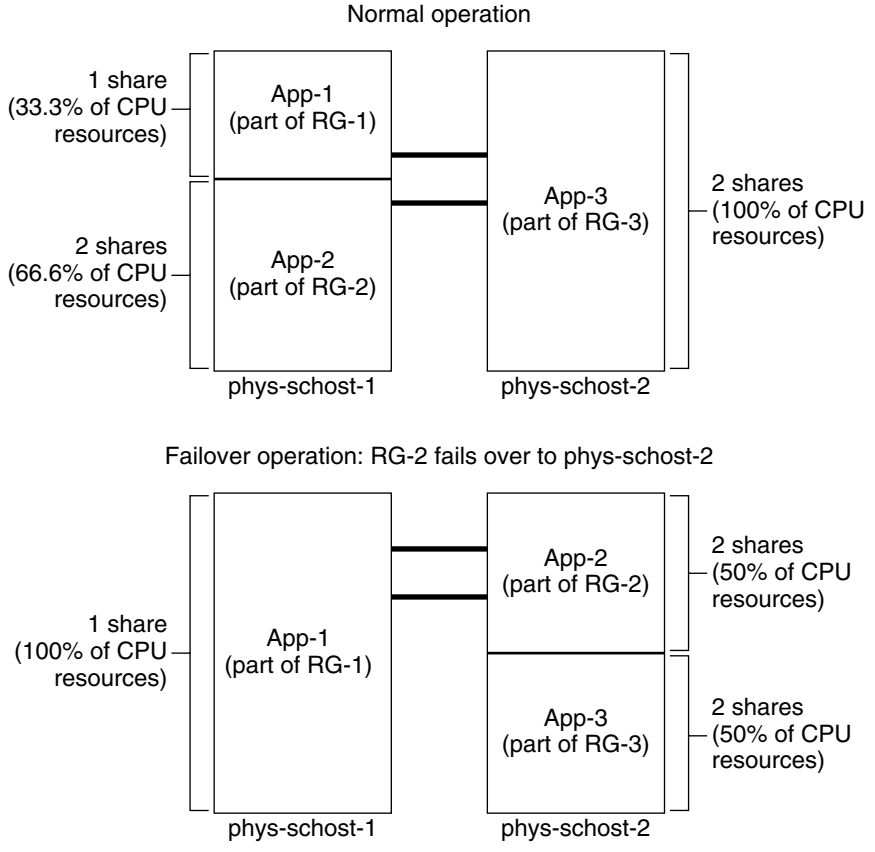
In a configuration in which multiple resource groups have the same default master, a resource group (and its associated applications) can fail over or be switched over to a secondary node or zone. Meanwhile, the default master is running in the cluster.

Note – During failover, the application that fails over is allocated resources as specified in the configuration file on the secondary node or zone. In this example, the project database files on the primary and secondary nodes have the same configurations.

For example, this sample configuration file specifies that Application 1 is allocated 1 share, Application 2 is allocated 2 shares, and Application 3 is allocated 2 shares.

```
Prj_1:106:project for App_1:root::project.cpu-shares=(privileged,1,none)
Prj_2:107:project for App_2:root::project.cpu-shares=(privileged,2,none)
Prj_3:108:project for App_3:root::project.cpu-shares=(privileged,2,none)
```

The following diagram illustrates the normal and failover operations of this configuration, where RG-2, containing Application 2, fails over to *phys-schost-2*. Note that the number of shares assigned does not change. However, the percentage of CPU time available to each application can change, depending on the number of shares that are assigned to each application that demands CPU time.



Public Network Adapters and Internet Protocol (IP) Network Multipathing

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

Solaris Internet Protocol (IP) Network Multipathing software on Sun Cluster 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 Internet Protocol (IP) Network Multipathing configuration, which can be different from the configuration on other cluster nodes.

Public network adapters are organized into *IP multipathing groups* (multipathing groups). Each multipathing group has one or more public network adapters. Each adapter in a multipathing group can be active. Alternatively, you can configure standby interfaces that are inactive unless a failover occurs.

The `in.mpathd` multipathing daemon uses a test IP address to detect failures and repairs. If a fault is detected on one of the adapters by the multipathing daemon, a failover occurs. All network access fails over from the faulted adapter to another functional adapter in the multipathing group. Therefore, the daemon maintains public network connectivity for the node. If you configured a standby interface, the daemon chooses the standby interface. Otherwise, the daemon chooses the interface with the least number of IP addresses. Because the failover occurs at the adapter interface level, higher-level connections such as TCP are not affected, except for a brief transient delay during the failover. When the failover of IP addresses completes successfully, ARP broadcasts are sent. Therefore, the daemon maintains connectivity to remote clients.

Note – Because of the congestion recovery characteristics of TCP, TCP endpoints can experience further delay after a successful failover. Some segments might have been lost during the failover, activating the congestion control mechanism in TCP.

Multipathing groups provide the building blocks for logical host name and shared address resources. You can also create multipathing groups independently of logical host name and shared address resources to monitor public network connectivity of cluster nodes. The same multipathing group on a node can host any number of logical host name or shared address resources. For more information about logical host name and shared address resources, see the *Sun Cluster Data Services Planning and Administration Guide for Solaris OS*.

Note – The design of the Internet Protocol (IP) Network Multipathing mechanism is meant to detect and mask adapter failures. The design is not intended to recover from an administrator's use of `ifconfig` to remove one of the logical (or shared) IP addresses. The Sun Cluster software views the logical and shared IP addresses as resources that are managed by the RGM. The correct way for an administrator to add or to remove an IP address is to use `clresource` and `clresourcegroup` to modify the resource group that contains the resource.

For more information about the Solaris implementation of IP Network Multipathing, see the appropriate documentation for the Solaris Operating System that is installed on your cluster.

Operating System	Instructions
Solaris 9 Operating System	Chapter 1, "IP Network Multipathing (Overview)," in *IP Network Multipathing Administration Guide*
Solaris 10 Operating System	Part VI, "IPMP," in *System Administration Guide: IP Services*

SPARC: Dynamic Reconfiguration Support

Sun Cluster 3.2 support for the dynamic reconfiguration (DR) software feature is being developed in incremental phases. This section describes concepts and considerations for Sun Cluster 3.2 support of the DR feature.

All the requirements, procedures, and restrictions that are documented for the Solaris DR feature also apply to Sun Cluster DR support (except for the operating environment quiescence operation). Therefore, review the documentation for the Solaris DR feature *before* by using the DR feature with Sun Cluster software. You should review in particular the issues that affect nonnetwork 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 10 on Sun Hardware* collection) are both available for download from `http://docs.sun.com`.

SPARC: Dynamic Reconfiguration General Description

The DR feature enables 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 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 that contains active components would result in system errors. Before removing a board, the DR subsystem queries other subsystems, such as Sun Cluster, to determine whether the components on the board are being used. If the DR subsystem finds that a board is in use, the DR remove-board operation is not done. Therefore, it is always safe to issue a DR remove-board operation because the DR subsystem rejects operations on boards that contain active components.

The DR add-board operation is also always safe. 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 to actively use 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 reports `Unknown error`. You can safely ignore this error.

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

SPARC: DR Clustering Considerations for CPU Devices

Sun Cluster software does not reject a DR remove-board operation because of the presence of CPU devices.

When a DR add-board operation succeeds, CPU devices on the added board are automatically incorporated in system operation.

SPARC: DR Clustering Considerations for Memory

For the purposes of DR, consider two types of memory:

- Kernel memory cage
- Non-kernel memory cage

These two types differ only in usage. The actual hardware is the same for both types. Kernel memory cage is the memory that is used by the Solaris Operating System. Sun Cluster software does not support remove-board operations on a board that contains the kernel memory cage and rejects any such operation. When a DR remove-board operation pertains to memory other than the kernel memory cage, Sun Cluster software does not reject the operation. When a DR add-board operation that pertains to memory succeeds, memory on the added board is automatically incorporated in system operation.

SPARC: DR Clustering Considerations for Disk and Tape Drives

Sun Cluster rejects DR remove-board operations on active drives in the primary node. DR remove-board operations can be performed on inactive drives in the primary node and on any drives in the secondary node. After the DR operation, cluster data access continues as before.

Note – Sun Cluster rejects DR operations that impact the availability of quorum devices. For considerations about quorum devices and the procedure for performing DR operations on them, see "SPARC: DR Clustering Considerations for Quorum Devices" on page 86.

See "Dynamic Reconfiguration With Quorum Devices" in *Sun Cluster System Administration Guide for Solaris OS* for detailed instructions about how to perform these actions.

SPARC: DR Clustering Considerations for Quorum Devices

If the DR remove-board operation pertains to a board that contains an interface to a device configured for quorum, Sun Cluster software rejects the operation. Sun Cluster software also 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-board operation.

See Chapter 6, "Administering Quorum," in *Sun Cluster System Administration Guide for Solaris OS* for detailed instructions about how administer quorum.

SPARC: DR Clustering Considerations for Cluster Interconnect Interfaces

If the DR remove-board operation pertains to a board containing an active cluster interconnect interface, Sun Cluster software rejects the operation. Sun Cluster software also identifies the interface that would be affected by the operation. You must use a Sun Cluster administrative tool to disable the active interface before the DR operation can succeed.

Caution – Sun Cluster software requires each cluster node to have 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.

See "Administering the Cluster Interconnects" in *Sun Cluster System Administration Guide for Solaris OS* for detailed instructions about how to perform these actions.

SPARC: DR Clustering Considerations for Public Network Interfaces

If the DR remove-board operation pertains to a board that contains an active public network interface, Sun Cluster software rejects the operation. Sun Cluster software also identifies the interface that would be affected by the operation. Before you remove a board with an active network interface present, switch over all traffic on that interface to another functional interface in the multipathing group by using the `if_mpadm` command.

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

See "Administering the Public Network" in *Sun Cluster System Administration Guide for Solaris OS* for detailed instructions about how to perform a DR remove operation on a public network interface.

◆ ◆ ◆ CHAPTER 4

Frequently Asked Questions

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

- "High Availability FAQs" on page 89
- "File Systems FAQs" on page 90
- "Volume Management FAQs" on page 91
- "Data Services FAQs" on page 91
- "Public Network FAQs" on page 92
- "Cluster Member FAQs" on page 93
- "Cluster Storage FAQs" on page 94
- "Cluster Interconnect FAQs" on page 94
- "Client Systems FAQs" on page 95
- "Administrative Console FAQs" on page 95
- "Terminal Concentrator and System Service Processor FAQs" on page 96

High Availability FAQs

Question: What exactly is a highly available system?

Answer: The Sun Cluster software defines high availability (HA) as the ability of a cluster to keep an application running. The application runs even when a failure occurs that would normally make a server system unavailable.

Question: What is the process by which the cluster provides high availability?

Answer: Through a process known as failover, the cluster framework provides a highly available environment. Failover is a series of steps that are performed by the cluster to migrate data service resources from a failing node or zone to another operational node or zone in the cluster.

Question: What is the difference between a failover and scalable data service?

Answer: There are two types of highly available data services:

- Failover

- Scalable

A failover data service runs an application on only one primary node or zone in the cluster at a time. Other nodes or zones might run other applications, but each application runs on only a single node or zone. If a primary node or zone fails, applications that are running on the failed node or zone fail over to another node or zone. They continue running.

A scalable data service spreads an application across multiple nodes or zones to create a single, logical service. A scalable data service that uses a shared address to balance the service load between nodes can be online in only one zone per physical node. Scalable services leverage the number of nodes or zones 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 (GIF) node. Multiple GIF nodes can exist in the cluster. Each GIF node hosts one or more logical interfaces that can be used by scalable services. These logical interfaces are called *global interfaces*. One GIF node 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 GIF node fails, the global interface fails over to a surviving node.

If any node or zone on which the application is running fails, the application continues to run on other nodes or zones with some performance degradation. This process continues until the failed node or zone returns to the cluster.

File Systems FAQs

Question: Can I run one or more of the cluster nodes as highly available NFS servers with other cluster nodes as clients?

Answer: No, do not do a loopback mount.

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

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

Question: Must all cluster file systems have a mount point under the `/global` directory?

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

Question: What are the differences between using the cluster file system and exporting NFS file systems?

Answer: Several differences exist:

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, the cluster file system caches files when a file is being accessed from multiple nodes for read, write, file locks, asynchronous I/O.
4. The cluster file system is built to exploit future fast cluster interconnects that provide remote DMA and zero-copy functions.
5. If you change the attributes on a file (using `chmod`, for example) in a cluster file system, the change is reflected immediately on all nodes. With an exported NFS file system, this change can take much longer.

Question: The file system `/global/.devices/node@`*nodeID* appears on my cluster nodes. Can I use this file system to store data that I want to be highly available and global?

Answer: These file systems store the global device namespace. These file systems are not intended for general use. While they are global, these file systems 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. These file systems should not be used to store data that needs to be globally accessible or highly available.

Volume Management FAQs

Question: Do I need to mirror all disk devices?

Answer: 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.

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

Answer: This configuration is supported with the Solaris Volume Manager software managing the local disks and VERITAS Volume Manager managing the multihost disks. No other combination is supported.

Data Services FAQs

Question: Which Sun Cluster data services are available?

Answer: The list of supported data services is included in "Supported Products" in *Sun Cluster 3.2 Release Notes for Solaris OS*.

Question: Which application versions are supported by Sun Cluster data services?

Answer: The list of supported application versions is included in "Supported Products" in *Sun Cluster 3.2 Release Notes for Solaris OS*.

Question: Can I write my own data service?

Answer: Yes. See the Chapter 11, "DSDL API Functions," in *Sun Cluster Data Services Developer's Guide for Solaris OS* for more information.

Question: When creating network resources, should I specify numeric IP addresses or host names?

Answer: The preferred method for specifying network resources is to use the UNIX host name rather than the numeric IP address.

Question: When creating network resources, what is the difference between using a logical host name (a `LogicalHostname` resource) or a shared address (a `SharedAddress` resource)?

Answer: Except in the case of Sun Cluster HA for NFS, wherever the documentation recommends the use of a `LogicalHostname` resource in a `Failover` mode resource group, a `SharedAddress` resource or `LogicalHostname` resource can 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` resource is demonstrated when you configure both scalable and failover data services, and want clients to be able to access both services by using the same host name. In this case, the `SharedAddress` resources along with the failover application resource are contained in one resource group. The scalable service resource is contained in a separate resource group and configured to use the `SharedAddress` resource. Both the scalable and failover services can then use the same set of host names and addresses that are configured in the `SharedAddress` resource.

Public Network FAQs

Question: Which public network adapters does the Sun Cluster software support?

Answer: Currently, the Sun Cluster software 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.

Question: What is the role of the MAC address in failover?

Answer: 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.

Question: Does the Sun Cluster software support setting `local-mac-address?=true`*?*

Answer: Yes. In fact, IP Network Multipathing requires that `local-mac-address?` *must* be set to `true`.

You can set `local-mac-address` with the `eeprom` command, at the OpenBoot PROM `ok` prompt in a SPARC based cluster. See the `eeprom`(1M) man page. You can also set the MAC address with the SCSI utility that you optionally run after the BIOS boots in an x86 based cluster.

Question: How much delay can I expect when Internet Protocol (IP) Network Multipathing performs a switchover between adapters?

Answer: The delay could be several minutes. The reason is because when an Internet Protocol (IP) Network Multipathing switchover is performed, the operation sends a gratuitous ARP broadcast.

However, you cannot be sure that the router between the client and the cluster uses the gratuitous ARP. So, until the ARP cache entry for this IP address on the router times out, the entry can use the stale MAC address.

Question: How fast are failures of a network adapter detected?

Answer: The default failure detection time is 10 seconds. The algorithm tries to meet the failure detection time, but the actual time depends on the network load.

Cluster Member FAQs

Question: Do all cluster members need to have the same root password?

Answer: 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 or zones.

Question: Is the order in which nodes are booted significant?

Answer: In most cases, no. However, the boot order is important to prevent 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 order prevents you from accidentally bringing up a node with outdated cluster configuration information. Refer to "About Failure Fencing" on page 48 for details about amnesia.

Question: Do I need to mirror local disks in a cluster node?

Answer: Yes. Though this mirroring is not a requirement, mirroring the cluster node's disks prevents a nonmirrored disk failure from taking down the node. The downside to mirroring a cluster node's local disks is more system administration overhead.

Question: What are the cluster member backup issues?

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

See Chapter 11, "Backing Up and Restoring a Cluster," in *Sun Cluster System Administration Guide for Solaris OS* for additional information about how to backup and restore data.

Question: When is a node healthy enough to be used as a secondary node?

Answer: Solaris 9 OS:

After a reboot, a node is healthy enough to be a secondary node when the node displays the login prompt.

Solaris 10 OS:

A node or zone is healthy enough to be a secondary node or zone if the `multi-user-server` milestone is running.

```
# svcs -a | grep multi-user-server:default
```

Cluster Storage FAQs

Question: What makes multihost storage highly available?

Answer: Multihost storage is highly available because it can survive the loss of a single disk, because of mirroring (or because of 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. In addition, redundant paths from each node to the attached storage provide tolerance for the failure of a host bus adapter, cable, or disk controller.

Cluster Interconnect FAQs

Question: Which cluster interconnects does the Sun Cluster software support?

Answer: Currently, the Sun Cluster software supports the following cluster interconnects:

- Ethernet (100BASE-T Fast Ethernet and 1000BASE-SX Gb) in both SPARC based and x86 based clusters
- Infiniband in both SPARC based and x86 based clusters
- SCI in SPARC based clusters only

Question: What is the difference between a "cable" and a transport "path"?

Answer: Cluster transport cables are configured by 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. These cables are not probed and therefore their state is unknown. You can obtain the state of a cable by using the `cluster status` command.

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." You can obtain the status of a transport path by using the `clinterconnect status` command. See the `clinterconnect`(1CL) man page.

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
```

Two possible transport paths can be formed from these four cables.

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

Client Systems FAQs

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

Answer: Client systems connect to the cluster as they would to 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 Chapter 1, "Planning for Sun Cluster Data Services," in *Sun Cluster Data Services Planning and Administration Guide for Solaris OS* for more information about client-side configuration requirements.

Administrative Console FAQs

Question: Does the Sun Cluster software require an administrative console?

Answer: Yes.

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

Answer: The Sun Cluster software 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

Question: Does the administrative console need to be located "close" to the cluster, for example, in the same room?

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

Question: Can an administrative console serve more than one cluster, if any distance requirements are also first met?

Answer: 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 FAQs

Question: Does the Sun Cluster software require a terminal concentrator?

Answer: Starting with Sun Cluster 3.0, Sun Cluster software does not require a terminal concentrator. Unlike Sun Cluster 2.2, Sun Cluster 3.0, Sun Cluster 3.1, and Sun Cluster 3.2 do not require a terminal concentrator. Sun Cluster 2.2 required a terminal concentrator for failure fencing.

Question: I see that most Sun Cluster servers use a terminal concentrator, but the Sun Enterprise E1000 server does not. Why not?

Answer: The terminal concentrator is effectively a serial-to-Ethernet converter for most servers. The terminal concentrator's console port is a serial port. The Sun Enterprise E1000 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 E1000 server, you always use the SSP for consoles.

Question: What are the benefits of using a terminal concentrator?

Answer: Using a terminal concentrator provides console-level access to each node from a remote workstation anywhere on the network. This access is provided even when the node is at the OpenBoot PROM (OBP) on a SPARC based node or a boot subsystem on an x86 based node.

Question: If I use a terminal concentrator that Sun does not support, what do I need to know to qualify the one that I want to use?

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

Question: Can I free a locked port on the terminal concentrator that Sun supports without rebooting it?

Answer: Yes. Note the port number that needs to be reset and type the following commands:

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

Refer to the following manuals for more information about how to configure and administer the terminal concentrator that Sun supports.

- "Overview of Administering Sun Cluster" in *Sun Cluster System Administration Guide for Solaris OS*

- Chapter 2, "Installing and Configuring the Terminal Concentrator," in *Sun Cluster 3.1 - 3.2 Hardware Administration Manual for Solaris OS*

Question: What if the terminal concentrator itself fails? Must I have another one standing by?

Answer: 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.

Question: If I do use a terminal concentrator, what about security?

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

Question: SPARC: How do I use dynamic reconfiguration with a tape or disk drive?

Answer: Perform the following steps:

- 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 "SPARC: DR Clustering Considerations for Disk and Tape Drives" on page 86.
- 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.

Index

A

adapters, *See* network, adapters
administration, cluster, 31-87
administrative console, 24
  FAQs, 95
administrative interfaces, 32
agents, *See* data services
amnesia, 46
APIs, 63-64, 66
application, *See* data services
application communication, 64-65
application development, 31-87
application distribution, 51
attributes, *See* properties

B

backup node, 93
board removal, dynamic reconfiguration, 86
boot disk, *See* disks, local
boot order, 93

C

cable, transport, 94
CCP, 24
CCR, 35
CD-ROM drive, 21-22
client-server configuration, 56
client systems, 23
  FAQs, 95
  restrictions, 95
`clprivnet` driver, 65
cluster
  administration, 31-87
  advantages, 11-12
  application developer view, 14-15
  application development, 31-87
  backup, 93
  board removal, 86
  boot order, 93
  configuration, 35, 74-83
  data services, 56-62
  description, 11-12
  file system, 41-43, 90-91
    FAQs
  See also file system
    `HAStoragePlus` resource type, 42-43
    using, 42
  goals, 11-12
  hardware, 12-13, 17-24
  interconnect, 18, 22
    adapters, 22
    cables, 22
    data services, 64-65
    dynamic reconfiguration, 87
    FAQs, 94
    interfaces, 22
    junctions, 22
    supported, 94
  media, 21-22
  members, 18, 34
    FAQs, 93
    reconfiguration, 34
  nodes, 18
  password, 93

cluster *(Continued)*
public network, 22-23
public network interface, 56
service, 12-13
software components, 19-20
storage FAQs, 94
system administrator view, 13-14
task list, 15-16
time, 32
topologies, 24-29, 29-30
Cluster Configuration Repository, 35
Cluster Control Panel, 24
Cluster Membership Monitor, 34
clustered pair topology, 25-26, 29-30
clustered-server model, 56
CMM, 34
failfast mechanism, 34
See also failfast
concurrent access, 18
configuration
client-server, 56
data services, 74-83
parallel database, 18
repository, 35
virtual memory limits, 77-78
configurations, quorum, 50
console
access, 23-24
administrative, 23-24, 24
FAQs, 95
System Service Processor, 23-24
Controlling CPU, 73
CPU, control, 73
CPU time, 74-83

D

daemons, `svc.startd`, 72
data, storing, 90-91
data services, 56-62
APIs, 63-64
cluster interconnect, 64-65
configuration, 74-83
developing, 63-64
failover, 58-59
data services *(Continued)*
FAQs, 91-92
fault monitor, 62
highly available, 33
library API, 64
methods, 58
resource groups, 65-68
resource types, 65-68
resources, 65-68
scalable, 59-60
supported, 91-92
`/dev/global/` namespace, 40-41
developer, cluster applications, 14-15
device
global, 35-36
ID, 36
device group, 36-39
changing properties, 38-39
device groups
failover, 37-38
multiported, 38-39
primary ownership, 38-39
devices
multihost, 20
quorum, 46-55
DID, 36
disk path monitoring, 43-46
disks
dynamic reconfiguration, 86
failure fencing, 48
global devices, 35-36, 40-41
local, 21, 35-36, 40-41
mirroring, 93
volume management, 91
multihost, 35-36, 36-39, 40-41
SCSI devices, 20-21
DR, *See* dynamic reconfiguration
driver, device ID, 36
DSDL API, 66
dynamic reconfiguration, 85-87
cluster interconnect, 87
CPU devices, 85-86
description, 85
disks, 86
memory, 86
public network, 87

dynamic reconfiguration *(Continued)*
quorum devices, 86
tape drives, 86

E

E10000, *See* Sun Enterprise E10000

F

failback, 62
failfast, 34-35, 49
failover
data services, 58-59
device groups, 37-38
scenarios, Solaris Resource Manager, 78-83
failure
detection, 33
failback, 62
fencing, 34, 48
recovery, 33
FAQs, 89-97
administrative console, 95
client systems, 95
cluster interconnect, 94
cluster members, 93
cluster storage, 94
data services, 91-92
file systems, 90-91
high availability, 89-90
public network, 92-93
System Service Processor, 96-97
terminal concentrator, 96-97
volume management, 91
fault monitor, 62
fencing, 34, 48
file locking, 41
file system
cluster, 41-43, 90-91
data storage, 90-91
FAQs, 90-91
global
See file system, cluster
high availability, 90-91
file system *(Continued)*
local, 42-43
mounting, 41-43, 90-91
NFS, 43, 90-91
`syncdir` mount option, 43
UFS, 43
VxFS, 43
file systems, using, 42
framework, high availability, 33-35
Frequently Asked Questions, *See* FAQs

G

global
device, 35-36, 36-39
local disks, 21
mounting, 41-43
interface, 57
scalable services, 59
namespace, 36, 40-41
local disks, 21
global file system, *See* cluster, file system
global interface node, 57
`/global` mount point, 41-43, 90-91
groups, device, 36-39

H

HA, *See* high availability
hardware, 12-13, 17-24, 85-87
See also disks
See also storage
cluster interconnect components, 22
dynamic reconfiguration, 85-87
`HAStoragePlus` resource type, 42-43, 65-68
high availability
FAQs, 89-90
framework, 33-35
highly available, data services, 33
host name, 56

I

ID
  device, 36
  node, 40
`in.mpathd` daemon, 84
interfaces
  See network, interfaces
  administrative, 32
ioctl, 49
IP address, 91-92
IP Network Multipathing, 83-84
  failover time, 92-93
IPMP, *See* IP Network Multipathing

K

kernel, memory, 86

L

load balancing, 60-62
local disks, 21
local file system, 42-43
`local_mac_address`, 92-93
local namespace, 40-41
logical host name, 56
  compared to shared address, 91-92
  failover data services, 58-59
`LogicalHostname` resource type, *See* logical host name

M

MAC address, 92-93
mapping, namespaces, 40-41
media, removable, 21-22
membership, *See* cluster, members
memory, 86
mission-critical applications, 54
monitoring
  disk path, 43-46
  object type, 72
  system resources, 73
  telemetry attributes, 73
mounting
  file systems, 41-43
  `/global`, 90-91
  global devices, 41-43
  with `syncdir`, 43
multi-initiator SCSI, 20-21
multihost device, 20
multipathing, 83-84
multiported device groups, 38-39

N

N+1 (star) topology, 27-28
N*N (scalable) topology, 28-29
namespaces, 40-41
network
  adapters, 22-23, 83-84
  interfaces, 22-23, 83-84
  load balancing, 60-62
  logical host name, 56
  private, 18
  public, 22-23
    dynamic reconfiguration, 87
    FAQs, 92-93
    interfaces, 92-93
    IP Network Multipathing, 83-84
  resources, 56, 65-68
  shared address, 56
Network Time Protocol, 32
NFS, 43
nodes, 18
  backup, 93
  boot order, 93
  global interface, 57
  nodeID, 40
  primary, 38-39, 57
  secondary, 38-39, 57
NTP, 32
`numsecondaries` property, 38

O

object type, system resource, 72
Oracle Parallel Server, *See* Oracle Real Application Clusters

Oracle Real Application Clusters, 63

P

pair+N topology, 26-27
panic, 34-35, 35, 49
parallel database configurations, 18
password, root, 93
path, transport, 94
per-node address, 64-65
Persistent Group Reservation, 49
PGR, *See* Persistent Group Reservation
`preferenced` property, 38
primary node, 57
primary ownership, device groups, 38-39
private network, 18
projects, 74-83
properties
  changing, 38-39
  resource groups, 68
  `Resource_project_name`, 76-77
  resources, 68
  `RG_project_name`, 76-77
proxy resource types, 72
public network, *See* network, public
pure service, 60

Q

quorum, 46-55
  atypical configurations, 54
  bad configurations, 54-55
  best practices, 50-51
  configurations, 49-50, 50
  device, dynamic reconfiguration, 86
  devices, 46-55
  recommended configurations, 51-53
  requirements, 50
  vote counts, 47-48

R

recovery
  failback settings, 62
  failure detection, 33
removable media, 21-22
reservation conflict, 49
Resource Group Manager, *See* RGM
resource groups, 65-68
  failover, 58-59
  properties, 68
  scalable, 59-60
  settings, 66-68
  states, 66-68
resource management, 74-83
`Resource_project_name` property, 76-77
resource types, 42-43, 65-68
  proxy, 72
  `SUNW.Proxy_SMF_failover`, 72
  `SUNW.Proxy_SMF_loadbalanced`, 72
  `SUNW.Proxy_SMF_multimaster`, 72
resources, 65-68
  properties, 68
  settings, 66-68
  states, 66-68
`RG_project_name` property, 76-77
RGM, 58, 65-68, 74-83
RMAPI, 66
root password, 93

S

scalable data services, 59-60
`scha_cluster_get` command, 65
`scha_privatelink_hostname_node` argument, 65
SCSI
  failure fencing, 48
  multi-initiator, 20-21
  Persistent Group Reservation, 49
  reservation conflict, 49
`scsi-initiator-id` property, 21
secondary node, 57
server models, 56
Service Management Facility (SMF), 72
shared address, 56
  compared to logical host name, 91-92

shared address *(Continued)*
global interface node, 57
scalable data services, 59-60
`SharedAddress` resource type, *See* shared address
shutdown, 34-35
single-server model, 56
SMF, *See* Service Management Facility (SMF)
SMF daemon `svc.startd`, 72
software components, 19-20
Solaris projects, 74-83
Solaris Resource Manager, 74-83
configuration requirements, 76-77
configuring virtual memory limits, 77-78
failover scenarios, 78-83
Solaris Volume Manager, multihost devices, 20
split brain, 46, 48
SSP, *See* System Service Processor
sticky service, 60
storage, 20
dynamic reconfiguration, 86
FAQs, 94
SCSI, 20-21
Sun Cluster, *See* cluster
Sun Cluster Manager, 32
system resource usage, 74
Sun Enterprise E10000, 96-97
administrative console, 24
Sun Management Center (SunMC), 32
SunPlex Manager, *See* Sun Cluster Manager
`SUNW.Proxy_SMF_failover`, resource types, 72
`SUNW.Proxy_SMF_loadbalanced`, resource types, 72
`SUNW.Proxy_SMF_multimaster`, resource types, 72
`svc.startd`, daemons, 72
`syncdir` mount option, 43
system resource, threshold, 73
system resource monitoring, 73
system resource usage, 72
system resources
monitoring, 73
object type, 72
usage, 72
System Service Processor, 23-24, 24
FAQs, 96-97

T

tape drive, 21-22
telemetry attribute, system resources, 73
terminal concentrator, FAQs, 96-97
threshold
system resource, 73
telemetry attribute, 73
time, between nodes, 32
topologies, 24-29, 29-30
clustered pair, 25-26, 29-30
N+1 (star), 27-28
N*N (scalable), 28-29
pair+N, 26-27

U

UFS, 43

V

VERITAS Volume Manager, multihost devices, 20
volume management
FAQs, 91
local disks, 91
multihost devices, 20
multihost disks, 91
namespace, 40
RAID-5, 91
Solaris Volume Manager, 91
VERITAS Volume Manager, 91
vote counts, quorum, 47-48
VxFS, 43

Z

zones, 68