SGI® InfiniteStorage Cluster Manager for Linux® Administrator's Guide

007–3800–006

CONTRIBUTORS

Written by Lori Johnson

Engineering contributions by Derek Guy Barnes, Dale Brantly, Jeff Cech, Susheel Gokhale, Ron Kerry, Tim Kirby, Edward Mascarenhas, Anibal Monsalve Salazar, Daniel Moore, LaNet Merrill, Nate Pearlstein, Kevan Rehm, Paddy Sreenivasan, Olaf Weber, Geoffrey Wehrman, James Yarbrough

Illustrated by Chrystie Danzer

New Features in this Guide

This release includes the following:

- Support for the following:
  - Cluster with as many as 4 members
  - SGI ProPack 5 for Linux with SUSE Linux Enterprise Server (SLES) 10
  - DMF 3.5
  - IBM 3494 tape silo and TMF (see Chapter 11, "Tape Management Facility (TMF) Failover Script" on page 111)
- The install, upgrade, and uninstall procedures now use YaST. See Chapter 4, "Software Installation" on page 35.
- Requirements and recommendations for using SGI Cluster Manager are now located in the new Chapter 2, "Best Practices" on page 11. This has resulted in some reorganization of information formerly located elsewhere in the guide.
- "Troubleshooting Strategy " on page 125
- "Maintaining the `smb.conf.`*sharename* File on Shared Storage" on page 99

Record of Revision

Version	Description
001	May 2004 Original publication to support SGI Cluster Manager 3.0 for Linux
002	August 2004 Supports SGI Cluster Manager 3.1 for Linux
003	April 2005 Supports SGI Cluster Manager 4.0 for Linux
004	December 2005 Supports SGI Cluster Manager 4.1 for Linux
005	May 2006 Supports SGI Cluster Manager 4.2 for Linux
006	March 2007 Supports SGI Cluster Manager 4.3 for Linux

Contents

Figures

Tables

About This Guide

This guide provides information about SGI Cluster Manager for Linux, which provides highly available services for SGI Altix servers. Plug-ins provide high-availability services for CXFS clustered filesystems, local XVM logical volumes, the Data Migration Facility (DMF), and the Tape Management Facility (TMF).

Related Publications

The following publications contain additional information that may be helpful:

- SGI ProPack for Linux and SGI Altix documentation:
  - *NIS Administrator's Guide*
  - *Personal System Administration Guide*
  - *SGI ProPack for Linux Start Here*
  - *SUSE LINUX Enterprise Server for SGI Altix Systems*
  - *SGI Altix 330 System User's Guide*
  - *SGI Altix 350 System User's Guide*
  - *SGI Altix 3000 User's Guide*
  - *SGI Altix 3700 Bx2 User's Guide*
  - *SGI Altix 450 System User's Guide*
  - *SGI Altix 4700 System User's Guide*
  - *Performance Co-Pilot for IA-64 Linux User's and Administrator's Guide*
  - *SGI L1 and L2 Controller Software User's Guide*
  - *SGI Altix Systems Dual-Port Gigabit Ethernet Board User's Guide*
- *CXFS Administration Guide for SGI InfiniteStorage*
- *DMF Administrator's Guide for SGI InfiniteStorage*
- *TMF Administrator's Guide*

- *XVM Volume Manager Administrator's Guide*
- *TPM Installation Instructions and User's Guide for SGI TP9100*
- *SGI InfiniteStorage TP9300 and TP9300S RAID User's Guide*
- *SGI® InfiniteStorage TP9400 and SGI® InfiniteStorage TP9500 and TP9500S RAID User's Guide*
- *SGI InfiniteStorage TP9500 and TP9700 RAID User's Guide*
- *SGI TPSSM Administration Guide*

SGI Cluster Manager man pages:

- `clulockd`(8)
- `clumembd`(8)
- `cluquorumd`(8)
- `clurmtabd`(8)
- `clusvcmgrd`(8)
- `sgicm-config-cluster`(8)
- `sgicm-config-cluster-cmd`(8)

Other man pages:

- `kermit`(1)
- `minicom`(1)
- `exports`(5)

For more information about Samba, see:

http://www.samba.org/samba/docs

Obtaining Publications

You can obtain SGI documentation as follows:

- See the SGI Technical Publications Library at http://docs.sgi.com. Various formats are available. This library contains the most recent and most comprehensive set of online books, release notes, man pages, and other information.
- View release notes on your system by accessing the README file(s) for the product. This is usually located in the `/usr/share/doc/`*productname* directory, although file locations may vary.
- View man pages by typing `man` *title* at a command line.

Conventions

The following conventions are used throughout this document:

Convention	Meaning
`command`	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
variable	Italic typeface denotes variable entries and words or concepts being defined.
`user input`	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. (Output is shown in nonbold, fixed-space font.)
[ ]	Brackets enclose optional portions of a command or directive line.
...	Ellipses indicate that a preceding element can be repeated.

GUI	This font denotes the names of graphical user interface (GUI) elements such as windows, screens, dialog boxes, menus, toolbars, icons, buttons, boxes, fields, and lists.

Reader Comments

If you have comments about the technical accuracy, content, or organization of this publication, contact SGI. Be sure to include the title and document number of the publication with your comments. (Online, the document number is located in the front matter of the publication. In printed publications, the document number is located at the bottom of each page.)

You can contact SGI in any of the following ways:

- Send e-mail to the following address:

  techpubs@sgi.com

- Contact your customer service representative and ask that an incident be filed in the SGI incident tracking system.

- Send mail to the following address:

  SGI
  Technical Publications
  1140 East Arques Avenue
  Sunnyvale, CA 94085–4602

SGI values your comments and will respond to them promptly.

Introduction

The SGI Cluster Manager for Linux provides *highly available services* that survive a single point of failure. It uses redundant components and special software to provide services for a cluster that contains up to four machines or system partitions, known as *members*.

Each highly available service is owned by one member at a time. Highly available services are monitored by the SGI Cluster Manager software. If one member fails, another member restarts the highly available applications of the failed member, known as the *failover process*.

To application clients, the services on the backup member are indistinguishable from the original services before failure occurred. It appears as if the original member has crashed and rebooted quickly. Clients that use User Datagram Protocol (UDP) for communication with the server will notice a brief interruption in the highly available service. Clients that use Transmission Control Protocol (TCP) for communication may have to reconnect to the server in case of failure.

Although SGI Cluster Manager for Linux provides similar functionality to IRIX FailSafe, there are differences; see Appendix A, "FailSafe and SGI Cluster Manager" on page 139.

This chapter discusses the following:

Highly Available Services

A highly available service consists of the following:

- Disks (such as XVM volumes)
- IP address
- Filesystem (such as XFS or CXFS)
- NFS (if used)
- Samba (if used)
- User applications (if used)

SGI Cluster Manager Base Product

The SGI Cluster Manager base product provides failover support for the following:

- Filesystems (including XFS)
- NFS
- Samba
- IP addresses
- User-defined applications (that is, applications that are not provided by the SGI Cluster Manager product)

SGI Software Storage Plug-In Product

A *plug-in* is the set of software that allows a service to be highly available without modifying the application itself. SGI supplies plug-ins for the following:

- CXFS clustered filesystems
- Data Migration Facility (DMF)
- XVM volume manager in local mode

SGI also provides a failover script for the Tape Management Facility (TMF). You can modify your application to use this script to provide highly available services for TMF.

Hardware Requirements

SGI Cluster Manager requires a cluster of up to four members.

The following servers are supported:

- An SGI Altix 330 server with a USB-to-Ethernet adapter connected to the L1 system controller so that the brick emulates an L2 controller and becomes an L1/L2 controller. (Separate physical L2 controllers are not used with the Altix 330 systems.) Access to the L2 functionality is made by way of an Ethernet connection to a PC or laptop. An Altix 330 server must use the L2 Ethernet reset configuration (`l2network`) for remote resets.
- An SGI Altix 350 server with an IO10 PCI card, which may use either of the following for remote resets:
  - Network connection (`l2network`), which requires an additional PCI network interface card must be purchased. (Preferred method.)
  - Serial connection (`l2`), which requires the following:
    - *Multiport serial adapter cable* (a device that provides four DB9 serial ports from a 36-pin connector), must be purchased (part number `CBL-SATA-SERIAL`)
    - Hardware L2 system controller (which must be purchased)
- An SGI Altix 350 server with an IO9 PCI card, which must use the L2 Ethernet reset configuration (`l2network`) for remote resets. This requires a hardware L2 system controller that must be separately purchased.

Note: Customers cannot replace the IO9 PCI card in the Altix 350 with the IO10 PCI card. This procedure requires a new interface board and cables as well as a drive swap from SCSI to SATA. This procedure can only be done by SGI service personnel.

- An SGI Altix 450 with an Ethernet connection on the system control board of an IRU; an RJ45 connection is typically labeled "L2 host". Access to the L2 functionality is made by way of an Ethernet connection to a PC or laptop. An Altix 450 server must use the L2 Ethernet reset configuration (`l2network`) for remote resets.

- An SGI Altix 3700 server, which can use either the L2 Ethernet reset configuration (`l2network`) or the L2 serial reset configuration (`l2`). These servers may be partitioned; each system partition is an individual member.
- An SGI Altix 3700 Bx2 server with a USB-to-Ethernet adapter connected to the L1 system controller so that the brick emulates an L2 controller and becomes an L1/L2 controller. (Separate physical L2 controllers are not used with the Altix 3700 Bx2 systems.) Access to the L2 functionality is made by way of an Ethernet connection to a PC or laptop. An Altix 3700 Bx2 server must use the L2 Ethernet reset configuration (`l2network`). See "`l2network` Ethernet Connection" on page 25.
- An SGI Altix 4700 with an Ethernet connection on the system control board of an IRU or on a Dense router; an RJ45 connection is typically labeled "L2 host". Access to the L2 functionality is made by way of an Ethernet connection to a PC or laptop. An Altix 4700 server must use the L2 Ethernet reset configuration (`l2network`) for remote resets.

SGI Cluster Manager also requires the following:

- Shared quorum partitions without filesystems where configuration, cluster, and service status information is kept by SGI Cluster Manager. For more information, see "Use Shared Quorum Partitions" on page 15.
- Network cabling: you can connect private network or cross-over cables between members. You have a choice between an Ethernet cable from server to hub or a 20-ft cross-over Ethernet cable.

Note: To use a private network, you must have a second NIC whether you use a cross-over cable or a switch/hub.

Figure 1-1 shows an example configuration using CXFS. A private network is recommended for SGI Cluster Manager. The SGI Cluster Manager members should be able to communicate with the SGI Cluster Manager tiebreaker via the network. The tiebreaker can be a machine or a router or any device that can be connected via the network. (For more information about tiebreakers, see "Step 6: Set the Tiebreakers" on page 54.)

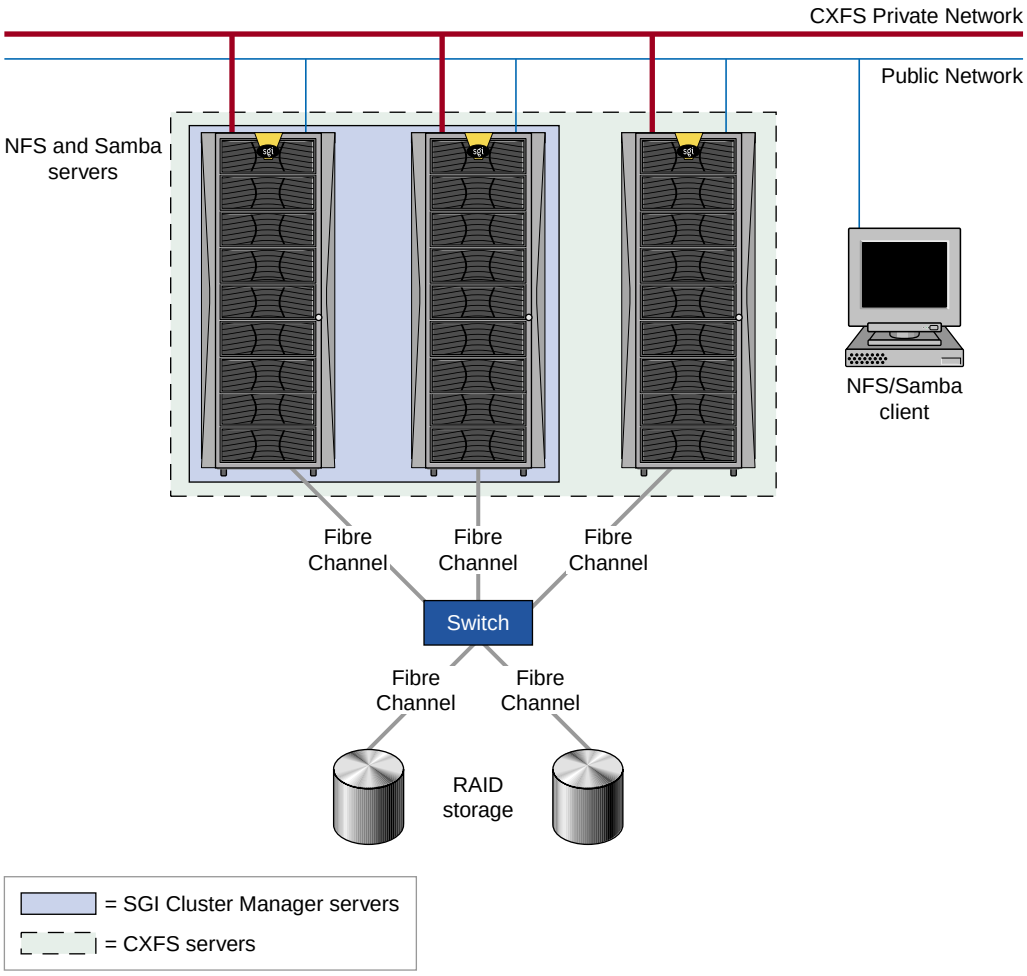

Figure 1-1 An Example CXFS and SGI Cluster Manager Configuration

Figure 1-2 shows several example configurations for application failover. In the two-member configuration, each member is the backup for the other. In 3+1 configuration, a single member serves as backup for all of the other members. In the ring configuration, backup duties are shared among all members.

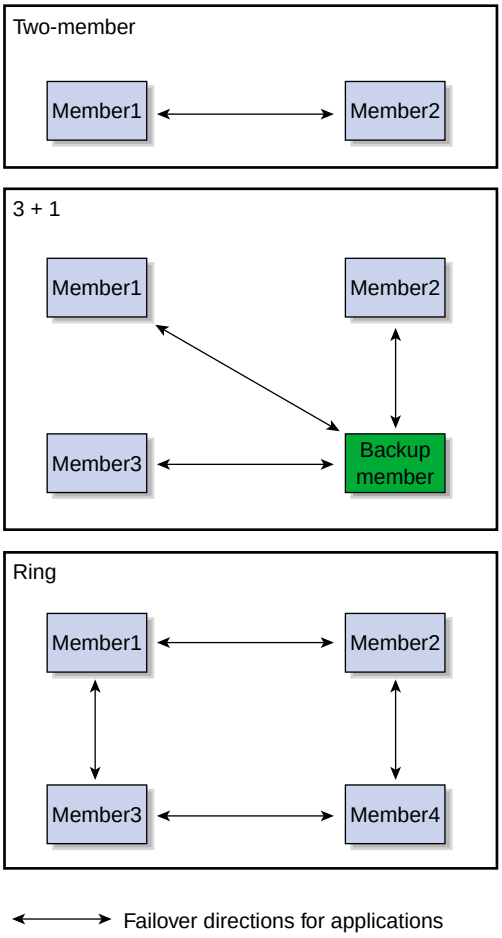

Figure 1-2 Example Application Failover Configurations

Software Requirements

SGI Cluster Manager requires the following:

- SGI ProPack 5 with SLES 10

Note: You cannot run the SLES `heartbeat` utility with SGI Cluster Manager.

This release also supports the following releases:

- CXFS release that supports metadata servers running SGI ProPack 5

  Note: Use of clustered XVM volumes with SGI Cluster Manager requires the CXFS plug-in. The SGI Cluster Manager XVM plug-in product supports local XVM volumes.

- DMF 3.5 or later
- TMF 1.4.6 or later

See the `README` file for a list of the RPMs included on CDs.

Failover Domains

The *failover domain* is the list of members in the cluster where a service can be online. Each failover domain has the following options that are considered when a new membership is formed or a failure occurs and a new target member for the service must be determined, or a member rejoins the cluster:

- *Restricted failover* permits failover only to the members listed. If all of the members in the domain are unavailable, the service will stop.

  If a domain is not restricted, a service can run on the member that is not in the domain if there is a failure and the member that is in the domain is unavailable. (However, administrative commands cannot relocate the service to a member that is not in the domain, whether or not this option is used.)

- *Ordered failover* causes the service to start on the first member defined (the lowest-ordered) if it is available; if that member is unavailable, the other member will be used. If controlled failback is not set, the service will automatically failback

from the second member to the original member when the original member is rebooted after a failure or maintenance period.

Note: *Lowest-ordered* means a higher preference for a service to be started on that member.

- *Controlled failback* prevents a service from being moved back to the original member when it rejoins the cluster even if it is the preferred member in the list (when ordered failover is used). The system administrator must manually relocate the service in order for it to run on the original member without an intervening failure. Only a new failure will cause a service to be automatically moved.

Suppose you have a cluster with members A, B, and C. Table 1-1 describes some of the possible results from using various options under different circumstances for the `nfs` service. (The failover domain options are also the same for 2 and 4 node clusters.)

Table 1-1 Failover Domain and Option Results

Failover Domain	Options	Circumstance	Results
(none)	(none)	Newly formed membership	The service will be started on A, B, or C (randomly chosen).
B	(none)	Newly formed membership	The service will be started on B if it is available. If B is not available, the service will be started on either A or C (randomly chosen).
B, A	(none)	Newly formed membership	The service will be started on either A or B (randomly chosen). If that member is unavailable, the other will be used. (A 3–member cluster will not run with only 1 member available.) This situation is similar to having no failover domain.
B	(none)	The service is running on B and then B fails	The service will be started on A or C (randomly chosen). The service will remain on A or C even after B restarts.

Failover Domain	Options	Circumstance	Results
B, A	Ordered	Newly formed membership	The service will be started on B if it is available. If B is not available, the service will be started on A. If neither A nor B is available, the cluster will not run. (A 3–member cluster will not run with only 1 member available.)
B, A	Restricted and controlled	Newly formed membership	The service will be started on either A or B (randomly chosen). If that member fails, the service will be restarted on the other member and will remain there until the system administrator manually intervenes.
B	Restricted	The service is running on B and then B fails	The service will stop.
B, A	Ordered	The service is running on B and then B fails	The service will be started on A. The service will be moved back to B as soon as it restarts.
B, A	Ordered and controlled	The service is running on B and then B fails	The service will be started on A. The service will remain on A, even after B restarts. To go back to B, the system administrator must manually move the service.
A, B, C	Restricted and ordered	Newly formed membership	The service will be started on A. If A is not available, the service will be started on B. If neither A nor B is available, the cluster will not run. (A 3–member cluster will not run with only 1 member available.)
A, B, C	Restricted and ordered	Started on A and then A fails	The service will be started on B. When A restarts, the service will return to A.

Failover Domain	Options	Circumstance	Results
A, B, C	Restricted and ordered	Started on A, A fails, service cannot start on B	The service will be started on C after failing to start on B. When A restarts, the service will return to A.
A, B, C	Restricted, ordered, and controlled	The service was running on A when A fails	The service will fail over to B. The service will remain on B, even after A restarts. To go back to A, the system administrator must manually move the service.

Cluster Daemons

Following is an overview of the cluster daemons:

- `clumembd`(8) is the cluster membership daemon. It performs network heartbeats and checks the liveliness of other members in the cluster.
- `cluquorumd`(8) is the cluster quorum daemon. It computes new membership and implements quorum. It also implements I/O fencing by resetting members that are in failed state and reads/writes membership information to the shared quorum partitions.
- `clurmtabd`(8) is the cluster remote NFS mount table daemon. It synchronizes NFS mount point entries by polling the `/var/lib/nfs/rmtab` file.
- `clusvcmgrd`(8) is the cluster service manager daemon. It starts/stops and checks the status of services running in the cluster.
- `clulockd`(8) is the cluster global lock manager daemon. The locks are stored on the shared quorum partitions.

For more information, see the man pages.

Best Practices

This chapter provides an overview of the best practices for system administration in an SGI Cluster Manager cluster. It discusses the following:

Configuration Best Practices

This section discusses the following:

Decide How You Want to Use SGI Cluster Manager

You must first decide how you want to use the SGI Cluster Manager cluster, what applications you want to run, and which of these should be made highly available (HA). This includes deciding how software and data will be distributed. You can then configure the disks and interfaces to meet the needs of the HA services that you want the cluster to provide.

Questions you must answer during the planning process are as follows:

- How do you plan to use the machines? Your answers might include uses such as offering home directories for users, running particular applications, supporting an Oracle database, providing Web services, and providing file service.
- Which of these uses will be provided as an HA service? SGI provides plug-ins for some HA applications; see "SGI Software Storage Plug-In Product" on page 2. To offer other applications as HA services, see Chapter 7, "Creating a New Highly Available Application" on page 89. If you need assistance, contact SGI Professional Services, which offers custom SGI Cluster Manager development and integration services.
- Which machine will be the primary member for each HA service? The primary member is the machine that provides the service, such as exporting the filesystem.
- For each HA service, how will the software and data be distributed on shared and non-shared disks? Each application has requirements and choices for placing its software on disks that are failed over (shared) or not failed over (non-shared).
- Are the shared disks going to be part of a RAID storage system or are they going to be disks in SCSI or Fibre Channel disk storage that have plexed logical volumes on them? Shared disks must be part of a RAID storage system or in a SCSI or Fibre Channel disk storage with plexed logical volumes on them.
- How will shared disks be configured?
  - As raw logical volumes?
  - As logical volumes with XFS filesystems on them?
  - As local XVM logical volumes with XFS filesystems on them?

  - As CXFS filesystems, which use XVM logical volumes? For information on using SGI Cluster Manager and CXFS, see Chapter 9, "CXFS Plug-In" on page 101.

  The choice of volumes or filesystems depends on the application that is going to use the disk space.

- Which IP addresses will be used by clients of HA services? Multiple interfaces may be required on each machine because a member could be connected to more than one network or because there could be more than one interface to a single network.
- How many HA IP addresses on each network interface will be available to clients of the HA services? At least one HA IP address must be available for each interface on each member that is used by clients of HA services.
- Which HA IP addresses on members in the failover domain are going to be available to clients of the HA services?
- For each HA IP address that is available on a member in the failover domain to clients of HA services, which interface on the other members will be assigned that IP address after a failover? Every HA IP address used by an HA service must be mapped to at least one interface in each member that can take over the resource group service. The HA IP addresses are failed over from the interface in the primary member of the resource group to the interface in the replacement member.

Use Redundant Hardware

SGI Cluster Manager runs on a specific set of SGI servers and supported disk storage devices. See "Hardware Requirements" on page 3. A cluster contains 2 members running SGI Cluster Manager.

You should provide multiple sources of the following:

- Power sources
- RAID disk devices and mirrored disks (SGI Cluster Manager supports Fibre Channel RAID and mirrored disks in direct-attach and SAN configurations)
- Paths to storage devices
- Networks
- Fibre Channel switches

- 100–MB hubs

At least two Ethernet interfaces on each member are required for the control network heartbeat connection, by which each member monitors the state of other members. The SGI Cluster Manager software also uses this connection to pass control messages between members. These interfaces have distinct IP addresses.

You can use 10/100baseT or 1–Gb ports in the system for heartbeat communication. All members should be on the same local network segment.

For each disk in a SGI Cluster Manager cluster, you must choose whether to make it a shared disk (allowing it to be failed over) or a non-shared disk. The system disk must be a non-shared disk. SGI Cluster Manager software must be on a non-shared disk and all system directories (such as `/tmp`, `/var`, `/usr`, `/bin`, and `/dev`) should be on a non-shared disk.

For more information about storage configuration, see the following:

- *SGI InfiniteStorage 10000 User's Guide*
- *SGI InfiniteStorage 6700 User's Guide*
- *SGI InfiniteStorage 4500 RAID User's Guide*
- *SGI InfiniteStorage 4000 RAID User's Guide*
- *SGI InfiniteStorage 350 Quick Start Guide*
- *SGI InfiniteStorage 120 Mass Storage Hardware Topics*
- *SGI InfiniteStorage S330 RAID User's Guide*
- *SGI InfiniteStorage RM610 and RM660 User's Guide*
- *SGI InfiniteStorage TP9500 and TP9700 RAID User's Guide*
- *SGI InfiniteStorage TP9300 and TP9300S RAID User's Guide*
- *TPM Installation Instructions and User's Guide for SGI TP9100*
- *SGI TPSSM Administration Guide*

Use a Private Network

For performance and security reasons, SGI recommends that the networks be private. Using a private network limits the traffic on the public network and therefore will help avoid unnecessary resets or disconnects. You many want to choose a numbering convention for private networks such as 192.168.50.*x* for primary network and 192.168.51.*x* for backup network, where *x* is the CXFS member ID in the cluster.

Use Shared Quorum Partitions

SGI Cluster Manager for Linux **requires** two shared 10–MB disk partitions to keep membership quorum: the *primary partition* and the *shadow partition* (used for backup purposes). You should use the block device to access these partitions.

The primary partition and the shadow partition should be in different storage devices connected to the members using different Fibre Channel (FC) cards. The two partitions should have independent I/O paths.

Note: If a data corruption occurs on the primary partition, the cluster members switch to the shadow partition.

A machine cannot join the cluster if it cannot write to both the primary and the shadow partition. If a cluster member is unable to write to both shared partitions, the member reboots and therefore leaves the cluster. (The other member can remotely power-cycle it as well.)

SGI Cluster Manager works on supported SGI RAID configurations. Each member in the cluster should be connected to storage using multiple paths so that service failovers are minimized. Ideally, the two shared quorum partitions should be on separate FC controllers at the front end, separate HBAs on the Altix, and on separate RAID logical units (LUNs) or RAID arrays if possible. They should be at least 10 MB in size and the partition type must be `linux`.

The device names for the shared quorum partitions must be identical on all cluster members. Use the `/usr/lib/clumanager/create_device_links` script to create the same device name on each member.

For more information, see the books listed at the beginning of this chapter.

Use the Heartbeat Network

SGI Cluster Manager uses hostnames for sending heartbeat and control messages to indicate that a member is up and running and to request operations or distribute information. Ethernet cables are provided that will allow the members to be connected directly or connected using a network hub.

You can use 10/100baseT or 1–Gb ports in the system for heartbeat communication. For more information, see *SGI Altix Systems Dual-Port Gigabit Ethernet Board User's Guide*.

Heartbeats are either broadcast on all networks or multicast on the network interface that hostname configured.

Fix Network Issues First

If there are any network issues on the private network, fix them before trying to use SGI Cluster Manager.

Use Network Reset for Power Control

You must use SGI L2 system controllers for power control. You may connect to the L2 over an Ethernet network or (depending on your particular hardware) you may connect directly to the L2 through a serial port.

The network connection method (`l2network`) is preferred because it is easier to set up and provides for greater flexibility while configuring a cluster.

However, if you use serial reset lines, use Cat5 wire with appropriate connectors and point-to-point connections between members. Be aware of the distance limitations for serial cables. You should have hardware flow control pins (RTS/CTS) connected in the serial cable.

For more information, see Chapter 3, "Power Control" on page 25.

Configure Firewalls for SGI Cluster Manager Use

SGI Cluster manager supports `iptables` for IP filtering. (It does not support `SuSEfirewall2`.)

Do one of the following:

- Configure firewalls to allow SGI Cluster Manager traffic:
  - If you are using `iptables`, the SGI Cluster Manager boot script (`init.d/clumanager`) will open the necessary ports when starting and close them when stopping.
  - If you are using something other than `iptables`, you must allow traffic on the following ports:
    - 1228 and 1229 for UDP multicast traffic
    - 34001-34004 for TCP
- Configure firewalls to allow all traffic on the private interfaces. This assumes that the public interface is not a backup metadata network.
- Disable firewalls.

For more information, see your firewall documentation.

Include Member Hostnames in `/etc/hosts`

SGI recommends that member hostnames and addresses be present in `/etc/hosts` so that communication between cluster daemons does not rely on the network information service (NIS) or the domain name service (DNS) being available.

Configure System Files Appropriately

You must configure the following system files appropriately in order to use SGI Cluster Manager:

- `/etc/hosts`
- `/etc/nsswitch.conf`
- `/etc/services`

In addition, you must ensure that the following have the correct hostname information in `/etc/HOSTNAME`.

The following hostname resolution rules and recommendations apply to SGI Cluster Manager clusters:

Caution: It is critical that you understand these rules before attempting to configure a SGI Cluster Manager cluster.

- The hostname must be configured on a network interface connected to the public network and should be resolved using `/etc/hosts`.
- Hostnames should not contain an underscore (_) or include any white-space characters.
- The `/etc/hosts` file has the following format, where *hostname* can be the simple hostname or the fully qualified domain name:

  IP_address hostname

  For example, suppose your `/etc/hosts` file contains the following:

  ```
  # The public interface:
  192.0.34.166  color-green.example.com color-green green

  # The private interface:
  192.168.1.1  color-green-private.example.com  color-green-private green-private
  ```

  The `/etc/HOSTNAME` file could contain either the hostname `color-green` or the fully qualified domain name `color-green.example.com`.

  In this case, you would enter the hostname `color-green` or the fully qualified domain name `color-green.example.com` for the member name.

- If you use the name service, you must configure your system so that local files are accessed before either NIS or DNS. That is, the `hosts` line in `/etc/nsswitch.conf` must list `files` first. For example:

  ```
  hosts:      files nis dns
  ```

  (The order of `nis` and `dns` is not significant to SGI Cluster Manager; `files` must be first.)

  The `/etc/sysconfig/network/ifcfg-eth*` file must have one of the interfaces with the value of the `IPADDR` corresponding to the IP address of `HOSTNAME` as found in `/etc/hosts`.

  For more information see the `nsswitch.conf` and the `nsd` man pages.

- If you change the `/etc/nsswitch.conf` or `/etc/hosts` files, you must restart `nsd` by using the `nsadmin restart` command, which also flushes its cache.

  The reason you must restart `nsd` after making a change to these files is that the `nsd` name service daemon actually takes the contents of `/etc/hosts` and places the contents in its memory cache in a format that is faster to search. Thus, you must restart `nsd` in order for it to see that change and place the new `/etc/hosts` information into RAM cache. If `/etc/nsswitch.conf` is changed, `nsd` must re-read this file so that it knows what type of files (for example, `hosts` or `passwd`) to manage, what services it should call to get information, and in what order those services should be called.

  The IP addresses on a running member in the cluster and the IP address of the first member in the cluster cannot be changed while cluster services are active.

- You should be consistent when using fully qualified domain names in the `/etc/hosts` file. If you use fully qualified domain names in `/etc/HOSTNAME` on a particular member, then all of the members in the cluster should use the fully qualified name of that member when defining the IP/hostname information for that host in their `/etc/hosts` file.

  The decision to use fully qualified domain names is usually a matter of how the clients (such as NFS) are going to resolve names for their client server programs, how their default resolution is done, and so on.

- If you change hostname/IP address mapping for a member in the cluster, you must recreate the member in the configuration database. You must remove the member from the cluster and the database, restart cluster processes on that member, and then define the member and add it to the cluster.

- The `/etc/HOSTNAME` file contains the hostname of the machine and should not be associated with an HA IP address.

Install the Software Appropriately

You must consult SGI managed services before installing an SGI Cluster Manager system. For more information, see:

http://www.sgi.com/services/managed_services/

You must install the SGI Cluster Manager software base components. You may want to install software for ESP, Performance Co-Pilot, accounting, and `expect` (for TMF).

You may wish to use `sendmail` with an alias to be used when reporting problems to the system administrator.

SGI recommends that you make configuration changes when the same version of SGI ProPack and the same version of SGI Cluster Manager is running on all members in the cluster.

For more information, see Chapter 4, "Software Installation" on page 35.

Installing SGI Cluster Manager Plug-ins

The basic process to install plug-in is as follows:

1. Install, configure, and test the base SGI Cluster Manager software as described in Chapter 4, "Software Installation" on page 35.
2. Install any required application software and the plug-in software.
3. Perform any system file configuration required by the plug-in.
4. Create the optional plug-in for the application that will be failed over. See Chapter 7, "Creating a New Highly Available Application" on page 89.
5. Test the failover.

Performing Upgrades

To perform an upgrade, do the following:

1. Stop SGI Cluster Manager daemons on each member.
2. Install the new software on each member.
3. Restart SGI Cluster Manager daemons on each member.
4. Verify that the SGI Cluster Manager configuration works on the upgraded cluster.

Use Cluster Configuration Tools Appropriately

At any given time, you must use either the `sgicm-config-cluster`(8) GUI or `sgicm-config-cluster`(8) command-line tool to perform configuration tasks; do not use both at the same time. The GUI and the CLI supply similar functionality, although there are a few exceptions.

After making modifications to the configuration using the GUI, you should save the information. See "Saving Changes" on page 43

If you are going to access the GUI remotely, SGI recommends that you use a virtual X server method such as Virtual Network Computing (VNC) for better performance. For more information, see the following RealVNC website:

http://www.realvnc.com/download.html

For more information, see "Cluster Configuration Tools" on page 41.

Use Consistent Filenames

The names of the device files for filesystems to store quorum information must be the same on all cluster members.

Use an Appropriate Failover Speed

An inappropriate member timeout will result in false failovers. An appropriate failover speed value will take time to determine; this can be the most difficult part of the SGI Cluster Manager configuration process. See "Step 5: Change the Heartbeat Interval, Timeout, and Failover Speed" on page 51.

Synchronize Configuration Changes

Each member has an `/etc/cluster.xml` file that contains cluster configuration information. If you make a change to this file on one member during initial configuration, you must copy the file to the other member using a command such as `scp`(1). (Do not edit the file directly.)

After making configuration changes, you must verify that the configuration files across the cluster are in synchronization.

For more information, see "Step 15: Verify that Configuration Changes are Synchronized" on page 64.

Modify the Logging Level as Needed

When you first install SGI Cluster Manager, you should set logging levels high to obtain enough information for troubleshooting. After the system is running

satisfactorily, you can reduce the log levels if the log files are filling too quickly. If problems occur, you an increase the logging level to help detect the cause. See "Message Logging" on page 87.

Test the Configuration

Test the system in three phases:

- Test individual components prior to starting SGI Cluster Manager software
- Test normal operation of the system
- Simulate failures to test the operation of the system after a failure occurs

During the first few weeks of operation, examine the failovers of each resource group to determine if they are due to inappropriately short timeout values; adjust the timeout values as needed.

Note: Performing a backup of the entire system may add stress to the system. You should consider this when determining resource group timeouts in order to avoid unnecessary failovers.

Administration Best Practices

This section covers best practices for the following:

Do Not Interrupt SGI Cluster Manager Commands

After a SGI Cluster Manager command is started, it may partially complete even if you interrupt the command by typing `Ctrl-c`. If you halt the execution of a command this way, you may leave the cluster in an indeterminate state and you may need to use the various status commands to determine the actual state of the cluster and its components.

Manage Log Files

If you are having problems with disk space, you may want to choose a less verbose log level.

You should rotate the log files at least weekly so that your disk will not become full. You may also want to archive and compress log files. See "Message Logging" on page 87

Start CXFS Daemons First

You should start CXFS cluster services and CXFS services before starting SGI Cluster Manager daemons. See "CXFS Start/Stop Order" on page 104.

Avoid Relocating CXFS Metadata Servers

Unless a service must run on the CXFS metadata server, you should configure SGI Cluster Manager so that it does not relocate the CXFS metadata server when it fails over a service.

Configure DMF Administrative Filesystems as Local XVM Filesystems

SGI recommends that you configure DMF administrative filesystems as local XVM filesystems, as discussed in "Configuring DMF for Local XVM Filesystems" on page 107. DMF cannot start until the DMF administrative filesystems are available. If they are CXFS filesystems, CXFS must recover them before they are accessible. See "Configuring DMF-Managed XFS Filesystems as CXFS Filesystems" on page 107.

Configure Tape Devices Consistently

If tape devices that are managed by the `helper_tmf` script are configured on more than one member in the cluster, they should be configured consistently. The same tape driver (for example, `ts`) should be used on each member where the tape device is configured. See "Configuring Tapes and TMF" on page 117.

Maintain the `/etc/tmf/sgicm_tmf.config` File on Each Member

You must maintain the `sgicm_tmf.config` file on each member; a change on one member is unknown to the other members. See "The `/etc/tmf/sgicm_tmf.config` File" on page 114.

Power Control

You must use SGI L2 system controllers for power control. An L2 controller is standard with each Altix 3700 rack. On some platforms, including the Altix 350, an L2 controller is a separate optional product that must be purchased in order to use SGI Cluster Manager. You may connect to the L2 over an Ethernet network or (depending on your particular hardware) you may connect directly to the L2 through a serial port. The network connection method (`l2network`) is preferred because it is easier to set up and provides for greater flexibility while configuring a cluster.

Note: If you have a system with an emulated L2 controller (such as an Altix 330 or Altix 3700 Bx2), or if you run CXFS with SGI Cluster Manger, you must use the `l2network` connection type. See "`l2network` Ethernet Connection" on page 25.

For information about configuring the power controller, see "Step 4: Add Power Controller Configuration" on page 47.

Note: Third-party network-based and serial-based power controllers are not supported for SGI Cluster Manager on SGI Altix servers. (However, network-based power controllers should not be confused with the `l2network` Ethernet connection.)

L2 Connections

You can use one of the following methods:

- "`l2network` Ethernet Connection" on page 25 (preferred)
- "`l2` Serial Connection" on page 27

`l2network` Ethernet Connection

The `l2network` Ethernet connection is the preferred L2 connection method. It is required if you have 3 or more members in the cluster.

The `l2network` Ethernet connection requires the following:

- An Ethernet port on each member.

- All members in the cluster and the L2 must be connected to the same network. SGI recommends using a private network for greater reliability. (If a private network is used, a PCI Ethernet card is required for each member.)
- The `l2network` designation in the cluster configuration
- The IP address of each member's L2 controller must be entered as the address of that member's power controller. For example, to specify the power controller for cluster member Machine-A, enter the IP address of the L2 of Machine-A.

Note: Multiple members within a partitioned system may share a single L2 as long as the system serial number on each L1 is the same.

Figure 3-1 shows the L2 Ethernet connection for an Altix 3700.

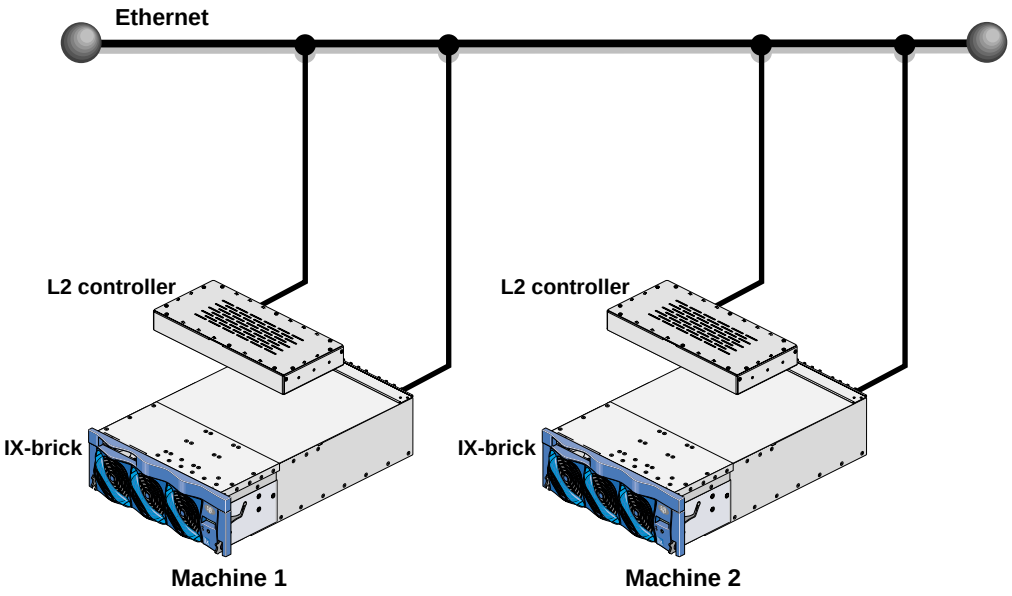

Figure 3-1 Altix 3700 L2 with an Ethernet Connection

Figure 3-2 shows the L1 USB port on an Altix 3700 Bx2 CR brick. Use a USB cable to connect the L1 USB port to the USB/network adapter mounted on the rack. The

USB/network adapter should be connected to the network and must be accessible from the other SGI Cluster Manager member via the network.

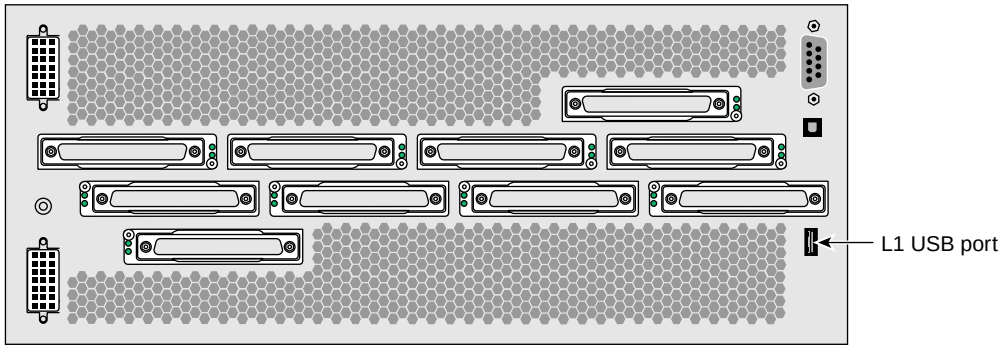

Figure 3-2 Altix 3700 Bx2 CR-brick Rear Panel

l2 Serial Connection

Note: The `l2network` Ethernet connection is preferred.

A serial connection requires the following:

- A two-member cluster
- Altix 350: serial ports on Altix 350 with IO10 and a `IO10 CBL-SATA-SERIAL` multiport serial adapter cable. You must also order the `LS-BASE-IO` serial ATA (SATA) drive option.

  Figure 3-3 shows the rear panel for an Altix 350. For information about using an Altix 350 with an IO9 PCI card, see "Hardware Requirements" on page 3.

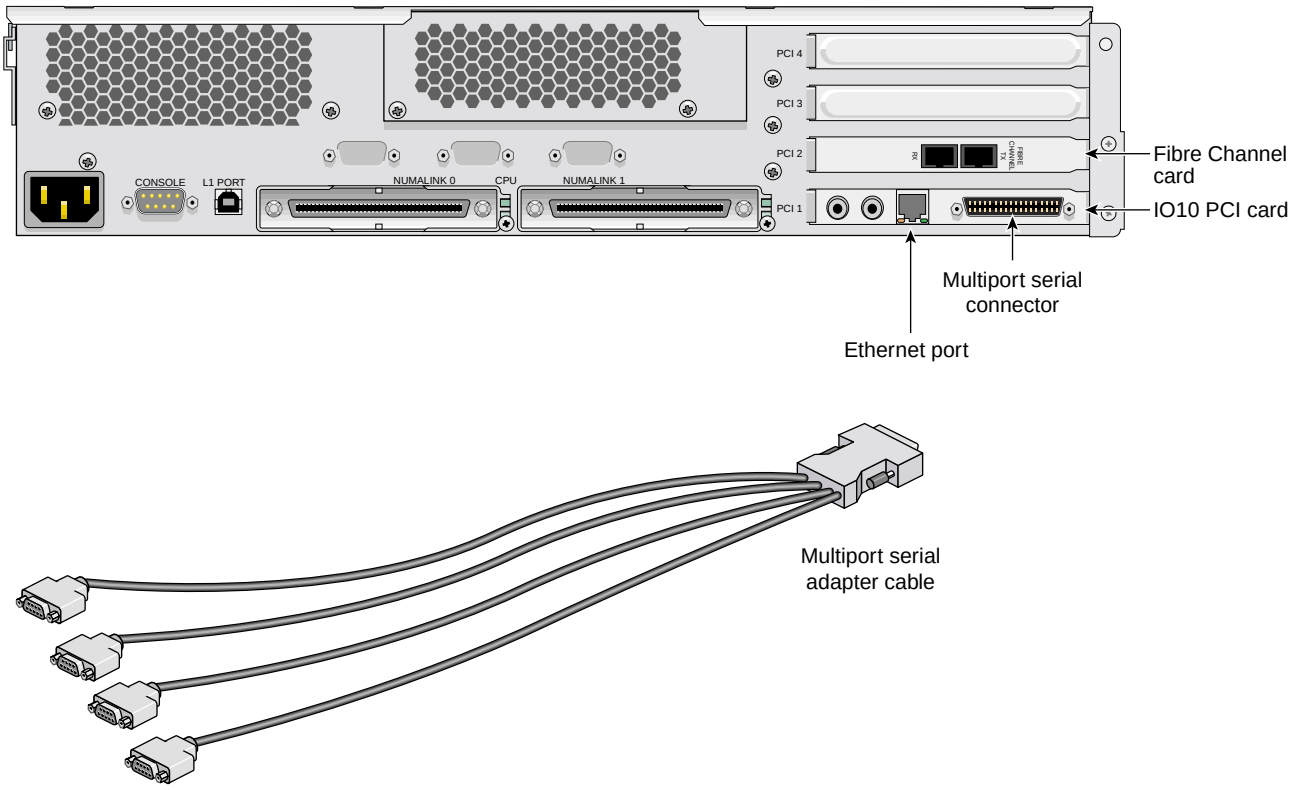

Figure 3-3 Altix 350 Rear Panel

- Altix 3700: DB9 serial ports on an IX-brick.
- Serial cables should use the remote modem port on the L2 system controller. Connect the serial cable to the remote modem port on one end and the `tty` port on the other end.
- The `l2` designation in the cluster configuration

Figure 3-4 and Figure 3-5 show the serial connections.

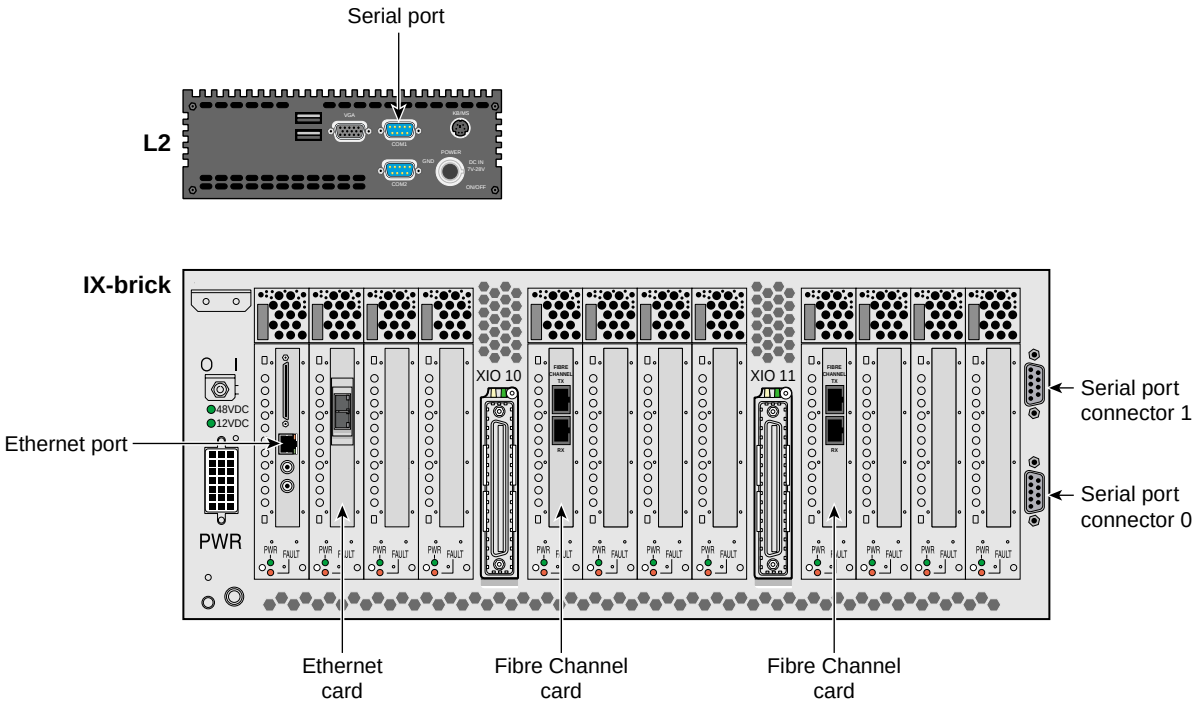

Figure 3-4 Altix 3700 Rear Panel

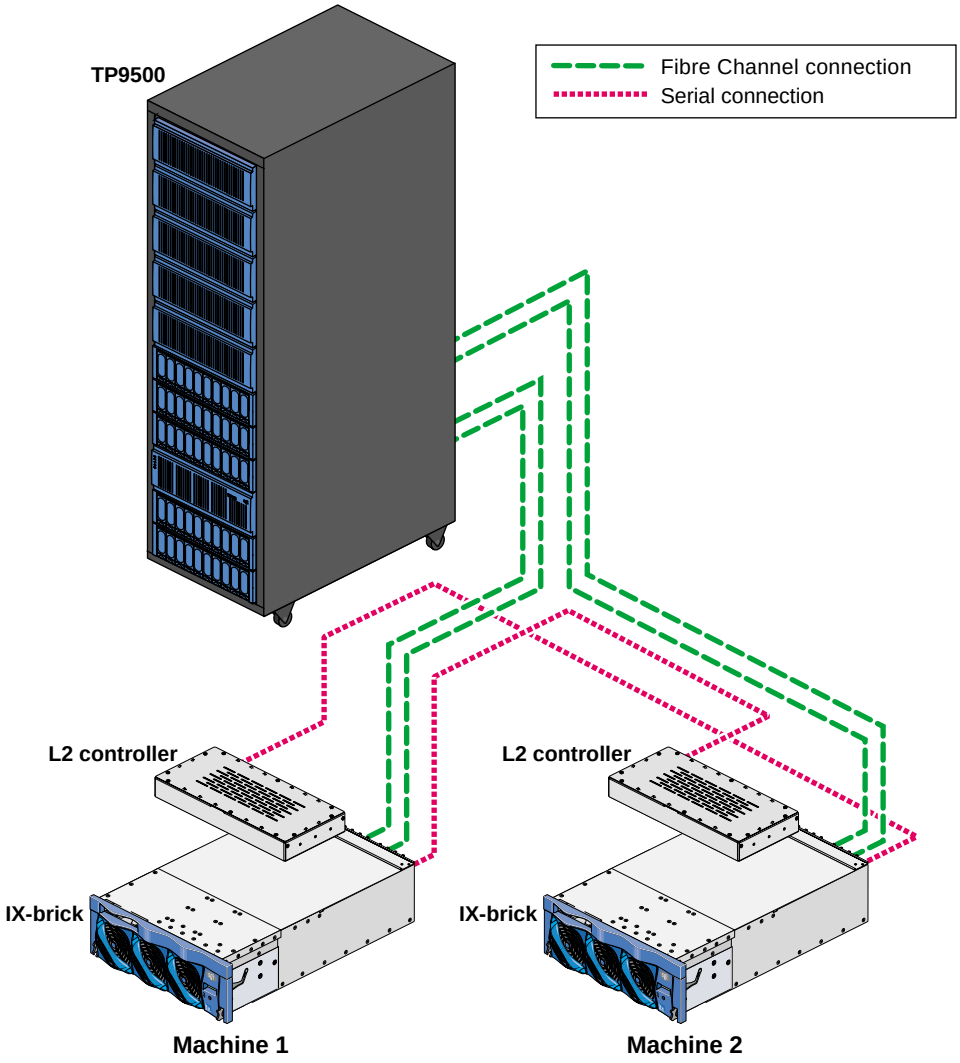

Figure 3-5 Altix 3700 L2 with Serial Cable Connection

Automatic Power for the L2

You must turn on the `apwr` automatic power variable on the L2.

To show the current status, use the following command on the L2:

```
L2> apwr
```

To turn on automatic power, set the `apwr` value to `on` on the L2. For example:

```
L2> apwr on
```

Connectivity Testing

Use the appropriate testing method:

- "Testing Ethernet Connectivity for the L2" on page 31
- "Testing Serial Connectivity for the L2" on page 32

Testing Ethernet Connectivity for the L2

To determine an L2's IP address or to configure an IP address for an L2, connect to the L2 using the serial port and use the L2 `ip` command.

For example, to show the current IP setting:

```
l2-foo-001-L2>ip
addr: 192.168.2.70   netmask: 255.255.255.0   broadcast addr: 192.168.2.255
```

Note: If you are using DHCP to assign the IP addresses of the L2s dynamically, you will see the following message:

```
l2-foo-001-L2>ip
static IP address not currently set
```

This is not an error, it just indicates that static IP addresses are not in use. To determine the IP address, use the `cfg` command from the L2 to show each brick and each L2 in the configuration.

To change the IP setting to `192.0.34.166`::

```
l2-foo-001-L2> ip  192.0.34.166 255.255.255.0 192.0.34.255
```

You can use the `ping` command to test connectivity to an L2. You can also use the L2 `l2find` command to find other L2s in the same subnet. For example:

```
[root@altix root]$ telnet l2-server.example.com
Trying 192.168.1.98...
Connected to l2-server.example.com.
Escape character is '^]'.

Linux 2.4.7-sgil2 (192.168.1.98) (ttyp2)

SGI SN1 L2 Controller

INFO: connection established to localhost, to quit enter <ctrl-]> <>
server-001-L2>help l2find
l2find
        print list of L2's on the same subnet as this one
server-001-L2>l2find
6 L2's discovered:

IP              SSN      NAME             RACK FIRMWARE
--------------- -------- ---------------- ---- ------------
[ L2's with different System Serial Numbers ]
  192.168.1.67  R2000016                     000 L3 controlle
  192.168.1.132 L1000487                     001 1.3.61
  192.168.1.96  N0000005  bar                002 1.24.2
  192.168.1.100 N0000005  bar2               003 1.24.2
  192.168.1.94  N0000005  bar3               004 1.24.2
  192.168.1.105 N0000005  bar_l2_2           001 1.22.0
server-001-L2>
```

Testing Serial Connectivity for the L2

You can use the `cu`(1) command to test the serial reset lines if you have installed the `uucp` RPM.

The `cu` command requires that the device files be readable and writable by the user `uucp`. The command also requires that the `/var/lock` directory be writable by group `uucp`.

Perform the following steps:

1. Assure that the `ioc4_serial` module is loaded:

   ```
   # lsmod | grep ioc4_serial
   ```

   If you do not see `ioc4_serial` in the output from the `lsmod` command, install the module with `modprobe`:

   ```
   # modprobe ioc4_serial
   ```

   If you intend to use the L2 serial connection permanently, the `ioc4_serial` module must be loaded automatically when the system boots. Typically, this is done by editing `/etc/sysconfig/kernel`. See Chapter 4, "Software Installation" on page 35.

2. Change ownership of the serial devices so that they are in group `uucp` and owned by user `uucp`.

   Note: The ownership change may not be persistent across reboots.

   For example, suppose you have the following TTY devices on the IO10:

   ```
   # ls -l /dev/ttyIOC*
   crw-rw----  1 root uucp 204, 50 Sep 15 16:20 /dev/ttyIOC0
   crw-rw----  1 root uucp 204, 51 Sep 15 16:20 /dev/ttyIOC1
   crw-rw----  1 root uucp 204, 52 Sep 15 16:20 /dev/ttyIOC2
   crw-rw----  1 root uucp 204, 53 Sep 15 16:20 /dev/ttyIOC3
   ```

   To change ownership of them to `uucp`, you would enter the following:

   ```
   # chown uucp.uucp /dev/ttyIOC*
   ```

3. Determine if group `uucp` can write to the `/var/lock` directory and change permissions if necessary.

   For example, the following shows that group `uucp` cannot write to the directory:

   ```
   # ls -ld /var/lock
   drwxr-xr-t  5 root uucp 88 Sep 19 08:21 /var/lock
   ```

   The following adds write permission for group `uucp`:

   ```
   # chmod g+w /var/lock
   ```

4. Join the `uucp` group temporarily, if necessary, and use `cu` to test the line.

   For example:

```
# newgrp uucp
# cu -l /dev/ttyIOC0 -s 38400
Connected
nodeA-001-L2>cfg
L2 192.168.0.1: - 001 (LOCAL)
L1 192.168.1.133:0:0    - 001c04.1
L1 192.168.1.133:0:1    - 001i13.1
L1 192.168.1.133:0:5    - 001c07.2
L1 192.168.1.133:0:6    - 001i02.2
```

For more information, see the `cu`(1) man page and the documentation that comes with the `uucp` RPM.

Other tools that may be useful when testing connectivity are `minicom`(1) and `kermit`(1).

After you have configured the cluster software, (see Chapter 5, "Configuration" on page 41), you can also use the `clufence`(8) command to test serial connectivity. See the `clufence`(8) man page for more information.

Software Installation

This chapter discusses the following:

Required Packages

The following packages are required:

- Required base product packages from the SGI Cluster Manager base product CD:
  - `clumanager-2.0.1-4.3.ia64.rpm`
  - `sgicm-config-cluster-2.0.1-4.3.noarch.rpm`
  - `sgi-cluster-manager-docs-4.3-1.noarch.rpm`

  Note: The `sgicm-config-cluster` RPM is dependent upon the `clumanager-2*` RPM. You must install the `clumanager-2*` RPM first.

- Optional high-availability plug-ins and scripts for CXFS, DMF, TMF, and local XVM from the SGI Cluster Manager storage software plug-ins CD:

  `clumanager-sgi-2.0.1-4.3.ia64.rpm`

For additional information, see the `README` file.

Installing the Software

You must install the SGI Cluster Manager base software before the plug-in software.

Note: This procedure assumes that you are running the supported level of SGI ProPack listed in "Software Requirements" on page 7. For more information about installing SGI ProPack, see *SGI ProPack for Linux Start Here*.

Installing the Base Software

To install the SGI Cluster Manager base software, do the following:

1. Insert the *SGI Cluster Manager 4.3 for Linux — Base Product* CD.
2. Start the YaST installation tool:

   # **yast2**

3. Click the **Installation Source** icon.
4. From the **Add** pull-down menu in the **Installation Source** window, select **CD ...**
5. Click **Finish**.
6. Click the **Software Management** icon.
7. In the **Software Management** window, select the following:

   Filters
   > Patterns

8. Locate **SGI ClusterManager** in the list that appears.
9. Click the **SGI ClusterManager** checkbox.
10. Click **Accept**.
11. If you intend to use L2 serial port connections in your cluster (see "`l2` Serial Connection" on page 27), you must assure that the `ioc4_serial` module is loaded when the system boots. Edit the file `/etc/sysconfig/kernel` and find the line that starts with `MODULES_LOADED_ON_BOOT`. It may look something like this:

```
MODULES_LOADED_ON_BOOT="job numatools xpmem fetchop mmtimer csa mca_recovery"
```

Change this line to include `ioc4_kernel`:

```
MODULES_LOADED_ON_BOOT="job numatools xpmem fetchop mmtimer csa mca_recovery ioc4_serial"
```

Note: This change will take effect at the next reboot.

Installing the Plug-In Software

Note: You must install the base software before installing the plug-in software. See "Installing the Base Software" on page 36.

To install the plug-in software, do the following:

1. Insert the *SGI Cluster Manager 4.3 for Linux — Storage Software Plug-ins* CD.
2. Start the YaST installation tool:

   # **yast2**

3. Click the **Installation Source** icon.
4. From the **Add** pull-down menu in the **Installation Source** window, select **CD ...**
5. Click **Finish**.
6. Click the **Software Management** icon.
7. In the **Software Management** window, select:

   Filters
   > Patterns

8. Locate **SGI ClusterManager Plugins** in the list that appears.
9. Click the **SGI ClusterManager Plugins** checkbox.
10. Click **Accept**

Upgrading the Software

You must upgrade the SGI Cluster Manager base software before the plug-in software.

Note: This procedure assumes that you are running the supported level of SGI ProPack listed in "Software Requirements" on page 7. For more information about installing SGI ProPack, see *SGI ProPack for Linux Start Here*.

Upgrading the Base Software

To upgrade the SGI Cluster Manager base software, do the following:

1. Insert the *SGI Cluster Manager 4.3 for Linux — Base Product* CD.
2. Start the YaST installation tool:

   # **yast2**

3. Click the **Installation Source** icon.
4. From the **Add** pull-down menu in the **Installation Source** window, select **CD ...**
5. Click **Finish**.
6. Click the **Software Management** icon.
7. In the **Software Management** window, select the following:

   Filters
   > Patterns

8. Locate **SGI ClusterManager** in the list that appears.
9. Click the **SGI ClusterManager** checkbox (which already has a check in it). This changes the icon to an upgrade icon.
10. Click **Accept**.

Upgrading the Plug-In Software

Note: You must upgrade the base software before installing the plug-in software. See "Installing the Base Software" on page 36.

To upgrade the plug-in software, do the following:

1. Insert the *SGI Cluster Manager 4.3 for Linux — Storage Software Plug-ins* CD.
2. Start the YaST installation tool:

   # **yast2**

3. Click the **Installation Source** icon.
4. From the **Add** pull-down menu in the **Installation Source** window, select **CD ...**
5. Click **Finish**.
6. Click the **Software Management** icon.
7. In the **Software Management** window, select:

   Filters
   > Patterns

8. Locate **SGI ClusterManager Plugins** in the list that appears.
9. Click the **SGI ClusterManager Plugins** checkbox (which already has a check in it). This changes the icon to an upgrade icon.
10. Click **Accept**.

Uninstalling the Software

If you uninstall the base software, you must also uninstall the plug-in software. However, you can uninstall the plugin software without uninstalling the base software.

To uninstall the software, do the following:

1. Start the YaST installation tool:

   # **yast2**

2. Click the **Software Management** icon.
3. In the **Software Management** window, select:

   Filters
   > Patterns

4. Locate **SGI ClusterManager** in the list that appears.
5. Click the **SGI ClusterManager** checkbox and the **SGI ClusterManager Plugins** checkbox twice to turn their icons into trash cans, indicating that the selected items will be uninstalled.

   (If you wish to uninstall only the plugin software, do not click the the **SGI ClusterManager** checkbox.)

6. Click **Accept**.

Configuration

This chapter provides an overview of the cluster configuration tools and the basic configuration process. It discusses the following:

Cluster Configuration Tools

SGI Cluster Manager supports the following tools to configure the cluster:

- Graphical user interface (GUI): `sgicm-config-cluster`(8)
- Command-line interface (CLI): `sgicm-config-cluster-cmd`(8)

At any given time, you must use only one of these tools to perform configuration tasks. The GUI and the CLI supply similar functionality, although there are a few exceptions.

Displaying Configuration Status

The GUI displays the current status of the cluster. To display more details about an item, select the item and click **Properties**. Figure 5-1 shows an example of the GUI.

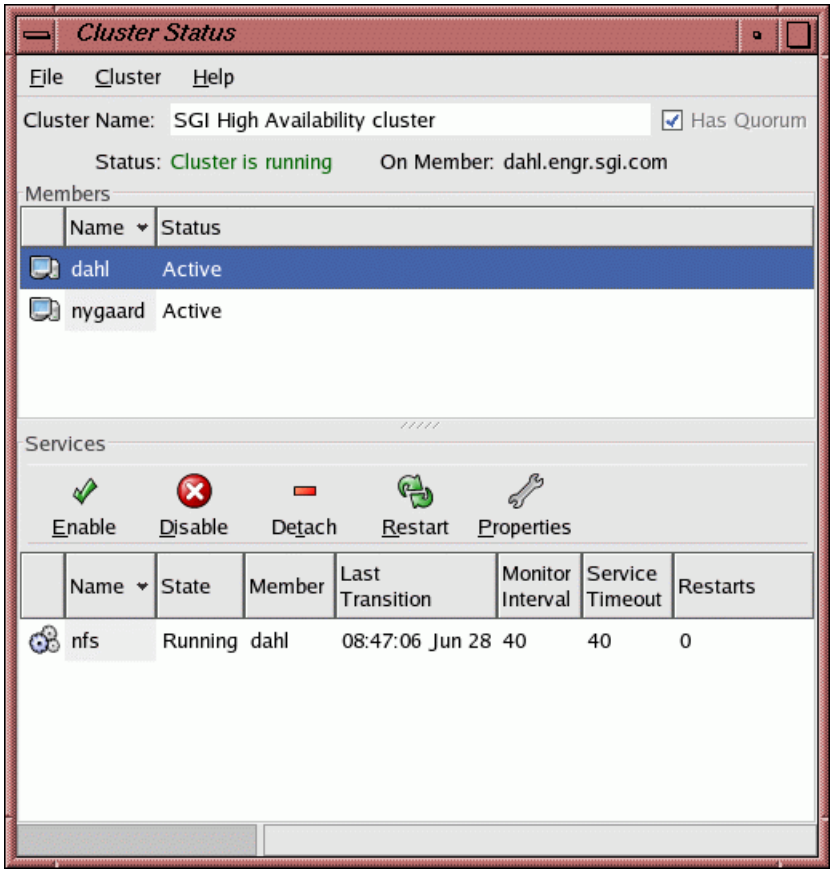

Figure 5-1 Cluster Status GUI

In the CLI, enter an argument with =*value* to assign a value or alone (without a =*value*) to display the current setting. For example, the following displays the name of the cluster and the number of times the configuration has been changed:

```
# sgicm-config-cluster-cmd --cluster

cluster:
  name = SGI High Availability cluster
  config_viewnumber = 14
```

Saving Changes

Caution: After making modifications to the configuration using the GUI, you should save the information using the following selections:

File
> Save

The information is written to a local `/etc/cluster.xml` file as well as to the shared quorum partitions. If the cluster is already configured and daemons are running on all nodes, the `/etc/cluster.xml` file on the other members is also updated.

Configuration Steps

This section discusses the configuration steps:

Step 1: Define the Shared Quorum Partitions

The names of the device files for filesystems to store quorum information must be the same on all cluster members. You must do the following:

- Ensure that the members have their disks attached identically.
- Create two unlabeled volumes of at least 10 MB in size on different physical devices. For example, you could use TPSSM on an SGI TP9500 RAID. For more information about TPSSM, see the *SGI TPSSM Administration Guide*.
- Run the `parted`(8) command to create two partitions of at least 10 MB in size with a 0x83 device type on the chosen volume. For more information, see the `parted`(8) man page.
- Create symbolic links (*symlinks*) so that the names of shared quorum partitions are the same on all cluster members. You must re-create the symlinks every time the machine reboots because `/dev` files are re-created. Therefore, you should modify the `/usr/lib/clumanager/create_device_links` script to add the device symlinks that are required.

  For example:

```
#!/bin/sh

#
# Create device links for shared quorum partitions if it does not exist
# if [ ! -h <device link> ]; then
#       ln -s /dev/....  /dev/shared ....
# fi
#

#
# Create device links for shared disks if the disks are not in the
# same I/O slot in all cluster members if it does not exist
#
 if [ ! -h /dev/shared1 ]; then
   ln
-s /dev/xscsi/pci02.02.0/node20000050cc00857a/port1/lun0/part1 /dev/shared1
```

```
 fi


 if [ ! -h /dev/shared2 ]; then
   ln
-s /dev/xscsi/pci02.02.1/node20000050cc00857a/port4/lun1/part2 /dev/shared2
 fi
```

SGI recommends that the two shared quorum partitions are on different Fibre Channel controllers; ideally, they should be on separate Fibre Channel controllers at the front end, on separate HBAs on the Altix, and on separate RAID logical units (LUNs) or RAID arrays if possible. They should be at least 10 MB in size and the partition type must be `Linux`. For more information, see Appendix B, "Setting the Partition Type to Linux" on page 143.

You can use the `lspci` command to see the Fibre Channel cards and you can use the `hwinfo --disk` command to see information about the shared disks. For example:

```
# lspci
...
0000:05:02.0 Fibre Channel: QLogic Corp. QLA2300 64-bit Fibre Channel Adapter (rev 01)
...

# hwinfo --disk --short

disk:
  /dev/sda             SGI ST373307LC
  /dev/sdb             SGI TP9100 FFX2
  /dev/sdc             SGI TP9100 FFX2
  /dev/sdd             SGI TP9100 FFX2
  /dev/sde             SGI TP9100 FFX2
  /dev/sdf             SGI TP9100 FFX2
  /dev/sdg             SGI TP9100 FFX2
  /dev/sdh             SGI TP9100 FFX2
```

For more information, see the `hwinfo`(8) man page.

Partition 1 from the disk at Fibre Channel target 2 and partition 1 from the disk at Fibre Channel target 3 will be used for storing shared state. To assure that the devices

have the same name on all cluster members, you could create symlinks to the block device files:

```
# ln -s /dev/sdb1  /dev/shared1
# ln -s /dev/sdc1  /dev/shared2
```

You should add these symlink commands to the `/usr/lib/clumanager/create_device_links` script on the cluster member. For more information, see the `ln`(1) man page.

The shared quorum partitions will then be referred to as `/dev/shared1` and `/dev/shared2`. These partitions will be the primary and shadow, respectively, in the following examples.

To define the shared quorum partitions in the GUI configuration window, select the following from the **Cluster Configuration** window:

Cluster
> Shared State

In the CLI:

```
sgicm-config-cluster-cmd --sharedstate \
                         --type=raw \
                         --rawprimary=path1 \
                         --rawshadow=path2
```

For example:

```
# sgicm-config-cluster-cmd --sharedstate --type=raw \
--rawprimary=/dev/shared1 --rawshadow=/dev/shared2
```

You should perform this step before defining the cluster ("Step 2: Create the Cluster" on page 46).

Do not modify the `/etc/init.d/clumanager` script.

Step 2: Create the Cluster

To create the cluster in the GUI, type the cluster name in the **Cluster Name** field in cluster configuration window. The default cluster name is `SGI High Availability cluster`.

In the CLI:

```
sgicm-config-cluster-cmd --cluster --name "clustername"
```

Step 3: Define the Members

To define a member in the GUI, do the following in the cluster configuration window:

- Click the **Members** tab.
- Click **New**.
- Enter the hostname of the new member. SGI recommends that member hostnames and addresses be present in `/etc/hosts` so that communication between cluster daemons does not rely on DNS or NIS being available.

In the CLI:

```
sgicm-config-cluster-cmd --add_member --name=membername
```

Step 4: Add Power Controller Configuration

For each member, you must provide information about its power controller. The SGI Cluster Manager supports SGI controllers using L2 with either serial cables or Ethernet cables for SGI Altix systems (do not confuse the `l2network` L2 Ethernet connection with network-based power controllers).

Note: The **Serial** power controller shown in the GUI refers to third-party products that are not supported by SGI Cluster Manager. (Although the other machine in the cluster will appear in the **Owner** field automatically, this value is not used.)

In the GUI **Cluster Configuration** window, select the member and click **Add Child**. The fields are as follows:

- SGI controllers:
  - **Type**: the power controller type of the member being defined (the local member):
    - `l2` for using L2 serial cables.

- `l2network` for using the L2 Ethernet connection. (This is the default in the GUI.)

Note: If you have a system with an emulated L2 controller (such as an Altix 3700 Bx2), or if you run CXFS with SGI Cluster Manger, you must use the `l2network` connection type. See "`l2network` Ethernet Connection" on page 25.

This is the `type` argument in the CLI; in the CLI, there is no default value.

– **Peer's TTY device file name**: the `tty` device filename on the **peer member** to which the local system controller is connected.

  This is the `device` argument in the CLI.

– **Altix partition**: the local member's system partition ID. If there are no partitions, partition ID is `0`. The default value in the GUI is `0`.

  This is the `partion` argument in the CLI.

Figure 5-2 shows an example for an L2 using the Ethernet network and Figure 5-3 shows an example in the GUI for an L2 using serial cables.

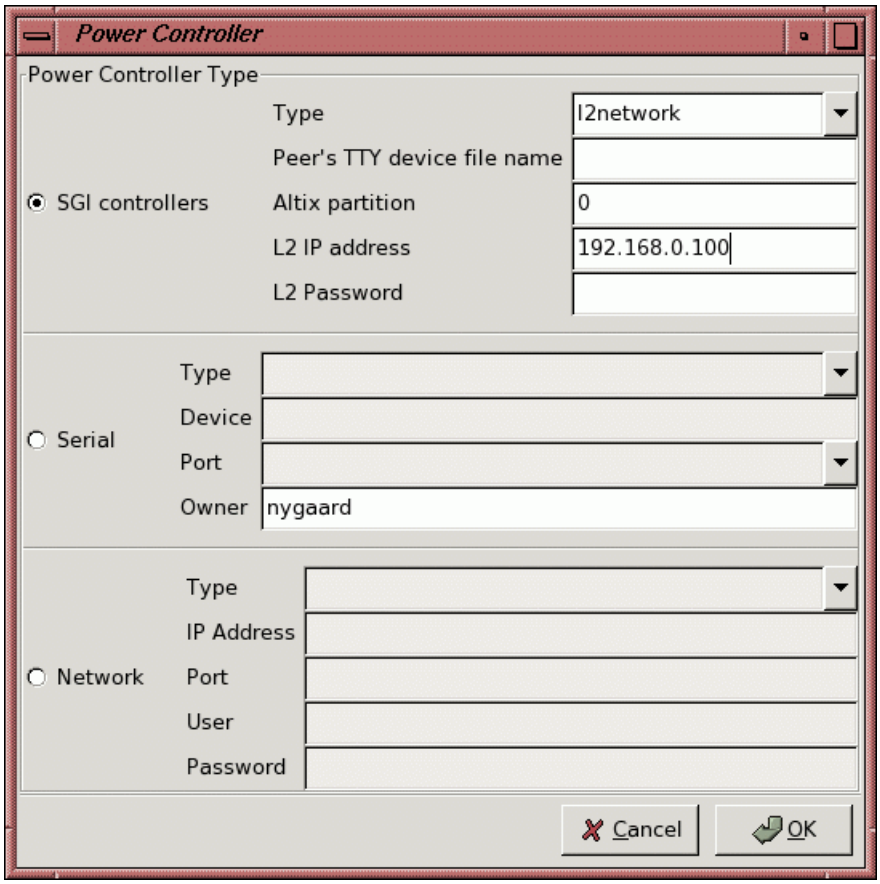

Figure 5-2 Power Controller Information for an L2 Using an Ethernet Network

---

Note: You can use a hostname in place of the IP address when configuring `l2network` reset provided that name resolution is in place; that is, the name is in `/etc/hosts` on all of the servers or is otherwise available via `gethostbyname`(2).

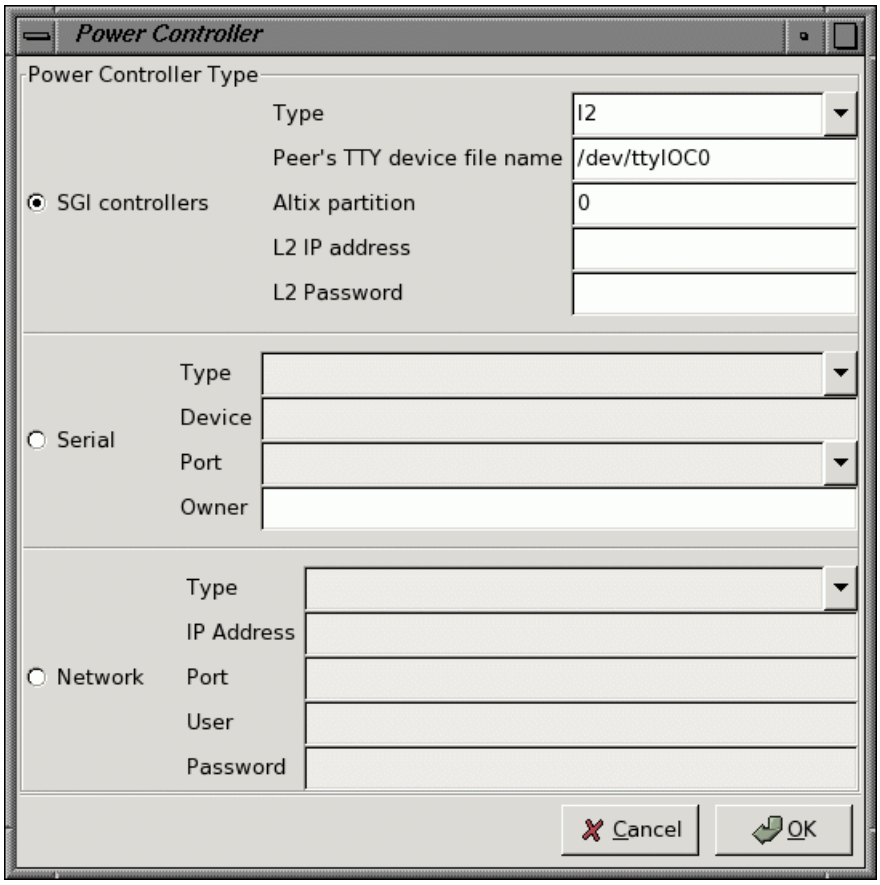

Figure 5-3 Power Controller Information for an L2 Using Serial Cables

In the CLI:

```
sgicm-config-cluster-cmd --member=membername \
                         --add_powercontroller \
                         --type=l2|l2network \
                               arguments for l2network:
                                   --ipaddress=L2_IPaddress_or_hostname \
                                   --password=L2_password_(if_defined) \
                                   --partition=Altix_partition_ID \
                          --device=/dev/ttyIOCx\
                          --partition=n
```

You can optionally set a password for the L2 to prevent unauthorized access to L2 functions via Ethernet. If you choose to use this security feature, SGI Cluster Manager must know the password in order to access L2 functionality. For more information, see *SGI L1 and L2 Controller Software User's Guide*.

For example, the following defines an L2 using the Ethernet method (therefore there is no `--device` argument):

```
# sgicm-config-cluster-cmd --member=member1  --add_powercontroller \
--type=l2network --ipaddress=192.168.0.100 --partition=3 --password=foo
```

For example, the following defines an L2 using the serial cable method:

```
# sgicm-config-cluster-cmd --member=member1  --add_powercontroller \
--type=l2 --device=/dev/ttyIOC0
```

For more information, see Chapter 3, "Power Control" on page 25.

Step 5: Change the Heartbeat Interval, Timeout, and Failover Speed

You can modify the time it takes to detect a member failure, known as the *failover speed*.

Note: The default failover speed differs depending upon which tool (GUI or CLI) you use to define the cluster. You cannot change the value for failover speed while the cluster daemons are running.

Failover Speed and the GUI

In the GUI, you can supply the failover speed directly:

1. In the **Cluster Configuration** window, select the following:

   Cluster
   > Daemon properties

2. Select the **clumembd** tab
3. Use the sliding bar to adjust failover speed, as shown in Figure 5-4. The GUI provides 15 seconds as the default failover speed value.
4. You can choose to enable either broadcast heartbeating or multicast heartbeating.

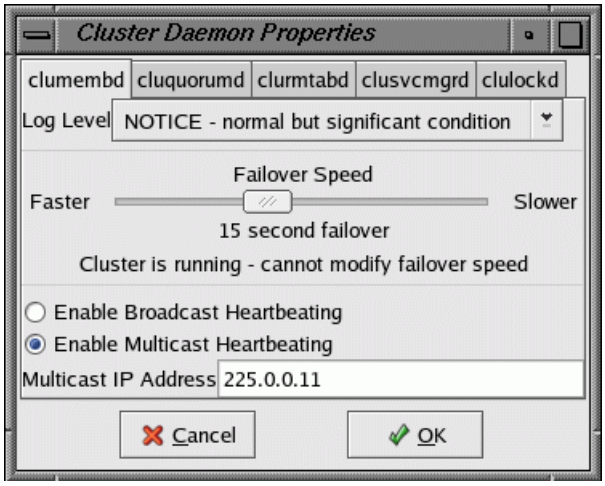

Figure 5-4 Adjusting Failover Speed

Failover Speed and the CLI

The `clumembd` daemon lets you specify the failover speed indirectly by defining the heartbeat interval and the timeout, from which the failover speed is automatically calculated:

- `interval` specifies the *heartbeat interval*, which is the number of microseconds before a heartbeat is sent to all other members in the cluster. The default value is 500000 (0.5 seconds).
- `tko_count` specifies the *heartbeat timeout*, which is the number of heartbeats missed before a member is declared as failed. The default value is 20.

Note: The GUI does not let you display or set the heartbeat interval or the heartbeat timeout individually.

The failover speed is calculated as follows:

interval_value `*` *tko_count_value* `=` *failover_speed*

Therefore, the default member failure detection time is 10 seconds (0.5 * 20 = 10).

Table 5-1 shows the failure detection times and parameter values that are supported.

Table 5-1 Supported Failure Detection Times and Parameter Values

Failover Speed (in seconds)	`interval` (in microseconds)	`tko_count`
30	`1000000`	`30`
25	`1000000`	`25`
20	`1000000`	`20`
15	`750000`	`20`
10	`500000`	`20`
5	`330000`	`15`

For example, the following command displays the heartbeat `interval` and `tko_count` values:

```
# sgicm-config-cluster-cmd --clumembd

clumembd:
  loglevel = 5
  interval = 500000
  tko_count = 20
  thread = yes
  broadcast = no
  multicast = yes
  multicast_ipaddress = 225.0.0.11
```

The failover speed is therefore 10 seconds. The following command changes the failover speed 15 seconds:

```
# sgicm-config-cluster-cmd --clumembd --interval=750000 --tko_count=20
```

Note: You cannot change the values for `interval` and `tko_count` while the cluster daemons are running.

For more information about using the command-line interface, see `sgicm-config-cluster-cmd` man page.

Step 6: Set the Tiebreakers

There are two types of tiebreakers:

- *Network tiebreaker* is used to avoid a *split-brain scenario,* in which two members attempt to form individual clusters. The network tiebreaker ensures that only the member that can contact the tiebreaker IP address is able to form a cluster. The network tiebreaker is the IP address of a machine or a router that **does not participate** in the cluster. Usually, it is the IP address of a network router that connects the members to the external world (clients). For clusters with more than two members, you must use a network tiebreaker.

Note: You must verify that the network tiebreaker can be accessed by the `ping`(1) command. (Some sites like to disable internet control message protocols at routers so the router or machines more than one hop away do not answer; such a router or machine could not be used as a tiebreaker.)

- *Disk tiebreaker*: If two members cannot talk to each other, they look at the status on the shared quorum partition disk to decide which member should survive and be part of the cluster membership. If the disk cannot be accessed or membership on the disk does not include a given machine, all SGI Cluster Manager processes on the machine exit. You can specify the number of seconds between the updates to the on-disk status. In the GUI, the default is 2 seconds.

In the GUI:

1. Select the following in the **Cluster Configuration** window:

   Cluster
   > Daemon properties

2. Select the **cluquorumd** tab.

3. Specify the desired values for the tiebreakers.

Figure 5-5 shows an example of the **cluquorumd** window.

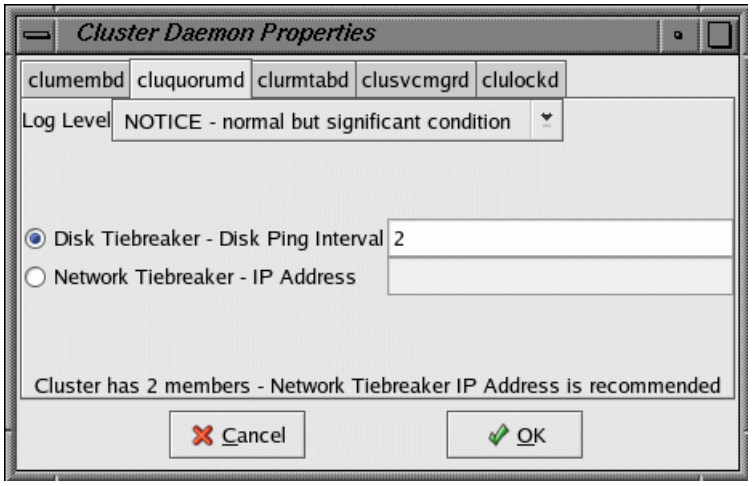

Figure 5-5 Tiebreakers

In the CLI:

```
sgicm-config-cluster-cmd --cluquorumd \
                         --tiebreaker_ip=IPaddress \
                         --pinginterval=seconds
```

Step 7: Create the Failover Domain

The failover domain is optional; if a failover domain is not defined, the service will be started on any member. For more information, see "Failover Domains" on page 7.

In the GUI **Cluster Configuration** window:

1. Select the **Failover Domains** tab.
2. Click **New**.
3. Enter the domain name and choose the desired failover and failback options.
4. Click **OK** to create the domain.

For information about the failover and failback options, see "Failover Domains" on page 7.

Figure 5-6 shows an example.

Figure 5-6 Failover Domain

In the CLI:

```
sgicm-config-cluster-cmd --add_failoverdomain \
                         --name=domainname   \
                         --restricted=yes|no  \
                         --ordered=yes|no  \
                         --controlled=yes|no

sgicm-config-cluster-cmd --failoverdomain=domainname  \
                         --add_failoverdomainnode \
                         --name=membername
```

The default for `--restricted`, `--ordered`, and `--controlled` is no.

Step 8: Configure the Service

You can specify the following for a service (the service must be disabled in order to configure it):

- Service name.

  Note: If you are using the GUI, you cannot include white space within a service name.

- Failover domain name (see "Step 7: Create the Failover Domain" on page 56).
- Monitor interval (in seconds).
- Service timeout (in seconds), which is common for all actions (start, stop, and status check) that apply to the service. A service timeout of `0` means that there is no timeout (the service action will never timeout).

  Note: You cannot specify individual timeouts for each resource within the service nor for each action (stop/start/monitor).

- Monitor level (for NFS and Samba only):
  - `Check for processes`

    NFS checks for `nfsd` processes.

    Samba checks for `smb` and `nmb` processes.
  - `Check as client`

    NFS sends null RPCs to the NFS server.

    Samba sends `smb` and `nmb` queries to the samba server.
- Restart count limit, which is the number of local restarts allowed for a service. When the limit is exceeded, the service is failed over to the other member. If there are no monitor failures for a day, the number of restart failures is reinitialized to 0. The maximum is 500.
- User application script or directory, if applicable. (If you are configuring NFS or Samba services, it is not necessary to put anything in this field.)

In this field, you can specify an individual script or a directory containing scripts. A script contains functions to implement service failover. The directory or script is specified as a service parameter.

Each function will be called with two parameters:

– An action: one of `start`, `stop`, or `status`

– A service ID

If successful, the function must return 0; if it fails, it must return a non-zero value.

For an example script, see "Sample User Application Script" on page 92.

In the GUI **Cluster Configuration** window:

1. Select the **Services** tab.
2. Click **New**.
3. Enter the desired values.
4. Click **OK** to create the service.

Figure 5-7 shows an example of configuring an NFS high-availability service.

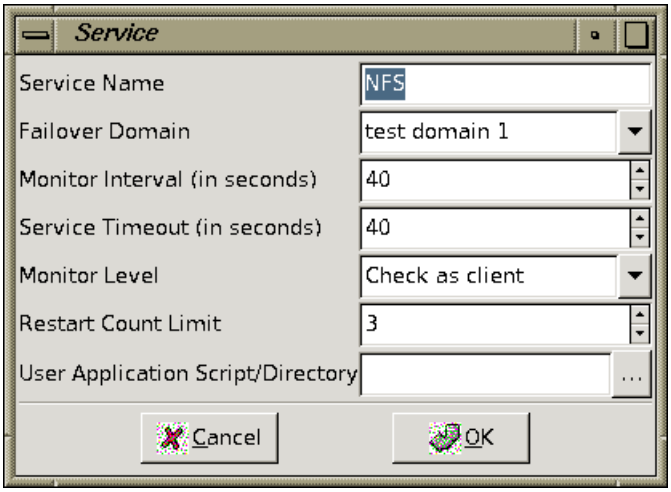

Figure 5-7 Configuring a High-Availability Service

In the CLI:

```
sgicm-config-cluster-cmd --add_service \
                         --name=servicename \
                         --failoverdomain=domainname \
                         --checkinterval=seconds \
                         --servicetimeout=seconds \
                         --monitorlevel="level" \
                         --restartcount=N \
                         --userscript=pathname
```

Note: The monitoring-level string values are case-sensitive and should be either of the following:

```
"Check for processes"
"Check as client"
```

Step 9: Add a Service IP Address

In the GUI **Cluster Configuration** window:

1. Select the **Services** tab.
2. Select the service name.
3. Click **Add Child**.
4. Choose **Add service IP address** and click **OK**.
5. Enter the IP address and optional netmask and broadcast address.
6. Click **OK**.

In the CLI:

```
sgicm-config-cluster-cmd --service=servicename \
                         --add_service_ipaddress \
                         --ipaddress=IPaddress \
                         --netmask=netaddress \
                         --broadcast=broadcastaddress
```

Step 10: Add the Disk and Filesystem Information to the Service *(Optional)*

In the GUI **Cluster Configuration** window:

1. Select the **Services** tab.
2. Select the service name.
3. Click **Add Child**.
4. Choose **Add Device** and click **OK**.
5. Enter information for the following, as appropriate:
   - Device special filename.
   - Samba share name.
   - Local XVM physical volumes (physvols). This must be a comma-separated list.
   - Mount point. If you are configuring a filesystem that requires the `dmi` mount option and are using local XVM, you must specify the mount point as follows:

     `mtpt=`*mountpoint*

     For example, if the mount point is `/dmfs`:

     ```
     mtpt=/dmfs
     ```

     When using CXFS, the mount point options are specified using CXFS tools.
   - Filesystem type `xfs` or `cxfs` (if using the CXFS plug-in). Default is `xfs`.

     Note: When configuring local XVM in the GUI, you must own the physvol for the XVM volume so that you may see the block device file for the volume in `/dev/lxvm/`.
   - Mount options

   Enable **Force Unmount**.
6. Click **OK**.

In the CLI:

```
sgicm-config-cluster-cmd --service=servicename \
                         --add_device \
                         --name=path

sgicm-config-cluster-cmd --service=servicename \
                         --device=path \
                         --mount   \
                         --mountpoint=mountpoint \
                         --fstype=xfs|cxfs \
                         --options=mountoptions \
                         --forceunmount=yes
```

Step 11: Add a Samba Share *(Optional)*

Samba share names must be unique within the cluster.

The **Samba Druid** is a configuration guide that lets you create a new Samba service or add Samba to an existing service. In the GUI **Cluster Configuration** window, start the **Samba Druid** by selecting the following:

Add Exports
> Samba

For more information, see "Samba Druid Example" on page 66.

In the CLI:

```
sgicm-config-cluster-cmd --service=servicename \
                         --device=path \
                         --sharename=sharename
```

Step 12: Define the NFS Information *(Optional)*

Define the NFS export point and NFS client information.

The **NFS Druid** is a configuration guide that lets you create a new NFS service or add NFS export points to an existing service. In the GUI **Cluster Configuration** window, start the **NFS Druid** by selecting the following:

Add Exports
> NFS

For more information, see "NFS Druid Example" on page 71.

In the CLI:

```
sgicm-config-cluster-cmd --service=servicename \
                         --device=path \
                         --add_nfsexport \
                         --name=exportdirectory

sgicm-config-cluster-cmd --service=servicename \
                         --device=path \
                         --nfsexport=exportpath \
                         --add_client \
                         --name=\* \
                         --options=options
```

The value * for the NFS client name means "all NFS clients." For better security, supply a list of NFS client systems instead of the * character. For more information, see the `exports`(5) man page.

Note: In general, you must use the `fsid` option to set the `fsid` value for the export. Any number in the range 1 through 65535 will work. See the `exports`(5) man page for further details on the `fsid` option.

Step 13: Save the Cluster Configuration *(GUI only)*

If you are using the GUI, you must explicitly save the configuration information as noted in "Cluster Configuration Tools" on page 41. Select the following from the **Cluster Configuration** window:

File
> Save

Step 14: Synchronize Configuration Changes Across the Cluster

During the initial configuration, you must manually copy the `/etc/cluster.xml` file to the other member in the cluster whether you use the GUI or the CLI.

Step 15: Verify that Configuration Changes are Synchronized

Each member has an `/etc/cluster.xml` file that contains cluster configuration information. If you make a change to this file on one member, you must copy the file to the other member using a command such as `scp`(1).

After making configuration changes, you must verify that the configuration files across the cluster are in synchronization. To do this, you can run the following command on each node and compare the `config_viewnumber` value on each, which lists the configuration file version number:

```
sgicm-config-cluster-cmd --cluster
```

The `config_viewnumber` value is updated each time a change is made to the configuration file.

For example, the following output from Machine1 and Machine2 shows that the configuration files are in synchronization for `test-cluster` because they both have the same `config_viewnumber` value (`10`):

- Machine1:

```
Machine1# sgicm-config-cluster-cmd  --cluster
cluster:
  name = test-cluster
  config_viewnumber = 10
```

- Machine2:

```
Machine2# sgicm-config-cluster-cmd  --cluster
cluster:
  name = test-cluster
  config_viewnumber = 10
```

In another example, the following output from Machine1 and Machine2 shows that the configuration files are out of synchronization because they have different `config_viewnumber` values (`10` and `11`):

- Machine1:

```
Machine1# sgicm-config-cluster-cmd  --cluster
cluster:
  name = test-cluster
  config_viewnumber = 10
```

- Machine2:

```
Machine2# sgicm-config-cluster-cmd  --cluster
cluster:
  name = test-cluster
  config_viewnumber = 11
```

If the `config_viewnumber` values are different, then configuration files are different. You should copy the configuration file with higher `config_viewnumber` number (which indicates the more recent configuration file) to the other member. In this case, you would copy the configuration file from Machine2 (which has the higher number of `11`) to Machine1 (which has the lower number of `10`).

Step 16: Start the Cluster Daemons

To automatically restart the SGI Cluster Manager daemons after a reboot, do the following in the CLI:

1. Enter the following command:

```
# chkconfig clumanager on
```

2. Start local cluster daemons on each member in the cluster by doing either of the following:

   - Enter `/etc/init.d/clumanager start`
   - In the GUI, select the following in the **Cluster Status** window:

     Cluster
     > Start Local Cluster Daemons

For more information, see Chapter 6, "Administration" on page 81.

Samba Druid Example

Figure 5-8 shows the **Samba Druid** initial window. Click **Forward** to configure the Samba service.

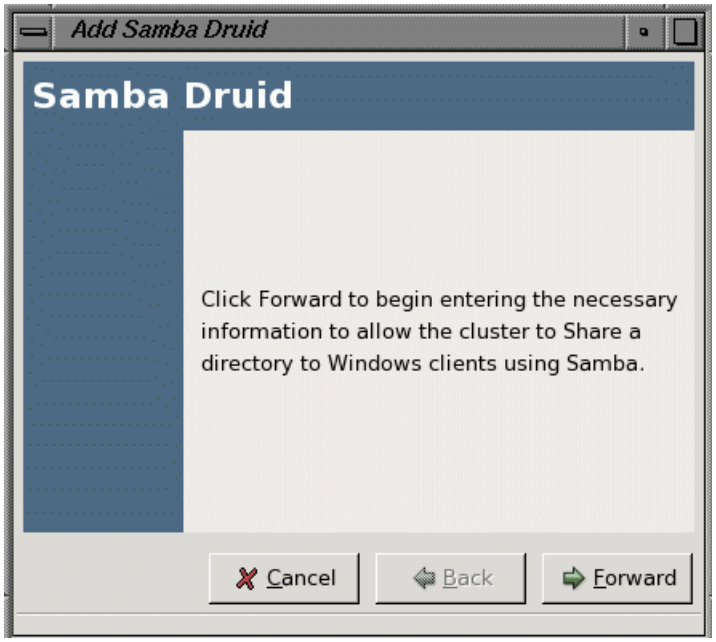

Figure 5-8 Samba Druid

You can choose to add Samba to an existing service or to create a new Samba service. In Figure 5-9, a new Samba service named `samba` with service IP address `192.168.0.3` is being created.

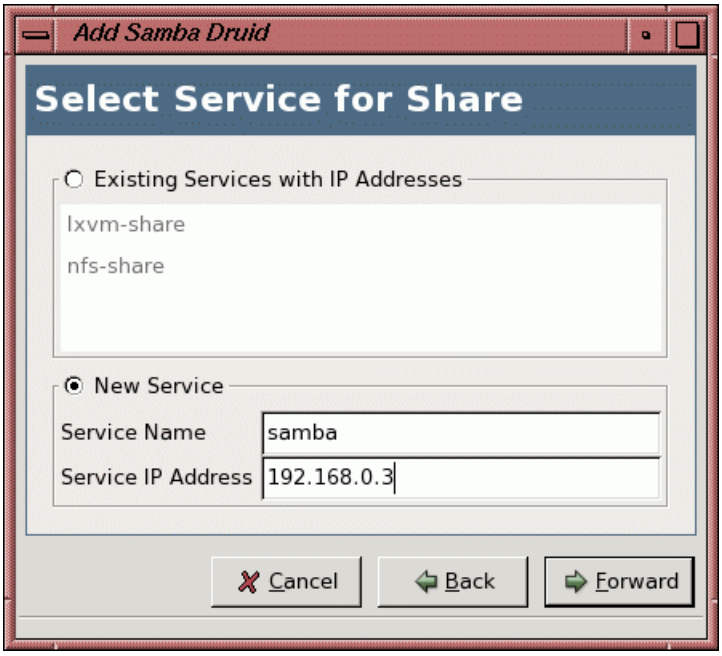

Figure 5-9 Samba Druid: Select Service for Share

You can choose the device and mount point of an existing service or add a new device and mount point. In Figure 5-10, a new device (`/dev/sdd`) and mount point (`/samba`) are being added. To change mount options, you must double-click on the device in the **Cluster Configuration** window after completing the **Samba Druid** configuration.

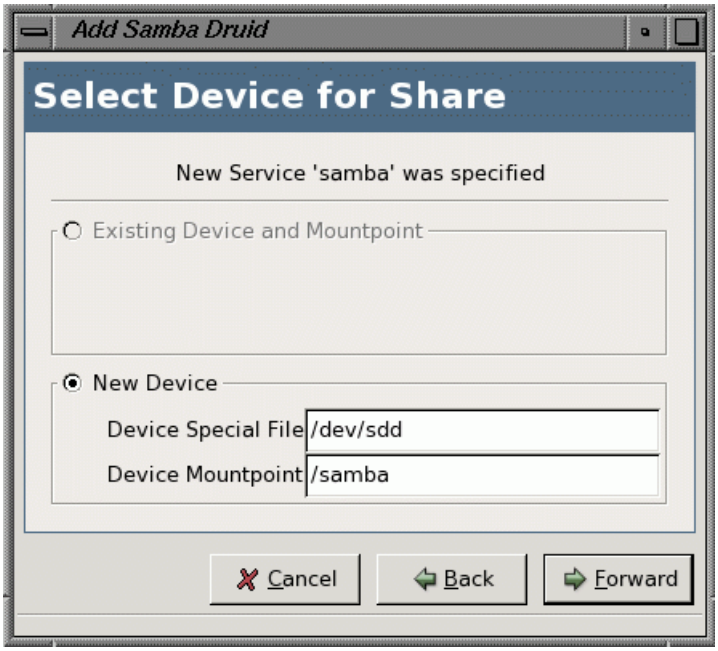

Figure 5-10 Samba Druid: Select Device for Share

In Figure 5-11, the name of the share is specified as `mysamba`. Only one share is configured at a time.

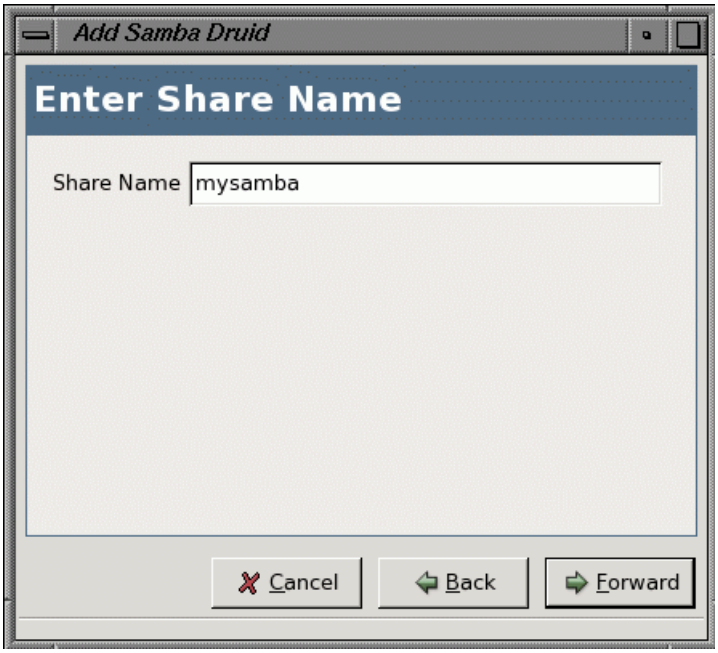

Figure 5-11 Samba Druid: Enter Share Name

Click **Apply** to complete the configuration of the Samba service. You must copy the `/etc/samba/smb.conf.mysamba` configuration file to the other member in the cluster. See Chapter 8, "Samba Plug-In" on page 95 for information about the newly created Samba configuration file.

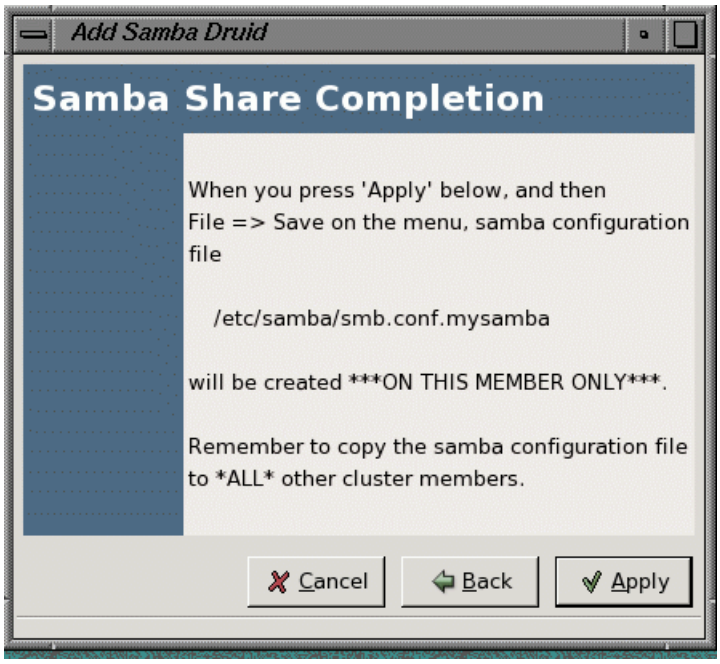

Figure 5-12 Samba Druid: Samba Share Completion

NFS Druid Example

Figure 5-13 shows the **NFS Druid** initial window. Click **Forward** to proceed.

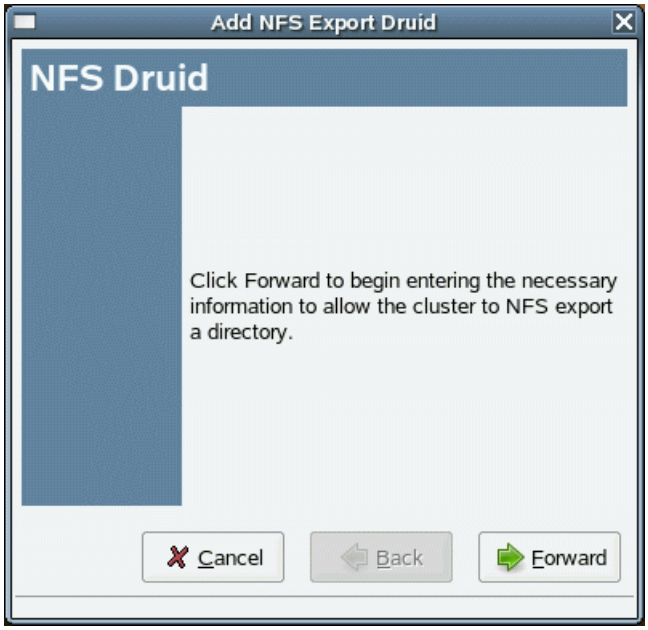

Figure 5-13 NFS Druid

Figure 5-14 shows the window that lets you enter the name of the export directory and its export options. You can add only one export directory at a time.

Note: In general, you must use the `fsid` option to set the `fsid` value for the export. Any number in the range 1 through 65535 will work. See the `exports`(5) man page for further details on the `fsid` option.

Figure 5-14 NFS Druid: Enter Directory to Export

You are given the choice of adding the export directory to an existing service or creating a new service for the export directory. In Figure 5-15, a new service `hanfs` is being created with service IP address `192.168.1.100`.

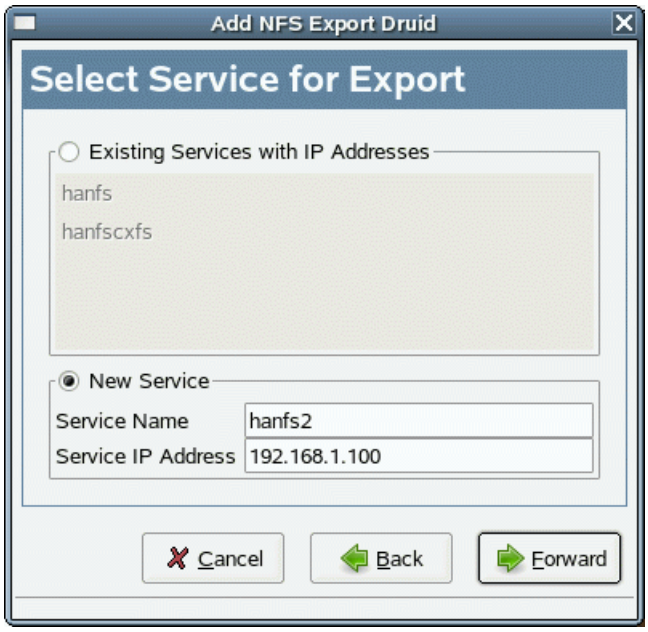

Figure 5-15 NFS Druid: Select Service for Export

You can add devices (filesystems) to the service. If you had chosen an existing service in the **Select Service for Export** window (Figure 5-15), you could choose an existing device mount point in the **Select Device for Export**. In Figure 5-16, a new device (`/dev/shared_xfs_2`) and mount point (`/share/xfs2`) are specified. To add filesystem mount options, you must double-click the device entry in the **Cluster Configuration** window after completing the NFS service configuration.

Figure 5-16 NFS Druid: Select Device for Export

Click **Apply** to complete the NFS configuration. If you want to modify service parameters, you must double-click the service in the **Cluster Configuration** window.

Figure 5-17 NFS Druid: NFS Export Completion

Cluster Configuration Example

The following example uses `sgicm-config-cluster-cmd` commands to create a two-member cluster with a service providing Samba shares and NFS service:

- `member1` is an Altix 350 system with no partitions that is connected to an L2 power controller
- `member2` is partition 3 of an Altix 3700 system that is connected to an L2 power controller using Ethernet, where `192.168.9.2` is the IP address of the L2 connected to `member2`
- The network tiebreaker is the IP address of a network router or another machine that determines which member should have connectivity to the public network

- `service1` is the IP address that will be used by clients to access the Samba share and NFS export point
- The service is allowed to restart four times within one day before a failover occurs

---

Note: Commands that modify the configuration file do not print anything if they are successful. The command exit status is `0` when successful.

Do the following:

1. Define shared state:

```
# sgicm-config-cluster-cmd --sharedstate --type=raw --rawprimary=/dev/shared_1 \
--rawshadow=/dev/shared_2
```

2. Create the cluster:

```
# sgicm-config-cluster-cmd --cluster --name "test-cluster"
```

3. Define the members:

```
# sgicm-config-cluster-cmd --add_member --name=member1 --watchdog=no

# sgicm-config-cluster-cmd --add_member --name=member2 --watchdog=no
```

4. Add power controller information for the members (`192.168.9.2` is the IP address of the L2):

```
# sgicm-config-cluster-cmd --member=member1 --add_powercontroller --type=l2 \
--device=/dev/ttyIOC0 --partition=0

# sgicm-config-cluster-cmd --member=member2 --add_powercontroller --type=l2network \
--ipaddress=192.168.9.2 --partition=3
```

5. Change the heartbeat timeout to 20 seconds with heartbeat interval of 1 second, resulting in a failover speed of 20 seconds:

```
# sgicm-config-cluster-cmd --clumembd --interval=1000000 --tko_count=20
```

6. Set up a network tiebreaker for the cluster:

```
# sgicm-config-cluster-cmd --cluquorumd --tiebreaker_ip=192.168.2.245
```

7. Create a failover domain with an ordered failover policy where the primary member is `member1` and the backup member is `member2`:

```
# sgicm-config-cluster-cmd --add_failoverdomain --name=domain1 \
--restricted=yes --ordered=yes

# sgicm-config-cluster-cmd --failoverdomain=domain1 --add_failoverdomainnode \
--name=member1

# sgicm-config-cluster-cmd --failoverdomain=domain1 --add_failoverdomainnode \
--name=member2
```

8. Create the service definition:

```
# sgicm-config-cluster-cmd --add_service --name=service1 --checkinterval=60  \
--servicetimeout=40 --monitorlevel="Check as client" \
--failoverdomain=domain1 --restartcount=4
```

9. Add a service IP address:

```
# sgicm-config-cluster-cmd --service=service1 --add_service_ipaddress \
--ipaddress=192.168.1.2 --netmask=255.255.255.0 \
--broadcast=192.168.1.255
```

10. Add the shared quorum partition and filesystem information to `service1`:

```
# sgicm-config-cluster-cmd --service=service1 --add_device --name=/dev/shared1

# sgicm-config-cluster-cmd --service=service1 --device=/dev/shared1 --mount  \
--mountpoint=/mnt1 --fstype=xfs --options=rw,sync \
--forceunmount=yes
```

11. Add a Samba share name:

```
# sgicm-config-cluster-cmd --service=service1 --device=/dev/shared1 \
--sharename=share1
```

12. Define the NFS export point and NFS client information. The directory is exported to all clients with read-only access:

```
# sgicm-config-cluster-cmd --service=service1 --device=/dev/shared1 \
--add_nfsexport --name=/shared1/export_dir

# sgicm-config-cluster-cmd --service=service1 --device=/dev/shared1 \
--nfsexport=/shared1/export_dir --add_client \
--name=\* --options=ro
```

Note: The value of * for the NFS client name means "all NFS clients." For better security, supply a list of NFS client systems instead of the * character. For more information, see the `exports`(5) man page.

13. (If you were using the GUI, you would have to save the configuration at this point.)

14. Synchronize the configuration changes. For example:

```
# scp /etc/cluster.xml root@member2:/etc/cluster.xml
root@member2's password:ENTER_ROOT_PASSWORD
cluster.xml                                   100% 3297    57.1MB/s   00:00
```

15. Verify that the changes are synchronized by running the following command on each member:

    ```
    # sgicm-config-cluster-cmd  --cluster
    ```

16. Start the SGI Cluster Manager daemons:

    a. Enter the following command:

    ```
    # chkconfig clumanager on
    ```

    b. Start local cluster daemons on each member in the cluster doing either of the following:

    ```
    # service clumanager start
    ```

    or

    ```
    # /etc/init.d/clumanager start
    ```

For more information and additional examples, see the `sgicm-config-cluster-cmd`(8) man page.

Administration

This chapter discusses the following:

Monitoring Status

To monitor status, use the following:

- The `sgicm-config-cluster` GUI to monitor the status of the cluster and the services
- `clustat` to monitor the cluster status

Figure 6-1 shows an example of the GUI.

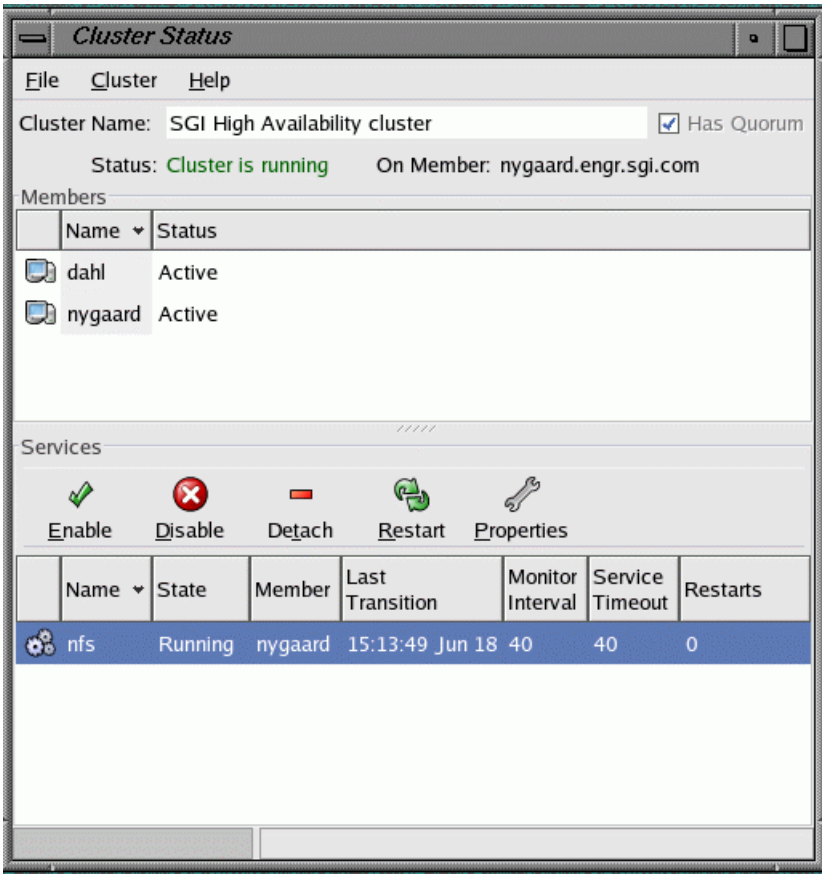

Figure 6-1 Status

Displaying Service Information

To display information about a service using the GUI, click on the service name in the **Cluster Status** window.

In the CLI:

```
sgicm-config-cluster-cmd --service=servicename
```

For example:

```
# sgicm-config-cluster-cmd --service=nfs
```

Figure 6-2 shows an example of the status window.

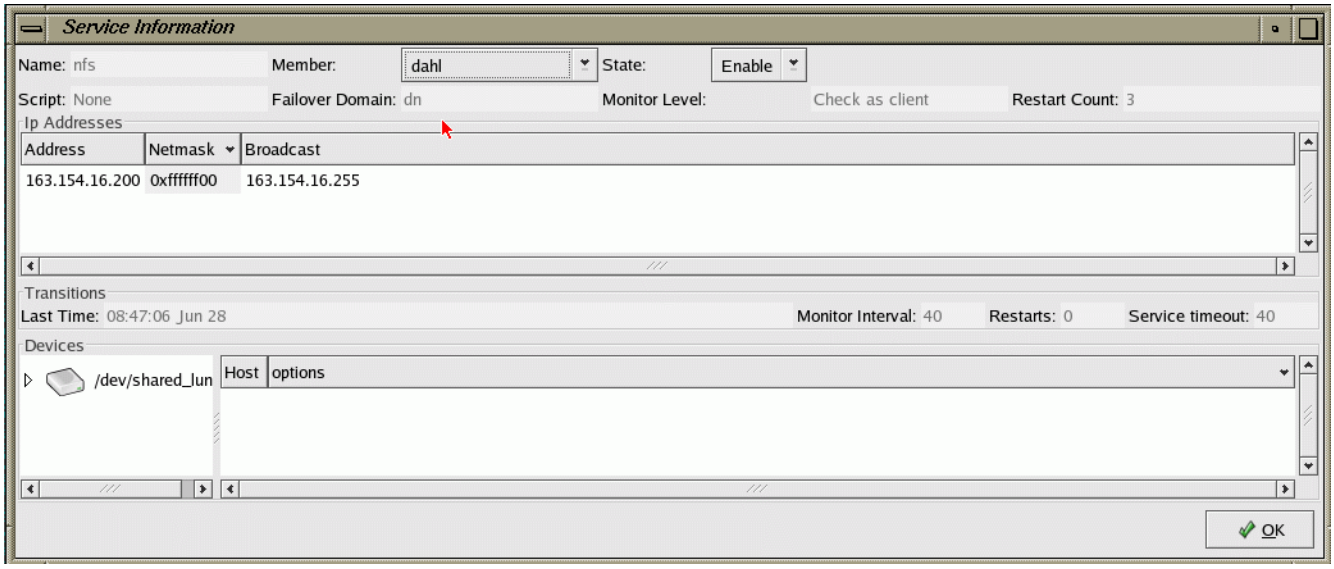

Figure 6-2 Service Information

Starting Cluster Processes

Use the following GUI selection in the **Cluster Status** window to start cluster daemons on the local member:

Cluster
> Start Local Cluster Daemons

In the CLI:

```
/etc/init.d/clumanager start
```

To start the daemons on other members, you must run the GUI or CLI on those other machines.

Stopping Cluster Processes

Use the following GUI selection in the **Cluster Status** window to stop cluster daemons on the local member:

Cluster
> Stop Local Cluster Daemons

In the CLI:

```
/etc/init.d/clumanager stop
```

To stop the daemons on other members, you must run the GUI or CLI on those other machines.

Service Administration

In the GUI, use the **Cluster Status** window to enable, disable, detach, restart, or stop services or to view service properties. You can also restart and relocate a service by using drag and drop on the service icon to the target node icon.

When you *enable* a service, you start it for the first time. The service will start on any member in the cluster based on the failover domain. When you *restart* the service, it restarts the service that was already running on the local node.

In a successful *detach* operation, the service is no longer monitored and is not part of the cluster, but continues to run on the member. (The difference between *detach* and *disable* is that the services are not stopped with a detach.)

You can also use the `clusvcadm` command as follows:

- Enable the service on the local member:

  `clusvcadm -e` *service*

- Enable the service on the specified member:

  `clusvcadm -e` *service* `-m` *member*

- Disable the service:

  `clusvcadm -d` *service*

- Detach the service:

  `clusvcadm -t` *service*

- Restart the service on the local member:

  `clusvcadm -R` *service*

- Relocate the service:

  `clusvcadm -r` *service* `-m` *member*

- Stop the service:

  `clusvcadm -s` *service*

To avoid seeing output, use the `-q` option.

Cluster Service States

A service can have one of the following states:

State	Description
Uninitialized	Transitioning when `clusvcmgrd` daemon starts
Pending	Transitioning to running or disabled
Running	Online and is being actively monitored
Disabled	Not online and service was stopped
Stopped	Disabled but will start when cluster processes are started again
Failed	Needs operator attention
Detached	Not online but the service was not stopped in the cluster

A detached service has no owner and last owner is the member where the service application is still running. The GUI and the `clustat` command display the last owner for services in the detached state, as shown in the Figure 6-3.

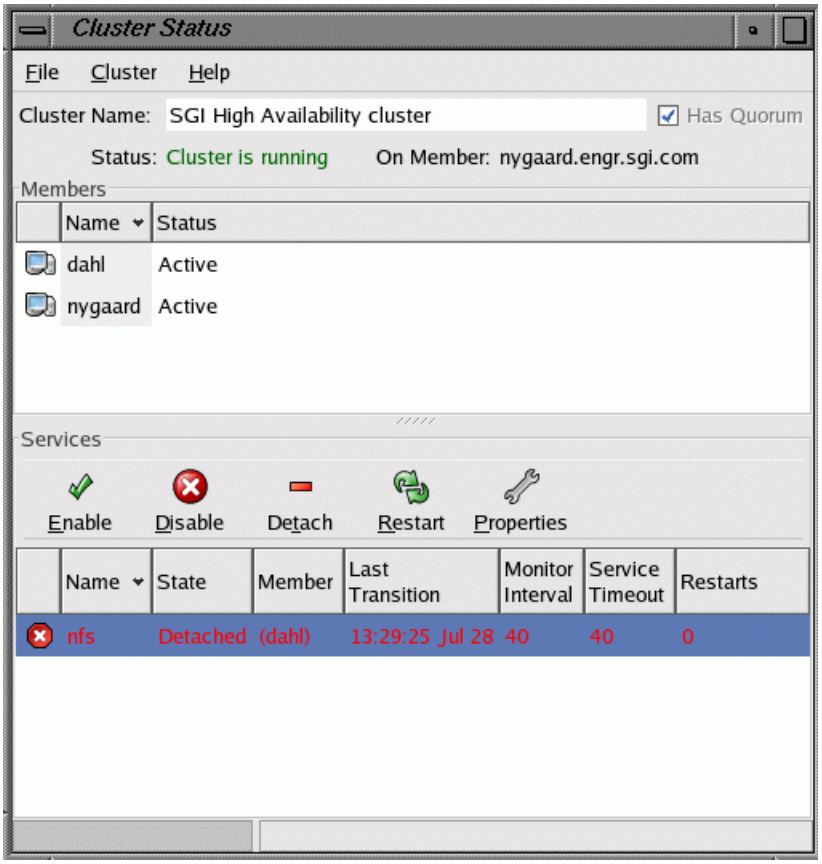

Figure 6-3 Detached State

To recover from detached state, you must disable the service and then enable it. When a disable action is performed on the service, the service's `stop` scripts are executed on the last owner. If you try to perform an enable or restart action on a service in the detached state, it will fail with the following error message:

`Service` *servicename* `is in detached state. Disable and then enable service.`

If the last owner of a service in detached state leaves cluster membership, or if the cluster daemons are stopped on the last owner of the service, the service will move to disabled state.

Caution: Although the service is in disabled state, the service application is still running on the last owner and is not stopped by SGI Cluster Manager. If you attempt to enable the service at this point, it will cause data integrity problems.

Message Logging

SGI Cluster Manager logs messages to `/var/log/messages` using the `syslog` facility `local4`. You can use `syslog.conf` to redirect messages to another location. To rotate logs, use `logrotate`(8).

SGI Cluster Manager uses the following message levels:

Level	Description
0	`EMERG` (emergency)
1	`ALERT`
2	`CRIT` (critical problem)
3	`ERROR`
4	`WARNING` (default)
5	`NOTICE`
6	`INFO` (informational)
7	`DEBUG`

Creating a New Highly Available Application

This chapter discusses the following:

The `clusvcmgrd` Daemon

All services in SGI Cluster Manager for Linux are managed by the `clusvcmgrd` daemon. The `clusvcmgrd` daemon does the following:

- Determines the cluster member where a service must run
- Processes service events
- Executes service scripts in a sequential manner

The `service` Script

The `service` script starts, stops, or determines the status of given service. The functions within the `service` script take the following parameters:

- An action, which can be one of `start`, `stop`, or `status`
- A *service ID* which is a number that identifies the service (the ID is automatically determined and is not user-configurable)

The functions within the `service` script run application scripts in the following order:

- `start` order:

  `device` *(including local XVM volumes)*
  `filesystem` *(including CXFS)*
  `nfs`
  `ip address`
  `samba`
  user-defined application (such as DMF and TMF applications)

- `stop` order:

  user-defined application (such as DMF and TMF applications)
  `ip address`
  `nfs`
  `samba`
  `filesystem` *(including CXFS)*
  `device` *(including local XVM volumes)*

You cannot change the order in which the application scripts are run.

The status of each application in a service is checked in a sequential manner. If the status of an application in the service fails, the status of other applications is not checked.

User application scripts are usually present in the `/usr/lib/clumanager/services` directory. These scripts return `$FAIL` (value `1`) on failure and `$SUCCESS` (value `0`) on success for each action.

Adding a Service

To add a new application, you must write a set of scripts that are specific to the user application. The user application script must be a `bash` shell script and should contain a shell function with a name that is the same as the application and should take an action (`start`, `stop`, or `status`) and a service ID as parameters. For example: a user application script for failing over an apache webserver could be called `apache_webserver` and it should have a shell function `apache` that takes an action and a service ID as parameters. The shell function will be called by the `service` script to execute appropriate action for a service.

The newly written script is configured as a user application script parameter. You must add all devices and IP addresses that the application depends on to the service. NFS export points and Samba shares can also be part of the service.

For general information about creating a service, see "Step 8: Configure the Service" on page 58.

Figure 7-1 shows the GUI screen to create a service. To get to this window, select the **Services** tab from the **Cluster Configuration** window.

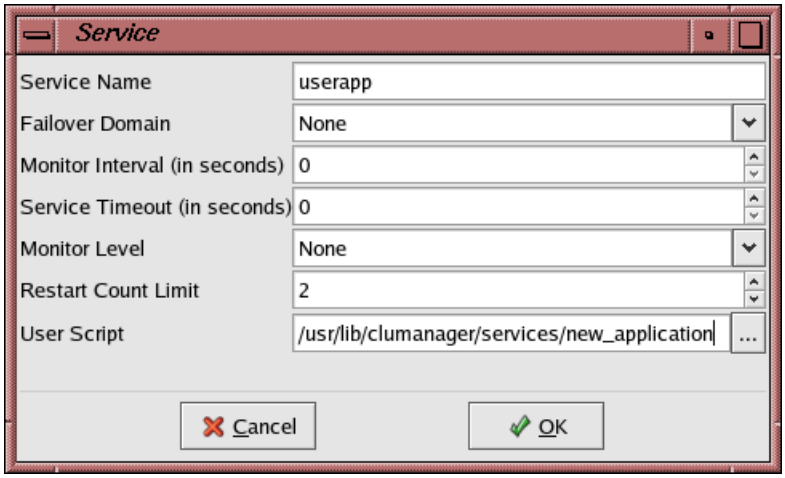

Figure 7-1 Creating a Service

The following command creates a service named `userapp` with the newly defined user script `new_application`:

```
# sgicm-config-cluster-cmd --add_service --name=userapp  \
--userscript=/usr/lib/clumanager/services/new_application   \
--checkinterval=40 servicetimeout=60
```

You must copy the newly created script to the following location in all members in the cluster:

`/usr/lib/clumanager/services/new_application`

Example of Failing Over Multiple User Applications

To fail over an apache webserver and mySQL database as part of a service, you must do the following:

- Create a directory for the service application scripts. For example:

  # **mkdir /usr/lib/clumanager/services/service1**

- Create a script within the directory for each application. For example, they could be named `apache_webserver` and `mySQL`:
  - `apache_webserver` should contain a shell function that takes an action and service ID as parameter. This function should perform start/stop/status operation on the apache server for `service1`.
  - `mySQL` should contain a shell function that performs start/stop/status operation on the mySQL database server for `service1`.

Note: The directory can contain symlinks to actual scripts that are present in some other directory.

Sample User Application Script

The following is an example user application script named `service_test`. This example script contains a shell function `test` that takes an action (`start`, `stop` or `status`) and service ID as parameters. The shell function `test` can call shell functions (as in the example) to perform the actions or to execute other scripts or commands to perform the action passed as parameters.

```
#
# startTest serviceID
#
startTest()
{

        if [ $# -ne 1 ]; then
          logAndPrint $LOG_ERR "Usage: startTest serviceID"
          return $FAIL
        fi
```

```
        typeset svcID=$1
        typeset svc_name=$(getSvcName $DB $svcID)

        logAndPrint $LOG_INFO "Running test start script for service $svc_name "

        return $SUCCESS
}

#
# stopTest serviceID
#
stopTest()
{

        if [ $# -ne 1 ]; then
          logAndPrint $LOG_ERR "Usage: stopTest serviceID"
          return $FAIL
        fi

        typeset svcID=$1
        typeset svc_name=$(getSvcName $DB $svcID)

        logAndPrint $LOG_INFO "Running test stop script for service $svc_name "
        return $SUCCESS
}

#
# statusTest serviceID
#
statusTest()
{

        if [ $# -ne 1 ]; then
          logAndPrint $LOG_ERR "Usage: statusTest serviceID"
          return $FAIL
        fi

        typeset svcID=$1
        typeset svc_name=$(getSvcName $DB $svcID)

        logAndPrint $LOG_INFO "Running test status script for service $svc_name "
```

```
        return $SUCCESS
}


# Given an action and service ID number run that action for that service.
test()
{
        if [ $# -ne 2 ]; then
          logAndPrint $LOG_ERR "Usage: test [start, stop, status] serviceID"
          return $FAIL
        fi

        typeset action=$1
        typeset svcID=$2

        case "$action" in
        'start')
          startTest $svcID
          return $?
          ;;

        'stop')
          stopTest $svcID
          return $?
          ;;

        'status')
          statusTest $svcID
          return $?
          ;;
        esac
}
```

Samba Plug-In

SGI Cluster Manager supports Samba as shipped with SGI ProPack for Linux. See "Software Requirements" on page 7 for the supported levels.

This chapter discusses the following:

For more information about Samba, see:

http://www.samba.org/samba/docs

Samba Process ID, Locks, and Password File

The Samba process ID (PID), locks, and password file are kept in the shared partitions and in the log file on the local disk. The lock directory is not removed during failover.

Note: The default location for these directories is not always the best one. For example, in a configuration that includes DMF, you do not want these directories to reside in a DMF-managed filesystem, although they must be in shared filesystems.

The default locations are as follows:

- PID directory: *mountpoint*`/.samba/`*sharename*`/pid`

  For example:

  ```
  pid directory = /mirror/.samba/SMBXVM/pid
  ```

- Lock directory: *mountpoint*`/.samba/`*sharename*`/locks`

  For example:

  ```
  lock directory = /mirror/.samba/SMBXVM/locks
  ```

- Log directory: `/var/log/samba`

  For example:

  ```
  log file = /var/log/samba/%m.log
  ```

- Password file: *mountpoint*`/.samba/`*sharename*`/private/smbpasswd`

  For example:

  ```
  smb passwd file = /mirror/.samba/SMBXVM/private/smbpasswd
  ```

See "Improving the Default `smb.conf.`*sharename* File" on page 97 and "Maintaining the `smb.conf.`*sharename* File on Shared Storage" on page 99.

Samba Share Configuration File

When you create a Samba service in SGI Cluster Manger, the `netbios name` entry is automatically added to the `/etc/samba/smb.conf.`*sharename* file with a value set to the IP name associated with highly available IP address configured for the Samba service. You can modify the `netbios name` value as long as it remains unique for the SGI Cluster Manager cluster.

When you create a Samba service in SGI Cluster Manger, the `smb passwd file` entry is automatically added to the `/etc/samba/smb.conf.`*sharename* file. However, when you use the `smbpasswd`(8) command, that entry is commented out and the `private dir` entry is added. For example:

```
#       smb passwd file = /lxvm2/.samba/samba_lxvm2/private/smbpasswd
        private dir = /lxvm2/.samba/samba_lxvm2/private
```

Samba Start/Stop Order

For the order in which Samba is started/stopped, see Chapter 7, "Creating a New Highly Available Application" on page 89.

Defining NFS Exports and Samba Exports

You can edit the `/etc/samba/smb.conf.`*sharename* Samba configuration file that is generated by SGI Cluster Manager to include additional Samba shares.

Whether or not you change this file, you must copy it to the other member in the cluster. For examples of using the Samba Druid and NFS Druid, see "Samba Druid Example" on page 66 and "NFS Druid Example" on page 71.

Improving the Default `smb.conf.`*sharename* File

If you name a share `SMBXVM`, the Samba Druid will create the following default `smb.conf.SMBXVM` file:

```
[global]
        workgroup = SGIHACLUSTER
        pid directory = /mirror/.samba/SMBXVM/pid
        lock directory = /mirror/.samba/SMBXVM/locks
        log file = /var/log/samba/%m.log
        netbios name = sauna
        smb passwd file = /mirror/.samba/SMBXVM/private/smbpasswd
        encrypt passwords = yes
        bind interfaces only = yes
        interfaces = 192.168.0.1/255.255.255.0

[SMBXVM]
        comment = High Availability Samba Service
        browseable = yes
        writable = no
        public = yes
        path = /mirror
```

You should modify this file as follows:

- Change the value for `workgroup`. It should be 8 or fewer characters for compatibility with older CIFS clients.
- Change the default `netbios name` value (which is determined by resolving the IP alias in the share) to uppercase in order to maintain compatibility with older CIFS clients.

- Ensure that the `smb passwd file` is not commented out after running the `smbpasswd`(8) command.
- Add the `private dir` parameter.
- Change the `writable` value to `yes` so that the share can be written to.
- Change the `log file` value to be on shared storage if you want the log files in the same place. (Otherwise, you must look on all members to see all of the logs.)
- Modify the default value of `max log size` to suit your site.

Following is an improved `smb.conf.SMBXVM` file after making changes to the default file created by the Samba Druid (bold indicates changed or added lines):

```
[global]
        workgroup = SGIHACL
        pid directory = /mirror/.samba/SMBXVM/pid
        lock directory = /mirror/.samba/SMBXVM/locks
        log file = /var/log/samba/%m.log
        netbios name = SAUNA
        smb passwd file = /mirror/.samba/SMBXVM/private/smbpasswd
        private dir = /mirror/.samba/SMBXVM/private
        encrypt passwords = yes
        bind interfaces only = yes
        interfaces = 192.168.0.1/255.255.255.0

[SMBXVM]
        comment = High Availability Samba Service
        browseable = yes
        writable = yes
        public = yes
        path = /mirror
```

If you name an NT domain `NT`, the Samba Druid will create the standard default `smb.conf.`*sharename* file (assuming that a machine account has already been set up that has never been bound to the domain). After making the appropriate changes, the file would for example be as follows (bold indicated changed or added lines):

```
[global]
        workgroup = MYDOMAIN
        security = domain
        wins server = 192.168.1.2
        pid directory = /mirror/.samba/SMBXVM/pid
```

```
        lock directory = /mirror/.samba/SMBXVM/locks
        log file = /var/log/samba/%m.log
        netbios name = SAUNA
        smb passwd file = /mirror/.samba/SMBXVM/private/smbpasswd
        private dir = /mirror/.samba/SMBXVM/private
        encrypt passwords = yes
        bind interfaces only = yes
        max log size = 0
        interfaces = 192.168.1.10/255.255.255.0


[SMBXVM]
        comment = High Availability Samba Service
        browseable = yes
        writable = yes
        public = yes
        path = /mirror
```

After Samba is running with this configuration, you must ensure that the `wins` server is set properly. Run the following:

```
# net join -s /etc/samba/smb.conf.SMBXVM
```

For more information, see the `smb.conf` man page.

Maintaining the `smb.conf.`*sharename* File on Shared Storage

To simplify maintenance, you might want to place the `smb.conf.`*sharename* file on shared storage using the following procedure:

1. Copy the `smb.conf.`*sharename* file to shared storage:

   ```
   # cp /etc/samba/smb.conf.sharename /shared_dir/samba/smb.conf.sharename
   ```

2. Remove the local `smb.conf.`*sharename* file:

   ```
   # rm -rf /etc/samba/smb.conf.sharename
   ```

3. On each member, create a symbolic link to the `smb.conf.`*sharename* file on shared storage:

   ```
   # ln -s /shared_dir/samba/smb.conf.sharename /etc/samba/smb.conf.sharename
   ```

Service Monitoring Levels

For information about service monitoring levels, see "Step 8: Configure the Service" on page 58.

Note: You cannot use a service monitor level set to `Check as client` if you use the Samba configuration parameter `browseable = no`. Doing so will result in immediate failovers (or restarts if they are configured) with each monitor pass because the `smbclient` check fails with `browseable = no`.

CXFS Plug-In

Using the CXFS clustered filesystem product with SGI Cluster Manager for Linux requires the value-add SGI product on the *SGI Cluster Manager 4.3 for Linux - Storage Software Plug-ins* CD and the supported level of CXFS (see "Software Requirements" on page 7).

You should configure the CXFS cluster, nodes, and filesystems according to *CXFS Administration Guide for SGI InfiniteStorage*.

You must give careful consideration when choosing nodes used for the SGI Cluster Manager cluster and nodes capable of being the CXFS metadata server. Certain services, such as DMF, require that the CXFS metadata server and the SGI Cluster Manager service be provided by the same node (see "Configuring DMF-Managed XFS Filesystems as CXFS Filesystems" on page 107). Others, such as NFS and Samba, do not have this requirement, but it may be desirable to enforce it in order to ensure that these services provide the best performance possible. (Some NFS and Samba workloads can cause significant performance problems when the NFS or Samba service is provided from a node that is not also the CXFS metadata server.) Unless a service must run on the CXFS metadata server, you should configure SGI Cluster Manager so that it does not relocate the CXFS metadata server when it fails over a service.

This chapter discusses the following:

Relocation Support

CXFS relocation is provided automatically by the CXFS plug-in. There is no need to modify the `cxfs_relocation_ok` parameter on the CXFS metadata servers.

Members and I/O Fencing

For I/O fencing, SGI Cluster Manager members should use `l2network` power controllers in order to prevent conflicts with CXFS I/O fencing methods. For more information, see "L2 Connections" on page 25.

Including a CXFS Filesystem in the Cluster Configuration

To include CXFS filesystems in the SGI Cluster Manager configuration, add filesystems as devices used by a service, as shown in Figure 9-1.

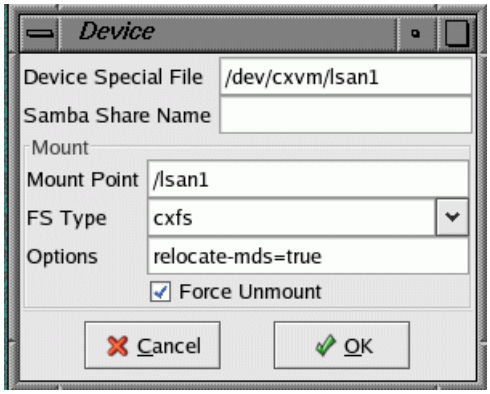

Figure 9-1 Adding a CXFS Filesystem as a Device

Enter the following:

- **Device Special File**: the block XVM device file
- **Mount Point**: the CXFS filesystem mount point
- **FS Type**: the filesystem type must be `cxfs`
- **Options**: one of the following:
  - `relocate-mds=true`, which allows the metadata server for the CXFS filesystem to be failed over when the service is failed over
  - `relocate-mds=false` (default)

Note: The **Force Unmount** item in the GUI and CLI is ignored for CXFS filesystems.

In the CLI, do the following:

```
sgicm-config-cluster-cmd --service=servicename \
                         --add_device \
                         --name=/dev/cxvm/volumename

sgicm-config-cluster-cmd --service=servicename \
                         --device=/dev/cxvm/volumename \
                         --mount \
                         --mountpoint=mountpoint \
                         --fstype=cxfs \
                         --options=relocate-mds=true|false
```

You can specify multiple CXFS filesystems by adding multiple devices to the service.

Note: You will not define a `--userscript` value when defining a service to failover CXFS filesystems. User scripts are used for failing over user-written applications. For more information on user scripts, see Chapter 7, "Creating a New Highly Available Application" on page 89.

For more information, see "Step 10: Add the Disk and Filesystem Information to the Service *(Optional)*" on page 61.

Members and Potential Metadata Servers

The members in the failover domain for the service that has CXFS filesystems should be same as the list of potential metadata servers for the CXFS filesystem. For example: machines `node1` and `node2` can be metadata servers for CXFS filesystem `/cxfs_san1`. The SGI Cluster Manager service `nfs1` that uses `/cxfs_san1` should have a failover domain of `node1` and `node2`.

CXFS Start/Stop Order

You should start CXFS cluster services and CXFS services before starting SGI Cluster Manager daemons. SGI Cluster Manager will wait for the CXFS filesystem to be mounted by CXFS before starting NFS, Samba, and other applications running on the CXFS filesystem. Therefore, service timeouts for all SGI Cluster Manager services that include CXFS filesystems should be carefully adjusted accordingly.

For the order in which CXFS is started/stopped, see Chapter 7, "Creating a New Highly Available Application" on page 89.

Data Migration Facility (DMF) Plug-In

Using the Data Migration Facility (DMF) with SGI Cluster Manager for Linux requires the value-add SGI product on the *SGI Cluster Manager 4.3 for Linux — Storage Software Plug-ins* CD and the supported level of DMF (see "Software Requirements" on page 7).

You should configure DMF according to *DMF Administrator's Guide for SGI InfiniteStorage*.

This chapter discusses the following:

Adding the DMF User Script to an Existing Service

The following command adds the DMF user script to an existing service. The script used is `/usr/lib/clumanager/services/svclib_dmf`:

```
# sgicm-config-cluster-cmd --service=service1   \
 --userscript=/usr/lib/clumanager/services/svclib_dmf
```

You could also add the script by modifying the service in the GUI. For more information, see "Step 8: Configure the Service" on page 58.

DMF Administrative Filesystems and Directories

To run DMF, you must configure the parameters shown in Table 10-1. A *required* parameter must be defined by all users of DMF. An *optional* parameter is needed only by users of certain MSPs or the library server. DMF cannot start unless the required filesystems and directories defined by these parameters are first mounted and available on shared disks.

Table 10-1 DMF Administrative Filesystem and Directory Parameters

Parameter	Status	Description
`HOME_DIR`	Required	Specifies the DMF databases
`JOURNAL_DIR`	Required	Specifies the DMF database journals
`SPOOL_DIR`	Required	Specifies the DMF log files
`MOVE_FS`	Optional	Moves files between MSPs
`CACHE_DIR`	Optional	Used by the library server as a cache for merging data from sparse tapes to new tapes
`FTP_DIRECTORY`	Optional	Used by the FTP MSP to store files
`STORE_DIRECTORY`	Optional	Used by the disk MSP to store files

In addition, the working directory used by the `dmaudit`(8) command must also be available when DMF starts. To configure the directory, run the `dmaudit` command and select the `<workdir>` item in the `<config>` menu.

You can configure the DMF administrative filesystems (also known as *support filesystems*) as local XVM filesystems. You must define them as instructed in "Configuring DMF for Local XVM Filesystems" on page 107. SGI Cluster Manager ensures that the DMF plug–in script is called after the necessary filesystems are mounted.

Note: You should only configure DMF administrative filesystems as CXFS filesystems if they are also using DMAPI.

To provide the best chance for database recovery, you should place the *JOURNAL_DIR* directory on a separate filesystem and a different physical device from the *HOME_DIR* directory.

If you use CXFS filesystems, you must define them as instructed in "Configuring DMF-Managed XFS Filesystems as CXFS Filesystems" on page 107.

Configuring DMF for Local XVM Filesystems

To configure the DMF administrative filesystems as local XVM filesystems, do the following:

1. Ensure that the DMF configuration is identical on all members.
2. Create the DMF administrative filesystems on shared disks as local XVM filesystems (`xvm` type). See "Step 10: Add the Disk and Filesystem Information to the Service *(Optional)*" on page 61.
3. Configure the SGI Cluster Manager local XVM volumes using the local XVM plug-in. See Chapter 12, "Local XVM Plug-In" on page 121.

Configuring DMF-Managed XFS Filesystems as CXFS Filesystems

SGI recommends that you configure DMF administrative filesystems as local XVM filesystems, as discussed in "Configuring DMF for Local XVM Filesystems" on page 107. DMF cannot start until the DMF administrative filesystems are available. If they are CXFS filesystems, CXFS must recover them before they are accessible.

To configure DMF-managed XFS filesystems as CXFS filesystems, do the following:

1. Ensure that the DMF configuration is identical on all members.
2. Create the DMF administrative filesystems as CXFS filesystems (`cxfs` type). See "Step 10: Add the Disk and Filesystem Information to the Service *(Optional)*" on page 61.

   Note: The optional `MOVE_FS` and `CACHE_DIR` DMF administrative filesystems require the `dmi` mount option. They should not be CXFS filesystems because they are only required on the node running `dmdaemon`.

   ---

3. Configure the SGI Cluster Manager CXFS filesystems using the CXFS plug-in. For DMF-managed filesystems, configure `relocate-mds=true` (on) because DMF must run on the CXFS metadata server for that filesystem. See Chapter 9, "CXFS Plug-In" on page 101.

The /etc/dmf/sgicm_dmf.config File

The /etc/dmf/sgicm_dmf.config file lets you configure other information required by SGI Cluster Manager. The sgicm_dmf.config file exists on all members in the cluster and should be edited as necessary on each member.

Note: You must maintain the sgicm_dmf.config file on each member; a change on one member is unknown to the other members.

You can specify the following directives in the sgicm_dmf.config file:

- The following directive lets you control how DMF behaves when a monitor failure occurs:

  email-only-on-monitor-failure true|false

  By setting email-only-on-monitor-failure to true, DMF will not return an error but will send a monitor failure mail message to the email address specified in email-addresses. The default is false. If email-only-on-monitor-failure is false (explicitly or by default), no email is sent; the monitor script will return an error and the service will be failed over.

- The following directive specifies the email addresses to which a monitor failure message will be sent:

  email-addresses *email1*[,*email2*[,*email3...*]]

  The default address is root. If you specify multiple email addresses, you must separate them with commas. (An improperly formatted directive will be ignored.)

For example, the following sends email messages about monitor failures to chris and pat and does not return an error:

```
% cat /etc/dmf/sgicm_dmf.config
email-only-on-monitor-failure true
email-addresses chris@mycompany.com,pat@mycompany.com
```

If the /etc/dmf/sgicm_dmf.config does not exist, no email is sent; the monitor script will return an error and the service will be failed over.

DMF Start/Stop Order

For the order in which DMF is started/stopped, see Chapter 7, "Creating a New Highly Available Application" on page 89.

Ensuring that Only SGI Cluster Manager Starts DMF

When the DMF service is to be managed by SGI Cluster Manager, it is important that only the Cluster Manager starts DMF. Perform these commands on each member of the cluster to ensure that only the Cluster Manager can start DMF:

```
# touch /etc/dmf_failsafe
# chkconfig dmf off
```

Using TMF with DMF

To use the Tape Management Facility (TMF) with DMF in an SGI Cluster Manager environment, you must configure the appropriate TMF device groups in the `/etc/tmf/sgicm_tmf.config` file according to the instructions in Chapter 11, "Tape Management Facility (TMF) Failover Script" on page 111.

If TMF is configured as a mount service in the `/etc/dmf/dmf.conf` file, the DMF plug-in will automatically call the `/usr/lib/clumanager/service/helper_tmf` TMF failover script and pass along the appropriate TMF device group names.

The service timeout value should be at least 100 seconds if DMF is being used with TMF-managed tape devices. The following command will set the service timeout to 100 seconds for the SGI Cluster Manager service `service1`:

```
# sgicm-config-cluster-cmd --service service1 --servicetimeout=100
```

To do this with the GUI, see "Step 8: Configure the Service" on page 58.

Tape Management Facility (TMF) Failover Script

Using the Tape Management Facility (TMF) with SGI Cluster Manager requires the value-add SGI product on the *SGI Cluster Manager 4.3 for Linux — Storage Software Plug-ins* CD and the supported level of TMF (see "Software Requirements" on page 7). The following hardware is supported:

- IBM 3494 hardware controlled by the Control Path Server (CPS) software
- Storage Technology Corporation (STK) hardware controlled by the Automated Cartridge System Library Software (ACSLS) software

For more information about TMF, see the *TMF Administrator's Guide*.

This chapter discusses the following:

The `helper_tmf` Script

If your application requires tape support via TMF, then your user application script should call the `/usr/lib/clumanager/service/helper_tmf` TMF failover script, passing the appropriate parameters. See "Using the TMF Failover Script from the User Application Script" on page 118.

The DMF plug-in will automatically call the `helper_tmf` script if a Library Server Drive Group uses TMF as a mounting service.

The `helper_tmf` script lets you manage one or more *TMF device groups*, which are sets of tape devices defined in the `/etc/tmf/tmf.config` TMF configuration file.

The following example is part of a `/etc/tmf/tmf.config` that defines a TMF device group named `EGLF`:

```
DEVICE_GROUP
                name = EGLF
        AUTOCONFIG
{
                DEVICE
                        NAME   = f9840f1 ,
                        device_group_name = EGLF ,
                        FILE    = /hw/tape/500104f000417a18/lun0/c4p1 ,
                        status = down ,
                        access = EXCLUSIVE ,
                        vendor_address = (1,0,0,2),
                        LOADER = l180
}
```

The `helper_tmf` script performs the following functions for the calling user service or `userapp` script:

- Starts TMF if it is not already running.
- Configures the associated loader up if it is not already up.
- Allows the monitoring of multiple TMF device groups and their associated tape devices.
- Monitors the number of tape devices that are available within each TMF device group. If the number of devices currently available is less than the minimum threshold level, a monitoring failure will occur.
- Releases previous reservations that are held by another member (if the tape device firmware supports this option).
- Lets you assign different TMF device groups to each instance of an SGI Cluster Manager service or `userapp` script.

TMF Stop/Start Order

For the order in which TMF is started/stopped, see Chapter 7, "Creating a New Highly Available Application" on page 89.

Configuring a TMF Device Group

The `helper_tmf` script lets you specify device groups to stop, start, and monitor. Each of these managed device groups must be defined in the following files:

- `/etc/tmf/sgicm_tmf.config` (SGI Cluster Manager configuration file for TMF)
- `/etc/tmf/tmf.config` (standard TMF configuration file)

The `resource` directive in the `/etc/tmf/sgicm_tmf.config` file specifies a TMF device group. This directive is required for each TMF device group that you plan to use within SGI Cluster Manager. See "The `resource` Directive" on page 114.

Optional Configuration Specifications

There are other optional configuration specifications associated with a TMF device group. These specifications provide information to the `helper_tmf` script that lets it communicate with the tape library. They also identify the tape devices within the library on which `helper_tmf` will force dismounts.

The `helper_tmf` script can force a dismount of tapes from devices within the library. There may be various reasons why you might want to do this when a failover occurs. In the case of DMF, you would want to ensure that any DMF tapes that were in use on a previous member are available to DMF on the new member after a failover. If these tapes were in tape devices assigned to the previous member, they must be ejected and returned to the library so that they are again accessible to DMF on the new member. You may want the `helper_tmf` script to dismount only tape devices associated with a particular TMF device group or you may not want the `helper_tmf` script to dismount any tapes at all.

Some of the functions of the `helper_tmf` script are performed through TMF; the script issues commands to the TMF daemon to use these functions. However, the `helper_tmf` script forces a dismount of a tape from a device by issuing a command to the library software controlling the loader/library. The `helper_tmf` script communicates its request to the ACSLS software that controls the loader. The

`helper_tmf` script uses an `expect` script that issues commands to login to the loader and issue a dismount request to a tape device.

The `/etc/tmf/sgicm_tmf.config` File

The `/etc/tmf/sgicm_tmf.config` file lets you configure other information required by the `helper_tmf` script. The `sgicm_tmf.config` file exists on all members in the cluster and should be edited as necessary on each member.

The contents of the `sgicm_tmf.config` file are dependent on the tape devices assigned to each member in the cluster. If all members in the failover domain are configured through TMF to use exactly the same tape devices, this file would be the same on each member in the failover domain.

Note: You must maintain the `sgicm_tmf.config` file on each member; a change on one member is unknown to the other members.

You can specify the following directives in the `sgicm_tmf.config` file:

- "The `resource` Directive" on page 114
- "The `loader` Directive " on page 115
- "The `remote_devices` Directive" on page 116

The `resource` Directive

The `resource` directive defines the TMF device groups that can be managed by the `helper_tmf` script:

`resource` *device-group devices-minimum devices-loaned email-addresses*

where:

device-group	The TMF device group that is to be monitored. This is a device group that is defined in `/etc/tmf/tmf.config`.
devices-minimum	The minimum number of devices of the specified *device-group* that you must have available to you on a member before you fail over.

devices-loaned	Currently unused; should be set to `0`.
email-addresses	List of addresses to send email when the monitor script detects that tape devices in the *device-group* have become unavailable. Corrective action can then be taken to repair the tape devices before the *devices-minimum* threshold is crossed. This may be a comma- or white-space-separated list of names.

The `loader` Directive

The `loader` directive provides information about a TMF loader, which controls one or more tape devices that are members of TMF device groups being managed by SGI Cluster Manager. There may be multiple `loader` directives in the `sgicm_tmf.config` file.

The loader information is used by the `helper_tmf` script to force a dismount of tapes from tape devices that cannot be made available (that is, that have `tmstat` states other than `assn`, `free`, `conn`, or `idle`) so that those tapes can be used via other tape devices in the same device group. The information is also used to force a dismount of tapes from devices that are only connected to the other member, not to this member (as described in "The `remote_devices` Directive" on page 116).

If the file does not contain a `loader` directive, the `helper_tmf` script will make no attempt to force a dismount of tapes from any devices.

The directive has the following format:

`loader` *lname ltype lhost luser lpassword*

where:

lname	Name of the loader as defined in `/etc/tmf/tmf.config`
ltype	Type of the loader as defined in `/etc/tmf/tmf.config`, which can be either `IBMTLD` or `STKACS`
lhost	Server name of the loader as defined in `/etc/tmf/tmf.config`
luser	User name of the loader's administrator account, which must be `acssa`

lpassword Password for the loader's administrator account

The `tmmls` command shows the name of the loader and the server associated with it. For example:

```
# /usr/sbin/tmmls
loader type status m  server   old  m_pnd  d_pnd  r_qd  comp  avg
 operator  OPERATOR      UP A  IRIX       0      0      0     0     0    0(sec)
 wolfy       STKACS    DOWN A  wolfcree   0      0      0     0     0    0(sec)
 panther     STKACS    DOWN A  stk9710    0      0      0     0     0    0(sec)
 l180        STKACS      UP A  stk9710    0      0      0     0     0    0(sec)
```

For example, suppose you want to have the `helper_tmf` script dismount tape devices that are in the `l180` loader/library listed above. That library has the `stk9710` server associated with it. The loader directive in the `sgicm_tmf.config` file would look like the following:

```
loader l180 STKACS stk9710 acssa acssapassword
```

If the initial attempt to configure the device up fails, the `helper_tmf` script would force a dismount for each tape device that is specified in the `tmf.config` file to be in the `l180` loader/library and in the TMF device group. If you do not want the script to dismount any tape devices associated with a particular TMF device group, you would not place a loader directive in the `sgicm_tmf.config` file.

The `remote_devices` Directive

The `remote_devices` directive provides information about one or more tape devices that are part of a TMF device group, but which are not visible on this member.

For example, suppose you have a library with four SCSI tape devices where two tape devices are connected to each of two cluster members. If member A should crash, member B must be able to force a dismount of any tapes in A's tape devices so that they can then be used from member B. Because the tape devices are not visible on member B, the `remote_devices` directive provides the information needed to force a dismount of unseen tape devices.

The directive has the following format:

`remote_devices` *device-group lname tape-device-ID* `...`

where:

device-group	Name of the TMF device group with which the *tape-device-IDs* are associated.
lname	Name of the loader as defined in `/etc/tmf/tmf.config`. There must be a `loader` directive for *lname* elsewhere in this file, or the `remote_devices` directive will be ignored.
tape-device-ID	The vendor ID of the drive on which to force a dismount. This is the unique name by which the loader identifies the tape device: • For `STKACS`, the *tape-device-ID* value must match the `vendor_address` parameter of the `DEVICE` statement in the `/etc/tmf/tmf.config` file. • For `IBMTLD`, the *tape-device-ID* value must match the `name` parameter of the `DEVICE` statement in the `/etc/tmf/tmf.config` file. **Note:** No blanks should exist within the ID.

You can specify multiple vendor IDs in the same `remote_devices` directive as long as they all pertain to the same loader. If all the vendor IDs will not fit on a single line, add additional `remote_devices` directives for the same loader. For example, to enable the `helper_tmf` script to force a dismount of the remote tape devices `0,0,1,0` , `0,0,1,1` , `0,0,1,2` , and `0,0,1,3` in the `l180` loader/library for TMF device group `tmf_eglf`, the directive would be:

```
remote_devices tmf_eglfl  180  0,0,1,0  0,0,1,1  0,0,1,2  0,0,1,3
```

If multiple TMF device groups are defined, only the TMF device group named `tmf_eglf` will force a dismount of these tape devices.

Configuring Tapes and TMF

If tape devices that are managed by the `helper_tmf` script are configured on more than one member in the cluster, they should be configured consistently. The same tape driver (for example, `ts`) should be used on each member where the tape device is configured.

When configuring the `helper_tmf` script, you should be aware of several parameters in the `/etc/tmf/tmf.config` file. The `helper_tmf` script will try to start the loader associated with its device-group if it is not up. However, if the configuration file specifies `status=UP` for the loader, this step may not be necessary and the devices may become available sooner.

A tape device that is managed by the `helper_tmf` script will be configured in `/etc/tmf/tmf.config` on one or more members within the cluster. It should be configured with `status=down`.

If the tape devices being used do not support persistent reserve, then they should each be configured in `/etc/tmf/tmf.config` with `access=shared`. If the tape devices do support persistent reserve, it is recommended that you use this feature when using the `helper_tmf` script. To use persistent reserve, you should set `access=exclusive` in `/etc/tmf/tmf.config` for each tape device. The access option should be consistent across all members in the cluster where the tape devices are configured.

The `-g` option of the `tmconfig` command reassigns a device to a different device group name. The `helper_tmf` script does not support reassigning a device into a device group. That is because, in case of failover, the `helper_tmf` script on the member we have failed over to would not have any knowledge of this reassigned tape device. It would not be able to dismount tapes that are in the tape device. If you use `tmconfig -g` to move devices out of a device group, that will decrease the number of available tape devices that the monitor function of the `helper_tmf` script can detect. Also, in the case of failover or stop, the tape device will be configured down.

Using the TMF Failover Script from the User Application Script

You must write a user application script in order to use the `helper_tmf` script. For information on how to write user application script, see Chapter 7, "Creating a New Highly Available Application" on page 89.

In order to manage TMF device groups in an SGI Cluster Manager environment, the user application script must pass the appropriate parameters to the TMF failover script. This script called via the following command line:

`/usr/lib/clumanager/scripts/helper_tmf` *action* *device-groups*

where:

action One of `start`, `stop`, or `status`

device-groups	One or more TMF device groups upon which the action should be taken

Note: It is more efficient to invoke the `helper_tmf` script once with several *device-group* arguments rather than invoking it several times, each with a single *device-group* argument. For example, the following:

```
# /usr/lib/clumanager/scripts/helper_tmf start 9840 9940 LTO2
```

is more efficient than the following:

```
# /usr/lib/clumanager/scripts/helper_tmf start 9840
# /usr/lib/clumanager/scripts/helper_tmf start 9940
# /usr/lib/clumanager/scripts/helper_tmf start LTO2
```

For example, to start the 9840 device group:

```
/usr/lib/clumanager/services/helper_tmf start 9840
if [ $? -ne 0 ]; then
               logAndPrint $LOG_ERROR "start of 9840 device group failed"
               return 1;
fi
```

To stop the 9840 device group:

```
/usr/lib/clumanager/services/helper_tmf stop 9840
if [ $? -ne 0 ]; then
               logAndPrint $LOG_ERROR "unable to stop 9840 device group"
               return 1;
fi
```

To check the status of the 9840 device group:

```
/usr/lib/clumanager/services/helper_tmf status 9840
if [ $? -ne 0 ]; then
               logAndPrint $LOG_ERROR "device group 9840 not running"
               return 1;
fi
```

Service Timeout

The service timeout for the calling `userapp` or user script should be at least 100 seconds. The following command will set the service timeout to 100 seconds for the SGI Cluster Manager service `service1`:

```
sgicm-config-cluster-cmd --service service1 --servicetimeout=100
```

Local XVM Plug-In

SGI Cluster Manager supports failover of XVM volumes in *local* mode. This support is available as part of the `clumanager-sgi` RPM in on the *SGI Cluster Manager 4.3 for Linux — Storage Software Plug-ins* CD.

Note: XVM in **cluster** mode is supported only with CXFS. See Chapter 9, "CXFS Plug-In" on page 101.

For more information about XVM, see *XVM Volume Manager Administrator's Guide*.

This chapter contains the following:

Local XVM Device Configuration

Local XVM devices are configured as a device for a service. You can specify multiple XVM devices for a service. For each local XVM volume device, specify the list of physical volumes that it contains, separating each element in the list by a comma (`,`) character.

Following is an example to fail over local XVM volume `m0`:

1. Install and configure XVM on all members in the cluster.
2. Find the physical volumes that are part of volume `m0`:

```
# xvm
xvm:local> show -topology -extend vol/m0
vol/m0                              0 online,open
    subvol/m0/data             497824768 online,open
        stripe/stripe0             497824768 online,tempname,open (unit size: 128)
            mirror/mirror8              35558944 online,tempname,open
                slice/dks5d1s0              35558944 online,open (dks5d1:/dev/rdsk/dks5d1vol)
                slice/dks4d1s0              35558944 online,open (dks4d1:/dev/rdsk/dks4d1vol)
            mirror/mirror4              35558944 online,tempname,open
```

```
slice/dks11d1s0            35558944 online,open (dks11d1:/dev/rdsk/dks11d1vol)
slice/dks7d1s0             35558944 online,open (dks7d1:/dev/rdsk/dks7d1vol)
```

The list of physical volumes that belong to volume `m0` are `dks5d1`, `dks4d1`, `dks11d1`, and `dks7d1`.

3. Add the device to the service using the `sgicm-config-cluster` GUI or the `sgicm-config-cluster-cmd` command-line interface. The device name will be `/dev/lxvm/m0` and the physical volumes will be `dks5d1,dks4d1,dks11d1,dks7d1`.

   For example, the following output from the CLI after the `device` item shows the information that has been added to service `nfs1`:

```
# sgicm-config-cluster-cmd --service=nfs1 \
--device=/dev/lxvm/m0 \
--physvols=dks5d1,dks4d1,dks11d1,dks7d1

device:
  name = /dev/lxvm/m0
  sharename =   physvols = dks5d1,dks4d1,dks11d1,dks7d1

mount:
  mountpoint = /mnt5
  fstype = xfs
  options = rw
  forceunmount = yes

nfsexport:
  name = /mnt5

client:
  name = challenger.example.com
  options = rw
```

Note: SGI Cluster Manager uses the `xvm` subcommands `give` and `steal` during failover for local XVM volumes. However, the list of physical volumes can be specified or modified only if the `clumanager-sgi` RPM is installed on the member.

Figure 12-1 shows an example in the GUI.

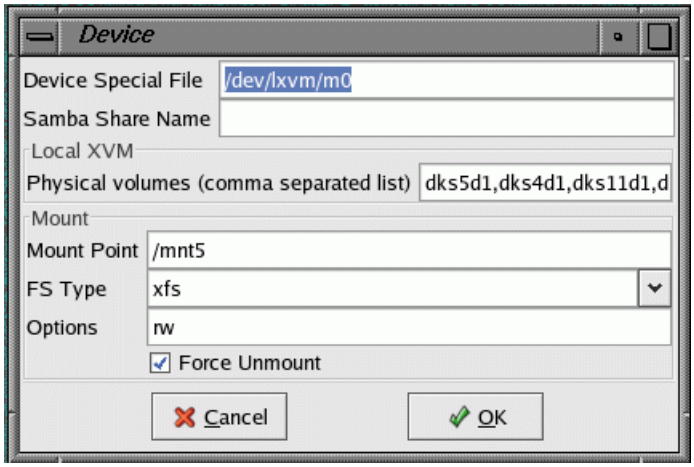

Figure 12-1 Adding an XVM Device

For more information on configuration, see "Step 8: Configure the Service" on page 58.

Local XVM Start/Stop Order

For the order in which local XVM is started/stopped, see Chapter 7, "Creating a New Highly Available Application" on page 89.

Troubleshooting

This chapter provides information about the following:

To avoid problems, see Chapter 2, "Best Practices" on page 11.

Troubleshooting Strategy

To troubleshoot SGI Cluster Manager problems, do the following:

Know the Troubleshooting Tools

This section provides an **overview** of the tools required to troubleshoot SGI Cluster Manager:

Caution: Many of the commands listed are beyond the scope of this book and are provided here for quick reference only. See the other guides and man pages referenced for complete information before using these commands.

Startup Scripts

Understand the startup scripts for CXFS, XVM, and your applications.

Physical Storage Tools

Understand the following physical storage tools:

- To display the hardware inventory:

  ```
  [root@linux root]# hwinfo --short
  ```

  If the output is not what you expected, do a probe for devices and perform a SCSI bus reset, using the following commands:

  – QLogic SCSI or Fibre Channel: use the following to probe the LUN on the specified *hostname*:

```
[root@linux root]# echo "- - -" > /sys/class/scsi_host/hostname/scan
```

  Each "-" character is a wildcard for bus, target, and LUN, respectively. Newer SCSI and all FC controllers have a single bus per function, but two functions in the dual-port controllers. For example, if you added a new LUN to a RAID (and the RAID is target 3) for a host named `host3`:

```
[root@linux root]# echo "0 3 -" > /sys/class/scsi_host/host3/scan
```

QLogic Fibre Channel: use the following to discover and build a new table for the LUN, where 3 is the host number:

```
[root@linux root]# echo "scsi-qlascan" >/proc/scsi/qla2xxx/3
```

– LSI: use the `lsiutil` tool to scan the HBA, selecting option 8 to scan for devices:

```
[root@linux root]# lsiutil

LSI Logic MPT Configuration Utility, Version 1.41, November 23, 2005

4 MPT Ports found

     Port Name         Chip Vendor/Type/Rev    MPT Rev  Firmware Rev
 1.  /proc/mpt/ioc0    LSI Logic 53C1030 B2      102      01032710
 2.  /proc/mpt/ioc1    LSI Logic 53C1030 B2      102      01032710
 3.  /proc/mpt/ioc2    LSI Logic FC949X A1       105      01030300
 4.  /proc/mpt/ioc3    LSI Logic FC949X A1       105      01030300

Select a device:  [1-4 or 0 to quit] 3

 1.  Identify firmware, BIOS, and/or FCode
 2.  Download firmware (update the FLASH)
 4.  Download/erase BIOS and/or FCode (update the FLASH)
 8.  Scan for devices
10.  Change IOC settings (interrupt coalescing)
13.  Change FC Port settings
16.  Display logged-in devices
20.  Diagnostics
21.  RAID actions
22.  Reset bus
23.  Reset target
30.  Beacon on
31.  Beacon off
60.  Show non-default settings
61.  Restore default settings
98.  Reset FC link
99.  Reset port

Main menu, select an option:  [1-99 or e for expert or 0 to quit] 8
```

```
FC949X's link is online, type is fabric direct attach, speed is 2 Gbaud

 B___T___L  Type       Vendor   Product          Rev         WWPN       PortId
 0 127   0  Disk       SGI      TP9300           0612  200d00a0b8131841 021500
 0 127   1  Disk       SGI      TP9300           0612
 0 127   2  Disk       SGI      TP9300           0612
 0 127  31  Disk       SGI      Universal Xport  0612
 0 128   0  Disk       SGI      TP9300           0612  200c00a0b8131841 021400
 0 128   1  Disk       SGI      TP9300           0612
 0 128   2  Disk       SGI      TP9300           0612

 0 128  31  Disk       SGI      Universal Xport  0612

 0 129   0  Disk       SGI      TP9100 F PSEUDO  5903  23000050cc007d2c 021300

 0 130   0  Disk       SGI      TP9100 F PSEUDO  5903  22000050cc007d2c 021200
            FC949X Port                                 100000062b0e4248 021700
            FCP Initiator                               210000e08b1058d4 021000
            FCP Initiator                               210100e08b3058d4 021100
            FCP Initiator                               100000062b0e4249 021600
            Non-FCP                                     20fc006069c021b6 fffffc
            Non-FCP                                     2007006069c021b6 fffffe
```

- To show the physical volumes, use the `xvm` command:

  # **`/sbin/xvm show -v phys/`**

  See the *XVM Volume Manager Administrator's Guide*.

Cluster Configuration Tools

Understand the following cluster configuration tools:

- To configure the cluster, use the `sgicm-config-cluster`(8) SGI Cluster Manager GUI or `sgicm-config-cluster-cmd`(8) command line. See "Cluster Configuration Tools" on page 41.
- To configure XVM volumes, use the `xvm` command. See the *XVM Volume Manager Administrator's Guide*.
- To configure CXFS nodes and cluster, use the CXFS GUI, the `cxfs_admin` command, or the `cmgr` command. See the *CXFS Administration Guide for SGI InfiniteStorage*.

Cluster Control Tools

Understand the cluster control tools:

- `sgicm-config-cluster`(8) GUI
- `/etc/init.d/clumanager`
- "Cluster Configuration Tools" on page 41

See Chapter 6, "Administration" on page 81.

Networking Tools

Understand the following networking tools:

- To send packets to network hosts:

  ```
  [root@linux root]# /bin/ping
  ```

- To show network status:

  ```
  [root@linux root]# /bin/netstat
  ```

Cluster/Node Status Tools

Understand the following cluster/node status tools:

- To monitor status, use the following:
  - The `sgicm-config-cluster` GUI to monitor the status of the cluster and the services
  - `clustat` to monitor the cluster status

  See "Monitoring Status" on page 81.

- To display information about a service, use the GUI or the following command:

  ```
  sgicm-config-cluster-cmd --service=servicename
  ```

  See "Displaying Service Information" on page 82.

- To see the mounted filesystems:

  ```
  [root@linux root]# /bin/mount
  [root@linux root]# /bin/df
  ```

```
[root@linux root]# cat /proc/mounts
_
```

You can also use the `df` command to report the number of free disk blocks

- To show volumes:

  # **/sbin/xvm show vol/**

  See the *XVM Volume Manager Administrator's Guide*.

Performance Monitoring Tools

Understand the following performance monitoring tools:

- To monitor system activity:

  # **/usr/bin/sar**

- To monitor system input/output device loading, use the `iostat`(1) command. For example, to monitor at 2–second intervals for 10000000 times:

  ```
  [root@linux root]# iostat -2 1000000
  ```

- To monitor process status, memory consumption, paging activity, block I/O operations, interrupts, context switches, and processor usage, use the `vmstat`(8) command. For example, to monitor at 1–second intervals for 1000 times:

  ```
  [root@linux root]# vmstat -a -n 1 1000
  ```

- To monitor the statistics for an XVM volume, use the `xvm` command:

  # **/sbin/xvm change stat on** {*concatname*|*stripename*|*physname*}

  See the *XVM Volume Manager Administrator's Guide*.

- To monitor system performance, use Performance Co-Pilot. See the *Performance Co-Pilot for IA-64 Linux User's and Administrator's Guide*, the *Performance Co-Pilot Programmer's Guide*, and the `pmie` and `pmieconf` man pages.

Use the `watch`(1) command to monitor the contents of a file. For example:

```
[root@linux root]# watch cat /proc/meminfo
```

Log Files

Monitor the messages in `/var/log/messages`. See "Message Logging" on page 87.

Identify Cluster Status

To identify the cluster status, do the following:

- Check the messages in `/var/log/messages` (see "Message Logging" on page 87)
- Use `shutil`(8) to see if shared quorum partitions are accessible
- Use `clufence`(8) to check the status of the reset cable
- Verify that the failover domain is defined correctly

Understand What Happens After a System Crash or Hang

Following is what happens after the system crashes or hangs:

1. The crashed system begins a dump and reboot.
2. SGI Cluster Manager detects that the system is not responsive because there is no heartbeat within the member-timeout interval.
3. SGI Cluster Manager issues a system reset.
4. SGI Cluster Manager forces the failover of the application on the crashed system to the backup system.
5. The formerly crashed system reboots.
6. The application moves back to the original system only if the failover domain is defined and the controlled failback option has not been specified for it.

Recovery from a `clulockd` Failure

If the `clulockd` daemon dies unexpectedly, it freezes all of the locks on the shared quorum partition. `clulockd` will write a message similar to the following in the logs:

```
Feb  6 17:25:14 3U:nygaard clulockd[6924]:  Signal 11 received; freezing
```

The `clusvcmgrd` daemon will not be able to monitor, start, or stop services. Logs on all members will have a message such as the following:

```
Feb  6 17:14:48 2U:dahl clusvcmgrd[3255]:  Couldn't connect to member #0: Connection timed out
Feb  6 17:14:48 3U:dahl clusvcmgrd[3255]:  Unable to obtain cluster lock: No locks available
```

To recover from this situation, do the following:

1. Stop cluster daemons on all members.
2. Reinitialize the shared state from one member in the cluster:

   ```
   shutil -i
   ```

3. Make sure that `/etc/cluster.xml` is same on all members.
4. Initialize the configuration on the shared quorum partition from one member in the cluster:

   ```
   shutil -s /etc/cluster.xml
   ```

5. Verify that the configuration has been initialized correctly from one member in the cluster:

   ```
   shutil -p /cluster/config.xml
   ```

For more information, see the `shutil`(8) man page.

Watchdog Errors

Software and hardware watchdog timers are not supported. If a watchdog has been enabled on a member, you may see the following errors when the cluster daemons are starting:

```
Creating /dev/watchdog: execvp: No such file or directory
^[[FAILED]
Loading Watchdog Timer (softdog): modprobe: Can't locate module softdog
^[[FAILED]
```

You may also see a message similar to the following in the Cluster Manager log:

```
clumembd[21355]:  clumembd_sw_watchdog_stop: watchdog is not running.
```

To disable the software watchdog on a member, enter the following:

```
sgicm-cluster-manager-cmd --member=member_name --watchdog=no
```

For example:

```
# sgicm-cluster-manager-cmd --member=member1 --watchdog=no
```

Shared Quorum Partitions

This section discusses the following:

For more information, see the `shutil`(8) man page.

Verify Accessibility

To see if shared quorum partitions are accessible, enter the following:

```
shutil
```

Read the Configuration File

To read the configuration file from the shared quorum partition, enter the following:

```
shutil -r -
```

You should use this command to compare the configuration files in the shared quorum partitions and the local copy.

Verify Metadata Information is Consistent

To verify that the service metadata information is the same on all members, run the following command at the same time on each member:

```
shutil -m /service/0/status
```

For example, the following output from member `jackhammer` and member `jackhammer2` indicates a problem:

- `jackhammer` output:

```
# shutil -m /service/0/status
Metadata information for /service/0/status

Data Length:   40 bytes
Data CRC:      0x2dae1205
Header CRC:    0x7c7185f1
Last modified: 12:34:58 Mar 31 2004
```

- `jackhammer2` output:

```
# shutil -m /service/0/status
Metadata information for /service/0/status

Data Length:   40 bytes
Data CRC:      0x80711487
Header CRC:    0x9ba9e2cf
Last modified: 12:34:51 Mar 31 2004
```

In this case, the service metadata information from both members is inconsistent (the `CRC` information and the `Last modified` time stamps are different). The information must be identical from all the members.

Write the Configuration File

To write the configuration file, use the following command:

```
shutil -s /etc/cluster.xml
```

You should use this command if one of the following is true:

- The configuration file in the shared quorum partitions is not consistent with the `/etc/cluster.xml` file
- The shared quorum partition was cleared using the `shutil -i` command

Displaying Metadata Remotely

To display the metadata information from the shared quorum partition, use the following command:

```
shutil -p /service/0/status
```

Last Resort: Clear Information

Caution: Do not run this command while the cluster is enabled.

To clear all cluster information, use the following command:

```
shutil -i
```

Serial Cable or Reset issues

The `clufence` command will fail with a nonzero error code for any of the following reasons:

- The serial cable is not connected
- The cable is faulty
- The system controller is not responding
- The tty device is not available because the serial port driver (`ioc4_serial`) is not loaded

The messages shown in the following output are also logged to `/var/log/messages`:

```
# clufence -s jackhammer2
[12314] info: STONITH: Power controller l2 connected to peer's /dev/ttyIOC1 controls jackhammer
[12314] info: STONITH: Power controller l2 connected to peer's /dev/ttyIOC1 controls jackhammer2
[12314] err: STONITH: Device at /dev/ttyIOC1 controlling jackhammer2 FAILED status check:
Timed out
```

Failed State for a Service

The following output indicates that the action to disable a service (in this case, `nfs_samba`) has failed, and the service is moved to failed state:

```
# clusvcadm -d nfs_samba
Member machine1 disabling nfs_samba...failed
Service nfs_samba might be running in the cluster. Stop the service manually.
```

To recover, do the following:

1. Fix the problem.
2. Stop the resources in the service manually using the `ifconfig`(8), `exportfs`(8), and `umount`(8) commands.
3. Disable the service using the `clusvcadm`(8) command or the SGI Cluster Manager GUI.

Note: SGI Cluster Manager does not verify that the service has been stopped before disabling.

For more information, see"Service Administration" on page 84 and "Cluster Service States" on page 85.

Error Messages

Following are common error messages.

```
[12314] err: STONITH: Device at /dev/ttyIOC1 controlling jackhammer2 FAILED status check:
Timed out
```

There is a problem with the serial cable or system controller. See "Serial Cable or Reset issues" on page 135.

```
clumembd[8431]:  No heartbeat channels available!
clumembd[8431]:  Heartbeat failed to initialize!
```

These messages (logged by the `clumembd` process when the local cluster manager daemons are started using GUI or command line) mean that the IP address for the hostname could not be determined or that the IP address assigned to the hostname is invalid. You can verify this by sending `ping` packets local machine's hostname. Fix the hostname IP address and restart the local cluster daemons.

```
Shared partition device file names must be defined.
```

An attempt was made to define the cluster before defining the shared state. Use the `--sharedstate` command line option or shared state GUI menu to define devices. See "Step 1: Define the Shared Quorum Partitions" on page 44.

```
Shared partition device file names primary /dev/shared1 and shadow /dev/shared2 are not valid.

Shared storage initialization failed.
Fix shared storage and write configuration file to shared storage.
Continuing ...
```

The shared quorum partitions are not accessible or not valid and a configuration change or query was made using the CLI.

```
Traceback (most recent call last):
   File "/usr/sbin/sgicm-config-cluster", line 47, in ?
     from clusterpkg.cluconfig_module  import cluconfig
   File "/usr/share/sgicm-config-cluster/configure/clusterpkg/cluconfig_module.py", line 2, in ?
     from clusterpkg.cluster_module    import cluster
   File "/usr/share/sgicm-config-cluster/configure/clusterpkg/cluster_module.py", line 2, in ?
     from xml.dom import minidom
 ImportError: No module named xml.dom
```

These messages will occur if you try to run SGI Cluster Manager without first installing the appropriate packages from the SUSE LINUX Enterprise Server 9 CDs. See the `README` file for a list of the RPMs.

Reporting Problems to SGI

If you encounter problems, collect the following data from each member:

- Output from the following commands:

  `exportfs` *(in NFS configurations)*
  `chkconfig --list`
  `clufence -s` *other_members*
  `clustat`
  `cxfsdump` *(in CXFS configurations)*
  `hwinfo`
  `ls -l` *each_shared_quorum_partition*
  `ls -lL` *each_shared_quorum_partition*
  `mount`
  `ps -ef | grep clu`
  `rpm -qa --last`
  `shutil -r -`
  `uname -a`

- Contents of the following files:

  `/etc/cluster.xml`
  `/usr/lib/clumanger/create_device_links`
  `/var/log/messages`
  `/etc/samba/smb.conf.`*SambaShareName* *(in Samba configurations)*

FailSafe and SGI Cluster Manager

Table A-1 summarizes the differences between IRIX FailSafe and SGI Cluster Manager for Linux for those readers who may be familiar with FailSafe.

Note: SGI Cluster Manager for Linux members and FailSafe nodes do not work together and cannot form a high-availability cluster.

Table A-1 Differences Between FailSafe and SGI Cluster Manager

Topic	FailSafe	SGI Cluster Manager
Operating system	IRIX	SGI ProPack 5 for Linux
Terminology	node resource	member application
Size of cluster	8 nodes	4 members
Node/member name	Hostname or private network address	Hostname only
NFS lock failover	Supported	Not supported
Network tiebreaker	A node that is participating the cluster membership. FailSafe tries to include the tiebreaker node in the membership in case of split-brain scenarios.	The IP address of machine or a router that **does not participate** in the cluster membership. Usually it is the IP address of a network router that connects the SGI Cluster Manager members to the external world (clients). In a split-brain scenario, only those members that can contact the tiebreaker IP address can form a cluster. There is also a disk tiebreaker.
Rolling upgrade	Supported	Not supported

Topic	FailSafe	SGI Cluster Manager
Configuration information storage	Information is stored in the cluster database. The cluster database is replicated on all nodes automatically and kept in synchronization.	Information is stored in the `/etc/cluster.xml` configuration file and in the shared partitions. For initial configuration, you must copy this file to all members, such as by using `scp`. After making configuration changes, you must verify that configuration files are in synchronization. See "Step 15: Verify that Configuration Changes are Synchronized" on page 64.
Making changes while the service is enabled	Device parameter, IP address parameters, and check interval can be changed.	Device parameter, IP address parameters, and check interval cannot be changed.
Script location for resources and resource types	`/var/cluster/ha/resource_types`	`/usr/lib/clumanager/services/service`
Heartbeat interval and timeout	You can specify cluster membership heartbeat interval and timeout (in milliseconds).	In the command line, you can specify the heartbeat interval (in microseconds) and the number of heartbeats that can be consecutively missed (`tko_count`). You can also specify the aggregate failover speed in the GUI.
Heartbeat networks	Allows multiple networks to be designated as heartbeat networks. You can choose a list of networks.	Allows heartbeat on all networks or as a multicast on the hostname network. However, you cannot choose a list of networks.
Action scripts	Separate scripts named `start`, `stop`, `monitor`, `restart`, `exclusive`.	A `bash` script that contains `start`, `stop`, and `status` parameters (see Chapter 7, "Creating a New Highly Available Application" on page 89). The equivalent for `restart` in SGI Cluster Manager is to perform a `stop` and then a `start`; there is no equivalent in SGI Cluster Manager for `exclusive`.
Resource timeouts	Timeouts can be specified for each action (`start`, `stop`, `monitor`, `restart`, `exclusive`) and for each resource type independently.	Timeout can be specified for each service irrespective of the action or the number of resources it contains.

Topic	FailSafe	SGI Cluster Manager
Resource dependencies	Resource and resource type dependencies are supported and can be modified by the user.	Applications have fixed dependencies. The start and stop order of applications cannot be modified.
Failover policies	The ordered and round-robin failover policies are predefined. User-defined failover policies are supported.	Only the predefined ordered policy is supported. No user-defined failover policies are supported.

Setting the Partition Type to Linux

When you create a new disk partition using `parted`(8), the partition type is automatically set based on the type of filesystem chosen for the partition. For most filesystems, such as `ext2` or `XFS`, the partition type will be hexadecimal 83 (`0x83`), or Linux.

You can also use `parted` to change the type of existing partitions. Use the `set` subcommand to change the type flag to `0x83`.

Note: The `parted` command expects a decimal number for most inputs. When entering a hexadecimal number would be more convenient, such as when setting the partition type flag, you must precede the number with `0x` to indicate hexadecimal input.

The following example shows the use of `parted` to change the type of partition 1 on disk `/dev/sde` from `0x82` (Linux swap) to `0x83` (Linux):

```
# parted /dev/sde
GNU Parted 1.6.21
Copyright (C) 1998 - 2004 Free Software Foundation, Inc.
This program is free software, covered by the GNU General Public License.
...

Using /dev/sde
(parted) print
Disk geometry for /dev/sde: 0.000-69424.000 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
1          0.031  69421.508  primary               type=82
(parted) set
Partition number? 1
Flag to change? type
New type?  [130]? 0x83
(parted) print
Disk geometry for /dev/sde: 0.000-69424.000 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
1          0.031  69421.508  primary               type=83
```

```
(parted) quit
Information: Don't forget to update /etc/fstab, if necessary.
#
```

Note: The `type` flag in the `parted` display of a partition table is equivalent to the `Id` field in the `fdisk`(8) display of a partition table.

Glossary

CLI

Command line interface (`sgicm-config-cluster-cmd`).

cluster global lock manager daemon

The `clulockd` daemon, which stores locks on the shared partition.

cluster membership daemon

The `clumembd` daemon, which performs network heartbeats and checks the liveliness of other members in the cluster.

cluster quorum daemon

The `cluquorumd` daemon, which computes new membership, implements quorum, enforces I/O fencing, and reads/writes membership information to the shared partition.

cluster remote NFS mount table daemon

The `clurmtabd` daemon, which synchronizes NFS mount point entries by polling the `/var/lib/nfs/rmtab` file.

cluster service manager daemon

The `clusvcmgrd` daemon, which starts/stops and checks the status of services running in the cluster.

control messages

Messages that SGI Cluster Manager software sends between the members to request operations or distribute information to ensure that services remain highly available.

controlled failback

The service will not be moved to a machine that has newly joined the cluster even if the new machine is the preferred member according to the failover domain. The system administrator must manually relocate the service in order to move it back to the preferred member.

disk tiebreaker

If two members cannot talk to each other, they look at the status on the shared partition disk to decide which member should survive and be part of the cluster membership. If the disk cannot be accessed or membership on the disk does not include a given machine, all SGI Cluster Manager processes on the machine exit.

failback option

A failover domain option that is considered when a member rejoins the cluster.

failover

The process by which one member restarts the highly available applications of a failed member.

failover domain

The list of members in the cluster where a service can be online.

failover option

A failover domain option that is considered when a failure occurs and a new target member for the service must be determined.

failover speed

The time it takes to detect a member failure.

GUI

Graphical user interface (`sgicm-config-cluster`).

heartbeat interval

The number of microseconds before a heartbeat is sent to all other members in the cluster.

heartbeat

Messages that SGI Cluster Manager software sends between the members that indicate a machine is up and running.

heartbeat timeout

The number of heartbeats missed before a member is declared as failed.

highly available services

Applications that are monitored by the SGI Cluster Manager software. If one member fails, the other member restarts the highly available applications of the failed member. To clients, the services on the replacement member are indistinguishable from the original services before failure occurred. It appears as if the original member has crashed and rebooted quickly. The clients notice only a brief interruption in the highly available service.

IO10

A full-size PCI expansion board that provides basic system I/O capabilities via the PCI bus.

IX brick

System component that provides the base I/O functionality for the system; it contains the electronics and hardware necessary to boot.

L2

SGI system controller used to monitor and manage the server.

local member

The machine being configured.

local XVM volumes

Logical volumes that are local to one member at a time and are not shared across the cluster. They may change ownership upon failover or moving a service.

lowest-ordered

A higher preference for a service to be started on that member.

member

A machine or system partition that is defined as part of a cluster.

multiport serial adapter cable

A device that provides four DB9 serial ports from a 36-pin connector.

network tiebreaker

Ensures that only the member that can contact the tiebreaker IP address can form a cluster. The tiebreaker is the IP address of a machine or a router that **does not participate** in the cluster. Usually, it is the IP address of a network router that connects the members to the external world (clients).

ordered failover

A failover domain option that causes the service to start on the first member defined if it is available.

partition

See *shared partitions* and *system partition*.

peer member

The other member in a 2–member cluster, to which the local system controller is connected.

plug-in

The set of software that allows a service to be highly available without modifying the application itself.

primary partition

One of the two disk partitions without filesystems where SGI Cluster Manager keeps configuration, cluster, and service status information. See also *shared partition* and *shadow partition*.

restricted failover

A failover domain option that permits failover only to the members listed.

SATA

Serial ATA disk.

service ID

A number that identifies the service (the ID is automatically determined and is not user-configurable).

shadow partition

One of the two disk partitions without filesystems where SGI Cluster Manager keeps configuration, cluster, and service status. The shadow partition is the backup partition. See also *primary partition* and *shared partitions*.

shared partitions

The two disk partitions without filesystems (primary partition and shadow partition) where SGI Cluster Manager keeps configuration, cluster, and service status information.

split-brain scenario

Network partition in which two members attempt to form individual clusters.

symlink

Symbolic link

system partition

A machine that is logically divided into multiple servers. Also referred to as an *Altix partition*.

Index

D

I

J

L

M

N