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

007–3800–002

CONTRIBUTORS

Written by Lori Johnson

Engineering contributions by Derek Guy Barnes, Dale Brantly, Jeff Cech, Susheel Gokhale, Ron Kerry, LaNet Merrill, Nate Pearlstein, Kevan Rehm, Paddy Sreenivasan

Illustrated by Chrystie Danzer

Production by Karen Jacobson

New Features in this Guide

This release supports the following:

- Support for the L2 power controller via Ethernet. See "L2 Power Controller" on page 12.
- Hardware diagrams describing L2 reset connections. See:
  - Figure 2-1 on page 13
  - Figure 2-2 on page 14
  - Figure 2-3 on page 15
  - Figure 2-4 on page 16
- Controlled failback option; see "Step 7: Create the Failover Domain" on page 36
- Detach capability for services, after which 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.) See "Service Administration" on page 52.
- Restart count limit; see "Step 8: Configure the Service" on page 38
- The ability to specify a script that contain shell functions to failover user applications. See "Step 8: Configure the Service" on page 38, "Example of Failing Over Multiple User Applications" on page 60, and "Sample User Application Script" on page 60
- Support for multiple user applications. See "Example of Failing Over Multiple User Applications" on page 60.

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

Contents

About This Guide . . . . . **xvii**
Related Publications xvii
Obtaining Publications xviii
Conventions xviii
Reader Comments xix

1. Introduction **1**
Base Product 2
Optional SGI Software Storage Plug-In Product 2
Highly Available Services 3
Hardware Requirements 3
Software Requirements 6
Failover Domains 6
Cluster Daemons 9

2. Hardware Installation **11**
Shared Partitions 11
Heartbeat Network 12
Power Controllers 12
L2 Power Controller 12
Testing Serial Connectivity for the L2 17
Testing Ethernet Connectivity for the L2 18

3. Software Installation **19**
Software Packages 19
Installing the Software 20

Upgrading 21
Uninstalling the Software 22

4. Configuration 23

Cluster Configuration Tools 23
Displaying Configuration Status 23
Saving Changes 25
Configuration Steps 25
Step 1: Define the Shared Partitions 26
Step 2: Create the Cluster 27
Step 3: Define the Members 28
Step 4: Add Power Controller Configuration 28
Step 5: Change the Heartbeat Interval, Timeout, and Failover Speed 32
Failover Speed and the GUI 32
Failover Speed and the CLI 33
Step 6: Set the Tiebreakers 35
Step 7: Create the Failover Domain 36
Step 8: Configure the Service 38
Step 9: Add a Service IP Address 40
Step 10: Add the Disk and Filesystem Information to the Service *(Optional)* 41
Step 11: Add a Samba Share *(Optional)* 42
Step 12: Define the NFS Information *(Optional)* 42
Step 13: Save the Cluster Configuration *(GUI only)* 43
Step 14: Synchronize Configuration Changes Across the Cluster 43
Step 15: Verify that Configuration Changes are Synchronized 43
Step 16: Start the Cluster Daemons 44
Example Cluster Configuration 44

5. Administration 49
Monitoring Status 49
Displaying Service Information 50
Starting Cluster Processes 51
Stopping Cluster Processes 52
Service Administration 52
Cluster Service States 53
Message Logging 55

6. Creating a New Highly Available Application 57
The `clusvcmgrd` Daemon 57
The `service` Script 57
Adding a Service 58
Example of Failing Over Multiple User Applications 60
Sample User Application Script 60

7. Samba Plug-In 63

8. CXFS Plug-In 65

9. Data Migration Facility (DMF) Plug-In 69
Adding the DMF User Script to an Existing Service 69
DMF Administrative Filesystems and Directories 69
Configuring DMF for Local XVM Filesystems 70
Configuring DMF for CXFS Filesystems 71
Start/Stop Order 71
Ensuring that Only SGI Cluster Manager Starts DMF 71
Using TMF with DMF 71

10. Tape Management Facility (TMF) Failover Script 73

The `helper_tmf` script 73

Configuring a TMF Device Group 75

Optional Configuration Specifications 75

The `/etc/tmf/sgicm_tmf.config` File 76

The `resource` Directive 76

The `loader` Directive 77

The `remote_devices` Directive 78

Configuring Tapes and TMF 79

Using the TMF Failover Script from the User Application Script 80

Service Timeout 81

11. Local XVM Plug-In 83

12. Troubleshooting 87

Best Practices 87

Recovery from a `clulockd` Failure 87

Watchdog Errors 88

Shared Partitions 89

Verify Raw Devices are Character Special Devices 89

Verify Accessibility 89

Read the Configuration File 90

Verify Metadata Information is Consistent 90

Write the Configuration File 91

Displaying Metadata Remotely 91

Last Resort: Clear Information 91

State Inconsistencies 91

Serial cable or Reset issues 92

Error Messages 92
Reporting Problems to SGI 93

Appendix A. Differences Between Red Hat Cluster Manager and SGI Cluster Manager 95

Appendix B. FailSafe and SGI Cluster Manager 97

Appendix C. Setting the Partition Type to `Linux` 101

Glossary 103

Index 109

Figures

Figure 1-1 An Example CXFS and SGI Cluster Manager Configuration 5

Figure 2-1 Altix 350 Rear Panel 13

Figure 2-2 Altix 3000 Rear Panel 14

Figure 2-3 Altix 3300 L2 with Serial Cable Connection 15

Figure 2-4 Altix 3300 L2 with an Ethernet Connection 16

Figure 4-1 Cluster Status GUI 24

Figure 4-2 Configuring the Power Controller Information for an L2 using Serial Cables . 30

Figure 4-3 Configuring the Power Controller Information for an L2 using an Ethernet Network 31

Figure 4-4 Adjusting Failover Speed 33

Figure 4-5 Tiebreakers 36

Figure 4-6 Failover Domain 37

Figure 4-7 Configuring a High-Availability Service 39

Figure 5-1 Status 50

Figure 5-2 Service Information 51

Figure 5-3 Detached State 54

Figure 6-1 Creating a Service 59

Figure 8-1 Adding a CXFS Filesystem as a Device 66

Figure 11-1 Adding an XVM Device 85

Tables

Table 1-1 Failover Domain and Option Results 8
Table 4-1 Supported Failure Detection Times and Parameter Values 34
Table 9-1 DMF Administrative Filesystem and Directory Parameters 69
Table A-1 Red Hat Cluster Manager and SGI Cluster Manager 95
Table B-1 Differences Between FailSafe and SGI Cluster Manager 97

About This Guide

This guide provides information about SGI Cluster Manager for Linux, which provides highly available services for SGI Altix servers. It is based on the Red Hat Cluster Manager product. An optional product provides 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:

- *Red Hat Cluster Suite: Configuring and Managing a Cluster*, which is available on the *SGI Cluster Manager x.x for Linux — Base Product* CD and at the following website:

  https://www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/cluster-suite/

- SGI ProPack for Linux and SGI Altix documentation:
  - *NIS Administrator's Guide*
  - *Personal System Administration Guide*
  - *SGI ProPack for Linux Start Here*
  - *SGI Altix 350 System User's Guide*
  - *SGI Altix 3000 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*

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.
- On IRIX systems, you can use InfoSearch (if installed), an online tool that provides a more limited set of online books, release notes, and man pages. Enter `infosearch` at a command line or select **Help > InfoSearch** from the Toolchest.
- On IRIX systems, you can view release notes by entering either `grelnotes` or `relnotes` at a command line.
- On Linux systems, you can 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.
- On IRIX and Linux systems, you can 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

- Use the Feedback option on the Technical Publications Library Web page:

  http://docs.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:

  Technical Publications
  SGI
  1500 Crittenden Lane, M/S 535
  Mountain View, California 94043–1351

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 multiple machines or system partitions, known as *members*. This release supports a cluster of two members.

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

To 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.

SGI Cluster Manager is based on Red Hat Cluster Manager, which is part of Red Hat Cluster Suite 3.0. Therefore, this guide will refer to the Red Hat documentation whenever possible. You must have access to *Red Hat Cluster Suite: Configuring and Managing a Cluster*, which is available on the SGI Cluster Manager CD and at the following website:

https://www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/cluster-suite/

This book provides additional information needed to use the SGI product for SGI Altix servers in a high-availability cluster environment. You should read through this guide before you begin configuring the cluster. For differences with Red Hat Cluster Manager, see Appendix A, "Differences Between Red Hat Cluster Manager and SGI Cluster Manager" on page 95.

Although SGI Cluster Manager for Linux provides similar functionality to IRIX FailSafe, there are differences; see Appendix B, "FailSafe and SGI Cluster Manager" on page 97.

This chapter discusses the following:

- "Base Product" on page 2
- "Optional SGI Software Storage Plug-In Product" on page 2

- "Highly Available Services" on page 3
- "Hardware Requirements" on page 3
- "Software Requirements" on page 6
- "Failover Domains" on page 6
- "Cluster Daemons" on page 9

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)

Optional 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. An optional value-add product supplies plug-ins for the following:

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

This optional product 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.

Highly Available Services

A highly available service consists of the following:

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

Hardware Requirements

SGI Cluster Manager requires the following:

- A cluster of up to two members. The following servers are supported:
  - SGI Altix 350 servers with an IO10 PCI expansion board and a *multiport serial adapter cable* (a device that provides four DB9 serial ports from a 36-pin connector). See "Power Controllers" on page 12.
  - SGI Altix 3700 servers or Altix 3300 servers (may be partitioned; each system partition is an individual member).
- The *shared partitions*, which are the raw disk partitions where configuration, cluster, and service status information is kept by SGI Cluster Manager. One is considered the *primary partition*, the other is the *shadow partition* and is used as a backup. The primary partition and the shadow partition should be in different storage devices connected to the members using different Fibre Channel cards. The two partitions should have independent I/O paths.
- 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 cross-over cabling for a private network, you must purchase another PCI Ethernet card.

- Serial cabling: For Altix 3000 systems, you must use the serial ports on the IX brick.
- Power control: an L2 system controller

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 35.)

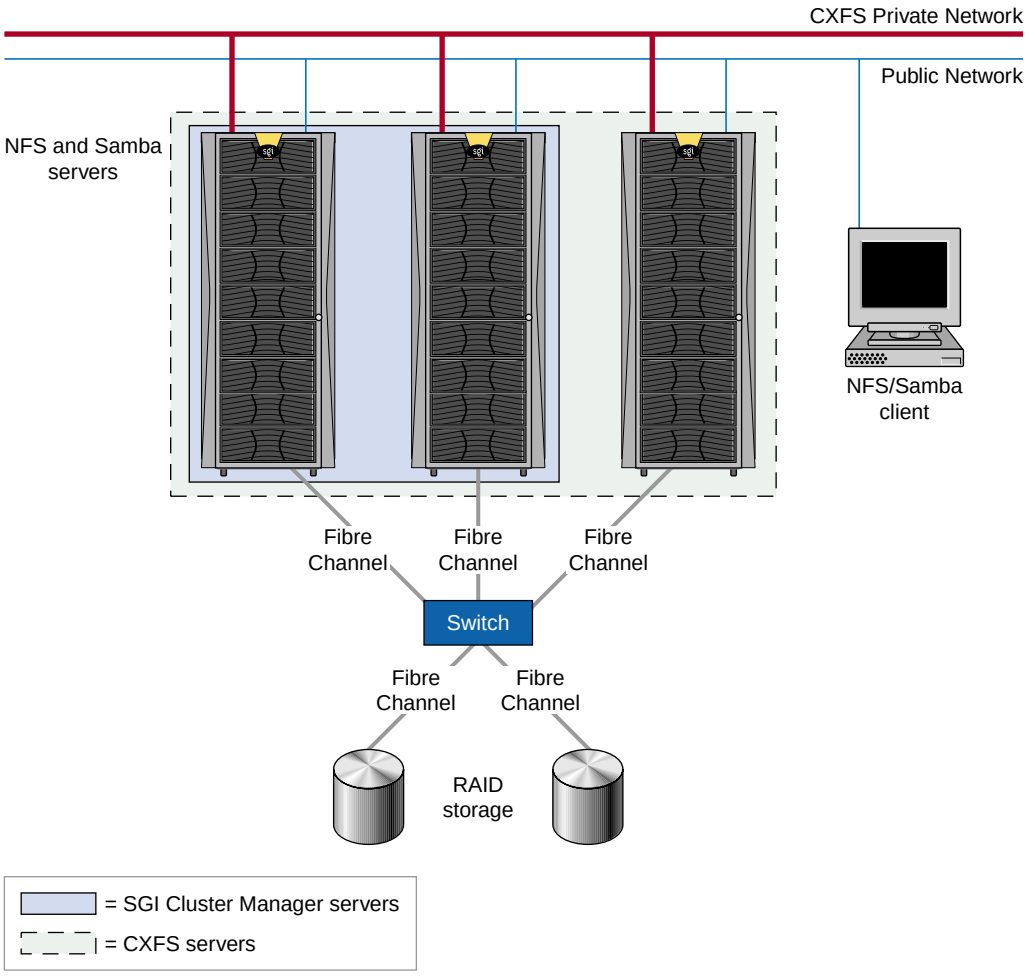

Figure 1-1 An Example CXFS and SGI Cluster Manager Configuration

Software Requirements

SGI Cluster Manager requires the following:

- SGI ProPack for Linux:
  - SGI ProPack 3 for Linux
  - SGI ProPack 3 Service Pack 1 for Linux
- Red Hat Enterprise Advanced Server 3.0

Note: The Linux virtual server (load-balancing software) that is part of Red Hat Cluster Suite is not supported on SGI Altix systems.

This release also supports the following releases:

- Samba 3.0 as shipped with Red Hat Enterprise Advanced Server 3.0
- CXFS 3.2 Altix server/client release

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

- DMF 3.0.1
- TMF 1.4.1

See "Software Packages" on page 19 and 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 two *failover options* that are considered when a failure occurs and a new target member for the service must be determined:

- *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 members that are not in domain if there is a failure and all members in the domain are 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 node to the original node when the original node is rebooted after a failure or maintenance period.

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

  If the failover is not ordered, a member from the list will be randomly chosen to run the service. If it fails, any other member from the list will be chosen.

Each failover domain also has a *failback option*, which is considered when a node rejoins the cluster. The *controlled failback* option says that a service will not be moved back to a 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 to manually relocate the service in order for it to run on the first member without an intervening failure. Only a new failure will cause a service to be automatically moved.

Suppose you have a cluster with two members, A and B. Table 1-1 describes some of the possible results from using various options under different circumstances for the `nfs` service.

Table 1-1 Failover Domain and Option Results

Failover Domain	Options	Circumstance	Results
(none)	(none)	Newly formed membership	The service will be started on any member in the cluster, 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 A.
B, A	(none)	Newly formed membership	The service will be started on one of the members A or B, randomly chosen. If that member is unavailable, the other will be used.
B	(none)	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.
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.
B, A	Restricted failover and controlled failback	Newly formed membership	The service will be started on either A or B, randomly chosen. If that member fails, the service restart on the other node 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 failover and controlled failback	The service is running on B and then B fails	The service will be started on A. The service remain on A even after B restarts. To go back to B, the system administrator must manually move the service.

Cluster Daemons

Following is an overview of the cluster daemons:

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

For more information, see the man pages.

Hardware Installation

This chapter discusses the following:

- "Shared Partitions"
- "Heartbeat Network"
- "Power Controllers" on page 12

Shared Partitions

SGI Cluster Manager for Linux **requires** two 10–MB partitions to keep membership quorum: the *primary partition* and the *shadow partition* (used for backup purposes). You should use the XSCSI raw device driver to access these partitions (do not use the Linux raw device driver).

SGI Cluster Manager supports SAN configurations using TP9300, TP9500, and TP9100 RAIDs. Each member in the cluster should be connected to storage using multiple paths so that service failovers are minimized. SGI recommends that the two shared partitions should be on different Fibre Channel (FC) controllers; ideally, they 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 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:

- *SGI® InfiniteStorage TP9400 and SGI® InfiniteStorage TP9500 and TP9500S RAID User's Guide*
- *SGI® InfiniteStorage TP9300 and TP9300S RAID User's Guide*
- *SGI TPSSM Administration Guide*

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 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.

Power Controllers

You must use SGI L2 system controllers for power control. Multiple members within a partitioned system may share a single L2 as long as the system serial number on each L1 is the same.

Network-based and serial-based power controllers are not supported for SGI Cluster Manager on SGI Altix servers.

For more information, see the following:

- *SGI Altix 3000 User's Guide*
- *SGI Altix 350 System User's Guide*

For information about configuring the power controller, see "Step 4: Add Power Controller Configuration" on page 28.

L2 Power Controller

The L2 is an optional product for the Altix 350 and Altix 3300 and must be purchased. An L2 is standard for each Altix 3700 rack. Use one of the following:

- Serial connection:
  - Altix 350: serial ports on Altix 350 with IO10 and a `IO10 CBL-SATA-SERIAL` multiport serial adapter cable. The customer must also order the `LS-BASE-IO` serial ATA (SATA) drive option.

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

Figure 2-1 shows the rear panel for an Altix 350.

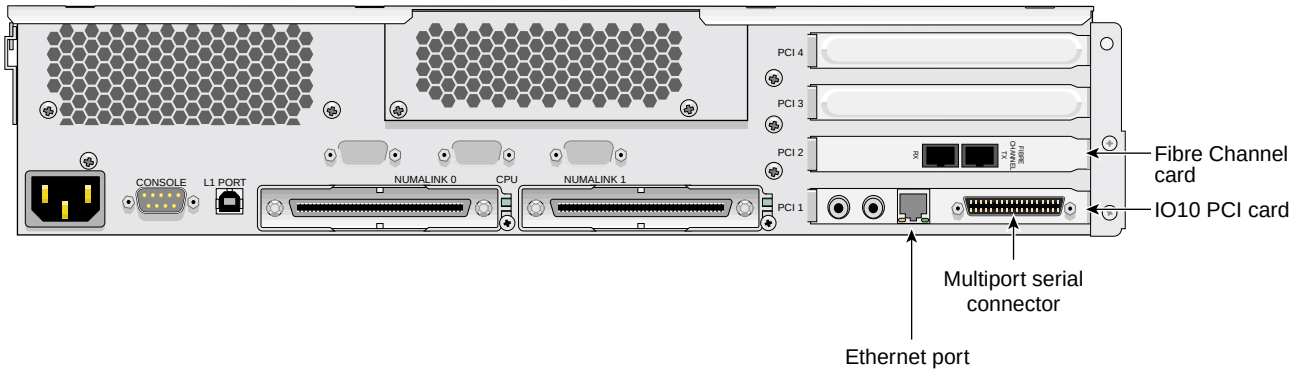

Figure 2-1 Altix 350 Rear Panel

- Altix 3300 and Altix 3700 should use 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.
- Requires the `l2` designation in the cluster database

Figure 2-2 on page 14 and Figure 2-3 on page 15 show the serial connections.

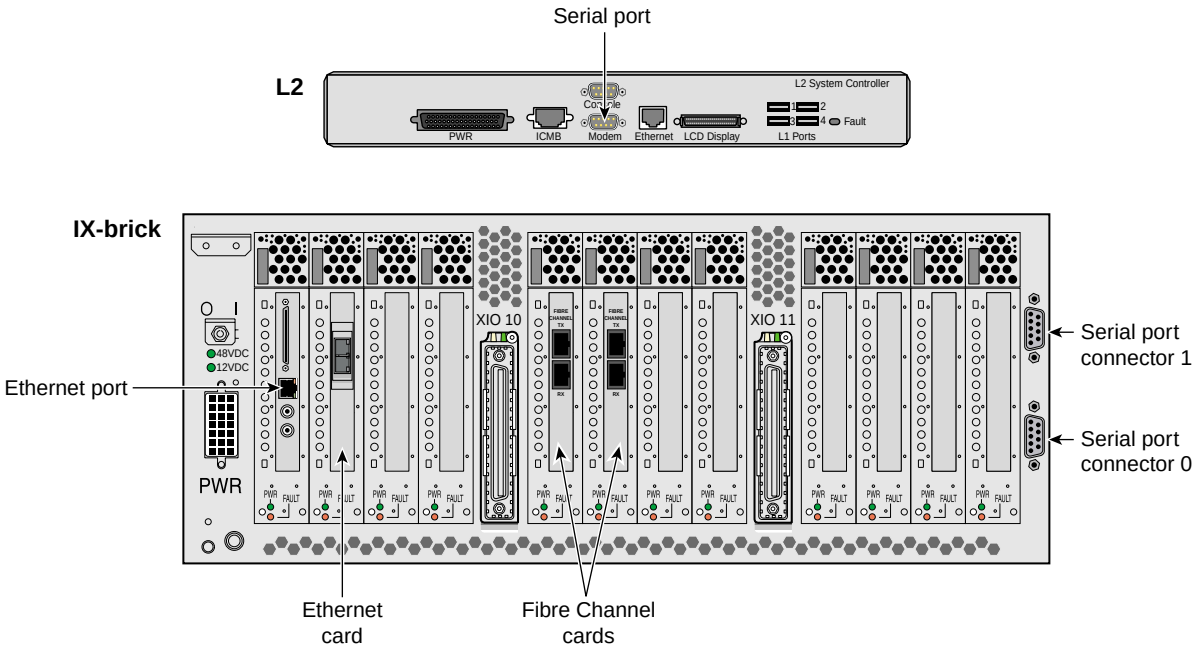

Figure 2-2 Altix 3000 Rear Panel

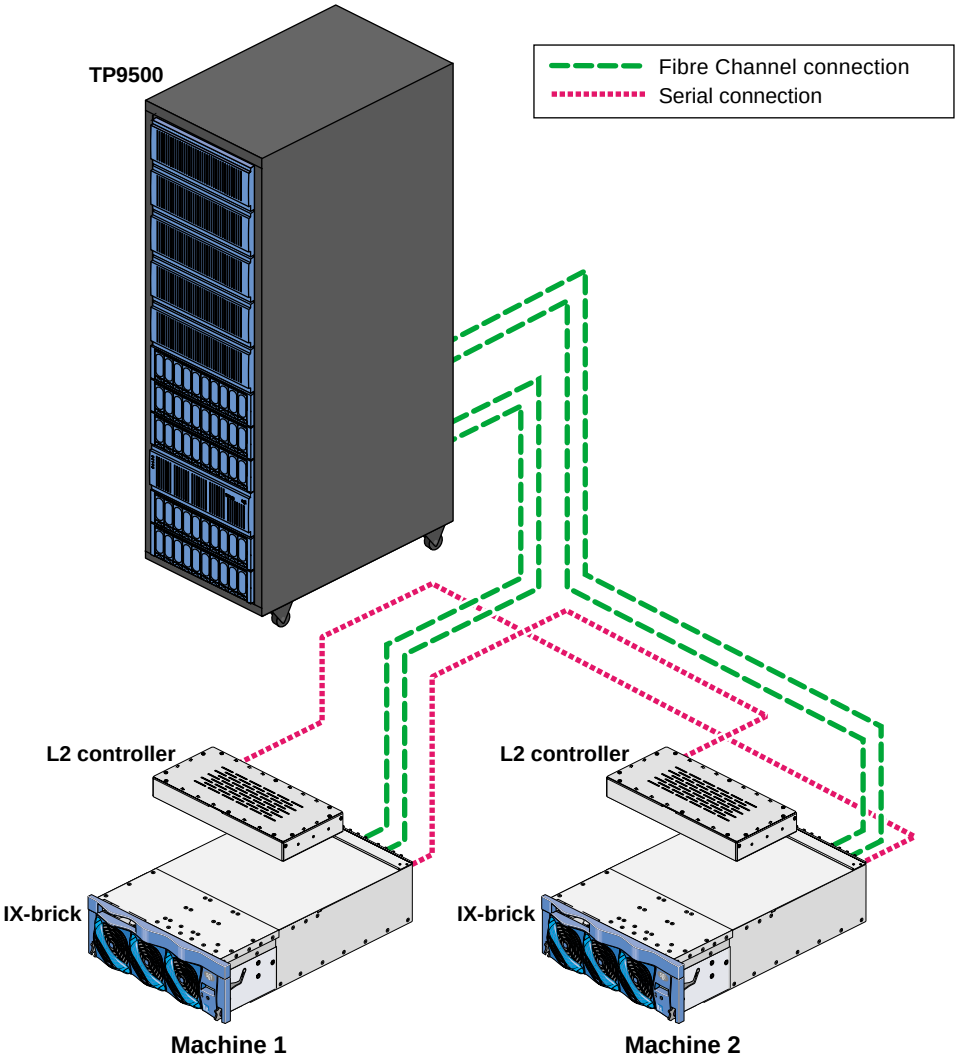

Figure 2-3 Altix 3300 L2 with Serial Cable Connection

- Ethernet connection:
  - An Ethernet network requires an Ethernet port on each member.
  - All members in the cluster and the L2 must be connected to the same network. For security reasons, SGI recommends a private network if the Ethernet method is used. If a private network is used, a PCI Ethernet card is required for each member.
  - Multiple members within a partitioned system may share a single L2 as long as the system serial number on each L1 is the same.
  - L2 via Ethernet is required if you are also running CXFS
  - Requires the `l2network` designation in the cluster database

  Figure 2-4 shows the Ethernet connection.

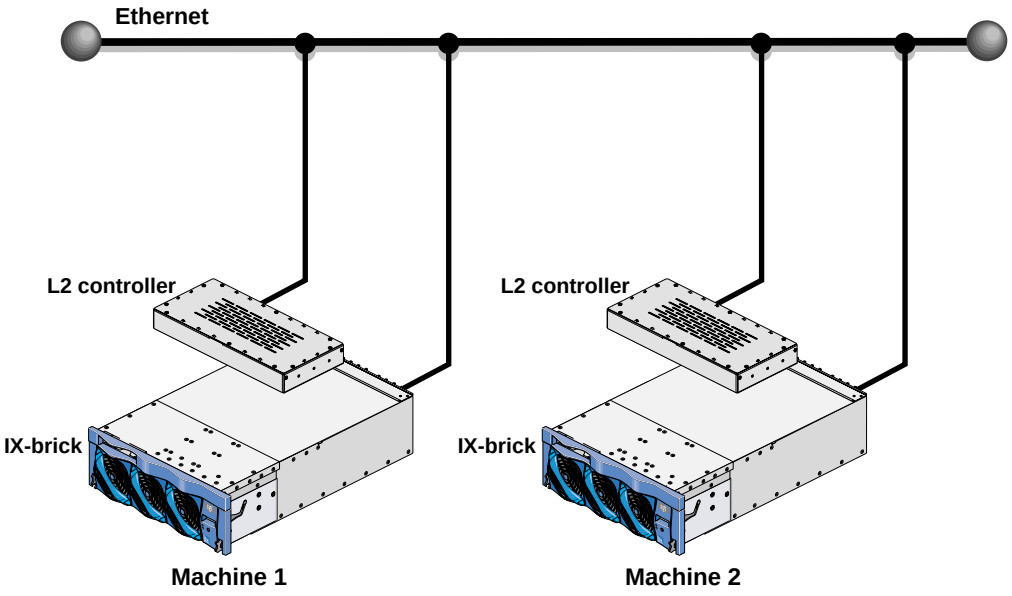

Figure 2-4 Altix 3300 L2 with an Ethernet Connection

Testing Serial Connectivity for the L2

To test the serial lines, do the following:

1. Create symlinks to the serial devices:

```
# ln -s /dev/ttyIOC4/0 /dev/ttyI0
# ln -s /dev/ttyIOC4/1 /dev/ttyI1
```

2. Use the `cu`(1) command to test the connection to the controller.

Note: By default, the `cu` program has a set user ID of `uucp`. Therefore, you must either change the permissions on the `tty` device file or change the ownership of the device file to `uucp` before running the `cu` command. For example:

```
# chown uucp.uucp /dev/ttyIOC4/0
```

The `cu` command requires the `tty` names to be in the following format:

`/dev/tty`*XXX*

You can use `/dev/ttyIOC4/0` to define power controllers in SGI Cluster Manager.

Note: You must use `parity=even`.

For example:

```
# cu -l /dev/ttyI0 -s 38400 --parity=even
Connected.

Jackhammer-001-L2>cfg
L2 192.0.1.133: - 001 (LOCAL)
L1 192.0.1.133:0:0    - 001c04.1
L1 192.0.1.133:0:1    - 001i13.1
L1 192.0.1.133:0:5    - 001c07.2
L1 192.0.1.133:0:6    - 001i02.2
```

For more information, see the `cu`(1) man page.

You can also use the `clufence`(8) command to test serial connectivity.

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: 192.0.2.70   netmask: 255.255.255.0   broadcast addr: 192.0.2.255
```

To change the IP setting to `63.154.16.7`:

```
l2-foo-001-L2> ip 63.154.16.7 255.255.255.0 63.154.16.255
```

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.acme.com
Trying 192.0.1.98...
Connected to l2-server.acme.com.
Escape character is '^]'.

Linux 2.4.7-sgil2 (192.0.1.98) (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 ]
  192.0.1.67  R2000016                     000 L3 controlle
  192.0.1.132 L1000487                     001 1.3.61
  192.0.1.96  N0000005  bar                002 1.24.2
  192.0.1.100 N0000005  bar2               003 1.24.2
  192.0.1.94  N0000005  bar3               004 1.24.2
  192.0.1.105 N0000005  bar_l2_2           001 1.22.0
server-001-L2>
```

Software Installation

This chapter describes how to install the SGI Cluster Manager for Linux software.

This chapter includes the following sections:

- "Software Packages"
- "Installing the Software" on page 20
- "Upgrading" on page 21
- "Uninstalling the Software" on page 22

Software Packages

The following packages are provided:

- Base product:
  - `clumanager-1.2.3-AS3.0sgi300`*XX*`.ia64.rpm`, which contains the basic monitoring and failover capabilities
  - `redhat-config-cluster-1.2.3-AS3.0sgi300`*XX*`.noarch.rpm`, which contains the configuration tools

    Note: The `redhat-config-cluster` RPM is dependent upon the `clumanager` RPM. You must uninstall `clumanager` first.

  - `rh-cs-en-3-AS3.0sgi300`*XX*`.noarch.rpm`, which contains the Red Hat documentation
  - `sgi-cluster-manager-docs-`*x.x*`-1.noarch.rpm`, which contains this SGI guide
- Optional high-availability plug-ins for CXFS, DMF, TMF, and local XVM: `clumanager-sgi-1.0.0`

Installing the Software

Do the following:

1. If necessary, upgrade to the supported level of SGI ProPack according to the directions in *SGI ProPack for Linux Start Here*. See "Software Requirements" on page 6.
2. Insert the *SGI Cluster Manager x.x for Linux — Base Product* CD and do the following to mount the CD and see its contents:

```
# mount /dev/cdrom /mnt/cdrom
# cd /mnt/cdrom
# ls
COPYING  README  RPM_MD5_SUMS  SGI  TRANS.TBL
```

   Read the README file to learn about any late-breaking changes in the installation procedure.

3. Install the software from the CD using the rpm(8) command:

```
# rpm -Uvh clumanager-1.2.3*.rpm redhat-config-cluster-1.0.0*.rpm rh-cs-en-3*rpm \
sgi-cluster-manager-docs*.rpm
      Preparing...                ########################################### [100%]
         1:clumanager             ########################################### [ 25%]
         2:redhat-config-cluster  ########################################### [ 50%]
         3:rh-cs-en               ########################################### [ 75%]
         4:sgi-cluster-manager-docs
```

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

4. If you have purchased the optional high-availability product for CXFS, DMF, TMF, and local XVM, insert the *SGI Cluster Manager x.x for Linux — Storage Software Plug-ins* CD. Do the following to mount the CD and see its contents:

```
# mount /dev/cdrom /mnt/cdrom
# cd /mnt/cdrom
# ls
COPYING  README  RPM_MD5_SUMS  SGI  TRANS.TBL
```

   Read the README file to learn about any late-breaking changes in the installation procedure.

5. Install the software from the CD:

```
# rpm -Uvh clumanager-sgi-1.0.0-AS3.0sgi300xx.ia64.rpm
```

Upgrading

To upgrade from the previous release, do the following:

1. Relocate all services to one member. For example:

   ```
   # clusvcadm -r service -m member
   ```

   For more information, see "Service Administration" on page 52.

2. Stop cluster daemons on the member to be upgraded. For example:

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

   For more information, see "Stopping Cluster Processes" on page 52.

3. Install the new software as described in "Installing the Software" on page 20.

4. Start the cluster daemons on the member. For example:

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

   For more information, see "Starting Cluster Processes" on page 51.

5. Relocate the services to the upgraded member if needed. For example:

   ```
   # clusvcadm -r service -m member
   ```

   For more information, see "Service Administration" on page 52.

6. Repeat steps 1 through 5 for the other member.

7. If you modified `/usr/lib/clumanager/create_device_links`, restore your modifications by copying the backup file that is automatically made:

```
# cp /usr/lib/clumanager/create_device_links.rpmsave /usr/lib/clumanager/create_device_links
```

Note: When you install the SGI Cluster Manager software, a new `/usr/lib/clumanager/create_device_links` file is installed. If the `create_device_links` file was modified, the changed file will be will be saved as `/usr/lib/clumanager/create_device_links.rpmsave`.

Uninstalling the Software

To uninstall the software, use the following command:

`rpm -e` *rpm_name*

Uninstalling the `clumanager` RPM will attempt to stop cluster daemons in the local node.

Note: The `redhat-config-cluster` RPM is dependent upon the `clumanager` RPM. You must uninstall `clumanager` first.

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

Configuration

This chapter provides an overview of the basic configuration process, plus specific information about SGI Cluster Manager for Linux that differs from the Red Hat Cluster Manager. For details, follow the information in Chapter 2, "Cluster Configuration," in the *Red Hat Cluster Suite: Configuring and Managing a Cluster* manual.

Cluster Configuration Tools

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

- Cluster Status graphical user interface (GUI): `redhat-config-cluster`
- Command-line interface (CLI): `redhat-config-cluster-cmd`

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 4-1 shows an example of the GUI.

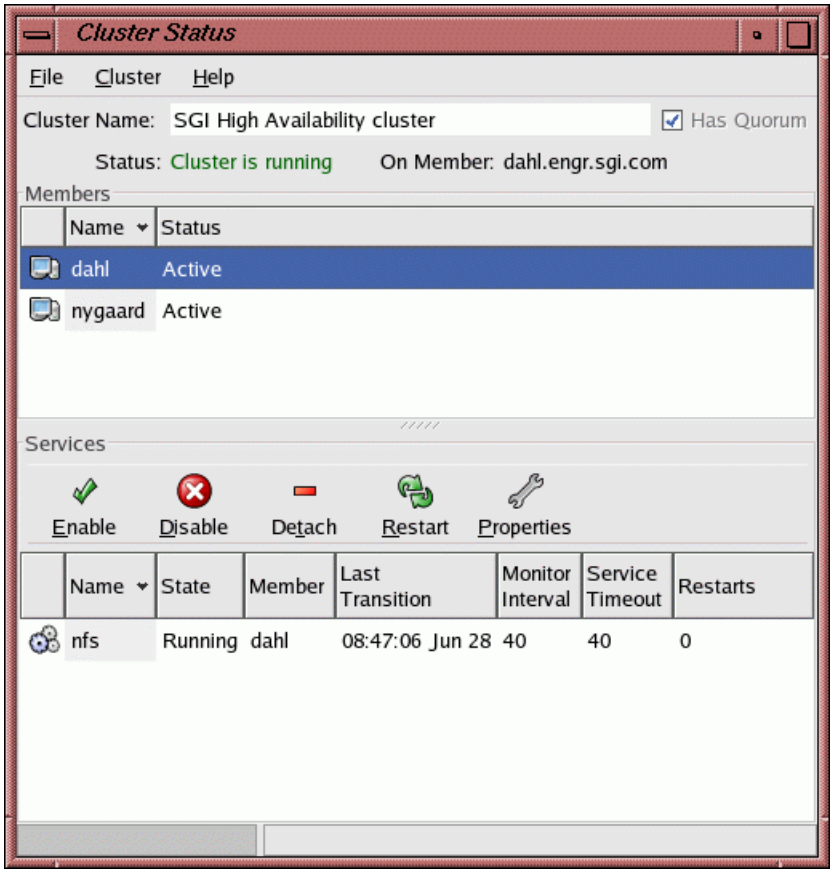

Figure 4-1 Cluster Status GUI

In the CLI, enter an argument without an = 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:

```
# redhat-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 partition.

Configuration Steps

This section discusses the configuration steps:

- "Step 1: Define the Shared Partitions" on page 26
- "Step 2: Create the Cluster" on page 27
- "Step 3: Define the Members" on page 28
- "Step 4: Add Power Controller Configuration" on page 28
- "Step 5: Change the Heartbeat Interval, Timeout, and Failover Speed" on page 32
- "Step 6: Set the Tiebreakers" on page 35
- "Step 7: Create the Failover Domain" on page 36
- "Step 8: Configure the Service" on page 38
- "Step 9: Add a Service IP Address" on page 40
- "Step 10: Add the Disk and Filesystem Information to the Service *(Optional)*" on page 41
- "Step 11: Add a Samba Share *(Optional)*" on page 42
- "Step 12: Define the NFS Information *(Optional)*" on page 42
- "Step 13: Save the Cluster Configuration *(GUI only)*" on page 43
- "Step 14: Synchronize Configuration Changes Across the Cluster" on page 43

- "Step 15: Verify that Configuration Changes are Synchronized" on page 43
- "Step 16: Start the Cluster Daemons" on page 44

Step 1: Define the Shared Partitions

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

- Ensure that the members have their disks attached identically.
- Create symbolic links (*symlinks*) on each member as appropriate. 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.

SGI recommends that the two shared partitions should be on different FC controllers; ideally, they should be on separate FC controllers at the front end, separate HBAs on the Altix, and on separate RAID 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 C, "Setting the Partition Type to `Linux`" on page 101.

For example, suppose you have the following output from the `hinv` command:

```
Integral SCSI controller pci04.01.0: Version Fibre Channel QLA2300 (Rev 1) pci04.01.0
    Disk Drive: unit   2 lun  0 on SCSI controller pci04.01.0  0
Integral SCSI controller pci05.01.0: Version Fibre Channel QLA2300 (Rev 1) pci04.01.0
    Disk Drive: unit   3 lun  0 on SCSI controller pci05.01.0  0
```

Partition 1 from disk on FC target 2 and FC target 3 are used for storing shared state. You could create symlinks to the raw XSCSI device names as follows (you must use character special devices):

```
# ln -s /dev/xscsi/pci04.01.0/target2/lun0/rpart1 /dev/shared_raw1
# ln -s /dev/xscsi/pci05.01.0/target3/lun0/rpart1 /dev/shared_raw2
```

You should add these symlink commands to the `/usr/lib/clumanager/create_device_links` script on the cluster member.

The shared raw partitions are `/dev/shared_raw1` and `/dev/shared_raw2`.

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

Cluster
> Shared State

In the CLI:

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

For example:

```
# redhat-config-cluster-cmd --sharedstate --type=raw  \
--rawprimary=/dev/shared_raw1 --rawshadow=/dev/shared_raw2
```

You should perform this step before defining the cluster ("Step 2: Create the Cluster" on page 27).

Note: Do not use the Red Hat Linux `raw`(8) interface for storing shared state. Disregard the information provided in sections 1.2.5, 1.4.4, and 1.4.6 of *Red Hat Cluster Suite: Configuring and Managing a Cluster*.

Instead, you should use the XSCSI raw partition device file and the `create_device_links` script. Use symlinks if the raw partition XSCSI device names on both members are not identical.

Files under `/dev` are re-created when the member reboots. If you are creating symlinks from devices to files under the `/dev` directory, you must create symlinks in the following script:

`/usr/lib/clumanager/create_device_links`

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:

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

Step 3: Define the Members

To define a member in the GUI, do the following:

- Select the following in the **Cluster Status** window:

  Cluster
  > Configure

- Click the **Members** tab.
- Click **New**.
- Provide the hostname to be used for communication.
- Disable the software watchdog.

Note: The software watchdog is not supported by the SGI Cluster Manager.

In the CLI (the watchdog is disabled by default):

```
redhat-config-cluster-cmd --add_member membername
```

Note: During the installation process, the Red Hat installer sets the hostname IP address to `127.0.0.1` in the `/etc/hosts` file unless you supply the hostname using the fully qualified domain name **and also** have a working primary domain name server that can resolve this name.

If `/etc/hosts` contains the default `127.0.0.1` address, you must change it to the actual IP address.

Step 4: Add Power Controller Configuration

For each member, you must provide information about its power controller. The SGI Cluster Manager supports the L2 system controller using either serial cables or Ethernet cables. (The **Serial** and **Network** power controllers shown in the GUI are not

supported by SGI Cluster Manager.) You can specify only one power controller for each member.

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

- **Type**: the power controller type of the node being defined (the local member):
  - `l2` for an L2 using serial cables (default in the GUI).
  - `l2network` for an L2 using the Ethernet connection. Altix partitions can share the same L2 in this configuration. You must use `l2network` if you are also running CXFS.

  This is the `type` field 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 . The default value in the GUI is `/dev/ttyIOC4/0`.

  This is the `device` field 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` field in the CLI.

Figure 4-2 shows an example in the GUI for an L2 using serial cables and Figure 4-3 shows an example for an L2 using the Ethernet network.

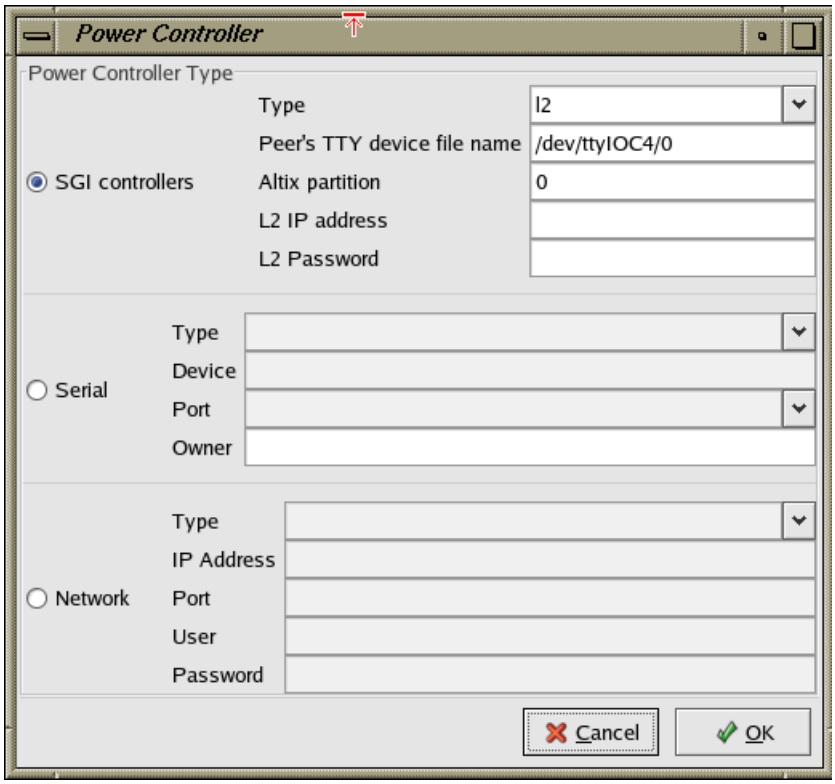

Figure 4-2 Configuring the Power Controller Information for an L2 using Serial Cables

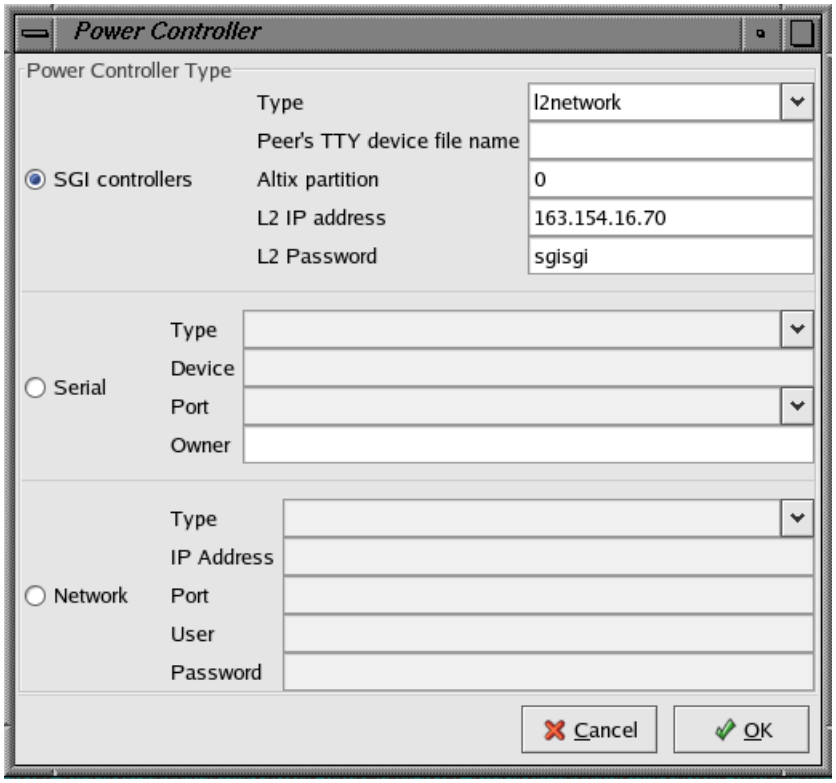

Figure 4-3 Configuring the Power Controller Information for an L2 using an Ethernet Network

In the CLI:

```
redhat-config-cluster-cmd --member=membername \
                          --add_powercontroller \
                          --type=l2|l2network \
                                required only for l2network:
                                --ipaddress L2_IPaddress
                                --password L2_password_(if_defined)
                                --partition Altix_partition_ID
                          --device=/dev/ttyIOC4/0 \
                          --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 example, the following defines an L2 using the Ethernet method:

```
# redhat-config-cluster-cmd --member=member1  --add_powercontroller \
--type=l2network --ipaddress=190.2.0.100 --partition=3 --password=foo
```

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

```
# redhat-config-cluster-cmd --member=member1  --add_powercontroller \
--type=l2
```

For hardware information, see "Power Controllers" on page 12.

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 4-4. The GUI provides 15 seconds as the default failover speed value.
4. You can choose to enable either broadcast hearbeating or multicast hearbeating.

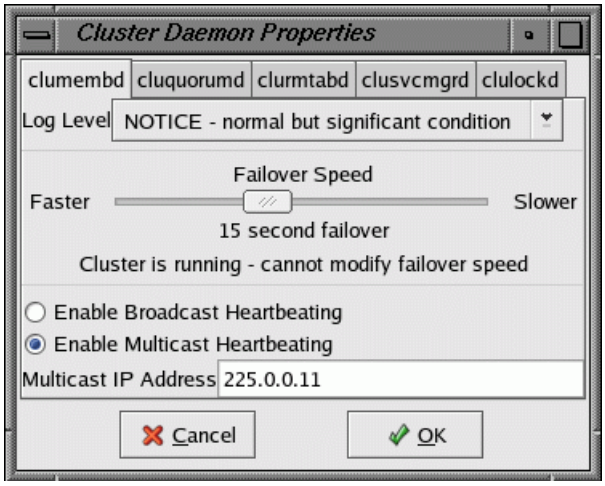

Figure 4-4 Adjusting Failover Speed

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 4-1 shows the failure detection times and parameter values that are supported.

Table 4-1 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:

```
# redhat-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:

```
# redhat-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 `redhat-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 both 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).

  Note: You must verify that the network tiebreaker can be accessed by the `ping` 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 nodes cannot talk to each other, they look at the status on the shared partition disk to decide which node 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 4-5 shows an example of the **cluquorumd** window.

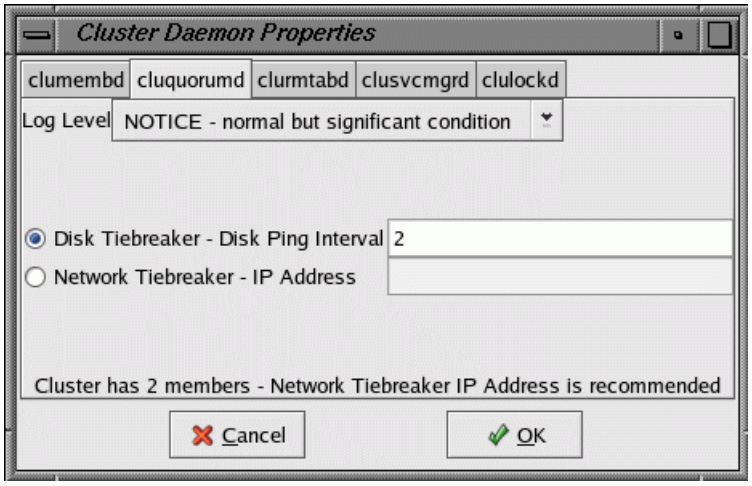

Figure 4-5 Tiebreakers

In the CLI:

```
redhat-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 6.

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 6.

Figure 4-6 shows an example.

Figure 4-6 Failover Domain

In the CLI:

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

redhat-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.
- Failover domain name (see "Step 7: Create the Failover Domain" on page 36).
- Monitor interval (in seconds).
- Service timeout (in seconds), which is common for all actions (start, stop, and status check) that apply to the service.

  Note: You cannot specify timeouts for each resource within the service.

- 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 another node. 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.

  In this field, you can specify a directory that contains scripts, or an individual script. The scripts contain shell functions to fail over user applications. This directory or a script is specified as service parameters.

The script name must be called `svclib_`*application*. This 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 60.

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 4-7 shows an example of configuring an NFS high-availability service.

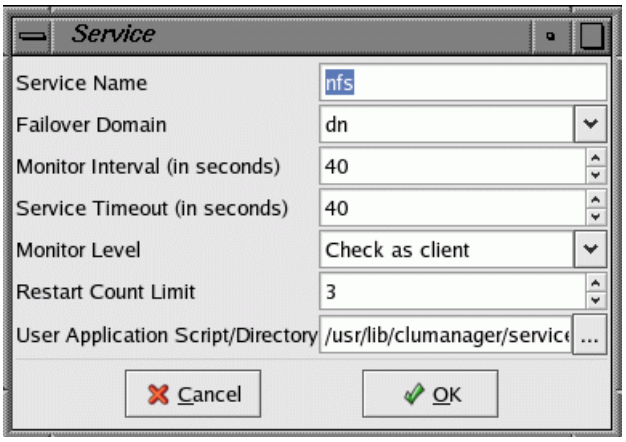

Figure 4-7 Configuring a High-Availability Service

In the CLI:

```
redhat-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:

```
redhat-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 the 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
   - Filesystem type (`xfs` or `cxfs` if using the CXFS plug-in)
   - Mount options

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

In the CLI:

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

redhat-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.

In the GUI **Cluster Configuration** window:

1. Select the following:

   Add Exports
   > Samba

2. Use the **Samba Druid** to enter the required information.

In the CLI:

```
redhat-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.

In the GUI **Cluster Configuration** window:

1. Select the following:

   Add Exports
   > Samba

2. Use the **NFS Druid** to enter the required information.

In the CLI:

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

redhat-config-cluster-cmd --service=servicename \
```

```
--device=path \
--nfsexport=exportpath \
--add_client \
--name=* \
--options=options
```

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 23. Select the following from the **Cluster Configuration** window:

File
> Save

Step 14: Synchronize Configuration Changes Across the Cluster

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, such as by running `scp`.

After making configuration changes, you must verify that the configuration files across the cluster are in synchronization. To do this, you can verify that they have the same configuration file version number (`config_viewnumber`). For example:

```
# redhat-config-cluster-cmd  --cluster
cluster:
  name = test-cluster
  config_viewnumber = 42
```

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 the `service clumanager start` command (or its equivalent `/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 5, "Administration" on page 49.

Example Cluster Configuration

The following example uses `redhat-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 `167.58.9.2` 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 will be used by clients to access the Samba share and NFS export point
- The service is allowed to restart 4 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:

```
# redhat-config-cluster-cmd --sharedstate --type=raw --rawprimary=/dev/shared_raw1 \
--rawshadow=/dev/shared_raw2
```

2. Create the cluster:

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

3. Define the members:

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

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

4. Add power controller information for the members (`167.58.9.2` is the IP address of the L2):

```
# redhat-config-cluster-cmd --member=member1 --add_powercontroller --type=l2 \
--device=/dev/ttyIOC4/0 --partition=0

# redhat-config-cluster-cmd --member=member2 --add_powercontroller --type=l2network \
--ipaddress=167.58.9.2 --partition=3
```

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

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

6. Set up a network tiebreaker for the cluster:

```
# redhat-config-cluster-cmd --cluquorumd --tiebreaker_ip=192.0.2.245
```

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

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

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

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

8. Create the service definition:

```
# redhat-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:

```
# redhat-config-cluster-cmd --service=service1 --add_service_ipaddress \
--ipaddress=163.154.17.200 --netmask=255.255.255.0 \
--broadcast=163.154.17.255
```

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

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

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

11. Add a Samba share name:

```
# redhat-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:

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

# redhat-config-cluster-cmd --service=service1 --device=/dev/shared1 \
--nfsexport=/shared1/export_dir --add_client \
```

```
--name=* --options=ro
```

13.

Note: 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:

```
# redhat-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 node 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 `redhat-config-cluster-cmd`(8) man page.

Administration

See Chapter 7, “Cluster Administration” in the *Red Hat Cluster Suite: Configuring and Managing a Cluster* manual.

This section discusses the following:

- "Monitoring Status"
- "Displaying Service Information" on page 50
- "Starting Cluster Processes" on page 51
- "Stopping Cluster Processes" on page 52
- "Service Administration" on page 52
- "Cluster Service States" on page 53
- "Message Logging" on page 55

Monitoring Status

To monitor status, use the following:

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

Figure 5-1 shows an example of the GUI.

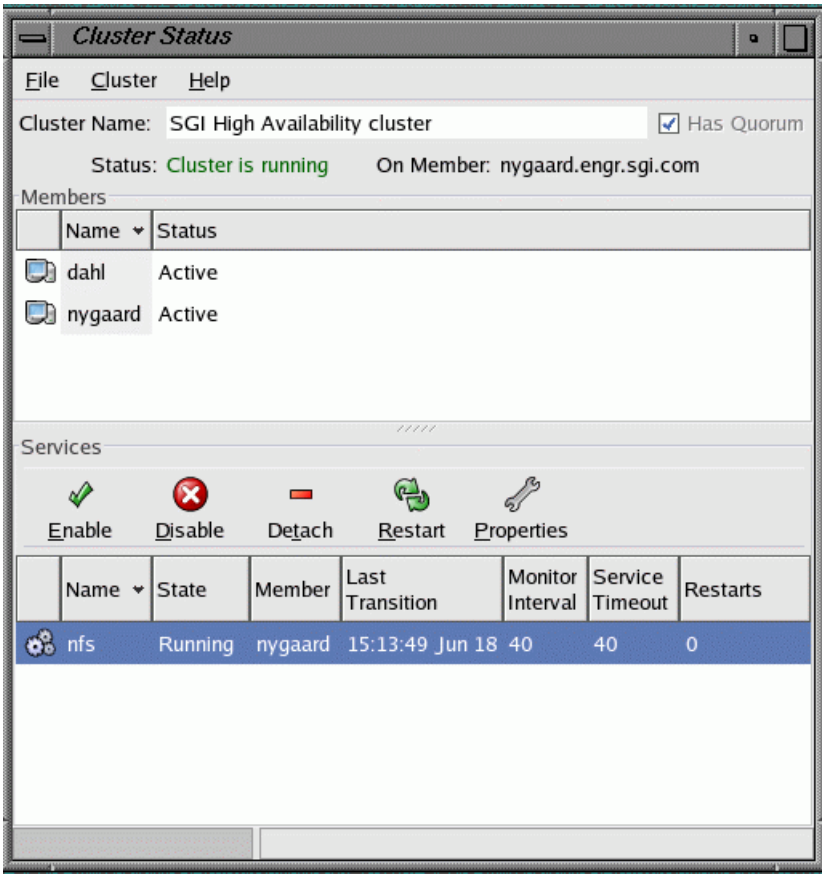

Figure 5-1 Status

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:

`redhat-config-cluster-cmd --service=`*servicename*

For example:

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

Figure 5-2 shows an example of the status window.

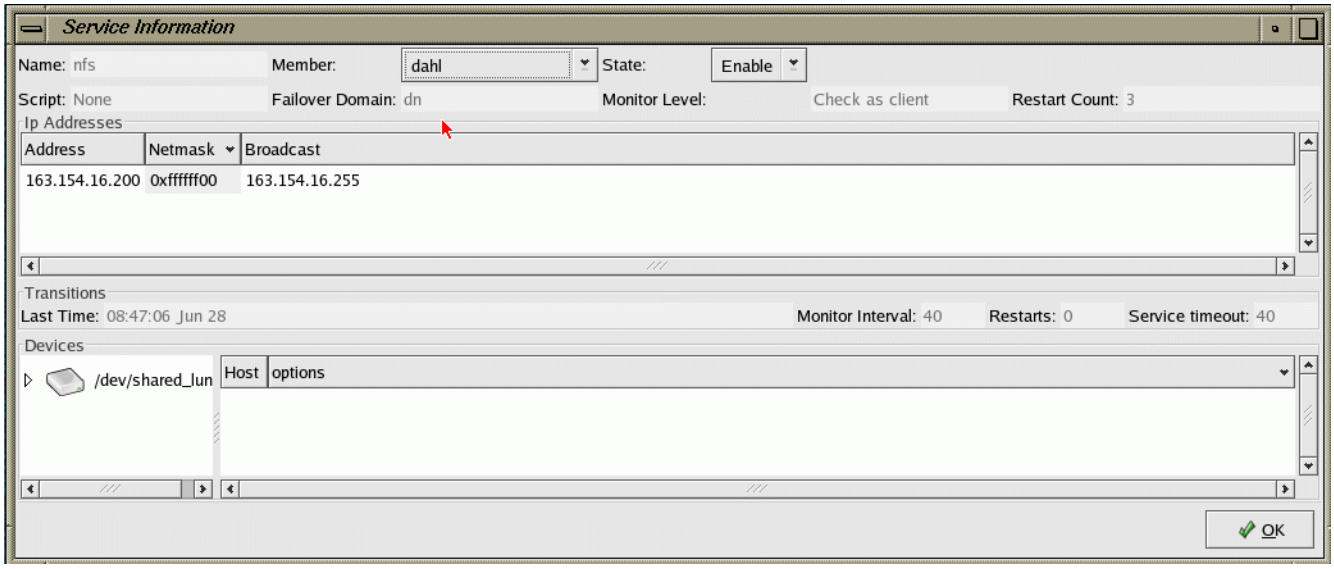

Figure 5-2 Service Information

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 use drag and drop to relocate services.

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*

  You could also drag and drop the service icon into the target node icon in the **Cluster Status** GUI window.

- 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 5-3.

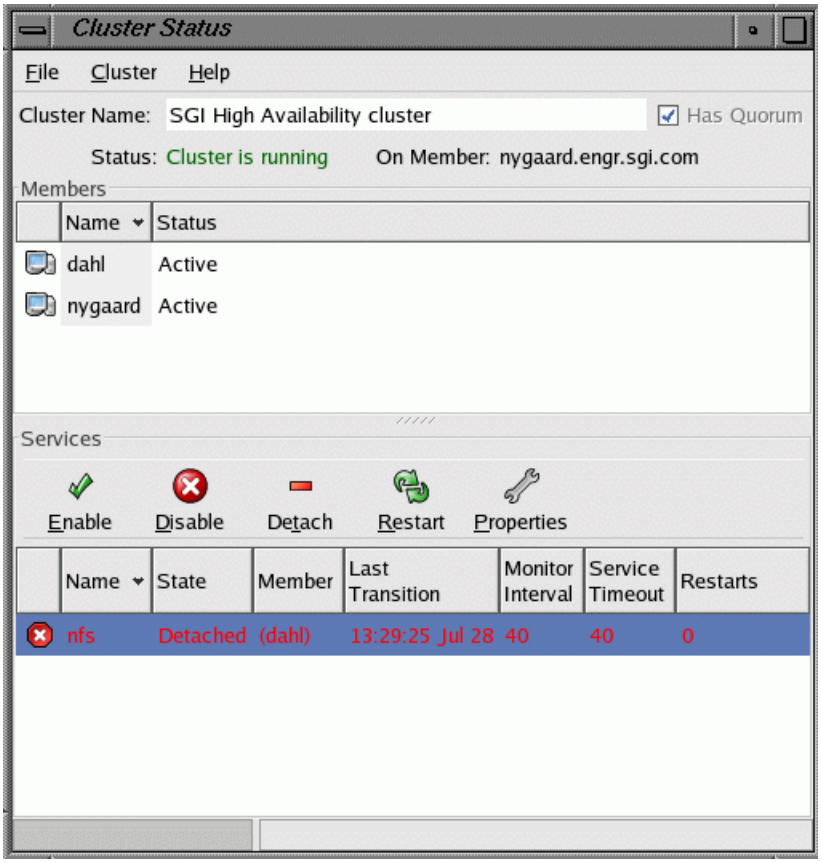

Figure 5-3 Detached State

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 disable 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"
- "The `service` Script"
- "Adding a Service" on page 58
- "Example of Failing Over Multiple User Applications" on page 60
- "Sample User Application Script" on page 60

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 `service` script takes 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 `service` script runs 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 should be called `svclib_apache` 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 38.

Figure 6-1 shows the GUI screen to create a service. To get to this window, select the **Services** tab from the **Cluster Configuration** window.

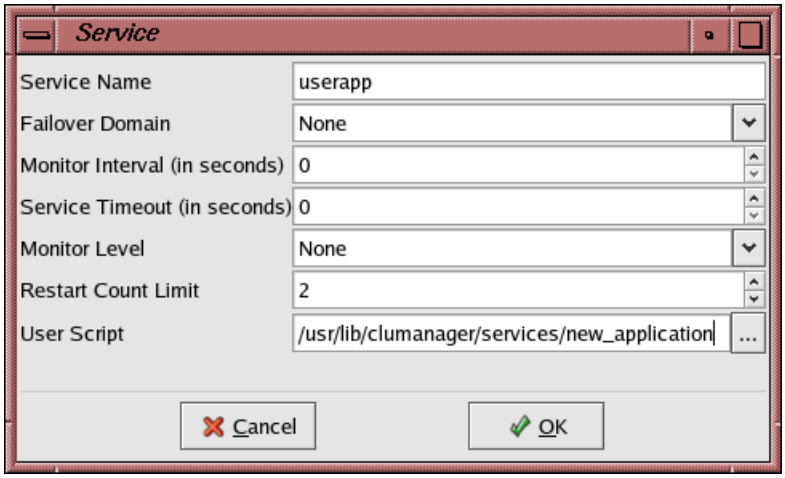

Figure 6-1 Creating a Service

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

```
# redhat-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, using the `svclib_`*application* name format. For example, they could be named `svclib_apachesvc1` and `svclib_mySQLsvc1`:

  – `svclib_apachesvc1` should contain a shell function `apachesvc1()` that takes an action and service ID as parameter. This function should perform start/stop/status operation on the apache server for `service1`.

  – `svclib_mySQLsvc1` should contain a shell function `mySQLsvc1()` 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 `svclib_test`. This example script must be called `svclib_test` and 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

The SGI Cluster Manager for Linux supports Samba as shipped with SGI ProPack for Linux. See "Software Requirements" on page 6 for the supported levels.

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.

The locations are as follows:

- PID directory: *mountpoint*`/.samba/`*sharename*`/pid`
- Log directory: `/var/log/samba`
- Lock directory: *mountpoint*`/.samba/`*sharename*`/locks`
- Password file: *mountpoint*`/.samba/`*sharename*`/private/smbpasswd`

For the order in which Samba is started/stopped, see Chapter 6, "Creating a New Highly Available Application" on page 57. For information about service monitoring levels, see "Step 8: Configure the Service" on page 38.

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 x.x for Linux - Storage Software Plug-ins* CD and the supported level of CXFS (see "Software Requirements" on page 6).

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

All CXFS metadata servers must also be members of the SGI Cluster Manager cluster.

Note: To use CXFS relocation and fail over the service from one member to another, you must set the `relocation_ok` parameter to `1` (enable) on all potential CXFS metadata servers as follows:

```
# echo 1 > /proc/sys/fs/cxfs/cxfs_relocation_ok
```

For more information about relocation support, see *CXFS Administration Guide for SGI InfiniteStorage*.

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 Power Controller" on page 12.

To include CXFS filesystems in the SGI Cluster Manager configuration, add filesystems as devices used by a service, as shown in Figure 8-1.

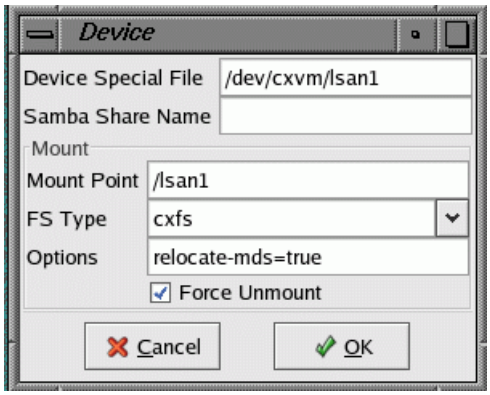

Figure 8-1 Adding a CXFS Filesystem as a Device

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:

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

redhat-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.

For more information, see "Step 10: Add the Disk and Filesystem Information to the Service *(Optional)*" on page 41.

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.

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`.

For the order in which CXFS is started/stopped, see Chapter 6, "Creating a New Highly Available Application" on page 57.

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 x.x for Linux — Storage Software Plug-ins* CD and the supported level of DMF (see "Software Requirements" on page 6).

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

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`:

```
# redhat-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 38.

DMF Administrative Filesystems and Directories

To run DMF, you must configure the parameters shown in Table 9-1. 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 9-1 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

Parameter	Status	Description
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` 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 as local XVM filesystems or as CXFS filesystems. SGI Cluster Manager ensures that the DMF plug–in script is called after the necessary filesystems are mounted.

If you use local XVM filesystems, you must define them as instructed in "Configuring DMF for Local XVM Filesystems" on page 70.

If you use CXFS filesystems, you must define them as instructed in "Configuring DMF for CXFS Filesystems " on page 71.

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 41.
3. Configure the SGI Cluster Manager local XVM volumes using the local XVM plug-in. See Chapter 11, "Local XVM Plug-In" on page 83.

Configuring DMF for CXFS Filesystems

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 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 41.
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 8, "CXFS Plug-In" on page 65.

Start/Stop Order

For the order in which DMF is started/stopped, see Chapter 6, "Creating a New Highly Available Application" on page 57.

Ensuring that Only SGI Cluster Manager Starts DMF

After installing `clumanager-sgi-1.0.0`, you should perform the following actions on each member of the cluster in the failover domain. These commands ensure that DMF can only be started via SGI Cluster Manager:

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

Using TMF with DMF

To use the Tape Management Facility (TMF) with DMF in a 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 10, "Tape Management Facility (TMF) Failover Script" on page 73.

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`:

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

To do this with the GUI, see "Step 8: Configure the Service" on page 38.

Tape Management Facility (TMF) Failover Script

Using the Tape Management Facility (TMF) with SGI Cluster Manager for Linux requires the value-add SGI product on the *SGI Cluster Manager 3.0 for Linux — Storage Software Plug-ins* CD and the supported level of TMF (see "Software Requirements" on page 6). Only Storage Technology Corporation (STK) hardware controlled by the Automated Cartridge System Library Software (ACSLS) software is supported.

For more information about TMF, see the *TMF Administrator's Guide*. For the order in which TMF is started/stopped, see Chapter 6, "Creating a New Highly Available Application" on page 57.

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 80.

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.

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 76.

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 76
- "The `loader` Directive " on page 77
- "The `remote_devices` Directive" on page 78

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 78).

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 must be `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:

```
# /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`, this will be a

comma-separated four-digit string listing the ACS, LSM, drive panel, and drive (for example, `0,0,1,3`).

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

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
```

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`:

```
redhat-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 x.x for Linux — Storage Software Plug-ins* CD.

Note: XVM in **cluster** mode is supported only with CXFS. See Chapter 8, "CXFS Plug-In" on page 65.

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 both 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 `redhat-config-cluster` GUI or the `redhat-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`:

```
# redhat-config-cluster-cmd --service=nfs1 \
--device=/dev/lxvm/m0

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.engr.sgi.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 11-1 shows an example in the GUI.

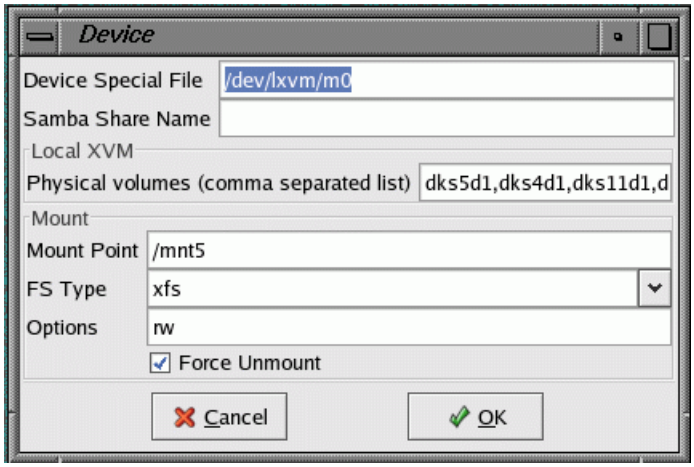

Figure 11-1 Adding an XVM Device

For more information on configuration, see "Step 8: Configure the Service" on page 38.

For the order in which local XVM is started/stopped, see Chapter 6, "Creating a New Highly Available Application" on page 57.

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

Troubleshooting

This chapter provides information about the following:

- "Best Practices"
- "Recovery from a `clulockd` Failure"
- "Watchdog Errors" on page 88
- "Shared Partitions" on page 89
- "State Inconsistencies" on page 91
- "Serial cable or Reset issues" on page 92
- "Error Messages" on page 92
- "Reporting Problems to SGI" on page 93

Best Practices

If you run into problems, do the following:

- Check the messages in `/var/log/messages` (see "Message Logging" on page 55)
- Use `shutil` to see if shared partitions are accessible
- Use `clufence` to check the status of the reset cable
- Verify that the failover domain is defined correctly

Recovery from a `clulockd` Failure

If the `clulockd` daemon dies unexpectedly, it freezes all of the locks on the shared partition. `clulockd` will log a message such as 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 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` man page.

Watchdog Errors

Software and hardware watchdog timers are not supported. If you enable software watchdog when configuring a member, you will see the following error messages when 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]
```

Disable the software watchdog.

Shared Partitions

This section discusses the following:

- "Verify Raw Devices are Character Special Devices"
- "Verify Accessibility"
- "Read the Configuration File"
- "Verify Metadata Information is Consistent" on page 90
- "Write the Configuration File" on page 91
- "Displaying Metadata Remotely" on page 91
- "Last Resort: Clear Information" on page 91

Verify Raw Devices are Character Special Devices

Use the following command to verify that the `rawprimary` and `rawshadow` devices are character special device (such as `rpart1`) :

`ls -lL` ***share_raw_device***

Using block special devices (such as `part1`) will lead to inconsistencies in cluster, member, and node states. The cluster appears to work but communication between members is affected. For an example, see "Verify Metadata Information is Consistent" on page 90.

See "Step 1: Define the Shared Partitions" on page 26.

Verify Accessibility

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

```
shutil
```

Read the Configuration File

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

```
shutil -r -
```

You should use this command to compare the configuration files in the shared 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. See "Verify Raw Devices are Character Special Devices" on page 89.

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

Displaying Metadata Remotely

To display the metadata information from the shared 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
```

State Inconsistencies

If you encounter state inconsistencies between members, verify that the shared raw devices are character special devices (such as `rpart1`) and not block special devices (such as `part1`). See "Verify Raw Devices are Character Special Devices" on page 89.

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 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/ttyIOC4/0 controls jackhammer
[12314] info: STONITH: Power controller l2 connected to peer's /dev/ttyIOC4/0 controls jackhammer2
[12314] err: STONITH: Device at /dev/ttyIOC4/0 controlling jackhammer2 FAILED status check:
Timed out
```

Error Messages

Following are common error messages.

```
Raw device file names must be defined.
```

An attempt was made to define the cluster before defining the shared state. You must define the shared state first. See "Step 1: Define the Shared Partitions" on page 26.

```
Raw device file names primary /dev/raw/raw1 and shadow /dev/raw/raw43 are not valid.

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

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

```
[12314] err: STONITH: Device at /dev/ttyIOC4/0 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 92.

Reporting Problems to SGI

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

- Output from the following commands:

  ```
  hinv
  chkconfig --list
  shutil -r -
  rpm -qa
  uname -a
  ps -ef | grep clu
  clustat
  ls -l each_shared_partition
  ls -lL each_shared_partition
  clufence -s other_members
  exportfs    (in NFS configurations)
  mount
  cxfs_dump    (in CXFS configurations)
  ```

- Contents of the following files:

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

Differences Between Red Hat Cluster Manager and SGI Cluster Manager

Table A-1 summarizes the differences between Red Hat Cluster Manager and SGI Cluster Manager. (You should not install Red Hat Cluster Manager on an SGI Altix system. Red Hat Cluster Manager is intended only for Linux 32–bit machines. Installing Red Hat Cluster Manager on an SGI Altix server will result in conflicts with SGI Cluster Manager.)

Table A-1 Red Hat Cluster Manager and SGI Cluster Manager

Topic	Red Hat Cluster Manager	SGI Cluster Manager for Linux
Service timeouts	Not supported.	Supported.
Check/monitor interval	Supports a check interval, which does not include execution time. That is, the check starts at time *t*, and the next check will be at *t+check_interval*, irrespective of execution time.	Supports a monitor interval. The next check will be done after *execution time + monitor interval*.
Software/hardware watchdog timers	Supported.	Not supported. Do not enable the software watchdog when configuring a member. If you enable software watchdog, you will see error messages when cluster daemons are started. For more information, see "Watchdog Errors" on page 88.

Topic	Red Hat Cluster Manager	SGI Cluster Manager for Linux
Samba service	Does not keep Samba files on shared storage. Locations: • Process ID (PID) directory: `/var/run/samba/`*netbiosname* • Log directory: `/var/log/samba` • Lock directory: `/var/cache/samba` • Password file: `/etc/samba/smbpasswd`	Keeps Samba files in the shared partition and in the log file on the local disk. The lock directory is not removed during failover. Locations: • PID directory: *mountpoint*`/.samba/`*sharename*`/pid` • Log directory: `/var/log/samba` • Lock directory: *mountpoint*`/.samba/`*sharename*`/locks` • Password file: `/.samba/`*sharename*`/private/smbpasswd` For more information, see Chapter 7, "Samba Plug-In" on page 63.
Power control	Network– or serial-based power controllers. There can be multiple controllers defined for a given member.	SGI L2 system controllers on SGI Altix servers There can be only one controller defined for a given member.
Shared partition quorum errors	Reboots the member.	Local cluster daemons exit and wait for the member to be reset by other members in the cluster.

FailSafe and SGI Cluster Manager

Table B-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 B-1 Differences Between FailSafe and SGI Cluster Manager

Topic	FailSafe	SGI Cluster Manager
Operating system	IRIX	Red Hat Advanced Server for Linux with SGI ProPack
Terminology	node resource	member application
Size of cluster	8 nodes	2 members
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	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. 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 43.
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.
Private network	Supports multiple networks for heartbeats and control messages and can failover from one network to another. Networks can be dedicated (private) or public networks.	Not supported. SGI Cluster Manager uses the public network (host names) for heartbeats and control messages.
Action scripts	Separate scripts named `start`, `stop`, `monitor`, `restart`, `exclusive`.	A `bash` script that contains `start`, `stop`, and `status` parameters (see Chapter 6, "Creating a New Highly Available Application" on page 57). 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`.

Topic	FailSafe	SGI Cluster Manager
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.
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`

To set a partition type to `Linux`, use `fdisk` command as follows:

1. Use the `t` subcommand to change a partition's system ID.
2. Enter hexadecimal code `83`, which represents `Linux`.
3. Enter `w` to write the change and exit the `fdisk` tool.

For example, to change partition 1 to `Linux`, do the following (some output is truncated):

```
# fdisk /dev/xscsi/pci04.01.0/target3/lun0/disc

The number of cylinders for this disk is set to 69424.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)
```

```
Command (m for help): p

Disk disc: 72.7 GB, 72796340224 bytes
64 heads, 32 sectors/track, 69424 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

Device Boot    Start       End    Blocks   Id  System
 part1             1        11     11248    5  Extended
 part2            12        22     11264   83  Linux
 part3            23     19096  19531776   83  Linux
 part4         19097     69424  51535872    5  Extended
 part5         19097     38170  19531760   83  Linux
 part6         38171     57244  19531760   83  Linux
 part7         57245     69424  12472304   83  Linux

Command (m for help): t
Partition number (1-7): 1
Hex code (type L to list codes): L

...
 5  Extended        40  Venix 80286     83  Linux           c7  Syrinx
 ...
Hex code (type L to list codes): 83

Command (m for help): w
```

To verify the partition type, use the p subcommand. For example:

```
Command (m for help): p

Disk disc: 72.7 GB, 72796340224 bytes
64 heads, 32 sectors/track, 69424 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

Device Boot    Start       End    Blocks   Id  System
 part1             1        11     11248   83  Linux
 part2            12        22     11264   83  Linux
 part3            23     19096  19531776   83  Linux
 part4         19097     69424  51535872    5  Extended
 part5         19097     38170  19531760   83  Linux
 part6         38171     57244  19531760   83  Linux
 part7         57245     69424  12472304   83  Linux
```

Glossary

CLI

Command line interface (`redhat-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 (`redhat-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, another 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.

IX brick

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

IO10

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

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 a member and not shared across the cluster.

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

Another member in the cluster that to which the local system controller is connected.

primary partition

One of the two raw disk partitions where SGI Cluster Manager keeps configuration, cluster, and service status information. See also *shared partition* and *shadow partition*.

plug-in

The set of software that allows a service to be highly available without modifying the application itself.

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 raw disk partitions 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 raw disk partitions (primary partition and shadow partition) where SGI Cluster Manager keeps configuration, cluster, and service status information.

split-brain scenario

Network partition in which both 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

A

action scripts, 98
actions in a service script, 57
administration, 49
ALERT message level, 55
Altix servers, 3
application, 57, 97
application-specific scripts, 58

B

base product, 2
bash, 99
best practices, 87
block special devices, 89

C

cables, 3
CACHE_DIR, 70
CDs, 19
character special devices, 89
check interval, 95
chkconfig, 44, 47, 71, 93
CLI, 23
clufence, 87, 92, 93
clulockd, 9, 87
clumanager, 19, 22, 44, 47, 51, 52
clumembd, 9, 33
cluquorumd, 9
clurmtabd, 9
clustat, 93
cluster configuration
  See "configuration", 23
cluster creation, 27, 45
cluster daemons, 9
cluster database, 98
cluster global lock manager daemon, 9
cluster membership daemon, 9
cluster process, 51, 52
cluster quorum daemon, 9
cluster remote NFS mount table daemon, 9
cluster service manager daemon, 9
cluster status GUI, 23
cluster.xml, 43, 47
clusvcadm, 52
clusvcmgrd, 9, 57, 88
command-line interface, 23
config_viewnumber, 43, 47
configuration, 42, 46
  cluster, 27, 45
  disks and filesystems, 41, 46
  example, 44
  failover domain, 36, 46
  failover speed, 32, 45
  heartbeat interval, 32, 45
  members, 28, 45
  power controller, 28, 45
  Samba share, 42, 46
  save cluster configuration, 43
  service, 38, 46
  service IP address, 40, 46
  shared partitions, 26, 45
  start cluster daemons, 44, 47
  status, 24
  steps, 25
  synchronize changes, 43, 47
  tiebreaker, 35, 45
  timeout, 32, 45
  tools, 23
configuration file, 91

connectivity test, 17
control messages, 12, 98
controlled failback, 7
create_device_links, 11, 27
CRIT message level, 55
cu, 17
CXFS, 65, 69, 73
  optional product, 2
  version requirements, 6
cxfs_dump, 93

D

daemons, 9
DB9 serial ports, 13
DEBUG message level, 55
dependencies, 59, 99
detached state, 54
/dev files and symlinks, 26
device driver, 11
device special file, 66
disabled state, 53
disk device naming, 26
disk tiebreaker, 35
disks, 41
dmaudit, 70
DMF, 69
  CXFS and, 71
  existing service, 69
  local XVM and, 70
  optional product, 2
  parameters, 69
  starting, 71
  TMF and, 72
  version requirements, 6
domain, 6, 36, 46
dual paths, 11

E

EMERG message level, 55
ERROR message level, 55
error messages, 92
/etc/cluster.xml, 25, 43, 47, 88, 91, 93, 98
/etc/init.d/clumanager, 27, 51, 52
/etc/samba/passwd, 96
/etc/samba/smb.conf*, 93
/etc/tmf/sgicm_tmf.config, 76
Ethernet connection, 16
exclusive, 99
exportfs, 93

F

failback option, 7
failed state, 54
failover, 1
failover domain, 6, 36, 46, 87
failover speed, 32, 45
FailSafe differences, 97
failure detection times, 34
fdisk, 101
filesystems, 41, 46
FS type, 66
FTP_DIRECTORY, 70

G

global lock manager daemon, 9
graphical user interface for configuration, 23
GUI, 23

H

hardware
  diagrams, 13

installation, 11
supported, 3
heartbeat interval, 32, 45
heartbeat network, 12
heartbeats, 9
helper_tmp script, 73
highly available applications, 57
hinv, 26, 93
HOME_DIR, 69

I

INFO message level, 55
installation
hardware, 11
software, 19
interval parameter, 33
IO10, 3
IO9, 13
ip, 18
IP address for service, 40, 46
IRIX FailSafe differences, 97
IX brick, 4

J

JOURNAL_DIR, 69

L

L2 power controller, 4, 12, 28, 45
L2 system controllers, 96
l2network, 16
Linux raw device driver, 11
Linux virtual server, 6
load-balancing software, 6
loader directive, 77
local XVM, 83
DMF, 70
local4 facility, 55
lock directory for Samba, 96
lock manager daemon, 9
log directory for Samba, 96
log levels, 55
logrotate, 55
lowest-ordered, 7
ls, 93

M

member, 97
member definition, 28, 45
member state inconsistencies, 91
members, 1
membership daemon, 9
message levels, 55
message logging, 55
messages, 92
metadata consistency among members, 90
metadata display, 91
monitor interval, 95
monitor level, 38
monitor levels, 38, 46
monitoring status, 49
mount, 93
mount point and CXFS, 66
mount table daemon, 9
mount the CD, 20
MOVE_FS, 70
multiple CXFS filesystems, 67
multiple user applications, 60
multiport serial adapter cable, 3

N

network cabling, 3
network heartbeats, 9
network tiebreaker, 35

network-based power controllers, 96
networks for heartbeat and control, 98
new applications, 57
NFS, 42, 46
NFS mount table daemon, 9
node, 97
NOTICE message level, 55

O

ordered failover, 7

P

packages, 19
parity setting, 17
partition size, 11
partition type, 101
password file for Samba, 96
pending state, 53
PID directory for Samba, 96
plug-in, 2
power controller, 96
  See "L2 power controller", 4
primary partition, 11
private network, 12, 98
ps, 93
public network, 98

Q

quorum daemon, 9

R

RAIDs supported, 11
raw device driver, 11
raw device filenames, 92
raw interface caution, 27
raw partitions, 26, 45, 89
README, 6, 20
Red Hat Cluster Manager
  differences, 95
  documentation, 1
Red Hat Cluster Suite, 6
Red Hat Enterprise Advanced Server, 6
redhat-config-cluster, 23, 25
redhat-config-cluster-cmd, 23
reinitialize the shared state, 88
relocate-mds, 66, 71
relocation_ok, 65
remote modem port, 13
remote NFS mount table daemon, 9
remote_devices directive, 78
removing software, 22
reporting problems to SGI, 93
requirements
  hardware, 3
  software, 6
reset cable
  status, 87
reset daemon, 9
resource, 97
resource dependencies, 99
resource directive, 76
restart, 99
restricted failover, 7
rolling upgrade, 97
rpm, 20, 93
RPMs, 6, 19
running state, 53

S

Samba, 63
  configuration, 42, 46
  version requirements, 6
samba, 96

save cluster configuration, 43
scp, 43
script location, 98
script order, 58
serial cable, 92
  problem, 92
serial ports, 4
serial-based power controllers, 96
servers supported, 3
service ID, 57
service IP address, 40, 46
service manager, 57
service manager daemon, 9
service script, 57
service states, 53
service timeout, 81, 95
service timeouts, 38, 46
service timeouts and CXFS, 67
SGI ProPack for Linux, 6
SGI ProPack version required, 6
sgicm_tmf.config, 76
shadow partition, 11
share name, 42, 46
shared partition quorum errors, 96
shared disks, 11
shared partitions, 87, 92
shared raw partitions, 89
shared state, 26, 45
shutil, 87–89
software installation, 19
software packages, 19
software requirements, 6
software watchdog, 28
special devices, 89
split-brain scenario, 35
SPOOL_DIR, 69
start order, 58
state inconsistencies, 91
status, 49
  configuration, 24
stop order, 58
stopped state, 53
STORE_DIRECTORY, 70
svclib_<application> user script, 60
symlinks, 17, 26
syslog, 55
system controller
  See "L2 power controller", 4
system controller problem
  problem, 92

T

Tape Management Facility
  See "TMF", 73
tapes and TMP, 79
TCP, 1
tiebreaker, 35, 45, 97
timeout, 32, 45, 99
tko_count parameter, 33
TMF
  configuration file, 76
  device group, 75
  failover script, 80
  helper_tmf script, 73
  loader directive, 77
  optional configuration specifications, 75
  optional product, 2
  remote_devices directive, 78
  resource directive, 76
  service timeout, 81
  tape configuration, 79
  version requirements, 6
TP9xxx RAID, 11
troubleshooting, 87
tty port, 13

U

UDP, 1
uname, 93

uninitialized state, 53
uninstalling the software, 22
upgrade, 97
user application script parameter, 59
/usr/lib/clumanager/create_device_links, 11, 26, 27
/usr/lib/clumanager/services, 58
/usr/lib/clumanager/services/new_application, 59
/usr/lib/clumanager/services/service, 98
/usr/lib/clumanger/create_device_links, 93

V

/var/cache/samba, 96
/var/cluster/ha/resource_types, 98
/var/lib/nfs/rmtab, 9
/var/log/messages, 55, 87, 93
/var/log/samba, 96
/var/run/samba/<netbiosname>, 96

W

WARNING message level, 55
watchdog, 28
watchdog errors, 88
watchdog timers, 95

X

XSCSI device names, 26
XSCSI raw device driver, 11
XVM, 6
XVM (local), 83
  DMF, 70