Sun HPC ClusterTools 3.0 Administrator's Guide: With LSF

901 San Antonio Road
Palo Alto, , CA 94303-4900
USA 650 960-1300 Fax 650 969-9131

Part No: 805-6280-10
June 1999, Revision A

Contents

Preface

The *Sun HPC ClusterTools 3.0 Administrator's Guide: With LSF* discusses various Sun HPC-specific administration issues that are not addressed in the LSF Batch documentation. In particular, it describes `hpc.conf`, a Sun HPC configuration file that extends the definition of Sun HPC system attributes beyond the parameters defined in the LSF configuration files.

Before reading this guide, system administrators and users should read the *LSF Batch Administrators Guide*, version 3.2, which is supplied with the LSF software.

Using Solaris Commands

This document may not contain information on basic Solaris™ commands and procedures, such as shutting down the system, booting the system, and configuring devices.

See one or more of the following for this information:

- AnswerBook™ online documentation for the Solaris 2.x and Solaris 7 software environment
- Other software documentation that you received with your system

Typographic Conventions

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output.	Edit your `--login` file. Use `ls --a` to list all files. `% You have mail.`
`AaBbCc123`	What you type, when contrasted with on-screen computer output.	`% su` `Password:`
AaBbCc123	Book titles, new words or terms, words to be emphasized. Command-line variable; replace with a real name or value.	Read Chapter 6 in the *User's Guide.* These are called *class* options. You *must* be `root` to do this. To delete a file, type `rm` *filename.*

Shell Prompts

TABLE P–2 Shell Prompts

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

Related Documentation

TABLE P–3 Related Documentation

Application	Title	Part Number
All	*Sun HPC ClusterTools 3.0 Read Me First*	805-6281-10
All	*Sun HPC ClusterTools 3.0 Product Notes*	805-6262-10
SCI	*Sun HPC SCI Guide*	805-6263-10
Installation	*Sun HPC ClusterTools 3.0 Installation Guide*	805-6264-10
Installation	*LSF 3.2 Installation Guide*	805-6265-10
Sun MPI Programming	*Sun MPI 4.0 User's Guide: With LSF*	805-7230-10
Sun MPI Programming	*Sun MPI 4.0 Programming and Reference Guide*	805-6269-10
Prism	*Prism 6.0 User's Guide*	805-6277-10
Prism	*Prism 6.0 Reference Manual*	805-6278-10
Sun S3L	*Sun S3L 3.0 Programming and Reference Guide*	805-6275-10
LSF	*LSF Batch Administrator's Guide*	805-6257-10
LSF	*LSF Batch User's Guide*	805-6258-10
LSF	*LSF Parallel User's Guide*	805-6259-10
LSF	*LSF Programmer's Guide*	805-6260-10

TABLE P–3 Related Documentation *(continued)*

Sun Documentation on the Web

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

`http://docs.sun.com`

CHAPTER 1

Introduction

This manual contains Sun HPC–specific system administration information that is not available in the *LSF Batch Administrator's Guide*.

Sun HPC system administrators should read the LSF documentation first and then read this manual to learn about issues not covered in the LSF documentation set.

The following list summarizes the topics covered in this manual:

- Chapter 2 explains how to edit the configuration file `hpc.conf`. This configuration file supplements the LSF configuration files described in the *LSF Batch Administrator's Guide*, providing Sun HPC–specific configuration data.
- Chapter 3 discusses some Sun HPC–specific LSF Batch queue-tuning issues.
- Chapter 4 describes the Sun Parallel File System (PFS) and includes a discussion of parallel file system parameters that can influence configuration decisions. If you will not be using PFS file systems, you can safely ignore this chapter.
- Chapter 5 explains how to start and stop PFS daemons. Again, if your system does not implement PFS file systems, ignore this chapter.
- Appendix A provides instructions for installing the Sun HPC ClusterTools packages using the command line instead of the graphical user interface.
- Appendix B describes the Cluster Console Manager (CCM), a set of tools that allow many administration tasks to be performed from a single window.

The balance of this chapter provides an overview of the Sun HPC ClusterTools 3.0 release.

Sun HPC System Overview

A Sun HPC ClusterTools 3.0 system can be a single Sun SMP (symmetric multiprocessor) server or a cluster of these SMPs running both the Sun HPC ClusterTools 3.0 software and LSF Base, Batch, and Parallel software.

Sun HPC System Hardware

A Sun HPC system configuration can range from a single Sun SMP (symmetric multiprocessor) server to a cluster of SMPs connected by any Sun-supported, TCP/IP-capable interconnect.

Note - An individual SMP server within a Sun HPC cluster is referred to as a *node*.

The recommended interconnect technology for clustering Sun HPC servers is the Scalable Coherent Interface (SCI). SCI's bandwidth and latency characteristics make it the preferred choice for the cluster's primary network. An SCI network can be used to create Sun HPC clusters with up to four nodes.

Larger Sun HPC clusters can be built using a Sun-supported TCP/IP interconnect, such as 100BaseT Ethernet or ATM. Individual parallel Sun HPC jobs can have up to 1024 processes running on as many as 64 nodes.

Any Sun HPC node that is connected to a disk storage system can be configured as a Parallel File System (PFS) I/O server. See Chapter 4 and Chapter 5 for additional information about PFS I/O servers and PFS file systems.

Sun HPC ClusterTools 3.0 Software and LSF Suite 3.2.3 Software

Sun HPC ClusterTools 3.0 software is an integrated ensemble of parallel development tools that extend Sun's network computing solutions to high-end distributed-memory applications. The Sun HPC ClusterTools products are teamed with LSF Suite 3.2.3, Platform Computing Corporation's resource management software.

The Sun HPC ClusterTools 3.0 software runs under Solaris 2.6 or Solaris 7 (32-bit or 64-bit).

Load Sharing Facility

LSF Suite 3.2.3 is a collection of resource-management products that provide distributed batch scheduling, load balancing, job execution, and job termination services across a network of computers. The LSF products required by Sun HPC ClusterTools 3.0 software are: LSF Base, LSF Batch, and LSF Parallel.

- LSF Base – Provides the fundamental services upon which LSF Batch and LSF Parallel depend. It supplies cluster configuration information as well as the up-to-date resource and load information needed for efficient job allocation. It also supports interactive job execution.
- LSF Batch – Performs batch job processing, load balancing, and policy-based resource allocation.
- LSF Parallel – Extends the LSF Base and Batch services with support for parallel jobs.

Refer to the *LSF Administrator's Guide* for a fuller description of LSF Base and LSF Batch and to the *LSF Parallel User's Guide* for more information about LSF Parallel.

LSF supports the concept of *interactive batch* execution of Sun HPC jobs as well the conventional batch method. Interactive batch mode allows users to submit jobs through the LSF Batch system and remain attached to the job throughout execution.

Sun MPI and MPI I/O

Sun MPI is a highly optimized version of the Message-Passing Interface (MPI) communications library. Sun MPI implements all of the MPI 1.2 standard as well as a significant subset of the MPI 2.0 feature list. For example, Sun MPI provides the following features:

- Integration with Platform Computing's Load Sharing Facility (LSF).
- Support for multithreaded programming.
- Seamless use of different network protocols; for example, code compiled on a Sun HPC system that has a Scalable Coherent Interface (SCI) network, can be run without change on a cluster that has an ATM network.
- Multiprotocol support such that MPI picks the fastest available medium for each type of connection (such as shared memory, SCI, or ATM).
- Communication via shared memory for fast performance on clusters of SMPs.
- Finely tunable shared-memory communication.
- Optimized collectives for symmetric multiprocessors (SMPs).
- Prism support – Users can develop, run, and debug programs in the Prism programming environment.
- MPI I/O support for parallel file I/O.
- Sun MPI is a dynamic library.

Sun MPI and MPI I/O provide full F77, C, and C++ support and Basic F90 support.

Parallel File System

Sun HPC ClusterTools's Parallel File System (PFS) component provides high-performance file I/O for multiprocess applications running in a cluster-based, distributed-memory environment.

PFS file s closely resemble UFS file s, but provide significantly higher file I/O performance by striping files across multiple PFS I/O server nodes. This means the time required to read or write a PFS file can be reduced by an amount roughly proportional to the number of file server nodes in the PFS file .

PFS is optimized for the large files and complex data access patterns that are characteristic of parallel scientific applications.

Prism

Prism is the Sun HPC graphical programming environment. It allows you to develop, execute, debug, and visualize data in message-passing programs. With Prism you can

- Control program execution, such as:
  - Start and stop execution.
  - Set breakpoints and traces.
  - Print values of variables and expressions.
  - Display the call stack.
- Visualize data in various formats.
- Analyze performance of MPI programs.
- Control entire multiprocess parallel jobs, aggregating processes into meaningful groups, called process sets or *psets*.

Prism can be used with applications written in F77, F90, C, and C++.

Sun S3L

The Sun Scalable Scientific Subroutine Library (Sun S3L) provides a set of parallel and scalable functions and tools that are used widely in scientific and engineering computing. It is built on top of MPI and provides the following functionality for Sun MPI programmers:

- Vector and dense matrix operations (level 1, 2, 3 Parallel BLAS).
- Iterative solvers for sparse s.
- Matrix-vector multiply for sparse s.
- FFT
- LU factor and solve.
- Autocorrelation.
- Convolution / deconvolution.
- Tridiagonal solvers.
- Banded solvers.
- Eigensolvers.
- Singular value decomposition.
- Least squares.
- One-dimensional sort.
- Multidimensional sort.
- Selected ScaLAPACK and BLACS application program interface.
- Conversion between ScaLAPACK and S3L.
- Matrix transpose.
- Random number generators (linear congruential and lagged Fibonacci).
- Random number generator and I / O for sparse s.
- Matrix inverse.
- Array copy.
- Safety mechanism.
- An array syntax interface callable from message-passing programs.
- Toolkit functions for operations on distributed data.
- Support for the multiple instance paradigm (allowing an operation to be applied concurrently to multiple, disjoint data sets in a single call).
- Thread safety.
- Detailed programming examples and support documentation provided online.

Sun S3L routines can be called from applications written in F77, F90, C, and C++.

SunCompilers

The Sun HPC ClusterTools 3.0 release supports the following Sun compilers:

- Sun Compilers C / C++ 4.2 (also included in Sun Visual WorkShop C++ 3.0)

- Sun WorkShop Compilers Fortran 4.2 (also included in Sun Performance WorkShop Fortran 3.0)
- Sun Visual WorkShop C++ 5.0
- Sun Performance WorkShop Fortran 5.0

Cluster Console Manager

The Cluster Console Manager is a suite of applications (`cconsole`, `ctelnet`, and `crlogin`) that simplify cluster administration by enabling the administrator to initiate commands on all nodes in the cluster simultaneously. When invoked, the selected Cluster Console Manager application opens a master window and a set of terminal windows, one for each node in the cluster. Any command entered in the master window is broadcast to all the nodes in the cluster. The commands are echoed in the terminal windows, as are messages received from the respective nodes.

Switch Management Agent

The Switch Management Agent (SMA) supports management of the Scalable Coherent Interface (SCI), including SCI session management and various link and switch states.

CHAPTER 2

hpc.conf – Sun HPC System Configuration File

This chapter discusses the Sun HPC configuration file `hpc.conf`, which defines attributes of Sun HPC clusters that are not defined in any LSF configuration files. A single `hpc.conf` file is shared by all the nodes in a cluster. It resides in the LSF shared directory `LSF_SHARED_LOC`.

The configuration file `hpc.conf` is organized into five sections which are summarized below and illustrated in Code Example 2–1.

- The `ShmemResource` section defines certain shared memory attributes. See “`ShmemResource` Section” on page 9 “`ShmemResource` Section” on page 9for details.
- The `Netif` section lists and ranks all network interfaces to which Sun HPC nodes are connected. See “`Netif` Section” on page 10 “`Netif` Section” on page 10for details.
- The `MPIQOptions` section allows the administrator to control certain MPI parameters by setting them in the `hpc.conf` file. See “`MPIOptions` Section” on page 13 “`MPIOptions` Section” on page 13 for details.
- The `PFSFileSystem` section names and defines all parallel file systems in the Sun HPC cluster. See “`PFSFileSystem` Section” on page 17 “`PFSFileSystem` Section” on page 17for details.
- The `PFSServers` section names and defines all parallel file system servers in the Sun HPC cluster. See “`PFSServers` Section” on page 19 “`PFSServers` Section” on page 19for details.
- The `HPCNodes` section can be used to define an HPC cluster that consists of a subset of the nodes contained in the LSF cluster. See “`HPCNodes` Section” on page 20 “`HPCNodes` Section” on page 20for details.

The `hpc.conf` file follows the formatting conventions of LSF configuration files. That is, each configuration section is bracketed by a `Begin/End` keyword pair and, when a parameter definition involves multiple fields, the fields are separated by spaces.

Sun HPC ClusterTools 3.0 software is distributed with an `hpc.conf` template, which you can edit to suit your particular configuration requirements. This chapter provides instructions for editing each part of `hpc.conf`.

Note - When any changes are made to `hpc.conf`, the system should be in a quiescent state. To ensure that it is safe to edit `hpc.conf`, shut down the LSF Batch daemons. See Chapter 3 “Managing LSF Batch” in the *LSF Batch Administrator’s Guide* for instructions on stopping and starting LSF Batch daemons.

CODE EXAMPLE 2–1 General Organization of hpc.conf File

```
Begin ShmemResource
  :     End ShmemResource

Begin Netif
NAME    RANK    MTU    STRIPE    PROTOCOL    LATENCY    BANDWIDTH
  :      :       :        :         :           :           :
End Netif

Begin MPIOptions queue=hpc
  :
End MPIQOptions

Begin HPCNodes
  :
End HPCNodes

Begin PFSFileSystem=pfs1 NODE            DEVICE            THREADS
  :               :                  :
End PFSFileSystem

Begin PFSServers
NODE            BUFFER_SIZE
  :               :                End PFSServers

Begin HPCNodes
 :
End HPCNodes
```

Note - If changes need to be made to the `PFSFileSystem` or `PFSServers` sections, all PFS file systems must be unmounted first. This requirement is in addition to the need to stop and restart the LSF Batch daemons, as described in the previous note. See Chapter 5 of this manual for instructions on how to perform these tasks.

ShmemResource Section

The ShmemResource section provides the administrator with two parameters that control allocation of shared memory and swap space: MaxAllocMem and MaxAllocSwap. This special memory allocation control is needed because some Sun HPC ClusterTools components use shared memory.

Code Example 2–2 shows the ShmemResource template that is in the hpc.conf file that is shipped with Sun HPC ClusterTools 3.0 software.

CODE EXAMPLE 2–2 ShmemResource Section Example

```
#Begin ShmemResource
#MaxAllocMem  0x7fffffffffffffff
#MaxAllocSwap 0x7fffffffffffffff
#End ShmemResource
```

To set MaxAllocMem and/or MaxAllocSwap limits, remove the comment character (#) from the start of each line and replace the current value, 0x7fffffffffffffff, with the desired limit.

The following section explains how to set these limits.

Guidelines for Setting Limits

Sun HPC's internal shared memory allocator permits an application to use swap space, the amount of which is the smaller of:

- The value (in bytes) given by the MaxAllocSwap parameter.
- 90% of available swap on a node

If MaxAllocSwap is not specified, or if zero or a negative value is specified, 90% of a node's available swap will be used as the swap limit.

The MaxAllocMem parameter can be used to limit the amount of shared memory that can be allocated. If a smaller shared memory limit is not specified, the shared memory limit will be 90% of available physical memory.

The following Sun HPC ClusterTools components use shared memory:

- The resource management software uses shared memory to hold cluster and job table information. Its memory use is based on cluster and job sizes and is not controllable by the user. Shared memory space is allocated for the runtime environment when the LSF daemon starts and is not affected by MaxAllocMem and MaxAllocSwap settings. This ensures that the runtime environment and LSF

Base subsystem can start up no matter how low these memory-limit variables have been set.

- MPI uses shared memory for communication between processes that are on the same node. The amount of shared memory allocated by a job can be controlled by MPI environment variables.
- Sun S3L uses shared memory for storing data. An MPI application can allocate parallel arrays whose subgrids are in shared memory. This is done with the utility `S3L_declare_detailed()`, described in the *Sun S3L Programming and Reference Guide*.

Note - Sun S3L supports a special form of shared memory known as *Intimate Shared Memory* (ISM), which reserves a region in physical memory for shared memory use. What makes ISM space special is that it is not swappable and, therefore, cannot be made available for other use. For this reason, the amount of memory allocated to ISM should be kept to a minimum.

Note - Shared memory and swap space limits are applied per-job on each node.

If you have set up your system for dedicated use (only one job at a time is allowed), you should leave `MaxAllocMem` and `MaxAllocSwap` undefined. This will allow jobs to maximize use of swap space and physical memory.

If, however, multiple jobs will share a system, you may wish to set `MaxAllocMem` to some level below 50% of total physical memory. This will reduce the risk of having a single application lock up physical memory. How much below 50% you choose to set it will depend on how many jobs you expect to be competing for physical memory at any given time.

Note - When users make direct calls to `mmap`(2) or `shmget`(2), they are not limited by the `MaxAllocMem` and `MaxAllocSwap` variables. These utilities manipulate shared memory independently of the `MaxAllocMem` and `MaxAllocSwap` values.

`Netif` Section

The `Netif` section identifies the network interfaces supported by the Sun HPC cluster and specifies the rank and striping attributes for each interface. The `hpc.conf` template that is supplied with Sun HPC ClusterTools 3.0 software contains a default list of supported network interfaces as well as their default ranking. Code Example 2–3 represents a portion of the default `Netif` section.

CODE EXAMPLE 2–3 Netif Section Example

```
Begin Netif
NAME            RANK      MTU       STRIPE     PROTOCOL     LATENCY WIDTHmidnn          0        1
idn             10        16384     0          tcp          20          150
scin            20        32768     1          tcp          20          150
  :             :           :       :           :           :           :
scid            40        32768     1          tcp          20          150
  :             :           :       :           :           :           :
scirsm          45        32768     1          rsm          20          150
  :             :           :       :           :           :           :
  :             :           :       :           :           :           :
smc             220       4096      0          tcp          20          150
End Netif
```

Interface Names

The first column lists the names of possible network interface types.

Rank Attribute

The rank of an interface is the order in which that interface is to be preferred over other interfaces. That is, if an interface with a rank of 0 is available when a communication operation begins, it will be selected for the operation before interfaces with ranks of 1 or greater. Likewise, an available rank-1 interface will be used before interfaces with a rank of 2 or greater.

Note - Because `hpc.conf` is a shared, cluster-wide configuration file, the rank specified for a given interface will apply to all nodes in the cluster.

Network ranking decisions are usually influenced by site-specific conditions and requirements. Although interfaces connected to the fastest network in a cluster are often given preferential ranking, raw network bandwidth is only one consideration. For example, an administrator might decide to dedicate one network that offers very low latency, but not the fastest bandwidth to all intra-cluster communication and use a higher-capacity network for connecting the cluster to other systems.

MTU Attribute

This is a placeholder column. Its contents are not used at this time.

Stripe Attribute

Sun HPC ClusterTools 3.0 software supports scalable communication between cluster nodes through striping of MPI messages over SCI interfaces, as described in the *Sun HPC 3.0 SCI Guide*. In striped communication, a message is split into smaller packets and transmitted in two or more parallel streams over a set of network interfaces that have been logically combined into a *stripe-group*.

The `STRIPE` column allows the administrator to include individual SCI network interfaces in a *stripe-group pool*. Members of this pool are available to be included in logical stripe groups. These stripe groups are formed on an as-needed basis, selecting interfaces from this stripe-group pool.

To include the SCI interface in a stripe-group pool, set its `STRIPE` value to `1`. To exclude an interface from the pool, specify `0`. Up to four SCI network interface cards per node can be configured for stripe-group membership.

When a message is submitted for transmission over the SCI network, an MPI protocol module distributes the message over as many SCI network interfaces as are available.

Stripe-group membership is made optional so you can reserve some SCI network bandwidth for non-striped use. To do so, simply set `STRIPE` = `0` on the SCI network interface(s) you wish to reserve in this way.

Protocol Attribute

This column identifies the communication protocol used by the interface. The `scirsm` interface employs the RSM (Remote Shared Memory) protocol. The others all use TCP (Transmission Control Protocol).

Latency Attribute

This is a placeholder column. Its contents are not used at this time.

Bandwidth Attribute

This is a placeholder column. Its contents are not used at this time.

MPIOptions Section

This section provides the means to control many MPI runtime parameters at the queue level. This is done by naming the LSF queue of interest and then listing the parameters to be defined, along with their desired values.

The hpc.conf file that is shipped with Sun HPC ClusterTools 3.0 software includes two MPIOptions templates—see Code Example 2–4. The first template, which contains the phrase queue=hpc, is designed for use by multiuser queue named hpc. The second template, which contains the phrase queue=performance, is intended by use by a dedicated (single-user) queue named performance.

To use either template, simply delete the comment character (#) from the beginning of each line in the template of interest.

Note - The options in the general-purpose template (queue=hpc) are the same as the default settings for the Sun MPI library. In other words, you will have the effects of this template even if you do not use it or change any of the library defaults. This template is provided in the MPIOptions section so you can see what options are most beneficial when operating in a multiuser mode.

CODE EXAMPLE 2–4 MPIOptions Section Example

```
# Following is an example of the options that affect the runtime
# environment of the MPI library. The listings below are identical
# to the default settings of the library. The "queue=hpc" phrase
# makes it an LSF-specific entry, and only for the queue named hpc.
# These options are a good choice for a multiuser queue. To be
# recognized by CRE, the "Queue=hpc" needs to be removed.
#
# Begin MPIOptions queue=hpc
# coscheduling  avail
# pbind         avail
# spindtimeout   1000
# progressadjust   on
# spin            off
#
# shm_numpostbox       16
# shm_shortmsgsize    256
# rsm_numpostbox       15
# rsm_shortmsgsize    401
# rsm_maxstripe         2
# End MPIOptions

# The listing below is a good choice when trying to get maximum
# performance out of MPI jobs that are running in a queue that
# allows only one job to run at a time.
#
```

(continued)

(Continuation)

```
# Begin MPIOptions Queue=performance
# coscheduling             off
# spin                      on
# End MPIOptions
```

Table 2–1 provides brief descriptions of the MPI runtime options that can be set in `hpc.conf`. Each description identifies the default value and describes the effect of each legal value. Refer to the *Sun MPI 4.0 User's Guide: With LSF* for fuller descriptions.

Note - Some MPI options not only control a parameter directly, they can also be set to a value that passes control of the parameter to an environment variable. Where an MPI option has an associated environment variable, Table 2–1 names the environment variable.

TABLE 2–1 MPI Runtime Options

Option	Values		Description
	Default	**Other**	
`coscheduling`	`avail`		Allows `spind` use to be controlled by the environment variable `MPI_COSCHED`. If `MPI_COSCHED=0` or is not set, `spind` is not used. If `MPI_COSCHED=1`, `spind` must be used.
		`on`	Enables coscheduling; `spind` is used. This value overrides `MPI_COSCHED=0`.
		`off`	Disables coscheduling; `spind` is not to be used. This value overrides `MPI_COSCHED=1`.

TABLE 2–1 MPI Runtime Options *(continued)*

Option	Values		Description
	Default	**Other**	
pbind	avail		Allows processor binding state to be controlled by the environment variable MPI_PROCBIND. If MPI_PROCBIND=0 or is not set, no processes will be bound to a processor. This is the default. If MPI_PROCBIND=1, all processes on a node will be bound to a processor.
		on	All processes will be bound to processors. This value overrides MPI_PROCBIND=0.
		off	No processes on a node are bound to a processor. This value overrides MPI_PROCBIND=1.
spindtimeout	1000		When polling for messages, a process waits 1000 milliseconds for spind to return. This equals the value to which the environment variable MPI_SPINDTIMEOUT is set.
		integer	To change the default timeout, enter an integer value specifying the number of milliseconds the timeout should be.
progressadjust	on		Allows user to set the environment variable MPI_SPIN.
		off	Disables user's ability to set the environment variable MPI_SPIN.
shm_numpostbox	16		Sets to 16 the number of postbox entries that are dedicated to a connection endpoint. This equals the value to which the environment variable MPI_SHM_NUMPOSTBOX is set.

TABLE 2–1 MPI Runtime Options *(continued)*

Option	Values		Description
	Default	**Other**	
		integer	To change the number of dedicated postbox entries, enter an integer value specifying the desired number.
shm_shortmsgsize	256		Sets to 256 the maximum number of bytes a short message can contain. This equals the default value to which the environment variable `MPI_SHM_SHORTMSGSIZE` is set.
		integer	To change the maximum-size definition of a short message, enter an integer specifying the maximum number of bytes it can contain.
rsm_numpostbox	15		Sets to 15 the number of postbox entries that are dedicated to a connection endpoint. This equals the value to which the environment variable `MPI_RSM_NUMPOSTBOX` is set.
		integer	To change the number of dedicated postbox entries, enter an integer value specifying the desired number.
rsm_shortmsgsize	401		Sets to 401 the maximum number of bytes a short message can contain. This equals the value to which the environment variable `MPI_RSM_SHORTMSGSIZE` is set.
		integer	To change the maximum-size definition of a short message, enter an integer specifying the maximum number of bytes it can contain.
rsm_maxstripe	2		Sets to 2 the maximum number of stripes that can be used. This equals the value to which the environment variable `MPI_RSM_MAXSTRIPE` is set.

TABLE 2–1 MPI Runtime Options *(continued)*

Option	Values		Description
	Default	**Other**	
		integer	To change the maximum number of stripes that can be used, enter an integer specifying the desired limit. This value cannot be greater than 2.
`spin`	`off`		Sets the MPI library to avoid spinning while waiting for status. This equals the value to which the environment variable `MPI_SPIN` is set.
		`on`	Sets the MPI library to spin.

`PFSFileSystem` Section

The `PFSFileSystem` section describes the parallel file systems that Sun MPI applications can use. This description includes

- The name of the parallel file system.
- The hostname of each server node in the parallel file system.
- The name of the storage device to be included in the parallel file system being defined.
- The number of PFS I/O threads spawned to support each PFS storage device.

A separate `PFSFileSystem` section is needed for each parallel file system that you want to create. Code Example 2–5 shows a sample `PFSFileSystem` section with two parallel file systems, `pfs-demo0` and `pfs-demo1`.

CODE EXAMPLE 2–5 PFSFileSystem Section Example

```
Begin PFSFileSystem=pfs-demo0
NODE            DEVICE              THREADS
hpc-node0       /dev/rdsk/c0t1d0s2  1
hpc-node1       /dev/rdsk/c0t1d0s2  1
hpc-node2       /dev/rdsk/c0t1d0s2  1
End PFSFileSystem
```

(continued)

(Continuation)

```
Begin PFSFileSystem=pfs-demo1
NODE            DEVICE              THREADS
hpc-node3       /dev/rdsk/c0t1d0s2  1
hpc-node4       /dev/rdsk/c0t1d0s2  1
End PFSFileSystem
```

Parallel File System Name

The first line shows the name of the parallel file system. PFS file system names must not include spaces.

Server Node Hostnames

The `NODE` column lists the hostnames of the nodes that function as servers for the parallel file system being defined. The example configuration in Code Example 2–5 shows two parallel file systems:

- `pfs-demo0` – three server nodes: `hpc-node0`, `hpc-node1`, and `hpc-node2`.
- `pfs-demo1` – two server nodes: `hpc-node3` and `hpc-node4`.

Storage Device Names

The second column gives the device name associated with each member node. This name follows Solaris device naming conventions. (But note the use of `rdisk` in the device names.)

Thread Limits

The `THREADS` column allows the administrator to specify how many threads a PFS I/O daemon will spawn for the disk storage device or devices it controls. The number of threads needed by a given PFS I/O server node will depend primarily on the performance capabilities of its disk subsystem.

- For a storage object with a single disk or a small storage array, one thread may be enough to exploit the storage unit's maximum I/O potential.
- For a more powerful storage array, two or more threads may be needed to make full use of the available bandwidth.

PFSServers Section

A PFS I/O server is a Sun HPC node that is connected to one or more disk storage units that are defined as part of a parallel file system in the hpc.conf file—that is, they are listed in a PFSFileSystem section of the hpc.conf file. Plus, the node itself must be listed in the PFSServers section of hpc.conf, as shown in Code Example 2–6.

CODE EXAMPLE 2–6 PFSServers Section Example

```
Begin PFSServers
NODE            BUFFER_SIZE
hpc-node0        150
hpc-node1        150
hpc-node2        300
hpc-node3        300
End PFSServers
```

In addition to being defined in hpc.conf, a PFS server also differs from other nodes in a Sun HPC cluster in that it has a PFS I/O daemon running on it.

PFS I/O Server Hostnames

The left column lists the hostnames of the nodes that are PFS I/O servers. In this example, they are hpc-node0 through hpc-node3.

Buffer Size

The second column specifies the amount of memory the PFS I/O daemon will have for buffering transfer data. This value is specified in units of 32-Kbyte buffers. The number of buffers that you specify will depend on the amount of I/O traffic you expect that server is likely to experience at any given time.

The optimal buffer size will vary with system type and load. Buffer sizes in the range of 128 to 512 provide reasonable performance on most Sun HPC Systems. You can use pfsstat to get reports on buffer cache hit rates. This can be useful for evaluating how well suited the buffer size is to the cluster's current I/O activity.

HPCNodes Section

This section allows you to define a Sun HPC cluster that consists of a subset of the nodes contained in the LSF cluster. To use this configuration option, enter the hostnames of the nodes that you want in the HPC cluster between in this section, one hostname per line. Code Example 2–7 shows a sample HPCNodes section with two nodes listed.

CODE EXAMPLE 2–7 HPCNodes Section Example

```
Begin HPCNodes
node1
node2
End HPCNodes
```

Propagate hpc.conf Information

Whenever hpc.conf is changed, the LSF daemons must be updated with the new information. After all required changes to hpc.conf have been made, restart the LSF base daemons LIM and RES. Use the lsadmin subcommands reconfig and resrestert as follows:

```
hpc-demo# lsadmin reconfig
hpc-demo# lsadmin resrestart all
```

Then, use the badmin subcommands reconfig to restart mbatchd and hrestart to restart the slave batch daemons:

```
hpc-demo# badmin reconfig
hpc-demo# badmin hrestart all
```

This only needs to be done from one node. See the *LSF Batch Administrator's Guide* for additional information about restarting LSF daemons.

CHAPTER 3

Notes on LSF Batch Queues and Sun HPC Jobs

This chapter discusses various LSF Batch queue issues that are of particular interest to Sun HPC system administrators. It also discusses a new, optional configuration variable that can be used to verify project-based accounting.

Note - The following discussion deals with LSF terms and concepts with which you are expected to be familiar. If you have not done so already, please read the *LSF Batch Administrator's Guide*, version 3.2.3, paying special attention to sections dealing with queues, job starters, and the queue configuration file, `lsb.queues`.

Creating Sun HPC–Specific Queues

Because HPC jobs distribute multiple processes across multiple nodes, their batch queue requirements are different from those of serial jobs. For this reason, you should specifically configure one or more batch queues for running Sun HPC jobs. This involves editing the queue configuration file, `lsb.queues`. Sections "Specify PAM as Job Starter" on page 21 and "Enable Interactive Batch Mode" on page 22 discuss the queue parameters of primary interest, `JOB_STARTER` and `INTERACTIVE`.

Specify PAM as Job Starter

The `JOB_STARTER` parameter allows an LSF batch queue to pass job launching control over to a special job-starting procedure rather than launching the job itself.

For Sun HPC applications, this job launching role is given to the Parallel Application Manager (PAM), which is a utility for starting and managing MPI jobs.

PAM should be specified as the job starter on all queues that will be used by Sun HPC jobs. To do this, simply edit the `JOB_STARTER` line in `lsb.queues` to read as follows:

```
JOB_STARTER=pam
```

When a Sun HPC job is submitted to a PAM-configured queue, the queue will start PAM running. PAM, in turn, will launch the Sun HPC job on the cluster.

Enable Interactive Batch Mode

LSF supports the concept of *interactive batch* job execution. When a job is submitted in interactive batch mode, it receives the same batch scheduling and host selection services as noninteractive batch jobs, but the terminal from which the job was submitted remains attached to the job as if it were launched interactively.

Note - Interactive batch mode is the *only* interactive mode Sun HPC ClusterTools 3.0 software supports.

By default, both batch mode and interactive batch mode are available. To select interactive batch mode, include the `--I` option on the `bsub` command line. Without this option, `bsub` invokes conventional batch mode.

The `INTERACTIVE` parameter in the `lsb.queues` file allows you restrict a queue to *accept only* interactive batch jobs or *exclude all* interactive batch jobs. Use it to restrict Sun HPC–dedicated queues to interactive batch jobs. Otherwise, noninteractive jobs could be added to the queue, which could make the queue less efficient for handling the interactive batch jobs. To impose this restriction, add the following line to the appropriate queue descriptor in the `lsb.queues` file:

```
INTERACTIVE=ONLY
```

All jobs submitted to a queue configured in this way must include the `--I` option on the `bsub` command line.

Note - Separate queues can be configured for batch-mode–only jobs as well.

Because interactive batch jobs need fast response times, there are other steps you should take to minimize job launch latencies normally associated with batch queue behavior. These are described in the next section.

Configuring for Fast Interactive Batch Response Time

There are several steps you can take to optimize the response time of an interactive batch queue. These steps are discussed in Sections “Set PRIORITY in lsb.queues” on page 23 through “Add Optimization Parameters to lsb.params” on page 23.

Set PRIORITY in lsb.queues

The PRIORITY parameter defines a batch queue’s priority relative to other batch queues. To ensure faster dispatching, assign a higher PRIORITY value to interactive batch queues than you give to noninteractive queues. A higher number equals a higher priority. For example, the following setting

```
PRIORITY=12
means that jobs on that queue will usually be serviced sooner than
jobs on queues with a setting of PRIORITY=11 or lower.
```

Set NICE in lsb.queues

Set the queue’s NICE parameter to 10. This will ensure that it receives the same CPU priority as other interactive queues.

Set NEW_JOB_SCHED_DELAY in lsb.queues

Set the NEW_JOB_SCHED_DELAY parameter to 0. This will allow a new job scheduling session to be started as soon as a job is submitted to this queue.

Add Optimization Parameters to lsb.params

During installation of the Sun HPC ClusterTools 3.0 packages, you are asked if you want to modify the lsb.params file to optimize interactive batch response time. If

you answered yes, the SUNWrte package makes the following changes to the lsb.params file:

```
MBD_SLEEP_TIME=1
MAX_SBD_FAIL=30
JOB_ACCEPT_INTERVAL=0
```

The first parameter, MBD_SLEEP_TIME, specifies the number of seconds LSF Batch will wait between attempts to dispatch jobs. The default is 60 seconds. SUNWrte changes the interval to 1 second.

The MAX_SBD_FAIL parameter specifies how many times LSF Batch will try to reach an unresponsive slave batch daemon before giving up. MBD_SLEEP_TIME controls the frequency of these attempts. If MAX_SBD_FAIL is not specified, its default value is three times the MBD_SLEEP_TIME value. SUNWrte sets MAX_SBD_FAIL to 30.

The JOB_ACCEPT_INTERVAL parameter specifies how many MBD_SLEEP_TIME periods LSF Batch will wait after successfully dispatching a job to a host before it dispatches another job to the same host. SUNWrte sets this parameter to 0, allowing the host to accept multiple jobs in each job dispatching period (MBD_SLEEP_TIME).

If you answered no during the installation, but now wish to enable these optimizations, simply edit these parameters in the lsb.params file as shown above.

Verifying Project-Based Accounting

One feature of LSF is project-based accounting—that is, individual jobs can be associated with particular projects and charges allocated accordingly. Projects can be specified in the following ways:

- with the bsub option, -P *proj_name*, where *proj_name* is the name of the project.
- with the environment variable LSB_DEFAULTPROJECT.
- with the parameter DEFAULT_PROJECT in the lsb.params file.

A new, optional configuration variable, CHECK_PROJECT_UGRPMEMBERSHIP, has been added to ensure the integrity of project-based accounting. To enable this feature, add the following line to the lsb.params file:

```
CHECK_PROJECT_UGRPMEMBERSHIP=y
```

When this entry is present in `lsb.params`, the software verifies that the person submitting the job is a member of the user group associated with *proj_name*. User groups are defined in the configuration file `lsb.users`. If the person submitting the job is not a member of the user group associated with that project, the job will be rejected.

CHAPTER 4

PFS Configuration Notes

As its name implies, the distinguishing characteristic of a parallel file system is the parallel layout of its files. Unlike serial file systems, such as UFS, which conduct file I/O in single, serial streams, PFS may distribute its files across two or more disks, each of which may be attached to a different PFS I/O server. This allows file I/O to be divided into multiple, parallel streams, yielding significant performance gains in every file read or write operation.

Note - Standard Solaris file system commands can be used to access and manipulate PFS files. However, the high-performance I/O capabilities of PFS can be fully exploited only through calls to MPI I/O library routines.

PFS Basics

PFS file systems are defined in the `hpc.conf` file. There, each file system is given a name and the list of hostnames of the PFS I/O servers across which it will be distributed.

A PFS I/O server is simply a Sun HPC node that has disk storage systems attached, has been defined as a PFS I/O server in the `hpc.conf` file, and is running a PFS I/O daemon. A PFS I/O server and the disk storage device(s) attached to it are jointly referred to as a *PFS storage system*.

Figure 4–1 illustrates a sample Sun HPC cluster with eight nodes:

- Four nodes function as compute servers only – CS0, 1, 2, and 4.
- Three nodes functions as PFS I/O servers only – IOS0, IOS 1, and IOS2.
- One node operates as both compute server and PFS I/O server – CS3-IOS3.

All four PFS I/O servers have disk storage subsystems attached. PFS I/O servers IOS0 and IOS3 each have a single disk storage unit, while IOS1 and IOS2 are each connected to two disk storage units.

The PFS configuration example in Figure 4–1 shows two PFS file systems, `pfs-demo0` and `pfs-demo1`.

Each PFS file system is distributed across three PFS storage systems. This means an individual file in either file system will be divided into three blocks, which will be written to and read from its storage subsystems in three parallel data streams.

Note that two PFS storage systems, IOS1 and IOS2, contain at least two disk partitions, allowing them to be used by both `pfs-demo0` and `pfs-demo1`.

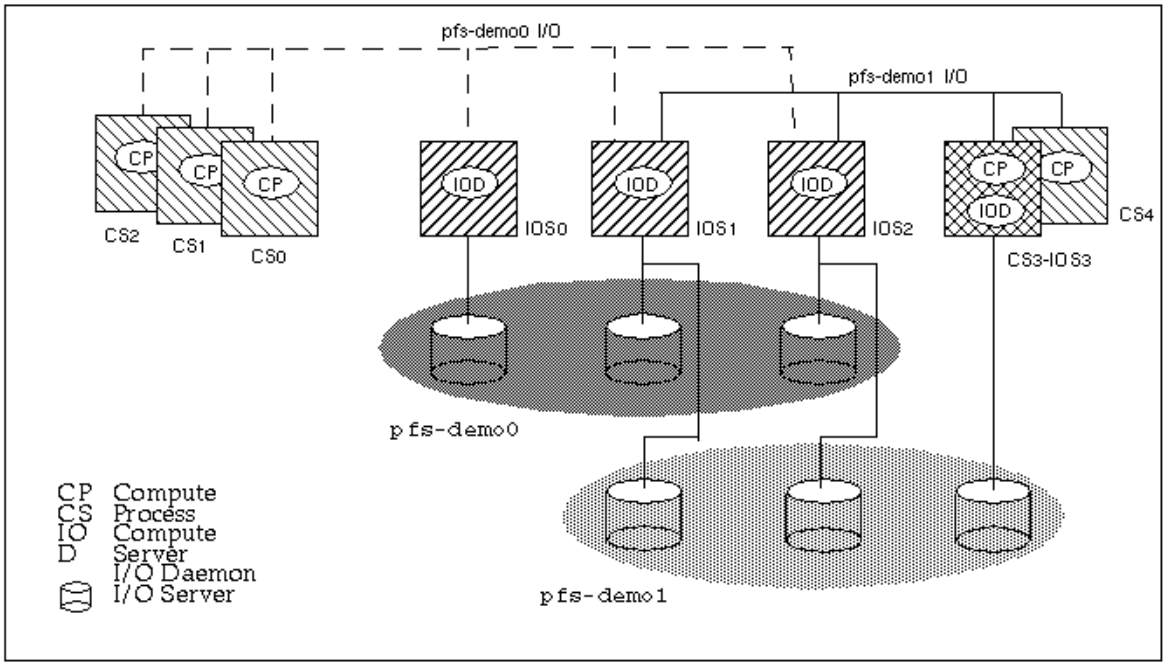

Figure 4–1 PFS Conceptual View

The dashed lines labeled `pfs-demo0` I/O indicate the data flow between compute processes 0, 1, and 2 and the PFS file system `pfs-demo0`. Likewise, the set of solid lines labeled `pfs-demo1` I/O represent I/O for the PFS file system `pfs-demo1`.

This method of laying out PFS files introduces some file system configuration issues not encountered with UFS and other serial file systems. These issues are discussed in the balance of this section.

Note - Although PFS files are distributed differently from UFS files, the same Solaris utilities can be used to manage them.

Applications and I/O Processes, Colocate or Run Separately?

If you plan to configure only a subset of the nodes on a cluster as PFS I/O servers, you will have the option of either colocating applications and I/O daemons on the same PFS I/O servers or segregating them onto separate nodes. If, however, you configure all the nodes in a cluster as PFS I/O servers, you will of necessity colocate applications and PFS I/O daemons.

Guidelines for making this choice are provided below.

Conditions That Favor Colocating

Each of the following conditions favors colocating applications with PFS I/O daemons.

- Large nodes (many CPUs per node).
- Fast disk-storage devices (storage arrays, for example) on each node.
- Lower-performance cluster interconnect, such as 10- or 100-BaseT Ethernet.
- Small number of applications competing for node resources.

When these conditions exist in combination, the network is more likely to be a performance-limiting resource than the relatively more powerful nodes. Therefore, it becomes advantageous to locate applications on the PFS I/O servers to decrease the amount of data that must be sent across the network.

Conditions That Favor Separating Applications and IODs

You should avoid running applications on I/O server nodes when some or all of the following conditions exist.

- Smaller nodes.
- Slow disk storage devices (single disks, for example) on each node.

- Relatively high-performance cluster interconnect, such as SCI or ATM.
- Large number of applications competing for node resources.

In this case, the competition for memory, bus bandwidth, and CPU cycles may offset any performance advantages local storage would provide.

Effect of Cluster Size

By itself, the size of a cluster (number of nodes) does not favor either colocating or not colocating applications and PFS I/O daemons. Larger clusters do, however, attenuate the benefits of colocating. This is because the amount by which colocating reduces network traffic can be expressed as

$$T_c = T_s - T_s / N$$

where T_c is the level of network traffic using colocating, T_s is the level of network traffic without colocating, and N is the number of nodes in the cluster. In other words, colocating reduces network traffic by 1/number-of-nodes. The more nodes there are in the cluster, the smaller the effect of colocating.

CHAPTER 5

Starting and Stopping PFS Daemons

A PFS I/O daemon must be running on each PFS I/O server, and a PFS proxy daemon must be running on each node that will access PFS file systems. This chapter describes the procedures for starting and stopping these daemons, as well as are how to create and mount PFS file systems.

Starting PFS I/O Daemons

The PFS I/O daemons will start automatically at boot time on nodes configured as PFS I/O servers. However, if you have just configured new PFS I/O servers, you can start their daemons manually without having to reboot. To do so, enter the following on each newly created PFS I/O server node.

```
pfs-srv0# /etc/init.d/sunhpc.pfs_server start
pfs-srv1# /etc/init.d/sunhpc.pfs_server start
 :
```

Alternatively, you can launch both the PFS proxy and I/O daemons on all the nodes in the cluster. To do this, execute the following command once, on any node in the cluster.

```
# /opt/SUNWhpc/etc/pfs/pfsstart
```

Starting PFS Proxy Daemons

The following command starts the proxy daemon. Execute it on each node that will need to access PFS file systems.

```
node0# /etc/init.d/sunhpc.pfs_client start
node1# /etc/init.d/sunhpc.pfs_client start
  :
```

Stopping PFS I/O Daemons

The following command stops the I/O server daemons. Execute it on each I/O server node.

```
node0# /etc/init.d/sunhpc.pfs_server stop
node1# /etc/init.d/sunhpc.pfs_server stop
         :
```

Stopping PFS Proxy Daemons

The following command stops the proxy daemon. Execute it on each node that has a proxy daemon running.

```
node0# /etc/init.d/sunhpc.pfs_client stop
node1# /etc/init.d/sunhpc.pfs_client stop
         :
```

Alternatively, you can manually stop both the PFS client and I/O server daemons on every node in the cluster by executing the following command on any one node in the cluster.

```
# /opt/SUNWhpc/etc/pfs/pfsstop
```

Create and Mount PFS File Systems

As with UFS file systems, you can use the Solaris utilities `mkfs` and `mount` to create and mount PFS file systems. For example, the following creates a 64-Kbyte PFS file system named `pfs-demo0`. Execute it on any server node.

```
adm# mkfs --F pfs pfs-demo0 64K
```

The **`-F`** option specifies the file system's type, which is `pfs`.

Next, mount the file system on each client node that has a PFS proxy daemon.

```
hpc-node0# mount --F pfs pfs-demo0 /pfs_demo0
hpc-node1# mount --F pfs pfs-demo0 /pfs_demo0
        :
```

Alternatively, you can execute the following on a single node. This will cause the PFS file system to be mounted on all nodes in the cluster.

```
# /opt/SUNWhpc/bin/pfsmount pfs-demo0 /pfs_demo0
```

You may also want to add an entry for each PFS file system in the file `/etc/vfstab`. This will make it unnecessary to include the **`--F`** option when making and mounting the file systems. Code Example 5–1 shows how a PFS file system entry might look in the file `/etc/vfstab`.

CODE EXAMPLE 5–1 Sample PFS Entry in /etc/vfstab

```
#device      device         mount         FS      fsck   mount     mount
#to mount    to fsck        point         type    pass   at boot   options
pfs-demo0    -              /pfs_demo0    pfs     -      no        -
```

Verify That PFS File System is Mounted

Before anyone attempts to use a newly created PFS file system, it is a good idea to verify that it is correctly mounted. This can easily be done by invoking `df --f PFS` on any node that has a PFS proxy daemon. Code Example 5–2 shows an example of this, with the PFS file system `pfs-demo0` included in the `df` output.

CODE EXAMPLE 5–2 Using df –f PFS to Verify PFS File System Is Mounted

```
hpc-node1# df --f PFS
/dev/pfs_psuedo (pfs_pseudo): 0 blocks 0 files
```

Alternatively, if you execute the `pfsmount` command without any arguments, it will list every mounted PFS file system in the cluster. This is illustrated in Code Example 5–3.

CODE EXAMPLE 5–3 Using pfsmount to List Mounted PFS File Systems

```
hpc-node1# /opt/SUNWhpc/bin/pfsmount
Mounted PFS filesystems:
   hpc-node4:
       pfs-demo0 on pfs-demo0

   hpc-node5:
       pfs-demo0 on  pfs-demo0
```

The PFS file system `pfs-demo0` is now ready to use. You can use Solaris utilities to create and delete PFS files and directories in the directory `/pfs-demo0`, just as you would in any UFS file system. To achieve best performance, however, applications should access the PFS facilities via MPI I/O calls.

APPENDIX **A**

Installing and Removing the Software

This appendix includes instructions for

- *Installing the software at the command line* – “Installing at the Command Line” on page 35
- *Removing the software* – “Removing the Software” on page 51
- *Removing and installing individual packages* – “Removing and Reinstalling Individual Packages” on page 53

Installing at the Command Line

The easiest way to configure and install Sun HPC ClusterTools 3.0 software is to use the configuration tool, `install_gui`, as described in the *Sun HPC ClusterTools 3.0 Installation Guide*. If you prefer, however, you may install the software from the command line as described in this appendix, with a few references to the installation guide.

Figure A–1 summarizes the steps involved. The solid lines identify tasks that are always performed. The dashed lines indicate special-case tasks.

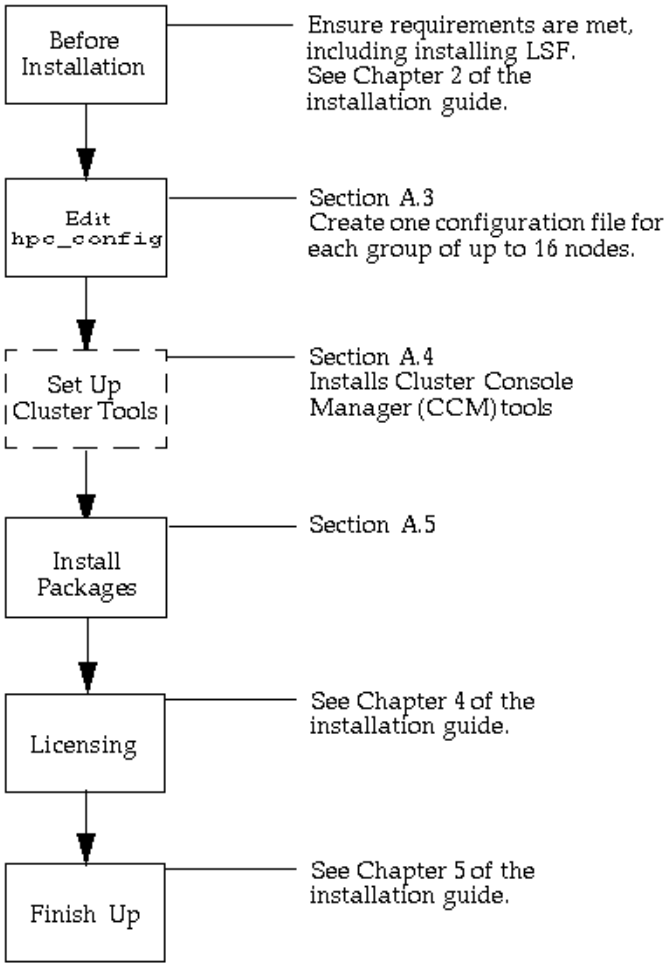

Figure A–1 Installing Sun HPC ClusterTools 3.0 Software at the Command Line

Before Installation

Before installing Sun HPC ClusterTools 3.0 software, you need to ensure that the hardware and software that make up your cluster meet certain requirements. You must have already installed LSF 3.2.3. Further requirements are outlined in the *Sun HPC ClusterTools 3.0 Installation Guide*. Review them before proceeding with the instructions in this appendix.

If you are installing the software on a cluster of more than 16 nodes, you will probably want to use the CCM tools to make installation easier. You can use these tools to install on up to 16 nodes at a time. If you need to install software on a cluster of more than 16 nodes, you must install it first on a group of up to 16 nodes, then add more nodes by repeating the installation process on additional groups of up to 16 until you have installed the software on all the nodes in the cluster. For each group of nodes, you will need to create a separate configuration file, each with a unique file name, such as `hpc_config1`, `hpc_config2`, and so on.

The `hpc_config` File

Many aspects of the Sun HPC ClusterTools 3.0 installation process are controlled by a configuration file called `hpc_config`, which is similar to the `lsf_config` file used to install LSF 3.2.3.

Instructions for accessing and editing `hpc_config` are provided in “Accessing `hpc_config`” on page 37 and “Editing `hpc_config`” on page 38.

Accessing `hpc_config`

Use a text editor to edit the `hpc_config` file directly. This file must be located in a directory within a file system that is mounted read / write / execute accessible on all the other nodes in the cluster. A template for `hpc_config` is provided on the Sun HPC ClusterTools 3.0 distribution CD-ROM to simplify creation of this file.

Before starting the installation process, you should copy this template to a directory on the node chosen to be the installation platform and edit it so that it satisfies your site-specific installation requirements. Choose a node to function as the installation platform and a home directory on that node for `hpc_config`.

Note - The directory containing `hpc_config` must be read / write / execute accessible (777 permissions) by all the nodes in the cluster.

The `hpc_config` template is located in

```
/cdrom/hpc_3_0_ct/Product/Install_Utilities/config_dir/hpc_config
```

To access `hpc_config` on the distribution CD-ROM, perform the following steps on the node chosen to be the installation platform:

1. **Mount the CD-ROM path on all the nodes in the cluster.**

2. **Load the Sun HPC ClusterTools distribution CD-ROM in the CD-ROM drawer.**

3. **Copy the configuration template onto the node.**

```
# cd config_dir_install
# cp /cdrom/hpc_3_0_ct/Product/Install_Utilities/config_dir/hpc_config .
```

config_dir_install is a variable representing the directory where the configuration files will reside; all cluster nodes must be able to read from and write to this directory.

4. **Edit the** `hpc_config` **file according to the instructions provided in the next section.**

If You Have Already Installed the Software

If you have already installed the software, you can find a copy of the `hpc_config` template in the directory `/opt/SUNWhpc/bin/Install_Utililites/config_dir`.

If you are editing an existing `hpc_config` file after installing the software using the graphical installation tool, the `hpc_config` file created by the tool will not contain the comment lines included in the template.

Editing `hpc_config`

Code Example A–1 shows the basic `hpc_config` template, but without most of the comment lines provided in the online template. The template is simplified here to make it easier to read and because each section is discussed in detail following Code Example A–1. Two examples of edited `hpc_config` files follow the general description of the template.

The template comprises five sections:

- *Supported Software Installation* – All installations must complete this first section. Since you will be using LSF, complete only Part A of this section.
- *General installation information* – All installations must complete this section. If you are installing the software locally on a single-node cluster, you can stop after completing this section.
- *Information for NFS and cluster-local installations* – If you are installing the software either on an NFS server for remote mounting or locally on each node of a multinode cluster, you need to complete this section, too.

CODE EXAMPLE A–1 hpc_config Template (With Most Comment Lines Removed)

```
Section I - Supported Software Information

LSF_SUPPORT="<choice>"

# PART A: Running HPC 3.0 ClusterTools software with LSF software.

# Do you want to modify LSF parameters to optimize HPC job launches?

MODIFY_LSF_PARAM="<choice>"

# Name of the LSF Cluster
LSF_CLUSTER_NAME="<clustername>"

# Section II - General Installation Information
# Type of Installation Configuration
INSTALL_CONFIG="<choice>"

# Installation Location
INSTALL_LOC="/opt"

# CD-ROM Mount Point
CD_MOUNT_PT="/cdrom/hpc_3_0_ct"

#Section III - For Cluster-Local and NFS Installation

# Installation Method
INSTALL_METHOD="<method>"

# Hardware Information
NODES="<hostname1> <hostname2> <hostname3>"

# SCI Support
INSTALL_SCI="<choice>"

# Section IV - For NFS Installation

# NFS Server NFS_SERVER=""

# Location of the Software Installed on the NFS ServerINSTALL_LOC_SERVER=""
```

- *Information for NFS installations* – You need to complete this section only if you are installing the software on an NFS server.

For the purposes of initial installation, ignore the fifth section.

Supported Software Installation

LSF Support

You will be using the software with LSF, so enter `yes` here.

```
LSF_SUPPORT="yes"
```

Since you will be using LSF, complete only Part A of this section.

LSF Parameter Modification

Allowing the Sun HPC installation script to modify LSF parameters optimizes HPC job launches. Your choice for this variable must be `yes` or `no`.

```
MODIFY_LSF_PARAM="choice"
```

Name of the LSF Cluster

Before installing Sun HPC ClusterTools software, you must have installed LSF 3.2.3. When you installed the LSF software, you selected a name for the LSF cluster. Enter this name in the `LSF_CLUSTER_NAME` field.

```
LSF_CLUSTER_NAME="clustername"
```

General Installation Information

All installations must complete this section. If you are installing the software locally on a single-node cluster, you can stop after completing this section.

Type of Installation

Three types of installation are possible for Sun HPC ClusterTools 3.0 software:

- `nfs` – Install the software on an NFS server for remote mounting.
- `smp--local` – For single-node clusters only: Install the software locally on the node.
- `cluster--local` – For multinode clusters only: Install the software locally on every node in the cluster.

Specify one of the installation types: `nfs`, `smp--local`, or `cluster--local`. There is no default type of installation.

```
INSTALL_CONFIG="config_choice"
```

Installation Location

The way the `INSTALL_LOC` path is used varies, depending on which type of installation you have chosen.

- *Local Installations* – For local installations (`smp--local` or `cluster--local`), `INSTALL_LOC` is the path where the packages will actually be installed.
- *NFS installations* – For NFS installations (`nfs`), `INSTALL_LOC` is the mount point for the software on the NFS *clients*.

You must enter a full path name. The default location is `/opt`. The location must have set (or mounted, if this is an NFS installation) read / write (755) permission on all the nodes in the cluster.

```
INSTALL_LOC="/opt"
```

If you choose an installation directory other than the default `/opt`, a symbolic link is created from `/opt/SUNWhpc` to the chosen installation point.

CD-ROM Mount Point

Specify a mount point for the CD-ROM. This mount point must be mounted on (that is, NFS-accessible to) all the nodes in the cluster. The default mount point is `/cdrom/hpc_3_0_ct`. For example:

```
CD_MOUNT_PT="/cdrom/hpc_3_0_ct"
```

Information for NFS and Cluster-Local Installations

If you are installing the software either on an NFS server for remote mounting or locally on each node of a multinode cluster, you need to complete this section.

Installation Method Options

Specify either `rsh` or `cluster--tool` as the method for propagating the installation to all the nodes in the cluster.

- `cluster--tool` – If you choose the `cluster--tool` option, you will be able to use one of the Cluster Console Manager (CCM) applications, `cconsole`, `ctelnet`, or `crlogin`, to greatly facilitate the installation of Sun HPC ClusterTools 3.0 software on all of the nodes in your cluster in parallel. See Appendix A B for information about using CCM tools.

Note - You can use the CCM tools to install on up to 16 nodes at a time. For clusters with more than 16 nodes, you will have to repeat the installation process on groups of up to 16 nodes at a time until you have installed the software on the entire cluster.

- `rsh` – If you choose the `rsh` method, the software will be installed serially on the cluster notes in the order in which they are listed in `hpc_config`. The CCM applications cannot be used to install the software in `rsh` mode.

Also note that this method requires that all nodes are trusted hosts—at least during the installation process.

```
INSTALL_METHOD="method"
```

Hardware Information

There are two ways to enter information in this section:

- If the cluster nodes are connected to a terminal concentrator, list each node in the following triplet format.

```
NODES="hostname1/termcon_name/port_id  hostname2/termcon_name/port_id  ..."
```

In each triplet, specify the host name of a node, followed by the host name of the terminal concentrator and the port ID on the terminal concentrator to which that node is connected. Separate the triplet fields with virgules (/). Use spaces between node triplets.

- If the cluster nodes are not connected to a terminal concentrator, simply list the node host names, separated by spaces, as follows.

```
NODES="hostname1  hostname2  hostname3  ..."
```

Every node in your Sun HPC cluster must also be in the corresponding LSF cluster. See the discussion of the `lsf.cluster.`*clustername* configuration file in the *LSF Batch Administrator's Guide* for information on LSF clusters.

Note - If you will not be using the CCM tools, you can allow the installation script to derive the node list from the LSF configuration file `lsf.cluster.`*clustername*. To do this, either set the `NODES` variable to `NULL` or leave the line commented out. You must be installing from one of the nodes in the LSF cluster.

SCI Support

This section tells the script whether to install the SCI-related packages. If your cluster includes SCI, replace *choice* with `yes`; otherwise, replace it with `no`.

```
INSTALL_SCI="yes"
```

A `yes` entry causes the three SCI packages and two RSM packages to be installed in the `/opt` directory. A `no` causes the installation script to skip the SCI and RSM packages.

Note - The SCI and RSM packages are installed locally on every node, not on an NFS server.

Information for NFS Installations Only

You need to complete this section only if you are installing the software on an NFS server.

NFS Server Host Name

The format for setting the NFS server host name is the same as for setting the host names for the nodes in the cluster. There are two ways to define the host name of the NFS server:

- If you have a terminal concentrator, describe the NFS server in the following triplet format.

```
NFS_SERVER="hostname/termcon_name/port_id"
```

- If you do not have a terminal concentrator, simply specify the host name of the NFS server.

```
NFS_SERVER="hostname"
```

The NFS server can be one of the cluster nodes or it can be external (but connected) to the cluster. If the server will be part of the cluster—that is, will also be an execution host for the Sun HPC ClusterTools software—it must be included in the `NODES` field described in “Hardware Information” on page 42. If the NFS server will *not* be part of the cluster, it must be available from all the hosts listed in `NODES`, but it should not be included in the `NODES` field.

Location of the Software on the Server

If you want to install the software on the NFS server in the same directory as the one specified in `INSTALL_LOC`, leave `INSTALL_LOC_SERVER` empty (`""`). If you prefer,

you can override INSTALL_LOC by specifying an alternative directory in INSTALL_LOC_SERVER.

```
INSTALL_LOC_SERVER="directory"
```

Recall that the directory specified in INSTALL_LOC defines the mount point for INSTALL_LOC_SERVER on each NFS client.

Sample hpc_config Files

Code Example A–2 and Code Example A–3 illustrate the general descriptions in the preceding sections with edited hpc_config files representing two different types of installations.

Local Install – Code Example A–2 shows how the file would be edited for a *local* installation on every node in a cluster. The main characteristics of the installation illustrated by Code Example A–2 are summarized below:

- Section 1:
  - The software will be used with LSF.

CODE EXAMPLE A–2 Sample Completed hpc_config File—Local Install via rsh

```
Section I - Supported Software Information

LSF_SUPPORT="yes"

# PART A: Running the software with LSF

# Do you want to modify LSF parameters to optimize HPC job
launches?
MODIFY_LSF_PARAM="yes"

# Name of the LSF Cluster
LSF_CLUSTER_NAME="italy"

#
Section II - General Installation Information

# Type of Installation Configuration
INSTALL_CONFIG="cluster-local"

# Installation Location
INSTALL_LOC="/export/home/opt2"

# CD-ROM Mount Point
CD_MOUNT_PT="/cdrom/hpc_3_0_ct"

#
```

(continued)

(Continuation)

```
Section III - For Cluster-Local and NFS Installation

# Installation Method
INSTALL_METHOD="rsh"

# Hardware Information
NODES="napoli pisa milano"

# SCI Support
INSTALL_SCI="no"#Section IV - For NFS Installation

# NFS Server
NFS_SERVER=""

# Location of the Software Installed on the NFS Server
INSTALL_LOC_SERVER=""
```

- The installation script will modify LSF parameters to optimize HPC launches.
- The set of clustered nodes that will be running LSF jobs is named `italy`.
- Section 2:
  - The packages will be installed locally. This means every node in the system will contain a copy of the packages that make up Sun HPC ClusterTools 3.0 software.
  - The base directory where the software will be installed is `/export/home/opt2/`.
  - The mount point for the software CD-ROM is `/cdrom/hpc_3_0_ct`.
- Section 3:
  - The software will be installed using the UNIX utility `rsh`.
  - The nodes are not connected to a terminal concentrator, so only the node host names are listed in the `Hardware Information` section.
  - The cluster in this example does not include SCI hardware.
- Section 4 is *not* completed, as the software is being installed locally on every node.

For the purposes of initial installation, ignore the fifth section.

NFS Install – Code Example A–3 shows an `hpc_config` file for an *NFS* installation. The main features of this installation example are summarized below:

- Section 1:
  - The software will be used with LSF.
  - The installation script will modify LSF parameters to optimize HPC launches.
  - The set of clustered nodes that will be running LSF jobs is named `italy`.
- Section 2:
  - The packages are being installed on an NFS server.
  - The mount point for the software on the NFS clients is `/opt/`.
  - The mount point for the software CD-ROM is `/cdrom/hpc_3_0_ct`.
- Section 3:
  - The software will be installed using Cluster Console Manager tools.
  - The nodes are connected to a terminal concentrator, and the cluster console facility will be used. Consequently, each host name must be part of a triplet entry that also includes the name of the terminal concentrator and the ID of the terminal concentrator port to which the node is connected.

This example shows the nodes `venice`, `napoli`, and `pisa` all connected to the terminal concentrator `rome` via ports 5002, 5003, and 5004.

- The cluster in this example does not include SCI hardware.

CODE EXAMPLE A–3 Sample Completed hpc_config File—NFS Install via cluster–tool

```
Section I - Supported Software Information

LSF_SUPPORT="yes"

# PART A: Running the software with LSF

# Do you want to modify LSF parameters to optimize HPC job launches?
MODIFY_LSF_PARAM="yes"

# Name of the LSF Cluster
LSF_CLUSTER_NAME="italy"

#
Section II - General Installation Information
# Type of Installation Configuration
INSTALL_CONFIG="nfs"
```

(continued)

(Continuation)

```
# Installation Location
INSTALL_LOC="/opt"

# CD-ROM Mount Point
CD_MOUNT_PT="/cdrom/hpc_3_0_ct"

#
Section III - For Cluster-Local and NFS Installation

# Installation Method
INSTALL_METHOD="cluster--tool"

# Hardware Information
NODES="venice/rome/5002 napoli/rome/5003 pisa/rome/5004"

# SCI Support
INSTALL_SCI="no"

#
Section IV - For NFS Installation

# NFS Server
NFS_SERVER="mars/rome/5005"

# Location of the Software Installed on the NFS Server
INSTALL_LOC_SERVER="/export/home/opt2"
```

- Section 4:
  - The host name of the NFS server must be supplied (in this example, `mars`). Because the NFS server is connected to a terminal concentrator, its host name must also be part of a triplet entry analogous to the entries in `NODES`.

In this case, the NFS server is not one of the nodes in the Sun HPC cluster. All the nodes in the cluster must be able to communicate with it over a network.

- The software will be installed on `mars` in the directory `/export/home/opt2`.

For the purposes of initial installation, ignore the fifth section.

Run cluster_tool_setup

Note - You can use the CCM tools to install on up to 16 nodes at a time. For clusters with more than 16 nodes, you will have to repeat the installation process on groups of up to 16 nodes at a time until you have installed the software on the entire cluster.

This step is optional. If you have chosen the `cluster--tool` method of installation and plan to use the CCM tools, you need to run the `cluster_tool_setup` script first. This loads the CCM administration tools onto a machine and creates a cluster configuration file that is used by CCM applications. See Appendix BB for a description of the three CCM applications, `cconsole`, `ctelnet`, and `crlogin`.

Note - `cconsole` requires the nodes to be connected to a terminal concentrator. The other two, `ctelnet` and `crlogin`, do not.

If you want to use `cconsole` to monitor messages generated *while rebooting the cluster nodes*, you will need to launch it from a machine *outside* the cluster. If you launch if from a cluster node, it will be disabled when the node from which it is launched reboots.

Perform the following steps, as root, to run `cluster_tool_setup`.

1. **Go to the `Product` directory on the Sun HPC ClusterTools 3.0 distribution CD-ROM.**

   Note that this directory must be mounted on (accessible by) all nodes in the cluster.

   ```
   # cd /cdrom/hpc_3_0_ct/Product/Install_Utilities
   ```

2. **If you are running on a node within the cluster, perform Step a. If you are running on a machine *outside* the cluster, perform Step b.**

   a. ***Within* the cluster, run `cluster_tool_setup --c`.**

   Run `cluster_tool_setup`; use the `--c` tag to specify the directory containing the `hpc_config` file.

   ```
   # ./cluster_tool_setup --c /config_dir_install
   ```

   b. ***Outside* the cluster, run `cluster_tool_setup --c --f`.**

Run `cluster_tool_setup`; use the `--c` tag to specify the directory containing the `hpc_config` file, plus a trailing `--f` tag.

```
# ./cluster_tool_setup --c /config_dir_install --f
```

3. **Set the `DISPLAY` environment variable to the machine on which you will be running the CCM tools.**

```
# setenv DISPLAY hostname:0
```

(This example uses C-shell syntax.)

4. **Invoke the CCM tool of your choice: `cconsole` (if the nodes are connected to a terminal concentrator), `ctelnet`, or `crlogin`.**

All three tools reside in `/opt/SUNWcluster/bin`. For example,

```
# /opt/SUNWcluster/bin/ctelnet clustername
```

where *clustername* is the name of the LSF cluster. All three CCM tools require the name of the cluster as an argument.

The CCM tool then creates a Common Window and separate Term Windows for all the nodes in the cluster.

5. **Position the cursor in the Common Window and press Return.**

This activates a prompt in each Term Window. Note that the Common Window does not echo keyboard entries. These appear only in the Term Windows.

You can now use CCM to remove previous release packages, as described in “Removing and Reinstalling Individual Packages” on page 53”, or to install the software packages, as described in “Installing Software Packages” on page 49.”

Installing Software Packages

This section describes the procedure for installing the Sun HPC ClusterTools packages. Note that the exact procedure for each step will depend on which installation mode you are in, `cluster-tool` or `rsh`.

- In `cluster-tool` mode, perform each step in the Common Window. Each entry will be echoed in every Term Window.
- In `rsh` mode, perform each step at the shell prompt of one of the nodes.

See Appendix BB for more information about the CCM tools that are available to you in `cluster-tool` mode.

Note - The `hpc_install` command writes various `SYNC` files in the directory containing its configuration file as part of the package installation. If the installation process stops prematurely—if, for example, you press `Ctrl--c`—some `SYNC` files may be left. You must remove these files before executing `hpc_install` again so they don't interfere with the next software installation session.

---

1. **Log in to each node as root.**

2. **Go to the Product directory on the Sun HPC ClusterTools 3.0 distribution CD-ROM.**

   Note, this directory must be mounted on (accessible by) all nodes in the cluster.

   ```
   # cd /cdrom/hpc_3_0_ct/Product/Install_Utilities
   ```

3. **Run** `hpc_install`**.**

   ```
   # ./hpc_install --c /config_dir_install
   ```

   where *config_dir_install* represents the directory containing the `hpc_config` file.

   The `--c` tag causes `hpc_install` to look for a file named `hpc_config` in the specified directory. If you want to install the software using a configuration file with a different name, you must specify a full path including the new file name after the `--c` tag.

   Note - If the `hpc_config` file contains an `INSTALL_SCI="yes"` entry, `hpc_install` will install the three SCI software packages along with the other Sun HPC ClusterTools packages. When the SCI packages are installed, the installation script will display a message telling you to reboot the nodes. Ignore this message. You must reboot the nodes only after any SCI interface cards are configured. If the system does not include SCI hardware, the nodes do not need to be rebooted.

Removing the Software

To remove LSF, see the documentation that came with the software.

The easiest way to remove Sun HPC ClusterTools 3.0 software is by using the configuration tool, `install_gui`. See the next section for details. If you prefer to remove the software at the command line, you may do so using the provided removal scripts. See “Removing the Software: Command Line” on page 52 for instructions.

Removing the Software: Configuration Tool

1. **Locate a configuration file or files for the cluster.**

   To remove the software from your cluster, you will need a configuration file that describes the cluster. Ideally you should use the configuration file you created when installing the software. If you cannot locate that file, you will have to create one. You can use the configuration tool to create the file. (See Chapter 3 of the *Sun HPC ClusterTools 3.0 Installation Guide*.)

   Note - The configuration tool will remove the software from up to 16 nodes at once. If you need to remove software from a cluster of more than 16 nodes, you must remove it first from a group of up to 16 of the nodes in your cluster. Then remove from more nodes by repeating the removal process on additional groups of nodes until you have removed the software from all the nodes in the cluster. The procedure is similar to installing the software on a cluster of more than 16 nodes. See Section 3.1.2 of the installation guide for more information.

   ---

2. **Load the Sun HPC ClusterTools 3.0 CD-ROM in the CD-ROM drawer.**

   The CD-ROM mount point must be mounted on all the nodes in the cluster.

3. **Enable root login access.**

   By default, most systems allow logins by root only on their console devices. To enable root login access during software removal, you must edit the `/etc/default/login` file on each node in the cluster. In this file on each node, find this line:

   ```
   CONSOLE=/dev/console
   ```

and make it into a comment by adding a # before it:

```
#CONSOLE=/dev/console
```

After removing the software, you should disable root login access again if your site's security guidelines require it.

1. **As root, launch the `install_gui` tool with the configuration file.**

   You can load the configuration file either from the command line or from within the tool after it has been launched.

   - At the command line, launch the configuration tool using the name of the configuration file as an argument:

     ```
     # /cdrom/hpc_3_0_ct/Product/Install_Utilities/install_gui hpc_config
     ```

   - Alternatively, you can load the configuration file after launching the tool by choosing Load from the File menu.

2. **Select the Remove task and click on the Go button.**

For help using the configuration tool, choose Help with Configuration Tool from the Help menu.

Removing the Software: Command Line

1. **Locate a configuration file or files for the cluster.**

   To remove the software from your cluster, you will need a configuration file that describes the cluster. Ideally you should use the configuration file you created when installing the software. If you cannot locate that file, you will have to create one.

   Note - You can use the CCM tools to install on up to 16 nodes at a time. For clusters with more than 16 nodes, you will have to repeat the installation process on groups of up to 16 nodes at a time until you have installed the software on the entire cluster.

   ---

2. **Place the Sun HPC ClusterTools 3.0 distribution CD-ROM in the CD-ROM drive.**

3. **Go to the directory on the CD-ROM containing the release packages.**

   This directory must be mounted with read/execute permissions (755) on all the nodes in the cluster:

```
# cd /cdrom/hpc_3_0_ct/Product/Install_Utilities/
```

4. **Run** `hpc_remove`; **use the** `--c` **option to specify the directory containing the** `hpc_config` **file.**

```
# ./hpc_remove --c /config_dir_install
```

The `--c` tag causes `hpc_remove` to look for a file named `hpc_config` in the specified directory. If you want to remove the software using a configuration file with a different name, you must specify a full path including the new file name after the `--c` tag.

Removing and Reinstalling Individual Packages

To remove a single package and install (or reinstall) another package in its place, perform the following steps:

```
#./hpc_remove --c hpc_config_file_path -d PACKAGE_NAME
#./hpc_install -c config_dir -d location_of_package/PACKAGE_NAME
```

For example:

```
# cd /cdrom/hpc_3_0_ct/Product
#./hpc_remove --c /home/hpc_admin -d SUNWhpmsc
#./hpc_install -c /home/hpc_admin -d /cdrom/hpc_2_0_sw/Product/SUNWhpmsc
```

APPENDIX **B**

Cluster Management Tools

This appendix describes a set of cluster administration tools that are installed with the Sun HPC ClusterTools 3.0 release.This toolset, called the Cluster Console Manager (CCM), allows you to issue commands to all nodes in a cluster simultaneously through a graphical user interface. The CCM offers three modes of operation:

- `cconsole` – This interface provides access to each node's console port through terminal concentrator links. To use this tool, the cluster nodes must be connected to terminal concentrator ports and those node / port connections must be defined in the `hpc_config` file. See the *Sun HPC ClusterTools 3.0 Installation Guide* for details.
- `ctelnet` – This interface initiates simultaneous `telnet` sessions over the network to all nodes in the cluster. Note that if passwords are required, every node must be able to accept the same password.
- `crlogin` – This interface uses `rlogin` to log you in to every node in the cluster. Note that if you launch `crlogin` while logged in as superuser, all `rlogin` sessions will be done as superuser. Likewise, if `crlogin` is launched from an ordinary user prompt, all remote logins will be done as user.

Each of these modes creates a command entry window, called the *Common Window*, and a separate console window, called a *Term Window*, for each node. Each command typed in the Common Window is echoed in all Term Windows (but not in the Common Window). Every Term Window displays commands you issue as well as system messages logged by its node.

Note - If the cluster nodes are not connected to a terminal concentrator (for example, the Sun HPC 10000 has no provision for a terminal concentrator), only `ctelnet` and `crlogin` can be used, not `cconsole`.

Launching Cluster Console Tools

All CCM tools are launched using the same command-line form:

```
% tool_name clsluster_name
```

where *tool_name* is `cconsole`, `ctelnet`, or `crlogin`, and *cluster_name* is a name given to the cluster at installation time. For example,

- Launch `ctelnet` by entering:

```
% ctelnet hpc_cluster
```

- Launch `crlogin` by entering:

```
% crlogin hpc_cluster
```

- Launch `cconsole` by entering:

```
% cconsole hpc_cluster
```

If you want to use `cconsole` to monitor messages generated while rebooting the cluster nodes, you will need to launch it from a machine outside the cluster. If you launch if from a cluster node, it will be disabled when the node from which it is launched reboots.

Note - Because `cconsole` accesses the console ports of every node in the cluster, no other accesses to any console in the cluster will be successful while the `cconsole` session is active.

Note that all three CCM commands take the standard X/Motif command-line arguments.

Common Window

The Common Window is the primary window used by the system administrator to send input to all the nodes. This window has a menu bar with three menus and a text field for command entry. The Common Window is always displayed when CCM is launched.

Menu Bar

The menu bar has three menus:

- Hosts
- Options
- Help

Note that in this manual, the CCM term *Hosts* refers to Sun HPC Cluster nodes.

Hosts Menu

The Hosts menu displays a list of the nodes contained in the cluster, plus two other entries, `Select Hosts` and `Exit`. Table B–1 describes these menu choices.

TABLE B–1 CCM Menu Entries

Entry	Function
Host toggle buttons	Selects whether or not the host gets input from the Common Window text field. There is a separate toggle button for each node currently connected to the CCM. ON — Enables input from the Common Window text field to the node. OFF — Disables input from the CCM.
Select Hosts	Displays the Select Hosts dialog window. See “Select Hosts Dialog Box” on page 57” for details.
Exit	Quits the CCM program.

Select Hosts Dialog Box

The Select Hosts dialog enables you to add or delete nodes during the current CCM session. The scrolled text window in the Select Hosts dialog displays a list of the nodes that are currently connected to CCM.

There are three Select Hosts dialog buttons, which are described in Table B–2

TABLE B–2 Select Hosts Dialog Buttons

Entry	Function
Insert	Opens a Term Window and establishes a connection to the specified hosts(s). Adds the host(s) specified in the Hostname text field to the list of accessible hosts. The inserted host name(s) are displayed in the hosts list in the scrolled text window and in the Common Window.
Remove	Deletes the host selected in the Hosts list in the scrolled text window.
Dismiss	Closes the Select Hosts dialog.

▼ To Add a Single Node

1. **Enter the *hostname* in the Hostname text field.**

2. **Select Insert.**

   Entering a valid host name opens a Term Window for the specified host and establishes a connection to that host. The name of the selected host appears in the scrolled text window and in the hosts list on the Hosts menu in the Common Window.

▼ To Add All Nodes in a Cluster

1. **Enter the *clustername* in the Hostname text field.**

2. **Select Insert.**

   CCM automatically expands the cluster name into its constituent host names and then opens one Term Window for each node. A connection is established for each of the constituent host names. CCM automatically displays the names of the hosts in the cluster in the scrolled text window and in the hosts list on the Hosts menu in the Common Window.

▼ To Remove a Node

1. **Select the name of the host in the list in the scrolled text window.**

2. **Select Remove.**

   This closes the corresponding Term Window and disconnects the host. The name of the removed host disappears from the scrolled text window and from the hosts list on the Hosts menu in the Common Window.

Options Menu

The Options menu has one entry, Group Term Window; see Table B–3 for a description.

TABLE B–3 Group Term Window Entry

Entry	Function
Group Term Windows	This is a toggle button that groups and ungroups the Common Window and the Term Window. ON — Group; the Term Windows follow the Common Window when the Common Window is moved. OFF — Ungroup: the Term Windows and the Common Window move independently.

Help Menu

The Help Menu has three entries; see Table B–4 for a description

TABLE B–4 Help Menu

Entry	Function
Help	Displays a Help window—the interface to the Sun online help system.
About	Displays the About box, which contains information on the CCM application, such as version number.
Comments	Displays the Comments box, which allows you to enter comments about the software and send them to the development team.

Text Field

The text field is where you enter commands you want executed simultaneously on multiple nodes. The state of the host toggle buttons under the Hosts menu determines which nodes receive this input.

Term Windows

The Term Window is just like a normal terminal window. To type on only one host, move the cursor to the Term Window of the desired host and type directly into it.

CCM Term Windows are like other terminal programs, such as `xterm`, `cmdtool`, and `shelltool`, except that they can also receive input from the Common Window. The Term Windows use VT220 terminal emulation.

The environment variable `TERM` informs your editor of your terminal type. If you are having display problems from `vi` or any other tools, set the environment variable using the appropriate commands for your shell.

The Term Window contains additional functionality, which you can access by positioning the pointer over the Term Window and pressing the right mouse button. This displays the menu described in Table B–5

TABLE B–5 Term Window Menu Entries

Entry	Function
Disable/Enable Scroll bar	Toggles the scroll bar display on and off in the Term Window.
Exit This Window	Closes the current Term Window.

Using CCM

To issue commands to multiple nodes simultaneously:

1. **Position the cursor in the text field of the Common Window and enter your command.**

   Every keystroke entered in this field is sent to all hosts that are currently selected for input.

To issue commands to a single node:

1. **Position the cursor in the corresponding Term Window and enter your command.**

   Alternatively, you can turn off all hosts in the Hosts menu, except the one you want to access. Then issue your commands from the Common Window.

Administering Configuration Files

Two configuration files are used by CCM tool: `clusters` and `serialports`. These files are created automatically by `cluster_tool_setup`, which places them in `/etc`. (These files are not updated automatically; if you later change cluster characteristics, you must update these files manually.)

The `clusters` File

The `clusters` configuration file maps a cluster name to the list of host names that make up the cluster Each line in this database corresponds to a cluster. The format is:

```
clustername hostname–1 hostname–2 [...] hostname–n
```

For example:

```
cities
chartres izmir tampico inchon essen sydney
```

The `clusters` file is used to map cluster names to host names on the command line and in the Select Hosts dialog.

The serialports File

The `serialports` file maps each host name to the terminal concentrator and the terminal concentrator serial port to which it is connected. Each line in this database specifies a separate serial port using the format:

```
hostname  terminal_concentrator  serial_port
```

For example:

```
chartres
cities-tc 5002
izmir
cities-tc 5003
```

The `serialports` file is used by `cconsole` to determine which terminal concentrator and serial ports to connect to for the various cluster nodes that have been specified on the command line or the Select Hosts dialog.