

# SGI™ GenomeCluster™ Administrator's Guide

007-4370-001

---

## CONTRIBUTORS

Written by Julie Boney

Production by Glen Traefald

Edited by Susan Wilkening

Illustrated by Chrystie Danzer

Technical contributions by Jonathan Sparks, Joe Landman, Dan Stevens

---

## COPYRIGHT

© 2001 Silicon Graphics, Inc. All rights reserved; provided portions may be copyright in third parties, as indicated elsewhere herein. No permission is granted to copy, distribute, or create derivative works from the contents of this electronic documentation in any manner, in whole or in part, without the prior written permission of Silicon Graphics, Inc.

---

## LIMITED RIGHTS LEGEND

The electronic (software) version of this document was developed at private expense; if acquired under an agreement with the USA government or any contractor thereto, it is acquired as "commercial computer software" subject to the provisions of its applicable license agreement, as specified in (a) 48 CFR 12.212 of the FAR; or, if acquired for Department of Defense units, (b) 48 CFR 227-7202 of the DoD FAR Supplement; or sections succeeding thereto. Contractor/manufacturer is Silicon Graphics, Inc., 1600 Amphitheatre Pkwy 2E, Mountain View, CA 94043-1351.

---

## TRADEMARKS AND ATTRIBUTIONS

IRIX is a registered trademark and SGI, the SGI logo, SGI GenomeCluster, SGI Linux, SGI ProPack for Linux, and Performance Co-Pilot are trademarks of Silicon Graphics, Inc.

Digi International is a trademark of Digi International, Inc. Intel is a registered trademark of the Intel Corporation. Linux is a registered trademark of Linus Torvalds. OpenPBS and Portable Batch System are trademarks of Veridian Systems, Inc. Red Hat and RPM are trademarks of Red Hat Software, Inc. RocketPort is a registered trademark and Control is a trademark of Control Corporation. SuSE is a trademark of SuSE.

Cover design by Sarah Bolles, Sarah Bolles Design, and Dany Galgani, SGI Technical Publications.

---

## Record of Revision

<b>Version</b>	<b>Description</b>
001	February 2001 Original publication.



---

# Contents

<b>About This Guide</b>	<b>xi</b>
Related Publications and Web Sites	xi
Obtaining Publications	xiii
Conventions	xiv
Reader Comments	xiv
<b>1. Introduction</b>	<b>1</b>
Terminology	1
Assumptions	2
Contents of the SGI GenomeCluster Release	2
Space Requirements	2
<b>2. CT-BLAST</b>	<b>5</b>
Invoking and Using CT-BLAST	5
CT-BLAST Options	6
CT-BLAST Arguments	8
Configuring the Code	9
<b>3. Topology</b>	<b>15</b>
<b>4. Verify the Installation</b>	<b>19</b>
What confidence Verifies	19
Using the confidence Script	19
<b>5. Troubleshooting</b>	<b>25</b>
No rsh/rcp Access	25
<b>007-4370-001</b>	<b>v</b>

Remove /etc/securetty File . . . . .	25
Change File Permissions . . . . .	25
Check PAM Authentication . . . . .	25
PBS Jobs Fail to Start . . . . .	26
Cannot Submit Jobs for PBS as root . . . . .	26
Getting PBS State Information . . . . .	27
Root Access for LSF . . . . .	27
PCP Errors . . . . .	28
Queue Priorities/Nice and PCP . . . . .	28
NFS Version Defaults to 2 . . . . .	29
<b>Index . . . . .</b>	<b>31</b>

---

## Figures

<b>Figure 3-1</b>	Standard Head/Execution Node Topology . . . . .	16
<b>Figure 3-2</b>	Optional Serial Console Topology . . . . .	17



---

## Tables

<b>Table 1-1</b>	Approximate Space Requirements Per Package (in MB)	. . . . .	3
------------------	--	-----------	---



---

## About This Guide

This document discusses SGI GenomeCluster, release 1.0, running on SGI 1100, SGI 1200, SGI 1400, or SGI 1450 systems. It assumes that one of the following Linux operating systems has already been installed on the systems in your cluster.

- SuSE 6.4
- Red Hat 6.2
- TurboLinux 6.0/6.02

The Load Sharing Facility (LSF) supports the following Linux operating systems:

- RedHat 6.1
- RedHat 7.0 (Kernel 2.2.16 and 2.4-test10)
- SuSE 6.4
- OpenLinux 2.4
- TurboLinux 6.0.4
- Debian Linux 2.2

This release also supports SGI ProPack for Linux version 1.4. By using SGI ProPack, you can avoid rebuilding the kernel for use with some products.

This document explains how to perform administrative tasks specific to SGI GenomeCluster.

## Related Publications and Web Sites

The following documents and Web sites contain additional information that may be helpful:

- SGI GenomeCluster man pages:
  - `clusterinfo(1)`, which displays information about the SGI GenomeCluster environment
  - `addnode(1)`, which adds a PBS client to the server's list

- `cadduser(1)`, which creates a user account on all cluster nodes
- `cdeluser(1)`, which deletes a user account on all cluster nodes
- `cping(1)`, which performs a clusterwide ping
- `cpush(1)`, which pushes a file or directory to all cluster nodes
- `crsh(1)`, which executes a command on all nodes in the cluster
- `creboot(8)`, which reboots all nodes in the cluster
- `chalt(8)`, which halts all nodes in the cluster
- `delnode(1)`, which removes a PBS client from the server's list
- SGI ProPack for Linux:
  - SGI ProPack site: <http://support.sgi.com/linux/index.html>
  - *SGI ProPack 1.4 for Linux Start Here* available from the SGI online Technical Publications Library at <http://techpubs.sgi.com/>
- General Linux clusters:
  - *Red Hat Linux 6.0 The Official Red Hat Linux Installation Guide*
  - *How to Build a Beowulf: A Guide to the Implementation and Application of PC Clusters*. Thomas L. Sterling, John Salmon, Donald J. Becker, Daniel F. Savarese. ISBN: 026269218X.
  - How to Build a Beowulf: a Tutorial  
<http://www.cacr.caltech.edu/beowulf/tutorial/tutorial.html>
  - *Beowulf Installation and Administration HOWTO*  
[http://buweb.parl.clemson.edu/doc\\_project/BIAA-HOWTO/Beowulf-Installation-and-Administration-HOWTO.html](http://buweb.parl.clemson.edu/doc_project/BIAA-HOWTO/Beowulf-Installation-and-Administration-HOWTO.html)
- Portable Batch System (PBS):
  - PBS Web site: <http://www.openpbs.org>
  - *PBS Administrator's Guide*
- LSF:
  - LSF Web site: <http://www.platform.com>

- Performance Co-Pilot (PCP) (orderable through SGI):
  - PCP open source Web site: <http://oss.sgi.com/projects/pcp>
  - PCP product web page: <http://www.sgi.com/software/co-pilot>
  - Manuals available from the SGI online Technical Publications Library <http://techpubs.sgi.com/>:
    - *Performance Co-Pilot User's and Administrator's Guide*
    - *Performance Co-Pilot Programmer's Guide*
- SGI SystemImager:
  - *SGI SystemImager Software Installation and Administration Guide* available from the SGI online Technical Publications Library <http://techpubs.sgi.com/>

The SGI SystemImager tool is based on the VA SystemImager product. For more information on VA SystemImager, see the following site:

<http://www.systemimager.org/>

  - Documentation in the `/mnt/cdrom/doc/sgi/ace-extras` directory on the CD and (after installation) in `/usr/share/doc/ace-extras/sgi-SystemImager-version/SIGUI_AG` directory
- Digi International: <http://www.digi.com/>
- Comtrol: <http://www.comtrol.com>

---

**Note:** The third-party documents listed here are not orderable from SGI.

---

## Obtaining Publications

To obtain SGI documentation, go to the SGI Technical Publications Library at:

<http://techpubs.sgi.com>.

## Conventions

The following conventions are used throughout this document:

<b>Convention</b>	<b>Meaning</b>
<code>command</code>	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
<i>variable</i>	Italic typeface denotes variable entries and words or concepts being defined.
<b>user input</b>	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. Output is shown in nonbold, fixed-space font.

## Reader Comments

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

You can contact us 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 World Wide Web page:

`http://techpubs.sgi.com`

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

- Send mail to the following address:

Technical Publications  
SGI  
1600 Amphitheatre Pkwy., M/S 535  
Mountain View, California 94043-1351

- Send a fax to the attention of “Technical Publications” at +1 650 932 0801.

We value your comments and will respond to them promptly.



## Introduction

This document discusses the software that SGI provides for SGI GenomeCluster computations. This software, called CT-BLAST, is a sequence similarity search tool that supports database analysis performed on Linux clusters. CT-BLAST works with a scheduler and other tools to provide a completely integrated and customized bioinformatics computing solution. CT-BLAST is based on the Basic Local Alignment Search Tool (BLAST), developed by the National Center for Biotechnology Information (NCBI). This document explains how to perform administrative tasks in the SGI GenomeCluster environment.

---

**Note:** Examples in this document may not show the current RPM numbers.

---

## Terminology

This document uses the following terminology:

- *Node.* An SGI 1100, SGI 1200, SGI 1400, or SGI 1450 computer. A node is a host.
- *Execution node.* A node on which a job is run.
- *Head node.* The node that performs administrative tasks for the cluster and is where Portable Batch System (PBS) requests are submitted. Users log into the head node. A head node can also be an execution node, but there can be only one head node.
- *File server.* An SGI Origin 200 server. The Origin 200 also acts as the batch server and scheduler for the cluster.
- *Cluster.* One head node and multiple execution nodes.
- *Remote administration node.* The node from which Lconsole administrative tasks are performed.
- *Serial multiplexer.* A hardware and software product that allows a remote machine to perform administrative tasks. The Digi International EtherLite 16 is the default product available from SGI; the Comtrol RocketPort is an alternative product required for use with the SGIconsole cluster manager.

## Assumptions

The instructions in this document assume the following:

- You have experience with system administration tasks.
- The software has already been installed.
- You know the IP addresses and host names of each node in the cluster. (The host name is the output of the `hostname(1)` command.)

## Contents of the SGI GenomeCluster Release

The SGI GenomeCluster release contains the following RPM packages:

- `sgi-cluster-tools`
- `sgi-ctblast-docs`
- `ctblast`
- `ncbitoolkit` (an open-source product shipped with SGI GenomeCluster)
- `data`

SGI GenomeCluster also supports two third-party batch schedulers: LSF and PBS.

## Space Requirements

To perform the installation, you must have sufficient amounts of free space in `/tmp` and `/usr` on each node in the cluster. For example, to install the software, you should have at least the following amounts of free space on each cluster node:

- 42 MB in `/tmp`
- 64 MB in `/usr` and `/etc`
- 5 MB in `/cache`

Table 1-1 shows the amount of space required by each package.

**Table 1-1** Approximate Space Requirements Per Package (in MB)

RPM Package	Head Node	Execution Node
sgi-cluster-tools	0.3	0.3
sgi-ctblast-docs	1.1	0
ctblast	5.9	0
ncbitoolkit	0	59.2
data	0	5.9

Ensure that you have enough disk space before installing SGI GenomeCluster.



## CT-BLAST

CT-BLAST is a cluster-aware implementation of the NCBI BLAST program. CT-BLAST does not modify any NCBI BLAST code. Rather, it restructures the flow of input and output from multiple copies of the NCBI BLAST code running on cluster computing nodes. CT-BLAST allows the user to minimize the time to return results by load balancing the queries across the cluster computing nodes. Since CT-BLAST does not require a modification of the source code to NCBI BLAST, new versions of NCBI BLAST are trivial to incorporate.

### Invoking and Using CT-BLAST

After CT-BLAST has been installed and configured on the cluster head node, you can invoke it by using the following command:

```
# ctblastall [options]
```

The usage syntax of the `ctblastall` command is a superset of the syntax of the NCBI `blastall` command. You can use your shell to alias the `blastall` command to the `ctblastall` command.

To use CT-BLAST successfully, use the following order of operations:

1. Move your database to the file server (`/data` on the head node).
2. Format your database by using the `formatdb` command.
3. Distribute the database to the compute nodes.
4. Move your input file to the head node and create a directory for the run.
5. Run the `ctblastall` command.
6. Gather the data from the directory.

If you do not need to move or format your data frequently, you may simply script steps 1 through 3 so that you need to run steps 4 through 6 only.

After moving the databases, you can use test data provided by SGI to run a comparison, as follows:

```
# ctblastall -i /usr/share/data/ctblast/small.fsa -o small.out -p blastx -d nr --verbose
```

Another method for populating the configuration file is described in "Configuring the Code", page 9.

## CT-BLAST Options

To obtain a list of options for the `ctblastall` command, simply enter the command and specify no options, as follows:

```
# /usr/local/bin/ctblastall
```

The resulting options list is identical to the options list of the `blastall` command on a single machine. That list should appear approximately as follows:

```
CT-BLAST:  copyright 2000,2001 by SGI
```

```
blastall  arguments:
```

```
-p Program Name [String]
-d Database [String]
  default = nr
-i Query File [File In]
  default = stdin
-e Expectation value (E) [Real]
  default = 10.0
-m alignment view options:
0 = pairwise,
1 = query-anchored showing identities,
2 = query-anchored no identities,
3 = flat query-anchored, show identities,
4 = flat query-anchored, no identities,
5 = query-anchored no identities and blunt ends,
6 = flat query-anchored, no identities and blunt ends,
7 = XML Blast output [Integer]
  default = 0
-o BLAST report Output File [File Out]  Optional
  default = stdout
-F Filter query sequence (DUST with blastn, SEG with others) [String]
  default = T
-G Cost to open a gap (zero invokes default behavior) [Integer]
  default = 0
-E Cost to extend a gap (zero invokes default behavior) [Integer]
  default = 0
```

-X X dropoff value for gapped alignment (in bits) (zero invokes default behavior) [Integer]  
    default = 0

-I Show GI?s in deflines [T/F]  
    default = F

-q Penalty for a nucleotide mismatch (blastn only) [Integer]  
    default = -3

-r Reward for a nucleotide match (blastn only) [Integer]  
    default = 1

-v Number of database sequences to show one-line descriptions for (V) [Integer]  
    default = 500

-b Number of database sequence to show alignments for (B) [Integer]  
    default = 250

-f Threshold for extending hits, default if zero [Integer]  
    default = 0

-g Perform gapped alignment (not available with tblastx) [T/F]  
    default = T

-Q Query Genetic code to use [Integer]  
    default = 1

-D DB Genetic code (for tblast[nx] only) [Integer]  
    default = 1

-a Number of processors to use [Integer]  
    default = 1

-O SeqAlign file [File Out] Optional

-J Believe the query defline [T/F]  
    default = F

-M Matrix [String]  
    default = BLOSUM62

-W Word size, default if zero [Integer]  
    default = 0

-z Effective length of the database (use zero for the real size) [Real]  
    default = 0

-K Number of best hits from a region to keep (off by default, if used a value of 100 is recommended) [Integer]  
    default = 0

-P 0 for multiple hits 1-pass, 1 for single hit 1-pass, 2 for 2-pass [Integer]  
    default = 0

-Y Effective length of the search space (use zero for the real size) [Real]  
    default = 0

```
-S Query strands to search against database (for blast[nx], and tblastx).
3 is both, 1 is top, 2 is bottom [Integer]
  default = 3
-T Produce HTML output [T/F]
  default = F
-l Restrict search of database to list of GI?s [String] Optional
-U Use lower case filtering of FASTA sequence [T/F] Optional
  default = F
-y Dropoff (X) for blast extensions in bits (0.0 invokes default behavior)
[Real]
  default = 0.0
-Z X dropoff value for final gapped alignment (in bits) [Integer]
  default = 0
--- The following are CT specific flags
--verbose          be very verbose about CT-BLAST settings, config, etc.
--debug           turn debugging options on (turns on verbose)
--test           do a test run. Create all files, just do not enqueue
jobs.
--loadbalance=method load balance the run using either the cosein or
mikhailov algorithms {String} Optional.
--chunk=size      number of sequences per chunk [Integer] Optional
```

## CT-BLAST Arguments

CT-BLAST accepts five arguments in addition to the regular `blastall` argument structure. These arguments are specific to CT-BLAST operations. They have been designed specifically not to collide with or overlap the arguments in the NCBI `blastall` code. The CT-BLAST arguments are as follows:

`--verbose`

Turns on verbose reporting of the work in progress. This argument is highly useful for problem detection. All of the output resulting from using this argument is directed to the standard error stream. The verbosity level is quite high, and the results are informative. Using this argument might be a preferred way to interactively run a job.

`--debug`

Turns on verbose reporting and additional tests and analysis. This argument gives you information that you might not need for

day-to-day runs, but it is essential for debugging the actions of CT-BLAST at a particular instant.

`--test`

Turns on verbose reporting and does all of the work except for submitting the job for execution. This argument does not submit the computation, but it does collect the results and test all of the features. It is useful for discovering a problem without overloading the cluster.

`--loadbalance=method`

Turns on the load balancer. The load balancer reorders the input to the jobs to get the best performance out of the cluster. The *method* option specifies a load balancing algorithm, either `cousein` or `mikhailov`. These algorithms give nearly identical performance for most work, but give somewhat different performance for pathological cases. You can also specify the load balancing algorithm in the configuration files by using the `sortingmethod=method` entry.

`--chunksize`

Divides the input file into a size work quantum. There are algorithms within the code to handle this function automatically, so usually, there is no need to use this option. However, you can use it to provide more granularity in controlling how the work gets divided, or you can use it as a cluster stress test (by using small sizes and large inputs).

## Configuring the Code

Configuring the code requires creating a world readable `/etc/ctblast` directory and, at a minimum, an `/etc/ctblast/ctblast.config` (global configuration) file. If you are missing this file or any needed contents, simply running the `ctblastall` command will inform you of what is missing and provide guidelines for setting it up. You can use more than one method to populate this file. One method is described at the beginning of this chapter. An easier method is to use the `/usr/local/bin/configure.pl` program to map a site configuration file into the global configuration file. To build the global configuration file, the root user can run the `configure.pl` program, as follows:

```
# /usr/local/bin/configure.pl -g
```

This program generates the configuration file in the correct location. Each user can run the `configure.pl` program to generate the `~/.ctblastrc` and `.ctblastrc` files, respectively, as follows:

```
# /usr/local/bin/configure.pl -u ; # generate user config file in ~/.ctblastrc
# /usr/local/bin/configure.pl -l ; # generate local config file in ${CWD}/.ctblastrc
```

The configuration files are read in the following order: global, user, local. The last configuration file can override previous configuration files. The `configure.pl` program generates these files from a site file named `/etc/ctblast/ctblast.parameters`. This file is a simple, comma-delimited, flat file database that describes the parameter, specifies which configuration file to put it in, specifies the value of the parameter, indicates where to place the configuration file, and provides a comment about the parameter. You can (and should) customize this file to your needs.

The following example is a sample of an existing `/etc/ctblast/ctblast.parameters` file:

```
# this is a comment
blastall,all,/usr/local/bin/blastall,all,"The absolute path to the blastall
binary?
#this is also a comment
group,all,ctblast,submission_node,"The queue to use for job submission?
sortingmethod,all,cousein,submission_node,"The load balancing method to use?
.
.
.
```

After running the `configure.pl` program, you will have a configuration file that looks approximately like the following example:

```
computenodetmp=/cache/run/
group=ctblast
hostn=/bin/hostname
killjob=/usr/local/bin/qdel
submit=/usr/local/bin/qsub
echo=/bin/echo
dbpath=/cache/db
status=qstat
sortingmethod=cousein
date=/bin/date
pwd=/bin/pwd
```

```
mkdir=/bin/mkdir
tmppath=/tmp
rcp=/usr/bin/rcp
rm=/bin/rm
statuspath=/etc/ctblast/
blastall=/usr/local/bin/blastall
dbhost=sgibio08:/data
scheduler=pbs
```

You can also view these entries by using the `verbose` option or the `debug` option with the `ctblastall` command.

Following is a description of each file entry:

```
computenodetmp=/cache/run/
```

Temporary path on each of the cluster compute nodes.

```
group=ctblast
```

Scheduling resource to use for allocating jobs. This resource may be a label to help the scheduler route the jobs properly or to help decide how resources are allocated based on group. Scheduling decisions are made by the scheduling entity. Therefore, the purpose of this entry is to pass this information on to the scheduler.

```
hostn=/bin/hostname
```

Path to the `hostname` command for the compute nodes.

```
killjob=/usr/local/bin/qdel
```

Path to the program that deletes jobs from a scheduler.

```
submit=/usr/local/bin/qsub
```

Path to the program that adds jobs to a scheduler.

```
echo=/bin/echo
```

Path to the `echo` command on the compute nodes.

```
dbpath=/cache/db
```

Location on the compute nodes at which the databases will reside.

`status=qstat`

Name of file in `/etc/ctblast` that contains a single line, detailing how to run a status inquiry against the current scheduler.

`sortingmethod=cousein`

Method selected for load balancing. Valid options are `cousein`, `mikhailov`, or `none`. If you have a variety of sequence lengths, using this parameter will have a large positive impact on performance. Note: As a result of the way this entry works, the output results will not be in the same order as the input sequences.

`date=/bin/date`

Path to the `date` command on the computational nodes.

`pwd=/bin/pwd`

Path to the `pwd` command (print working directory) on the computational nodes.

`mkdir=/bin/mkdir`

Path to the `mkdir` command on the computational nodes.

`tmppath=/tmp`

Path to local temporary storage used for submitting and processing results.

`rcp=/bin/rcp`

Path to the `rcp` command on the computational nodes.

`rm=/bin/rm`

Path to the `rm` command on the computational nodes.

`statuspath=/etc/ctblast`

Path to the location on the head node at which the `status` command resides (see `status=qstat` above). This means that there is a file named `${statuspath}/${status}` that contains a simple query command about the scheduler for job status. This file is written by the `configure.pl` program.

`blastall=/usr/local/bin/blastall`

Path on the computational nodes to the `blastall` command.

`dbhost=sgibio08:/data`

Name of the file server and the path to the location at which the database files are stored.

`scheduler=pbs`

Name of the scheduling server being used. Currently supported values are `pbs` and `lsf`.



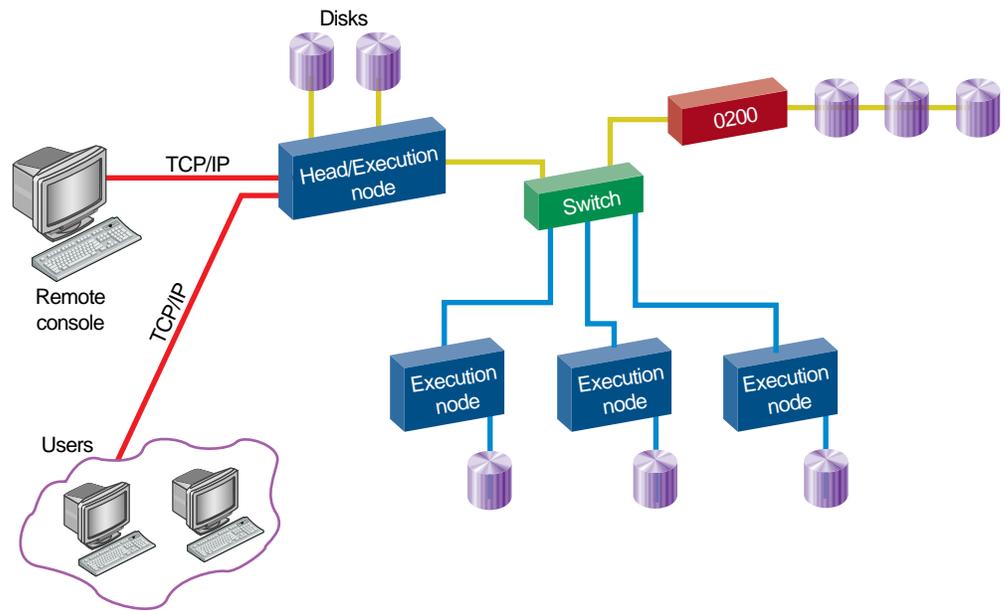
## Topology

SGI GenomeCluster supports the following topologies:

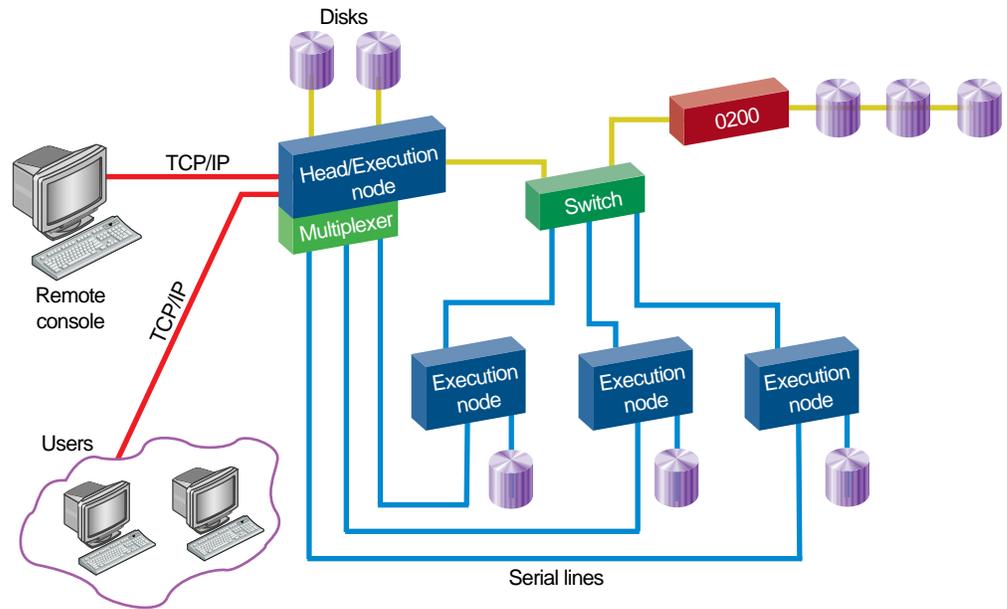
- A head node and multiple execution nodes
- A node that is both the head node and an execution node, plus other execution nodes
- A single node that is both the head node and the execution node (used for testing purposes)

You will also have an SGI Origin 200 *file server* as part of the cluster. Because of the enhanced I/O capabilities of IRIX and the Origin 200 system architecture, the file server serves as the cluster file server and the PBS server/scheduler for the whole cluster. The maximum internal data storage of a single Origin 200 system is approximately 756 GB. This capacity is expandable to 1,699,456 GB.

Figure 3-1, page 16, depicts the standard topology. The nodes must be SGI 1100, SGI 1200, SGI 1400, or SGI 1450 systems. Figure 3-2, page 17, depicts the topology with the optional Lconsole serial-port console feature. Notice that you must use a serial multiplexer when you use this optional feature. In addition, if you use the SGIconsole cluster manager, you must use a RocketPort multiplexer.



**Figure 3-1** Standard Head/Execution Node Topology



**Figure 3-2** Optional Serial Console Topology



## Verify the Installation

After completing the installation, you can run the `confidence` script to verify that the installation process has completed successfully. All of the administration scripts are in the `/usr/local/bin` directory.

### What confidence Verifies

The `confidence` script checks the following:

- That the user can use the `rsh(1)` command among all nodes (in any combination) within the cluster
- That the batch system is correctly configured and functioning
- That a sample test job completes and is correct

### Using the confidence Script

You must run the `confidence` script on the head node.

To run this script, perform the following steps:

1. Log in to the head node as the application user (preferably, not as root).
2. Execute the `confidence` script.
3. Make sure that `/usr/local/bin` is in your path. For example,

(For bash shell)

```
PATH=$PATH:/usr/local/bin; export PATH
```

(For csh shell)

```
setenv PATH "${PATH}:/usr/local/bin"
```

The following example shows output for a successful test on a two-node cluster:

#### 4: Verify the Installation

---

```
[sgigenomecluster@windy test]$ confidence
PBS processes currently running.
LSF processes currently running.
CT-BLAST scheduler selected: lsf
Running schedulers: PBS [yes] LSF [yes]
Testing that user has rsh access to and from all hosts.
Shipping a script to each host that will rsh to all other hosts.
PASS All hosts can rsh to every other host.
INFO hostnames match SGIGenomeCluster node names.
Executing LSF tests
PASS 1 node(s) configured and available to LSF.

Testing that LSF can execute a simple script ...
Submitting test job to default LSF queue. may take some time
Checking that output matches cluster node list ...
PASS LSF bsub output contains node windy output.
PASS LSF bsub output contains node gusty output.
Basic LSF test complete.
Executing ctblast test, this will take some time ~5 minutes
PASS CT-BLAST successful.
confidence: No errors detected. Success!!!
```

To verify the packages and configuration information of a particular node in the cluster, execute the `clusterinfo` command, as follows:

```
$ /usr/local/bin/clusterinfo
```

Output is as follows:

```
##### SGI GenomeCluster info on gusty #####
hosts uname -a info:
Linux gusty 2.2.16-4SGI_39smp #1 SMP Fri Sep 29 00:32:45 EDT 2000 i686 unknown

SGI GenomeCluster RPMS
-----
data-1.0-1
ncbitoolkit-6.1-1
package ctblast is not installed

SGI GenomeCluster files
-----
/etc/nodes file does not exist.
```

```
/etc/ctblast/ctblast.config file exists :
    statuspath=/etc/ctblast/qstat
dbpath=/cache/database
dbhost=sgibio07:/data
submit=/usr/local/bin/qsub
status=qstat
sortingmethod=cousein
killjob=/usr/local/bin/qdel
group=ctblast
tmppath=/tmp
scheduler=pbs
blastall=/usr/local/bin/blastall
End of file
/etc/ctblast/ctblast.parameters file exists :
    # format is as follows
# parameter name,which configuration file it is allowed in, the value of the parameter,
where to place the configuration file, a comment about the parameter
blastall,all,/usr/local/bin/blastall,all,"The absolute path to the blastall binary"
group,all,ctblast,submission_node,"The queue to use for job submission"
sortingmethod,all,cousein,submission_node,"The load balancing method to use"
dbhost,all,sgibio08:/data,all,"The name of the machine:/path acting as a
central repository for the databases"
dbpath,all,/cache/db,all,"The path to the databases on the on the compute node"
tmppath,all,/tmp,all,"The location of a temporary directory to use for job
construction/submission"
computenodetmp,all,/cache/run/,all,"the location of the remote temporary run
directory"
scheduler,all,pbs,submission_node,"The name of the job scheduler to use"
submit,all,/usr/local/bin/qsub,submission_node,"The program used to enqueue a job"
status,all,qstat,submission_node,"The method used to query the run status"
statuspath,all,/etc/ctblast/,submission_node,"The location of the status inquiry method"
getstatus,all,/usr/local/bin/qstat -f ,submission_node,"The command used for status inquiry"
killjob,all,/usr/local/bin/qdel,submission_node,"How to kill errant jobs"
mkdir,all,/bin/mkdir,all,"the mkdir command"
rcp,all,/usr/bin/rcp,all,"the rcp command"
pwd,all,/bin/pwd,all,"the pwd command"
hostn,all,/bin/hostname,all,"the hostname command"
date,all,/bin/date,all,"the date command"
echo,all,/bin/echo,all,"the echo command"
rm,all,/bin/rm,all,"the rm command"
End of file
```

#### 4: Verify the Installation

---

```
~/ctblastrc file does not exist.
.ctblastrc file exists :
  dbhost=windy:/data
status=bjobs
rcp=/usr/bin/rcp
sortingmethod=cousein
killjob=/usr/local/lsf/bin/bkill
pwd=/bin/pwd
hostn=/bin/hostname
group=ctblast
blastall=/usr/local/bin/blastall
statuspath=/etc/ctblast/
dbpath=/cache/db
submit=/usr/local/lsf/bin/bsub
compuodenotmp=/cache/run/
mkdir=/bin/mkdir
tmppath=/data/RUNS
date=/bin/date
rm=/bin/rm
echo=/bin/echo
scheduler=lsf
shell=/bin/bash
End of file
CTBLASTALL user is using lsf scheduler

Checking configuration files ...
No errors found.

LSF 4.0.1, Jun  2 2000
Copyright 1992-2000 Platform Computing Corporation

My cluster name is SGIGenomeCluster
My master name is windy

Checking configuration files ...

There are warning errors.

Do you want to see detailed messages? [y/n] y
Checking configuration files ...
Jan  4 12:17:27 2001 19444 6 4.0.1 minit: Trying to call LIM to get cluster name ...
```

```
Jan  4 12:17:27 2001 19444 6 4.0.1 Batch is enabled
Jan  4 12:17:27 2001 19444 6 4.0.1 JobScheduler is disabled
Jan  4 12:17:27 2001 19444 6 4.0.1 autoAdjustInit: Auto-adjustment is disabled
Jan  4 12:17:27 2001 19444 4 4.0.1 do_Groups: File
/usr/local/lsf/mnt/conf/lsbatch/configdir/lsb.hosts at line 54: Empty hostgroup section
Jan  4 12:17:27 2001 19444 3 4.0.1 main: Local host is not master
Jan  4 12:17:27 2001 19444 6 4.0.1 Checking Done
```

-----  
No fatal errors found.

Warning: Some configuration parameters may be incorrect.  
They are either ignored or replaced by default values.

SGI GenomeCluster processes

```
-----
root      1573      1 0 2000 ?      00:53:09 /usr/local/lsf/etc/lim
root      1575      1 0 2000 ?      00:00:02 /usr/local/lsf/etc/res
root      1577      1 0 2000 ?      00:00:24 /usr/local/lsf/etc/sbatchd
root      1578 1573 0 2000 ?      00:00:34 /usr/local/lsf/etc/pim
root       898      1 0 2000 ?      00:00:00 /usr/share/pcp/bin/pmcd
root     1042      1 0 2000 ?      00:00:00 pmlogger -P -c config.default 20
[biopak@gusty biopak]$
```

To obtain help for confidence, use the `-h` option with the `confidence` command.  
If you want debug information, use the `-v` option.



## Troubleshooting

This chapter highlights possible solutions to common problems.

### No rsh/rcp Access

If you are denied access through `rcp(1)` or `rsh(1)`, one of the following remedies might work for you.

#### Remove `/etc/securetty` File

If the `install` script does not have `rsh` or `rcp` access, it may be because a `/etc/securetty` file exists. Move the file to `/etc/securetty.save` and rerun `install`.

#### Change File Permissions

If you see errors related to `.rhosts` in `/var/log/messages`, it may be because the file permissions are not set properly. Change the file permissions so that `group` and `other` do not have write permission, as in the following example:

```
# chmod 600 .rhost
```

#### Check PAM Authentication

If you run into problems with pluggable authentication modules (PAM), you may need to do one or more of the following:

- If you are using '+' as a wild host name in the `.rhosts` file and the hosts are not being recognized as expected, check the `pam.d/rshd` file entry in the `pam_rhosts_auth` file and ensure that you have specified `promiscuous` in that entry. This specification must be present for the hosts marked '+' to be granted access.
- Change the `rhost.auth.so` entry from `required` to `sufficient`.

If authentication fails, the `required` control token specifies that PAM will call all other modules before denying authentication.

If authentication is successful, the `sufficient` control token specifies that PAM will grant authentication even though a previously required module failed.

- Reorder the `auth` lines in the `/etc/pam.d/rlogin` and `/etc/pam.d/rsh` files as follows:

- Original order:

```
auth required /lib/security/pam_securetty.so
auth sufficient /lib/security/pam_rhosts_auth.so promiscuous
```

- Corrected order:

```
auth sufficient /lib/security/pam_rhosts_auth.so promiscuous
auth required /lib/security/pam_securetty.so
```

For more information about PAM, see the *Red Hat Linux 6.0 The Official Red Hat Linux Installation Guide*.

## PBS Jobs Fail to Start

If PBS jobs fail to start, it is likely that the execution nodes cannot communicate with the head node. Do the following:

- Check that `pbs_mom` is running on the execution nodes.
- Check that the PBS server node is listed in the `mom_priv/config` file.
- Check that the queues are enabled on the server.

## Cannot Submit Jobs for PBS as root

If the root user is prevented from submitting jobs, use another user ID. The default installed SGI GenomeCluster PBS will not be configured so that root can submit requests.

To configure PBS to allow root to submit requests, execute the following on the head node:

```
# /usr/local/bin/qmgr -c 'set server acl_roots="root@*" '
```

## Getting PBS State Information

To get PBS state information on all nodes, execute the following on the head node:

```
# /usr/local/bin/pbsnodes -a
```

If a node is missing, add it to the `/usr/spool/pbs/server_priv/nodes` file and restart PBS by entering the following on the head node:

```
# /etc/rc.d/init.d/pbs restart
```

If a node is down, ensure that the execution node's `/usr/spool/pbs/server_name` file contains the name of the head node. You must restart `pbs_mom` by entering the following on the execution node:

```
# /etc/rc.d/init.d/pbs_mom restart
```

It may take a few moments for the head node to synchronize up with the execution node.

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

## Root Access for LSF

By default, root access across the cluster is not allowed. To allow root access from a remote cluster, specify `LSF_ROOT_REX=all` in the `lsf.conf` file. This specification implies that root jobs from both the local and remote clusters are accepted. This applies to both interactive and batch jobs.

If you want clusters `clus1` and `clus2` to allow root access executions for local jobs only, you must specify `LSF_ROOT_REX=local` in the `lsf.conf` files of both `clus1` and `clus2`. However, if you want `clus2` to also allow root access execution from any cluster, change the line in the `lsf.conf` file for `clus2` to `LSF_ROOT_REX=all`.

---

**Note:** The `lsf.conf` file is host-type specific and not shared across different platforms. You must make sure that the `lsf.conf` file for all of your host types are changed consistently.

---

## PCP Errors

PCP applications will return the following error if the PCP daemons have quit or have not previously been started:

```
application: hinv.ncpu: Unknown metric name
```

```
application: Failed to get the number of CPUs from host "test"
```

To resolve this problem, restart the PCP daemons as follows:

```
# /etc/rc.d/init.d/pcp stop
# /etc/rc.d/init.d/pcp start
```

The PCP daemons must be running on every execution node in the cluster (optionally on the head node).

For information about other problems, see the following files:

- /usr/doc/pcp-*version*/README
- /usr/doc/pcp-ace-*version*/README
- /usr/doc/pcp-pro-*version*/README

## Queue Priorities/Nice and PCP

PCP does not measure the nice priority. The LSF default queue has a nice priority of 20. You should change the queue nice priority to something that will not be preempted and that can be monitored. For example, the contents of the /usr/local/lsf/mnt/conf/lsbatch/bio/configdir/lsb.queues file should be similar to the following:

```
Begin Queue
QUEUE_NAME = normal
PRIORITY   = 1
NICE       = 0
MEMLIMIT   = 5000 # jobs bigger than this (5M) will be niced
STACKLIMIT = 2048
DESCRIPTION = For normal low priority jobs, running only if hosts are lightly loaded.
#QJOB_LIMIT = 60 # job limit of the queue
#UJOB_LIMIT = 5  # job limit per user
#PJOB_LIMIT = 2  # job limit per processor
#RUN_WINDOW = 5:19:00-1:8:30 20:00-8:30
```

```
#rlm = 0.7/2.0    # loadSched/loadStop
#r15m = 1.0/2.5
#pg = 4.0/8
#ut = 0.2
#io = 50/240
#CPULIMIT = 180/apple # 3 hours of host apple
#FILELIMIT = 20000
#DATA LIMIT = 20000    # jobs data segment limit
#CORELIMIT = 20000
#PROCLIMIT = 5        # job processor limit
#USERS = all          # users who can submit jobs to this queue
#HOSTS = all          # hosts on which jobs in this queue can run
#PRE_EXEC = /usr/local/lsf/misc/testq_pre >> /tmp/pre.out
#POST_EXEC = /usr/local/lsf/misc/testq_post |grep -v "Hey"
#REQUEUE_EXIT_VALUES = 55 34 78
End Queue
```

## NFS Version Defaults to 2

The `nfs(5)` man page incorrectly says that the NFS version defaults to version 2. This is a problem when integrating NFS into a standard Linux environment. To get around this problem, you must mount the other Linux filesystems as NFS version 2 by using `nfsvers=2` in the `/etc/fstab` file.

For example:

```
voila:/users /users nfs    exec,dev,suid,rw,nfsvers=2 1 1
```

For more information, see the `fstab(5)` man page.



---

## Index

### A

administration scripts, 19

### B

batch schedulers, 2

### C

/cache requirement, 2

cluster, definition, 1

confidence script

  debug information, 23

  help, 23

  verification, 19

configuration

  CT-BLAST code, 9

  hardware, 15

CT-BLAST

  arguments, 8

  configuration files, 9

  definition, 5

  invoking and using, 5

ctblastall command, 5

### E

errors

  cannot submit jobs as root, 26

  in /var/logs/messages, 25

  PBS jobs fail to start, 26

  PCP, 28

  with pluggable authentication modules  
  (PAM), 25

  /etc/pam.d/rlogin, 26

  execution node, definition, 1

### F

file server, cluster component, 15

file server, definition, 1

### H

hardware configuration, 15

head and execution node, 15

head node, definition, 1

hostname, 2

### I

install utility, 2

installation verification

  single node, 20

  steps, 19

  two-node cluster, 19

introduction, 1

### L

LSF root access, 27

**M**

mom\_priv/config file, 26

**N**

networks, 16  
node, definition, 1

**P**

PBS state information, 27  
pbs\_mom, 26  
PCP errors, 28  
pluggable authentication modules (PAM), 25  
products in SGI GenomeCluster, 2

**Q**

queue priorities, 28

**R**

remote administration node, definition, 1  
required/sufficient control tokens, 25  
rhost.auth.so, 25  
.rhosts errors, 25  
root access for LSF, 27

**S**

serial multiplexer, definition, 2

SGI 1450, SGI 1400, SGI 1200, SGI 1100 systems, 15  
space requirements, 2

**T**

terminology, 1  
/tmp requirement, 2  
topology, 15  
troubleshooting  
    getting PBS state information, 27  
    NFS version defaults to 2, 29  
    no rsh/rcp access, 25  
    PAM problems, 25  
    PBS jobs fail to start, 26  
    PCP errors, 28  
    promiscuous mode, 25  
    queue priorities, 28  
    removing the /etc/securetty file, 25  
    .rhosts errors, 25  
    root access for LSF, 27  
    root cannot submit jobs, 26

**U**

/usr and /etc requirement, 2

**V**

/var/logs/messages, 25