Linux® Cluster Manager
User's Guide

0898016-100
December 2006

The information contained in this document is believed to be correct at the time of publication. It is subject to change without notice. Concurrent makes no warranties, expressed or implied, concerning the information contained in this document.

To report an error or comment on a specific portion of the manual, photocopy the page in question and mark the correction or comment on the copy. Mail the copy (and any additional comments) to Concurrent Computer Corporation, 2881 Gateway Drive, Pompano Beach, Florida, 33069. Mark the envelope **"Attention: Publications Department**." This publication may not be reproduced for any other reason in any form without written permission of the publisher.

Printed in U. S. A.

Revision History:

Date	Level	Effective With
September 2006	000	RedHawk Linux Release 4.1.*x*
December 2006	100	RedHawk Linux Release 4.1.*x*

Contents

List of Figures

List of Tables

Preface

Scope of Manual

This manual is intended for users responsible for the installation and use of the RedHawk Linux Cluster Manager product.

Structure of Manual

This guide consists of the following sections:

- Chapter 1, *Getting Started*, provides an overview of the RedHawk Linux Cluster Manager product and detailed procedures for installing and configuring Cluster Manager on the master system and cluster nodes.
- Chapter 2, *Grid Engine Software*, describes the Grid Engine software used to manage resources and submit jobs and provides instructions for configuring your cluster.
- Appendix A, *Node Information Worksheet*, is an easy-to-use worksheet for recording information needed when configuring the cluster.
- The *Index* contains an alphabetical reference to key terms and concepts and the pages where they occur in the text.

Syntax Notation

The following notation is used throughout this manual:

italic	Books, reference cards, and items that the user must specify appear in *italic* type. Special terms may also appear in *italic*.
`list bold`	User input appears in **`list bold`** type and must be entered exactly as shown. Names of directories, files, commands, options and man page references also appear in **`list bold`** type.
`list`	Operating system and program output such as prompts, messages and listings of files and programs appears in `list` type.
`[]`	Brackets enclose command options and arguments that are optional. You do not type the brackets if you choose to specify these options or arguments.
hypertext links	When viewing this document online, clicking on chapter, section, figure, table and page number references will display the corresponding text. Clicking on Internet URLs provided in **blue** type will launch your web browser and display the web site. Clicking on publication names and numbers in **red** type will display the corresponding manual PDF, if accessible.

Related Publications

RedHawk Linux Operating System Documentation	**Pub No.**
RedHawk Linux Release Notes Version x.x	0898003
RedHawk Linux User's Guide	0898004
RedHawk Linux Frequency-Based Scheduler (FBS) User's Guide	0898005
Real-Time Clock and Interrupt Module (RCIM) PCI Form Factor User's Guide	0898007
iHawk Optimization Guide	0898011
RedHawk Linux FAQ	N/A
Partner Documentation	**Pub No.**
N1 Grid Engine 6 Administration Guide	817-5677-20
N1 Grid Engine 6 Release Notes	817-5678-20
N1 Grid Engine 6 User's Guide	817-6117-20
N1 Grid Engine 6 Installation Guide	817-6118-20

where *x.x* = release version

For more information about Grid Engine, see Chapter 2.

1
Getting Started

1
Getting Started

This chapter describes RedHawk Linux Cluster Manager and provides procedures for installing and configuring the product.

What is RedHawk Cluster Manager?

RedHawk™ Linux® Cluster Manager contains everything needed to install and configure Concurrent's iHawk™ systems into a highly integrated, high performance computer cluster and the user interface to effectively utilize the cluster's full capabilities.

A cluster contains a master host and multiple nodes. Each node contains its own CPU, memory, operating system and I/O subsystem and is capable of communicating with each other. Clusters are used to run parallel programs for time-intensive computations, such as simulations and other CPU-intensive programs that would take an inordinate amount of time to run on regular hardware.

Nodes can contain a hard disk or can be diskless. Any Concurrent iHawk system can be configured as a node in a cluster.

Cluster Manager includes Grid Engine, an open source batch-queuing system, developed by Sun Microsystems, that manages and schedules the allocation of distributed resources such as processors, memory, disk-space, and software licenses. Grid Engine is designed for use on computer clusters and is responsible for accepting, scheduling, dispatching and managing the remote execution of large numbers of standalone, parallel or interactive user jobs.

Cluster Manager is an optional product that can be installed on systems running the corresponding version of the RedHawk Linux operating system; for example, Cluster Manager 4.1on a RedHawk 4.1[.*x*] system.

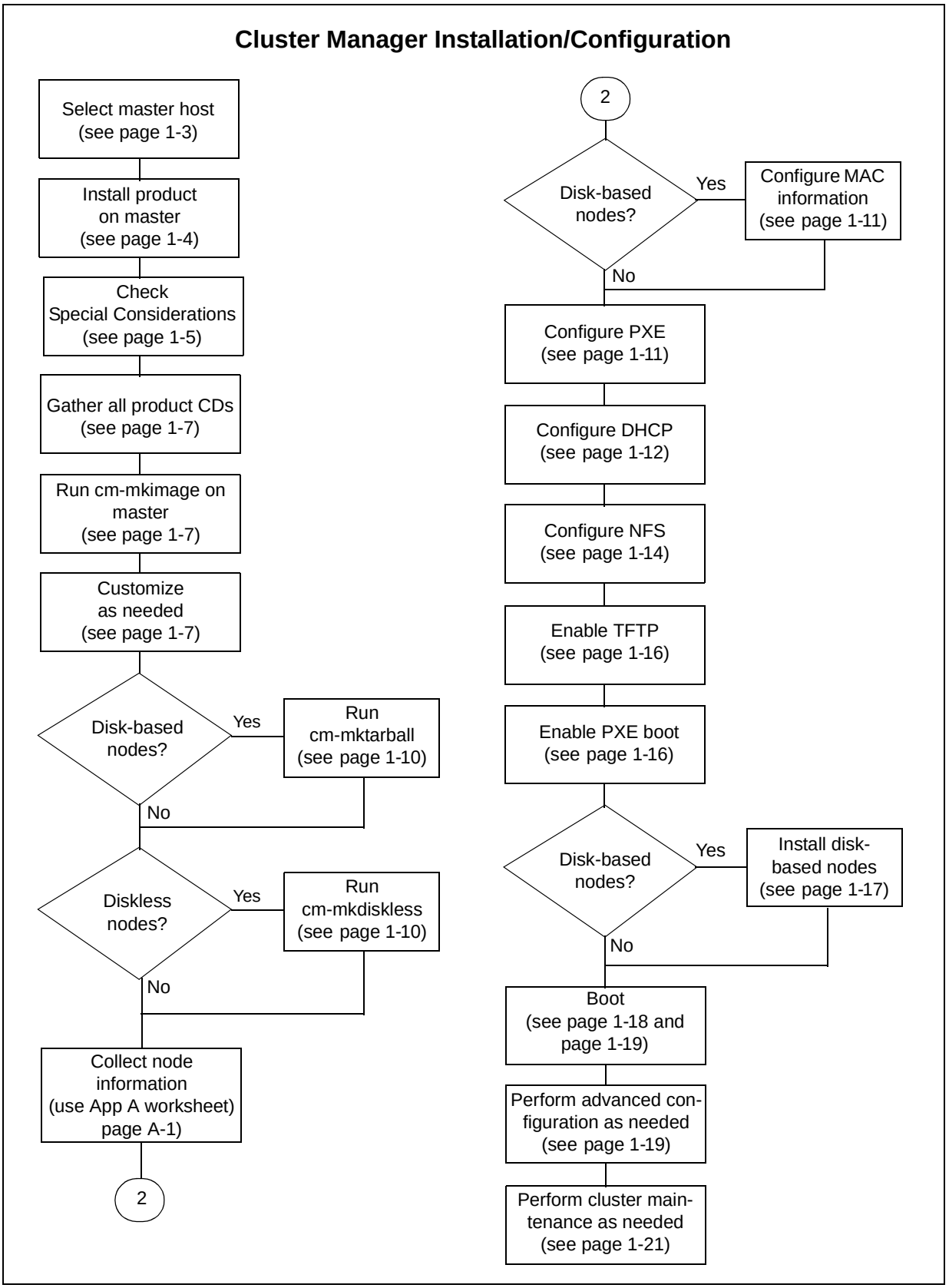
Note that Cluster Manager is based on the open source YACI (Yet Another Cluster Installer) project (pronounced Yak-E) developed at Lawrence Livermore National Laboratories. The string "yaci" is mentioned in various places within this document and while running the Cluster Manager installation and configuration programs. More information about YACI is available at the official YACI web site:
http://www.llnl.gov/linux/yaci/yaci.html

Procedure Flow Chart

Figure 1-1 is a flow chart that illustrates the procedure for installing and configuring your cluster with RedHawk Cluster Manager.

The page numbers in the flow chart are hyperlinks to the appropriate sections where complete information about the step can be found.

Figure 1-1 Cluster Manager Installation/Configuration Flow Chart



Installing Cluster Manager

Prerequisites

Hardware

- Cluster nodes may be any Concurrent iHawk system with at least one NIC that supports PXE booting.

  NOTE: Some BIOSes do not provide an option to boot with PXE. You may use the etherboot utility to work around this. Concurrent does not support this configuration.

- Cluster nodes may contain a local hard disk or may be diskless.
- The cluster master requires that **`/tftpboot/yaci`** have a minimum of 11 GB of free space for creating the cluster file system image.

  NOTE: It is recommended (but not required) that all nodes in the cluster have matching hardware.

Software

- RedHawk Linux Version 4.1.*x* or later

Before Installing

Before installing the software, you should identify one system to be the cluster “master”. It is on this system that you will install the Cluster Manager software. This system will be the master for all of the cluster nodes. It should be a separate dedicated system that can also be used as the Grid Engine master system. If you build a cluster containing mixed architectures (i386 and x86_64), each architecture should have its own master host.

The cluster master requires that **`/tftpboot/yaci`** have a minimum of 11 GB of free space for creating the cluster file system image. This differs from the standard RedHawk configuration. If RedHawk is already installed on the master host, a new disk should be added and mounted under **`/tftpboot/yaci`** ***before*** installing Cluster Manager.

Installing the Product CD

Follow these steps to install Cluster Manager on the master host.

1. On the system designated as the cluster master with RedHawk Linux Version 4.1.*x* or later running, insert the disc labeled "RedHawk Linux Cluster Manager Version 4.1" appropriate to your system's architecture and insert it into the CD-ROM drive.
2. To mount the cdrom device, execute the following command:

   NOTE: **`/media/cdrom`** is used in the examples that follow. Depending on the type of drive attached to your system, the actual mount point may differ. Check **`/etc/fstab`** for the correct mount point.

   ```
   # mount /media/cdrom
   ```

3. To install, execute the following commands:

   ```
   # cd /media/cdrom
   # ./install-cm
   ```

4. When the installation completes, execute the following commands:

   ```
   # cd /
   # umount /media/cdrom
   # eject
   ```

5. Remove the disc from the CD-ROM drive and store.

Product Updates

As Cluster Manager updates are issued, they will be made available for downloading from Concurrent's RedHawk Updates website, **http://redhawk.ccur.com**.

Configuring the Cluster Manager Master System

Configuration Summary

Setting up a cluster involves the following steps, which are described in detail in the sections that follow:

1. Create the file system image which will be used by each of the cluster nodes.
2. Configure various network services (e.g., PXE, DHCP, NFS) on the master system for all of the cluster nodes.
3. Enable TFTP on the master system.

Creating a Cluster File System Image

Creating a cluster file system image involves the following steps:

- run **cm-mkimage** to create a file system image directory
- customize the file system image directory (if desired)
- run **cm-mktarball** to create a compressed tar file used to install disk-based nodes
- run **cm-mkdiskless** to create a root file system ramdisk used to boot diskless nodes

When you run **cm-mkimage**, you are effectively performing a full installation of RedHawk within the cluster's image directory. The installation almost exactly mirrors the process of installing RedHawk onto an actual iHawk system. As such, you should be somewhat familiar with the RedHawk installation process (see the *RedHawk Linux Release Notes* for more information).

Running **cm-mktarball** is only necessary if you plan to install cluster nodes with local hard disks.

Running **cm-mkdiskless** is only necessary if you plan to boot diskless cluster nodes.

The minimum disk space requirements in **/tftpboot/yaci** are as follows:

- **cm-mkimage** 8.2 GB
- **cm-mktarball** 2.8 GB
- **cm-mkdiskless** 256 MB

The actual disk space requirements may increase depending on what software you decide to install in the cluster file system image.

Before You Begin: Special Considerations

The default settings for Cluster Manager should be acceptable for most cluster installations. However, the following sections describe areas that may need site-specific modification. If this is the case for your site, these need to be addressed before running **cm-mkimage**.

For additional information, refer to the configuration files and scripts in the **/tftpboot/yaci/etc** and **/tftpboot/yaci/scripts** directories on the master system.

Node Types

cm-mkimage, **cm-mktarball**, and **cm-mkdiskless** take a 'type' argument. The type argument is optional and will default to 'redhawk' if unspecified. Generally, you should not need to specify a type name explicitly; only specify a type name if you are creating a cluster with multiple node types (see "Multiple Node Types" on page 1-19).

Disk Partitioning

Disk partitioning applies only to disk-based nodes. The disk partitioning of the hard disks on each of the nodes is controlled by the **/tftpboot/yaci/etc/***type***/partition_list** file on the master system. By default, the file **/tftpboot/yaci/etc/partition_list** is used and contains the following settings:

```
#Device MountPoint      Format  SizeMB  Bootable
sda     /boot           ext3    512     *
sda     /               ext3    16384
sda     swap            swap    4096
sda     /home           ext3    rest
```

If you wish to change the default, copy **partition_list** to *type***/partition_list** and edit this new file before running **cm-mkimage**. **cm-mkimage** will copy **partition_list** to *type***/partition_list** (and use the defaults) only if *type***/partition_list** does not exist.

The contents should be fairly self-explanatory. Note that you must ***not*** specify the disk partition number in the device column; that is, use "sda" and not "sda1". The YACI installer will automatically determine the optimal physical partition mapping required.

You are free to modify this file as needed by the requirements of your specific cluster, however be sure to use "sd*" style device names for SCSI and SATA disks and "hd*" style device names for IDE disks.

For more information on disk partitioning, see the **fdisk(8)**, **sfdisk(8)** and **fstab(5)** man pages.

Variables

A few default parameters for the nodes may be changed prior to running **cm-mkimage**. These include:

- the baud rate of the serial console; default is 115200
- the serial port of the serial console; default is ttyS0
- the Ethernet device to broadcast DHCP requests on; default is eth0
- additional space (in megabytes) to allocate in diskless ramdisk
- static configuration of the hostname and booting network interface on disk-based nodes when they are installed; default is to use DHCP each time the nodes boot

If you wish to change these defaults, you may set shell variables in the file **/tftpboot/yaci/etc/***type***/variables** before running **cm-mkimage**. Below are some examples:

```
BAUD="9600"
SERIAL_PORT="ttyS1"
DHCP_DEVICE="eth1"
EXTRA_RAMDISK_MB=50
STATIC_NETWORK="yes"
```

Building the Cluster File System

The **cm-mkimage** script prompts you to insert several CDs during the installation. These CDs were supplied with your original iHawk system or in a later optional purchase.

The CDs that are required are:

- Red Hat Enterprise Linux 4.0 with Update 2 Install Discs 1-4 (i386) or Discs 1-5 (x86_64)
- Red Hat Enterprise Linux 4.0 Updates
- RedHawk Linux 4.1 OS

The following CDs are optional products that you may also choose to install:

- RedHawk Linux 4.1 Documentation
- RedHawk Linux 4.1 Cluster Manager (needed only if SGE will be used on the nodes)
- RedHawk Linux 4.1 Frequency-Based Scheduler
- RedHawk Linux 4.1 PCI-to-VME Bridge Library
- NightStar Tools for RedHawk Linux

To build the cluster file system, invoke the following command as root:

```
# cm-mkimage [type]
```

Insert the CDs as requested and follow the on-screen instructions.

After you have completed the installation of the above CDs, **cm-mkimage** will prompt you whether you wish to install other packages into the cluster file system image. If you choose to do additional installations, you will be placed into a configuration shell where you may install other CDs or download and install updates.

Additional customization, including software installation, may be done at any time. The section “Customizing the Cluster File System” discusses this in more detail.

When **cm-mkimage** is finished, the cluster file system image will be in the directory **/tftpboot/images/***type*.

Customizing the Cluster File System

A cluster file system may be customized by modifying the files in the **/tftpboot/images/***type* directory. This may include manually editing or overwriting configuration files as well as installing additional software. This section discusses some common customizations you may wish to do.

Users and Groups

cm-mkimage automatically copies the **/etc/passwd**, **/etc/shadow**, and **/etc/group** files from the master’s root file system to the cluster file system. You may wish to edit or overwrite these files.

Time Zone

cm-mkimage automatically copies **/etc/sysconfig/clock** from the master's root file system to the cluster file system. You may wish to edit or overwrite this file.

Default Run Level

By default, cluster nodes will boot to run level 3. To change this, edit **/etc/inittab** in the cluster file system directory.

Default Kernel

cm-mkimage configures **/boot/grub/grub.conf** to boot the RedHawk trace kernel by default on disk-based nodes. Edit this file to change the default kernel that is booted on disk-based nodes. See "Kernel Selection" on page 1-20 for more information.

Network Configuration

cm-mkimage configures the cluster file system so that networking is entirely configured using the DHCP protocol when a cluster node boots. (See "Configuring DHCP" on page 1-12 to configure the DHCP server on the master host.)

To configure a static **/etc/hosts** file, edit or overwrite this file under the cluster file system directory.

To statically configure DNS, create the file **/etc/sysconfig/resolv.conf** under the cluster file system directory and configure it appropriately.

To statically configure a default gateway, add the following line to the file **/etc/sysconfig/network** under the cluster file system directory:

```
GATEWAY=nnn.nnn.nnn.nnn
```

The network interface used to install and boot cluster nodes will be configured using the DHCP protocol. To configure additional network interfaces with DHCP, create the file **/etc/sysconfig/network-scripts/ifcfg-eth***N* (where *N* is the appropriate interface number) to contain:

```
DEVICE=ethN
BOOTPROTO=dhcp
```

It is not possible to statically configure network interfaces on diskless nodes, but this is possible on disk-based nodes once the node is installed. To statically configure network interfaces on disk-based nodes, edit the **/etc/sysconfig/network-scripts/ifcfg-eth***N* files on the node's disk after it is installed.

Installed Software

Software may be installed, removed, and updated in the cluster file system directory by using the **cm-chroot** command. This command runs a shell with the root directory being the cluster image directory. All changes made to system files (including software installed or removed) will be done in the cluster image directory only. The master's root file system will not be affected.

cm-chroot must be run as root and has the following usage:

```
cm-chroot [-x command] [type]
```

The default type is 'redhawk'. If a command is given with the **-x** option, that command is executed; otherwise, an interactive bash shell is started.

The following example demonstrates using a software CD to install software in the 'redhawk' cluster image directory:

```
# cm-chroot
redhawk-cluster-image# mount /dev/cdrom /media/cdrom
redhawk-cluster-image# cd /media/cdrom

     (follow CD-specific instructions)

redhawk-cluster-image# umount /dev/cdrom
redhawk-cluster-image# exit
```

Running X Applications

Running X applications under **cm-chroot** requires the following special X configuration on the master host:

1. Edit **/etc/X11/gdm/gdm.conf** and ensure that it has the following line:

   ```
   DisallowTCP=false
   ```

2. Restart the X server.

3. Run the command:

   ```
   $ xhost +
   ```

Updating Software with NUU

Concurrent's Network Update Utility (NUU) is a graphical user interface which allows a user to install, update and remove RPM packages on a system. The packages are downloaded from a remote yum repository. For more information about NUU, see **http://redhawk.ccur.com/nuu**.

NUU may be used to maintain the software installed in a cluster image directory. To do so, perform the following configuration steps:

1. Configure the master host to run X applications (i.e. NUU) under **cm-chroot** (see "Running X Applications" above).

2. Ensure that **/etc/resolv.conf** is configured correctly under the cluster image directory so that NUU can resolve external domain names. One way to do this is to copy **/etc/resolv.conf** from the master host. For example:

   ```
   # cp /etc/resolv.conf /tftpboot/yaci/images/redhawk/etc
   ```

Once configuration is complete, you may use **cm-chroot** to run **nuu**. For example:

```
# cm-chroot -x nuu
```

Building a Compressed Tar File for Disk-Based Nodes

Before installing disk-based cluster nodes, you must first create a compressed tar file of the cluster file system image. This tar file will be used to image the hard disk of disk-based cluster nodes.

To create the compressed tar file, invoke the following command as root:

```
# cm-mktarball [type]
```

When **cm-mktarball** is finished, the cluster file system image will be compressed and placed in the **/tftpboot/tarfiles/***type***.tgz** file where it will be made available to install on nodes with hard disks.

Building a Ramdisk Image for Diskless Nodes

In order to boot diskless nodes, you must first create a ramdisk image containing the root file system used by diskless nodes. Other file systems will be mounted from the master host over NFS once a diskless node is booted.

To create the ramdisk image, invoke the following command as root:

```
$ cm-mkdiskless [type]
```

When **cm-mkdiskless** is finished, the ramdisk image will be compressed and placed in the **/tftpboot/images/***type***-ramdisk.gz** file where it will be made available to diskless nodes.

Configuring Network Services

The following sections provide the steps needed to configure network services on the master system for all of the cluster nodes:

- "Collect Node Information"
- "Configuring the MAC Information File"
- "Configuring PXE"
- "Configuring DHCP"
- "Configuring NFS"

Collect Node Information

Use the **Node Information Worksheet** provided in Appendix A to record the following information.

1. Collect the following information for the entire cluster:
    - The subnet to use for the cluster (e.g. 192.168.1.0)
    - The netmask to use for the cluster (e.g. 255.255.255.0)
    - Broadcast address (e.g. 192.168.1.255)
    - Default router(s) (if any)
    - Domain name server(s) and domain name (if any)

Note that a cluster can exist on a network that includes other systems that are not in the cluster.

2. For each node in the cluster, collect the following information:
   - MAC address of the PXE-capable Ethernet controller in the node
   - IP address to assign to the node
   - Hostname to assign to the node
3. If you choose to have more than one node type in your cluster, establish that scheme at this time. See "Multiple Node Types" on page 1-19 for more information.

Configuring the MAC Information File

This step is only required for nodes having a local hard disk. Diskless nodes need not be listed in this file.

Create a file in **`/tftpboot/yaci/etc`** named **`MAC.info`** with the mapping of MAC addresses to host names. The MAC addresses can be obtained by entering the system's BIOS or, if the cluster node has already been loaded, by booting the system and running **`ifconfig -a`** .

Provide the mapping entries in **`MAC.info`** in the following form:

hostname MAC type

For example:

```
node1       00:0D:56:BA:CE:CF       redhawk
node2       00:0D:56:BA:CE:D2       redhawk
node3       00:0D:56:BA:CE:D4       redhawk
...
```

The *type* field should normally be 'redhawk', though for advanced cluster installations with more than one node type defined it may be something other than 'redhawk' (see "Multiple Node Types" on page 1-19).

Configuring PXE

PXE must be configured to instruct all nodes what to do when they boot.

Once the cluster file system image is created, the following configuration files will reside in the **`/tftpboot/yaci/pxelinux.cfg`** directory. For the 'redhawk' node type these files would be:

`redhawk.install`	instructs a node to (re)install its local hard disk
`redhawk.local`	instructs a node to boot from its local hard disk
`redhawk.diskless`	instructs a node to boot diskless

To determine which nodes perform which of the above actions, create symbolic links to these files according to the file naming rules used by PXE-linux. These rules are:

- It will search for a config file using its own IP address in upper case hexadecimal; e.g., 192.168.1.1 -> C0A80101

- If that file is not found, it will repeatedly remove one hex digit and try again.
- If that fails, it will look for a file named **default**.

As an example, if the IP address is 192.168.1.1 it will try (in order):

```
pxelinux.cfg/C0A80101
pxelinux.cfg/C0A8010
pxelinux.cfg/C0A801
pxelinux.cfg/C0A80
pxelinux.cfg/C0A8
pxelinux.cfg/C0A
pxelinux.cfg/C0
pxelinux.cfg/C
pxelinux.cfg/default
```

You may use the **gethostbyname** program included with Cluster Manager to discover the IP addresses of nodes and convert them to hexadecimal. Here are some examples:

```
$ gethostbyname node1
192.168.1.1
$ gethostbyname -x node1
C0A80101
$ gethostbyname -x 192.168.1.1
C0A80101
```

Examples

To boot all nodes diskless, create the following symbolic link in the **pxelinux.cfg** directory:

```
default -> redhawk.diskless
```

To have all nodes install their local hard disk, create the following symbolic link in the **pxelinux.cfg** directory:

```
default -> redhawk.install
```

Note that once a node installs its local hard disk, a symbolic link is automatically created to boot that system from its local disk. For example, after a few node installations, your **pxelinux.cfg** directory may look like:

```
default -> redhawk.install
C0A80101 -> redhawk.local
C0A80102 -> redhawk.local
C0A80103 -> redhawk.local
```

Configuring DHCP

1. On the master system, copy the example **dhcpd.conf** file to **/etc**:

   ```
   $ cp /usr/share/doc/ccur-yaci-4.1/dhcpd.conf /etc/dhcpd.conf
   ```

   Edit the file by providing the values shown in italics below with those appropriate to your cluster as recorded on your **Node Information**

Worksheet. This example shows a master and three cluster nodes. Refer to **`dhcpd.conf(5)`** if necessary for more information about configuring DHCP.

```
ddns-update-style ad-hoc;
server-name "master-name";
allow bootp;

subnet subnet netmask netmask {
    option subnet-mask netmask;
    option broadcast-address broadcast-address;

    # default gateway
    option routers routers;

    # DNS setup
    option domain-name-servers domain-name-servers;
    option domain-name "domain-name";

    group {
        filename "yaci/pxelinux.0";
        use-host-decl-names on;

        host node1 {
            hardware ethernet node1_MAC;
            fixed-address node1_ipaddress;
        }
        host node2 {
            hardware ethernet node2_MAC;
            fixed-address node2_ipaddress;
        }
        host node3 {
            hardware ethernet node3_MAC;
            fixed-address node3_ipaddress;
        }
    }
}
```

Each node in the cluster must have a unique “host” entry. If you are already serving DHCP from the master server, entries for other machines that are not part of the cluster are allowed and will not interfere with the cluster operation.

2. If cluster nodes have multiple network interfaces which must be configured, they can also be configured from the master’s DHCP server provided that the master is also on the same networks.

   For each additional network, add a subnet declaration and configure IP addresses for the cluster node network interfaces on that network. For example:

```
subnet 192.1.0.0 netmask 255.255.0.0 {
    option subnet-mask 255.255.0.0;
    option broadcast-address 192.1.255.255;

    group {
        host node1-if2 {
            hardware ethernet 00:30:48:59:F7:B7;
            fixed-address 192.1.1.3;
        }
        host node2-if2 {
            hardware ethernet 00:30:48:59:6B:15;
            fixed-address 192.1.1.4;
        }
        host node3-if2 {
            hardware ethernet 00:30:48:59:F7:A3;
            fixed-address 192.1.1.5;
        }
    }
```

3. Once you have the contents of **dhcpd.conf** updated correctly, enable and start the DHCP service by issuing the following commands as the root user:

   ```
   $ chkconfig dhcpd on
   $ service dhcpd start
   ```

   The service should start up properly. If it does not, the most common reason is a syntax error in **/etc/dhcpd.conf**. See the **dhcpd.conf(5)** man page for more information.

Configuring NFS

1. On the master system, copy the example **exports** file to **/etc**. This file controls which machines have access to the NFS exported filesystems of the master server.

   ```
   $ cp /usr/share/doc/ccur-yaci-4.1/exports /etc/exports
   ```

   Edit the file to include entries for each of the nodes in the cluster. The entries are different depending on whether a node is disk-based or diskless.

   For each disk-based node, add the following:

   /tftpboot/yaci *node1*(rw,no_root_squash,sync)

   For each diskless node, add the following entries:

   /tftpboot/yaci/images/redhawk/boot *node1*(ro,no_root_squash,sync)
   /tftpboot/yaci/images/redhawk/usr *node1*(ro,no_root_squash,sync)
   /tftpboot/yaci/images/redhawk/lib *node1*(ro,no_root_squash,sync)
   x86_64 only --> /tftpboot/yaci/images/redhawk/lib64 *node1*(ro,no_root_squash,sync)
   /tftpboot/yaci/images/redhawk/var/lib/rpm *node1*(ro,no_root_squash,sync)
   /tftpboot/yaci/images/redhawk/home *node1*(rw,no_root_squash,sync)

Multiple nodes may be listed in a single entry. Make sure to use the backslash continuation character at the end of each line. For example:

```
/tftpboot/yaci node1(rw,no_root_squash,sync) \
                  node2(rw,no_root_squash,sync) \
                  node3(rw,no_root_squash,sync)
or
/tftpboot/yaci/images/redhawk/boot \
                  node1(ro,no_root_squash,sync) \
                  node2(ro,no_root_squash,sync) \
                  node3(ro,no_root_squash,sync)
/tftpboot/yaci/images/redhawk/usr \
                  node1(ro,no_root_squash,sync) \
                  node2(ro,no_root_squash,sync) \
                  node3(ro,no_root_squash,sync)
/tftpboot/yaci/images/redhawk/lib \
                  node1(ro,no_root_squash,sync) \
                  node2(ro,no_root_squash,sync) \
                  node3(ro,no_root_squash,sync)
x86_64 only --> /tftpboot/yaci/images/redhawk/lib64 \
-->               node1(ro,no_root_squash,sync) \
-->               node2(ro,no_root_squash,sync) \
-->               node3(ro,no_root_squash,sync)
/tftpboot/yaci/images/redhawk/var/lib/rpm \
                  node1(ro,no_root_squash,sync) \
                  node2(ro,no_root_squash,sync) \
                  node3(ro,no_root_squash,sync)
/tftpboot/yaci/images/redhawk/home \
                  node1(rw,no_root_squash,sync) \
                  node2(rw,no_root_squash,sync) \
                  node3(rw,no_root_squash,sync)
```

2. Once you have the contents of **/etc/exports** updated correctly, enable and start the NFS daemons. Issue the following commands as the root user:

   ```
   $ chkconfig nfs on
   $ service nfs start
   ```

   or

   If you already have NFS configured and running on the master system, you may simply issue the following command as root to cause the NFS daemons to re-read the **/etc/exports** file:

   ```
   $ exportfs -rv
   ```

   The NFS service should start (or restart) properly. If it does not, the most common reason is a syntax error in the **/etc/exports** file. See the **exports(5)** man page for more information.

Enabling TFTP

Cluster Manager uses TFTP to download files to the cluster nodes when booting. Enable TFTP on the master server by issuing the following commands as the root user:

```
$ chkconfig tftp on
$ service xinetd restart
```

No separate configuration step is required for this operation, however if your site is concerned with security, you may wish to tighten the access controls of the **tftpd** daemon. For more information, refer to the **tftpd(8)** man page and the **/etc/xinetd.d/tftp** control file.

Booting Cluster Nodes

The cluster nodes can be booted once the master system is configured and the cluster file system image is built. Disk-based nodes will be installed the first time they are booted. The following subsections discuss the steps involved in disk-based node installation and subsequent booting of disk-based and diskless nodes.

Enabling PXE Booting

Pre-Execution Environment (PXE) booting must be enabled on every cluster node.

NOTE

If the BIOS on your system does not support PXE booting, 'etherboot' may be an option; however, Concurrent does not support this configuration. The following instructions apply to systems with a BIOS that supports PXE booting.

To enable PXE booting, do the following:

1. Reboot the cluster node and stop the system immediately after POST (Power-On Self-Test), normally by pressing F2, to get into the BIOS settings menu.
2. Each iHawk machine type has a slightly different BIOS settings menu, however the general rule is to navigate to the 'PCI Device' or the 'Integrated Devices' section of the BIOS menu and enable PXE boot on the first Ethernet interface that is present. Ensure that the chosen interface is connected to a switch that is present on the same network as the master system.

NOTE

The MAC address of the Ethernet interface on which you choose to enable PXE booting must match the MAC address for the node that you placed into the **`MAC.info`** file (see "Collect Node Information" on page 1-10).

3. Verify that the 'Boot Device Order' is set so that the system will attempt to PXE boot ***first*** before it attempts to boot from either the floppy, CD-ROM or hard-disk. This step is very important as the node will not successfully perform auto-installation unless PXE booting is the first boot method tried.

NOTE

If there are no PXE devices listed for Boot Device Order, save the BIOS settings, exit the BIOS settings menu, restart the system and re-enter the BIOS menu in order to make the PXE device options enabled in step 2 available for this step.

4. Once the BIOS settings are correct, save the settings and exit the BIOS settings menu.

Understanding the Boot Sequence

Once PXE is enabled, a cluster node will perform the following sequence of events when booting:

1. Send a DHCP broadcast
2. Receive a DHCP response from the master system—response specifies that the 'pxelinux.0' boot-loader should be booted
3. Boots the pxelinux.0 boot-loader and searches for a PXE configuration file to use for this node
4. pxelinux.0 follows the instructions in the PXE configuration found on the master system

Installing Disk-Based Nodes

To install a cluster image on a disk-based node, verify that the **`/tftpboot/yaci/pxelinux.cfg`** directory on the master system is configured so that the node will use the '*type*`.install`' PXE file (see "Configuring PXE" on page 1-11 for more information). Then reboot the node.

When the system boots, the pxelinux.0 boot-loader does the following:

1. Downloads and boots the YACI installation kernel from the master system
2. Zeroes the entire contents of the local hard disk
3. Partitions the local hard disk
4. Installs the cluster file system image on the local hard disk
5. Configures per-node system files (networking, hostname, etc.)
6. Installs grub into the Master Boot Record of the local hard disk

7. Creates a new PXE configuration file for this node on the master system such that the next boot will be off the local hard disk
8. Reboots

During installation, the node's system console output is redirected to the first serial communications port, known as COM1 or **/dev/ttyS0**. In order to view the node's console output, you must connect a serial terminal device to the correct serial port connector on the node.

Installation Logs

Cluster node installation generally completes without problems once the cluster master is properly configured. However, during the initial configuration of the master system it is possible that a master system configuration error will result in early cluster node installations failing.

Normally, during cluster node installation the serial console of the node displays an ASCII picture of a yak with text printed below it detailing the installation progress. If no text is being output, the installation has almost certainly run into a snag. Fortunately, a log file containing installation progress is written to the master system for each node in the cluster. The log files are located and named according to the following template:

```
/tftpboot/yaci/log/$NODENAME.log
```

By examining the contents of the node-specific log file, you can view the progress made during the node installation and see where the installer stopped if a problem occurred. The most common problems are mis-configurations of **MAC.info**, **dhcpd.conf** and **/etc/exports**. Also, verify that the NFS, DHCP and TFTP servers are running on the master system.

Booting Disk-Based /Nodes

To boot a disk-based node from the local hard disk, verify that the **/tftpboot/yaci/pxelinux.cfg** directory on the master system is configured so that the node will use the '*type***.local**' PXE file (see "Configuring PXE" on page 1-11 for more information). Then reboot the node.

When the system boots, the pxelinux.0 boot-loader boots the grub boot-loader from the local hard disk.

Grub will pause for 10 seconds and display "Press Any Key To Continue" on both the first serial port and the node's attached VGA console (if any). If no key is pressed on the VGA console's keyboard, the node's console will be automatically re-directed to the first serial port. If a key is pressed, the system's console will display on the VGA console's attached monitor.

Grub will then display a menu that presents a choice of kernels to boot on the system's console. If no key is pressed within 10 seconds, the default kernel will be booted (see "Kernel Selection" on page 1-20 for more information). You can use the menu to select an alternate kernel, or edit kernel command line options. See the help text printed below the on-screen menu for more information.

Note that this entire process happens automatically following the installation of a disk-based node.

Booting Diskless Nodes

No installation is required to boot a diskless node. The kernel and file system image are loaded directly from the master system.

To boot a diskless node, verify that the **`/tftpboot/yaci/pxelinux.cfg`** directory on the master system is configured so that the node will use the '*type*`.diskless`' PXE file (see "Configuring PXE" on page 1-11 for more information).

When the system boots, the pxelinux.0 boot-loader displays a prompt on the system's console presenting a choice of kernels to boot. If no key is pressed within 7 seconds, the default kernel will be booted (see "Kernel Selection" on page 1-20 for more information). You can type an alternate kernel name at the prompt. The kernel and root ramdisk are then downloaded from the master system and the kernel is booted.

Advanced Configuration

The following sections discuss more advanced configuration issues that may be suitable to your cluster.

Multiple Node Types

The default node type is 'redhawk'. Additional node types may be used if some nodes must use a different file system image. Each node type uses one file system image. You may create as many file system images as you like, provided you have enough disk space.

NOTE

The creation of each cluster file system image takes considerable disk space. Be sure to configure the master system so that the **`/tftpboot/yaci`** directory is on a large disk partition if you plan to define and create several node types. Refer to "Creating a Cluster File System Image" on page 1-5 for sizing information.

To switch the node type of a disk-based node that has already been installed, refer to "Reinstalling Disk-based Nodes" on page 1-21.

If you decide to create additional node types, for each additional node type desired:

1. Repeat the steps in "Creating a Cluster File System Image" on page 1-5 using the desired type name wherever a type name is optional.
2. Repeat the steps in "Configuring Network Services" on page 1-10 using the desired type name instead of 'redhawk'.

Red Hat Kernels

This section applies only to disk-based nodes. During the creation of the cluster file system image, Cluster Manager assumes that the hardware configuration of the master system will exactly match that of the cluster nodes. In practice, this is not always true (e.g. one node may have a RAID controller for increased disk performance). If the cluster contains non-uniform hardware configuration, the root image on a given cluster node may not be able to successfully boot the Red Hat kernels that are supplied in the root image.

In this case, you will need to manually create an **initrd** file that contains the correct kernel modules needed to boot the Red Hat kernel on the non-uniform node. To do this,

1. First boot the node with the RedHawk kernel.
2. Then, log into the node as the root user and issue the following command:

```
# mkinitrd /boot/initrd-2.4.21-27.0.2.ELsmp.img 2.4.21-27.0.2.EL
```

This command will examine the current hardware configuration and produce an updated and customized **initrd** for the Red Hat kernel that will allow the kernel to successfully boot on the current node.

Note that the specific kernel version numbers may vary based on the version of Cluster Manager being used. Look in the **/boot** directory on the node to see exactly which Red Hat kernel versions are available in the root image.

Kernel Selection

By default, the RedHawk 'trace' kernel is automatically booted on each of the cluster nodes. You can change this default.

For Disk-based Nodes

You can change the default kernel boot setting by editing the **/boot/grub/grub.conf** file.

The **grub.conf** file has a 'default' line that selects which kernel to boot. Normally, the default setting looks as follows:

```
default=#
```

The following table shows how the 'default' setting can be used to select different kernels.

#	Kernel Suffix	Trace	Debug
0	trace	yes	no
1	debug	yes	yes
2	(none)	no	no

Changing this value will change the kernel that is booted by default on each of the cluster nodes.

Note that once a node is installed, it is always possible to log into the node and change the **`/boot/grub/grub.conf`** file on that node individually (just like almost every other aspect of the node's configuration).

For Diskless Nodes

The kernel that is booted on diskless nodes is configured in the **`/tftpboot/yaci/pxelinux.cfg/`***type*`.diskless`** PXE configuration file (see "Configuring PXE" on page 1-11).

You may choose which kernel to boot by default by changing:

```
DEFAULT redhawk-trace
```

to be one of:

```
DEFAULT redhawk
DEFAULT redhawk-debug
```

Cluster Maintenance

Successful long-term deployment of a cluster requires the ability to maintain cluster file system images. The following sections describe how to perform the following tasks:

- adding nodes to a cluster
- re-installing the hard disk on disk-based nodes
- recreating a cluster file system image
- updating software and/or configuration files on a cluster file system image

Adding Nodes to a Cluster

Nodes can be added to a cluster at any time. To add a node, follow the directions beginning with "Configuring Network Services" on page 1-10.

Reinstalling Disk-based Nodes

Disk-based nodes may be reinstalled at any time. To reinstall a disk-based node, simply repeat the procedure detailed in "Installing Disk-Based Nodes" on page 1-17. Be sure to remove any symbolic links in the **`/tftpboot/yaci/pxelinux.cfg`** directory that are directing the node to boot from the local hard disk. For example, your **`pxelinux.cfg`** directory may look like this:

```
default -> redhawk.install
C0A80101 -> redhawk.local
C0A80102 -> redhawk.local
C0A80103 -> redhawk.local
```

To reinstall the C0A80101 node, simply remove the C0A80101 symbolic link. On the next reboot, that node will reinstall its local hard disk.

Recreating a Cluster File System Image

A new cluster file system image may be created from scratch at any time. Note that disk-based nodes will have to be reinstalled and diskless nodes will have to be rebooted in order to use the new image.

NOTE

All diskless nodes using a cluster file system image must be shutdown prior to creating the new image.

To create a new cluster file system image, simply repeat the procedure detailed in "Creating a Cluster File System Image" on page 1-5.

Updating a Cluster File System Image

It is possible to modify a cluster file system image once it is created. Note that disk-based nodes will have to be re-installed and diskless nodes will have to be rebooted in order to use the modified image.

To modify a cluster file system image, perform the following steps:

1. Modify files in the cluster file system image directory (see "Customizing the Cluster File System" on page 1-7).
2. Create a new tar image for disk-based nodes (see "Building a Compressed Tar File for Disk-Based Nodes" on page 1-10).
3. Create a new ramdisk image for diskless nodes (see "Building a Ramdisk Image for Diskless Nodes" on page 1-10).
4. Reconfigure PXE on the master so that disk-based nodes will be reinstalled on next boot (see "Configuring PXE" on page 1-11).
5. Reboot all nodes.

2
Grid Engine Software

2
Grid Engine Software

This chapter describes Grid Engine, the software that manages and schedules jobs across the cluster, and provides instructions for installing the product and configuring your cluster.

Overview

RedHawk Cluster Manager includes Grid Engine, an open source batch-queuing system developed by Sun Microsystems that accepts, schedules, dispatches and manages the remote execution of large numbers of user jobs. Grid Engine, also referred to as SGE, integrates multiple clusters into a grid environment.

A *grid* is a collection of computing resources that perform tasks. It can provide a single point of access to a large powerful distributed resource, or it may provide many access points, but to the user, it appears as a single computational resource.

Grid Engine does the following:

- Accepts jobs from the outside world. Jobs are users' requests for computer resources.
- Puts jobs in a holding area until the jobs can be run.
- Sends jobs from the holding area to an execution device.
- Manages running jobs.
- Logs the record of job execution when the jobs are finished.

The administrator configures the grid with customized resource management policies that schedule the jobs to be run on appropriate systems in the grid. Users can submit millions of jobs at a time without being concerned about where the jobs run. Jobs are submitted using Grid Engine commands or the QMON graphical user interface.

This chapter provides the information needed to construct a grid cluster of iHawk systems running RedHawk Linux and Cluster Manager that is appropriate to your needs. It does not include the specifics of using Grid Engine to administer the grid or submit jobs. For those details, refer to the resources provided in the section "Grid Engine Documentation" below.

Grid Engine Documentation

This section provides documentation and other resources you will need to administer and use Grid Engine on your cluster.

Manuals

The following manuals, produced by Sun Microsystems, are applicable to the Grid Engine open source software package included with Cluster Manager:

Manual Name	Publication Number
N1 Grid Engine 6 Administration Guide	817-5677-20
N1 Grid Engine 6 Release Notes	817-5678-20
N1 Grid Engine 6 User's Guide	817-6117-20
N1 Grid Engine 6 Installation Guide	817-6118-20

These document PDFs are supplied with RedHawk Cluster Manager. They can also be viewed or downloaded from the Internet at:
http://gridengine.sunsource.net/documentation.html

Refer to the section "Functionality Differences" below for differences between the revision of Grid Engine included in Cluster Manager and the documentation.

Web Sites

Additional information in the form of HowTo's can be found at:
http://gridengine.sunsource.net/project/gridengine/howto/howto.html

The following web sites contain additional information that may be helpful:

http://gridengine.sunsource.net/	Grid Engine Project Home
http://gridengine.info/	Tracking Grid Engine news, bugs, howtos and best practices

Man Pages

All Grid Engine man pages are available online. To view a man page, type: **`man`** *command.*

A summary of the man pages is provided in Table 2-1.

Table 2-1 Grid Engine Man Page Summary

Command Name	Description
Grid Engine User Commands (/usr/man/man1)	
gethostbyaddr	get hostname via IP address
gethostbyname	get local host information for specified hostname
gethostname	get local hostname
getservbyname	get configured port number of service
qacct	report and account for Grid Engine usage
qsub	submit a batch job to Grid Engine
qsh	submit an interactive X-windows session to Grid Engine
qlogin	submit an interactive login session to Grid Engine
qrsh	submit an interactive rsh session to Grid Engine
qalter	modify a pending batch job of Grid Engine
qresub	submit a copy of an existing Grid Engine job
qconf	Grid Engine Queue Configuration
qdel	delete Grid Engine jobs from queues
qhold	hold back Grid Engine jobs from execution
qhost	show the status of Grid Engine hosts, queues, jobs
qmake	distributed parallel make, scheduling by Grid Engine
qmod	modify a Grid Engine queue
qmon	X-Windows OSF/Motif graphical user interface for Grid Engine
qping	check application status of Grid Engine daemons
qrls	release Grid Engine jobs from previous hold states
qselect	used to modify queue attributes on a set of queues
qstat	show the status of Grid Engine jobs and queues
qtcsh	tcsh v6.09 with transparent remote execution by use of qrsh
sge_ckpt	Grid Engine checkpointing mechanism and checkpointing support
sge_intro	a facility for executing UNIX jobs on remote machines
sgepasswd	Modify the Grid Engine password file of Grid Engine
sge_types	Grid Engine type descriptions
submit	describes Grid Engine User Commands

Grid Engine File Formats (/usr/man/man5)	
access_list	Grid Engine access list file format
accounting	Grid Engine accounting file format
bootstrap	Grid Engine bootstrap file
calendar_conf	Grid Engine calendar configuration file format
checkpoint	Grid Engine checkpointing environment configuration file format
complex	Grid Engine complexes configuration file format
host_aliases	Grid Engine host aliases file format
host_conf	Grid Engine execution host configuration file format
hostgroup	host group entry file format
project	Grid Engine project entry file format
qtask	file format of the qtask file
queue_conf	Grid Engine queue configuration file format
reporting	Grid Engine reporting file format
sched_conf	Grid Engine default scheduler configuration file
sge_aliases	Grid Engine path aliases file format
sge_conf	Grid Engine configuration files
sgepasswd	Modify the Grid Engine password file of Grid Engine
sge_pe	Grid Engine parallel environment configuration file format
sge_priority	Grid Engine job priorities
sge_qstat	Grid Engine default qstat file format
sge_request	Grid Engine default request definition file format
share_tree	Grid Engine share tree file format
user	Grid Engine user entry file format
usermapping	user mapping entry file format
Grid Engine Administrative Commands (/usr/man/man8)	
sge_execd	Grid Engine job execution agent
sge_qmaster	Grid Engine master control daemon
sge_schedd	Grid Engine job scheduling agent
sge_shadowd	Grid Engine shadow master daemon
sge_shepherd	Grid Engine single job controlling agent

Functionality Differences

The N1 Grid Engine documentation by Sun Microsystems is used by the open source Grid Engine project as applicable documentation. Concurrent's distribution of Grid Engine is the default build of Grid Engine version 6 update 8 as documented in those documents with the following differences:

- The Windows operating system is not supported.
- Paralleled Environments (PE) MPI and PVM are not supported.
- Java language bindings are not included.
- DRMAA support is not included.

Configuring Your Cluster

Overview

It is suggested that the person responsible for installing Grid Engine and setting up the cluster have the N1 Grid Engine 6 documents available and that they familiarize themselves with the Grid Engine cluster architecture. For the example presented in this document, however, it should not be necessary to devote a great deal of study to the N1 Grid Engine 6 documents before attempting to set up the basic cluster, which is defined to be a single "master host" and one or more "execution hosts".

The ccur-sge binary rpm installation makes it easy to quickly configure any number of cluster nodes with minimal effort. When the rpm is installed on a system, that system becomes capable of assuming any role in the cluster. It is only a matter of configuring individual nodes to assume the role(s) they are assigned by running a handful of configuration scripts and by making some common configuration files accessible to the appropriate group of nodes.

It is possible to assign any number of roles to a given node. It is generally true that the master host should be dedicated to the job of being a master host. The master host is the brains of the cluster and should be left to the complex task of coordinating the efforts of the execution hosts.

The ccur-sge rpm also installs BerkeleyDB 4.4.20, which is used by Grid Engine for spooling.

Users are expected to make their own decisions on how best to configure their cluster based on individual needs. The examples provided here are designed to be simple and do not necessarily represent an ideal configuration.

Grid Engine is a complex application and can be configured in many different ways. An in depth study of the N1 Grid Engine 6 documentation will be necessary in order to fully optimize a cluster.

Grid Engine File System Requirements

Disk-based nodes should have the following minimums:

- Master host: 100 MB memory, 500 MB disk space
- Execution host: 20 MB memory, 50 MB disk space
- File server: 20 MB disk space + 20 MB per architecture

Diskless nodes should have the following minimums:

- Master host: 1 GB RAM; 2 GB is recommended.
- Execution host: 512 MB RAM; 1 GB is recommended.

Procedures

Procedures for configuring Grid Engine are given in this section.

Before starting this procedure, note the following:

- This is a simple procedure using 1 master host (**master**) and 3 execution hosts (**node1**, **node2** and **node3**).
- The services below are added to **/etc/services** by the ccur-sge rpm installation and setup questions about them during the process of configuring your cluster can be ignored:

  $SGE_QMASTER_PORT (if you haven't added the service >sge_qmaster<)
  $SGE_EXECD_PORT (if you haven't added the service >sge_execd<)

- You may also ignore suggestions to run any **qconf** commands during the installation process.
- The rpm installs the 'sgeadmin' user and the 'sge' group. Any member of the sge group and the sgeadmin need to have the SGE_ROOT environment variable included in their path, as well as the path to the grid engine interface "/usr/local/sge/ bin/$ARCH".

  To set SGE_ROOT:

  ```
  # SGE_ROOT=/usr/local/sge
  # export SGE_ROOT
  ```

  Use the **arch** command to determine which architecture your system has, then set the path. Below are examples for setting the appropriate path for both 32-bit and 64-bit systems, respectively:

  ```
  # arch
  i686
  # PATH=$PATH:$SGE_ROOT/bin/lx26-x86
  # export PATH

  # arch
  x86_64
  # PATH=$PATH:$SGE_ROOT/bin/lx26-amd64
  # export PATH
  ```

- For details beyond the scope of this quick setup example, refer to the *N1 Grid Engine 6 Installation Guide*.

Follow the steps below to configure Grid Engine.

1. Create the **`/etc/hosts`** entries for this cluster.

   Below is the example **`/etc/hosts`** for this cluster:

```
[root@master sge]# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1            localhost.localdomain localhost

192.168.1.15         master             # master host
192.168.1.1          node1              # execution host
192.168.1.2          node2              # execution host
192.168.1.3          node3              # execution host
```

2. Copy the **`/etc/hosts`** files to each node in the cluster.

```
[root@master sge]# scp /etc/hosts root@node1:/etc/hosts
hosts
100% 306 2.8MB/s 00:00
[root@master sge]# scp /etc/hosts root@node2:/etc/hosts
hosts
100% 306 2.8MB/s 00:00
[root@master sge]# scp /etc/hosts root@node3:/etc/hosts
hosts
100% 306 2.4MB/s 00:00
```

3. Run the master host installation script.

   a. Log in as root on the system selected to be the master host for this cluster and change into the SGE_ROOT directory **`/usr/local/sge`**.

   b. Run the **`install_qmaster`** script.

```
[root@master sge]# ./install_qmaster
```

   Below is the sequence of screens presented by this script.

   NOTE: We will be taking all the default choices except for the prompt which calls for the “shadow host” configuration. We will say “no” at this point.

```
Welcome to the Grid Engine installation
---------------------------------------
Grid Engine qmaster host installation
-------------------------------------

Before you continue with the installation please read these hints:

   -  Your terminal window should have a size of at least 80x24
      characters

   -  The INTR character is often bound to the key Ctrl-C.
      The term >Ctrl-C< is used during the installation if you
      have the possibility to abort the installation

The qmaster installation procedure will take approximately 5-10 minutes.

Hit <RETURN> to continue >>
```

```
 Grid Engine admin user account
 ------------------------------
 The current directory

    /usr/local/sge

 is owned by user

    sgeadmin

 If user >root< does not have write permissions in this directory on
 *all* of the machines where Grid Engine will be installed (NFS
 partitions not exported for user >root< with read/write permissions) it
 is recommended to install Grid Engine that all spool files will be
 created under the user id of user >sgeadmin<.

 IMPORTANT NOTE: The daemons still have to be started by user >root<.

 Do you want to install Grid Engine as admin user >sgeadmin< (y/n) [y]>>

 Installing Grid Engine as admin user >sgeadmin<
 Hit <RETURN> to continue >>
```

```
Checking $SGE_ROOT directory
----------------------------
The Grid Engine root directory is not set!
Please enter a correct path for SGE_ROOT.

If this directory is not correct (e.g. it may contain an automounter
prefix) enter the correct path to this directory or hit <RETURN> to use
default [/usr/local/sge] >>
Your $SGE_ROOT directory: /usr/local/sge

Hit <RETURN> to continue >>
```

```
Grid Engine TCP/IP service >sge_qmaster<
----------------------------------------
Using the service

   sge_qmaster

for communication with Grid Engine.

Hit <RETURN> to continue >>
```

```
Grid Engine TCP/IP service >sge_execd<
--------------------------------------
Using the service

   sge_execd

for communication with Grid Engine.

Hit <RETURN> to continue >>
```

```
Grid Engine cells
---------------

Grid Engine supports multiple cells.

If you are not planning to run multiple Grid Engine clusters or if you
don't know yet what is a Grid Engine cell it is safe to keep the default
cell name

   default

If you want to install multiple cells you can enter a cell name now.

The environment variable

   $SGE_CELL=<your_cell_name>

will be set for all further Grid Engine commands.

Enter cell name [default] >>

Using cell >default<.
Hit <RETURN> to continue >>
```

NOTE: If you choose to change your cell name, replace ‘default’ with the new cell name in all future references.

```
Grid Engine qmaster spool directory
-----------------------------------

The qmaster spool directory is the place where the qmaster daemon stores
the configuration and the state of the queuing system.

The admin user >sgeadmin< must have read/write access
to the qmaster spool directory.

If you will install shadow master hosts or if you want to be able to start
the qmaster daemon on other hosts (see the corresponding sectionin the
Grid Engine Installation and Administration Manual for details) the
account on the shadow master hosts also needs read/write access to this
directory.

The following directory

[/usr/local/sge/default/spool/qmaster]

will be used as qmaster spool directory by default!

Do you want to select another qmaster spool directory (y/n) [n] >>
```

```
Windows Execution Host Support
------------------------------

Are you going to install Windows Execution Hosts? (y/n) [n] >>
```

```
Verifying and setting file permissions
--------------------------------------

Did you install this version with >pkgadd< or did you already verify and
set the file permissions of your distribution >>

We do not verify file permissions. Hit <RETURN> to continue >>
```

```
Select default Grid Engine hostname resolving method
----------------------------------------------------

Are all hosts of your cluster in one DNS domain? If this is the case the
hostnames

   >hostA< and >hostA.foo.com<

would be treated as equal, because the DNS domain name >foo.com< is ignored
when comparing hostnames.

Are all hosts of your cluster in a single DNS domain (y/n) [y] >>

Ignoring domainname when comparing hostnames.

Hit <RETURN> to continue >>
```

```
Making directories
------------------

creating directory: default
creating directory: default/common
creating directory: /usr/local/sge/default/spool/qmaster
creating directory: /usr/local/sge/default/spool/qmaster/job_scripts
Hit <RETURN> to continue >>
```

```
Setup spooling
--------------
Your SGE binaries are compiled to link the spooling libraries
during runtime (dynamically). So you can choose between Berkeley DB
spooling and Classic spooling method.
Please choose a spooling method (berkeleydb|classic) [berkeleydb] >>
```

```
The Berkeley DB spooling method provides two configurations!

Local spooling:
The Berkeley DB spools into a local directory on this host (qmaster host)
This setup is faster, but you can't setup a shadow master host

Berkeley DB Spooling Server:
If you want to setup a shadow master host, you need to use Berkeley DB
Spooling Server!
In this case you have to choose a host with a configured RPC service. The
qmaster host connects via RPC to the Berkeley DB. This setup is more
failsafe, but results in a clear potential security hole. RPC communication
(as used by Berkeley DB) can be easily compromised. Please only use this
alternative if your site is secure or if you are not concerned about
security. Check the installation guide for further advice on how to achieve
failsafety without compromising security.

Do you want to use a Berkeley DB Spooling Server? (y/n) [n] >>

Hit <RETURN> to continue >>
```

```
Berkeley Database spooling parameters
-------------------------------------

Please enter the Database Directory now, even if you want to spool locally,
it is necessary to enter this Database Directory.

Default: [/usr/local/sge/default/spool/spooldb] >>

creating directory: /usr/local/sge/default/spool/spooldb
Dumping bootstrapping information
Initializing spooling database

Hit <RETURN> to continue >>
```

```
Grid Engine group id range
--------------------------

When jobs are started under the control of Grid Engine an additional group
id is set on platforms which do not support jobs. This is done to provide
maximum control for Grid Engine jobs.

This additional UNIX group id range must be unused group id's in your
system. Each job will be assigned a unique id during the time it is
running. Therefore you need to provide a range of id's which will be
assigned dynamically for jobs.

The range must be big enough to provide enough numbers for the maximum
number of Grid Engine jobs running at a single moment on a single host.
E.g. a range like >20000-20100< means, that Grid Engine will use the group
ids from 20000-20100 and provides a range for 100 Grid Engine jobs at the
same time on a single host.

You can change at any time the group id range in your cluster
configuration.

Please enter a range >> 20000-20100

Using >20000-20100< as gid range. Hit <RETURN> to continue >>
```

```
Grid Engine cluster configuration
---------------------------------

Please give the basic configuration parameters of your Grid Engine
installation:

   <execd_spool_dir>

The pathname of the spool directory of the execution hosts. User >sgeadmin<
must have the right to create this directory and to write into it.

Default: [/usr/local/sge/default/spool] >>
```

```
Grid Engine cluster configuration (continued)
---------------------------------------------

<administrator_mail>

The email address of the administrator to whom problem reports are sent.

It is recommended to configure this parameter. You may use >none< if you do
not wish to receive administrator mail.

Please enter an email address in the form >user@foo.com<.

Default: [none] >>
```

```
The following parameters for the cluster configuration were configured:

   execd_spool_dir /usr/local/sge/default/spool
   administrator_mail none

Do you want to change the configuration parameters (y/n) [n] >>
```

```
Creating local configuration
----------------------------
Creating >act_qmaster< file
Adding default complex attributes
Reading in complex attributes.
Adding default parallel environments (PE)
Reading in parallel environments:
   PE "make".
   PE "make.sge_pqs_api"
Adding SGE default usersets
Reading in usersets:
   Userset "deadlineusers".
   Userset "defaultdepartment".
Adding >sge_aliases< path aliases file
Adding >qtask< qtcsh sample default request file
Adding >sge_request< default submit options file
Creating >sgemaster< script
Creating >sgeexecd< script
Creating settings files for >.profile/.cshrc<


Hit <RETURN> to continue >>
```

```
qmaster/scheduler startup script
--------------------------------

We can install the startup script that will
start qmaster/scheduler at machine boot (y/n) [y] >>

cp /usr/local/sge/default/common/sgemaster /etc/init.d/sgemaster
/usr/lib/lsb/install_initd /etc/init.d/sgemaster

Hit <RETURN> to continue >>
```

```
Grid Engine qmaster and scheduler startup
-----------------------------------------

Starting qmaster and scheduler daemon. Please wait ...
   starting sge_qmaster
   starting sge_schedd
Hit <RETURN> to continue >>
```

```
Adding Grid Engine hosts
------------------------

Please now add the list of hosts, where you will later install your
execution daemons. These hosts will be also added as valid submit hosts.

Please enter a blank separated list of your execution hosts. You may press
<RETURN> if the line is getting too long. Once you are finished simply
press <RETURN> without entering a name.

You also may prepare a file with the hostnames of the machines where you
plan to install Grid Engine. This may be convenient if you are installing
Grid Engine on many hosts.

Do you want to use a file which contains the list of hosts (y/n) [n] >>
```

```
Adding admin and submit hosts
-----------------------------

Please enter a blank seperated list of hosts.

Stop by entering <RETURN>. You may repeat this step until you are entering
an empty list. You will see messages from Grid Engine when the hosts are
added.

Host(s): node1
node1 added to administrative host list
node1 added to submit host list
Hit <RETURN> to continue >>
```

```
Adding admin and submit hosts
-----------------------------

Please enter a blank seperated list of hosts.

Stop by entering <RETURN>. You may repeat this step until you are entering
an empty list. You will see messages from Grid Engine when the hosts are
added.

Host(s): node2
node2 added to administrative host list
node2 added to submit host list
Hit <RETURN> to continue >>
```

```
Adding admin and submit hosts
-----------------------------

Please enter a blank seperated list of hosts.

Stop by entering <RETURN>. You may repeat this step until you are entering
an empty list. You will see messages from Grid Engine when the hosts are
added.

Host(s): node3
node3 added to administrative host list
node3 added to submit host list
Hit <RETURN> to continue >>
```

```
Adding admin and submit hosts
-----------------------------

Please enter a blank seperated list of hosts.

Stop by entering <RETURN>. You may repeat this step until you are entering
an empty list. You will see messages from Grid Engine when the hosts are
added.

Host(s):
Finished adding hosts. Hit <RETURN> to continue >>
```

```
If you want to use a shadow host, it is recommended to add this host to the
list of administrative hosts.

If you are not sure, it is also possible to add or remove hosts after the
installation with <qconf -ah hostname> for adding and <qconf -dh hostname>
for removing this host

Attention: This is not the shadow host installationprocedure. You still
have to install the shadow host separately

Do you want to add your shadow host(s) now? (y/n) [y] >> n
```

```
Creating the default <all.q> queue and <allhosts> hostgroup
-----------------------------------------------------------

root@master added "@allhosts" to host group list
root@master added "all.q" to cluster queue list

Hit <RETURN> to continue >>
```

```
Scheduler Tuning
----------------

The details on the different options are described in the manual.

Configurations
--------------
1) Normal

   Fixed interval scheduling, report scheduling information,
   actual + assumed load

2) High

   Fixed interval scheduling, report limited scheduling
   information, actual load

3) Max

   Immediate Scheduling, report no scheduling information,
   actual load

Enter the number of your prefered configuration and hit <RETURN>!
Default configuration is [1] >>

We're configuring the scheduler with >Normal< settings!
Do you agree? (y/n) [y] >>
```

```
Using Grid Engine
-----------------
You should now enter the command:

   source /usr/local/sge/default/common/settings.csh

if you are a csh/tcsh user or

   # . /usr/local/sge/default/common/settings.sh

if you are a sh/ksh user.

This will set or expand the following environment variables:

   - $SGE_ROOT             (always necessary)
   - $SGE_CELL             (if you are using a cell other than >default<)
   - $SGE_QMASTER_PORT     (if you haven't added the service >sge_qmaster<)
   - $SGE_EXECD_PORT       (if you haven't added the service >sge_execd<)
   - $PATH/$path           (to find the Grid Engine binaries)
   - $MANPATH              (to access the manual pages)

Hit <RETURN> to see where Grid Engine logs messages >>
```

NOTE: If you are a **`bash`** user, enter the command line for sh/ksh users.

```
Grid Engine messages
--------------------

Grid Engine messages can be found at:

   /tmp/qmaster_messages (during qmaster startup)
   /tmp/execd_messages (during execution daemon startup)

After startup the daemons log their messages in their spool directories.

   Qmaster: /usr/local/sge/default/spool/qmaster/messages
   Exec daemon: <execd_spool_dir>/<hostname>/messages

Grid Engine startup scripts
---------------------------

Grid Engine startup scripts can be found at:

   /usr/local/sge/default/common/sgemaster (qmaster and scheduler)
   /usr/local/sge/default/common/sgeexecd (execd)

Do you want to see previous screen about using Grid Engine again (y/n) [n]
```

```
Your Grid Engine qmaster installation is now completed
------------------------------------------------------

Please now login to all hosts where you want to run an execution daemon and
start the execution host installation procedure.

If you want to run an execution daemon on this host, please do not forget
to make the execution host installation in this host as well.

All execution hosts must be administrative hosts during the installation.
All hosts which you added to the list of administrative hosts during this
installation procedure can now be installed.

You may verify your administrative hosts with the command

   # qconf -sh

and you may add new administrative hosts with the command

   # qconf -ah <hostname>

Please hit <RETURN> >>
```

4. After the **`install_qmaster`** script completes.

   You now should have the "default" directory created in the SGE_ROOT. Below this directory are the common files needed by the cluster plus the spooling area for the "master host."

   Below shows the "default" directory structure for this cluster. All references to the "default" cell name should be changed to *your_cell_name* where *your_cell_name* is the name of the cell that you chose in a previous installation step on page 2-9.

```
[root@master sge]# ls -R default
default:
common spool

default/common:
act_qmaster bootstrap qtask settings.csh settings.sh sge_aliases
sgeexecd sgemaster sge_request

default/spool:
qmaster spooldb

default/spool/qmaster:
heartbeat job_scripts jobseqnum messages qmaster.pid schedd

default/spool/qmaster/job_scripts:

default/spool/qmaster/schedd:
messages schedd.pid

default/spool/spooldb:
__db.001 __db.002 __db.003 __db.004 __db.005 __db.006 log.0000000001
sge sge_job
```

5. Setting up the environment variables for Grid Engine.

   In order for Grid Engine to operate, it looks for some important environment variables in the shell from which any Grid Engine command is executed.

   The scripts **`settings.sh`** and **`settings.csh`** will be located under **`SGE_ROOT/default/common/`**.

   If you followed the above procedure for the **`install_qmaster`** script, the important environment variables below will be exported when you execute the script **`settings.sh`** (or **`settings.csh`** if you are a C shell user).

   Example:

```
[root@master sge]# . default/common/settings.sh
```

   You should put this in the login profile for any member of the sge group or the sgeadmin on any submit host.

```
SGE_CELL=default
LD_LIBRARY_PATH=/usr/local/sge/lib/lx26-x86
PATH=/usr/local/sge/bin/lx26-x86:/usr/kerberos/sbin:/usr/kerberos/bin:/
bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:/usr/bin/X1
1:/usr/X11R6/bin:/root/bin
SGE_ROOT=/usr/local/sge
```

You should also add the following LANG=C to the environment to prevent an error when starting QMON. For example:

```
[sgeadmin@master sgeadmin]$ cat .bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
     .~/.bashrc
fi

. /usr/local/sge/default/common/settings.sh

LANG=C
export LANG

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH
unset USERNAME
```

6. Copy common files to each execution host in the cluster.

   Now that you have set up your master host, you will need to copy the common configuration files for the “default” cluster to each execution host. The example below shows a convenient method (only one node shown).

   NOTE: The default sgeadmin password is “sgeadmin”. It is recommended that you use a more secure password.

   The directory tree under **SGE_ROOT/default/common** on each node in this cluster MUST be the same.

   Change into SGE_ROOT on the master host and execute the following set of commands:

```
[root@master sge]# ssh sgeadmin@node1 mkdir -p /usr/local/sge/default
sgeadmin@node1's password:
/usr/X11R6/bin/xauth: creating new authority file
/home/sgeadmin/.Xauthority

[root@master sge]# scp -r default/common sgeadmin@node1:
/usr/local/sge/default
sgeadmin@node1's password:
bootstrap       100%        372        4.1MB/s      00:00
act_qmaster     100%          8       98.8KB/s      00:00
sge_aliases     100%       1608       19.6MB/s      00:00
qtask           100%       1994        4.9MB/s      00:00
sge_request     100%       2164        5.1MB/s      00:00
sgemaster       100%       13KB       17.2MB/s      00:00
sgeexecd        100%       8199       42.7MB/s      00:00
settings.csh    100%        725        7.6MB/s      00:00
settings.sh     100%        676        9.2MB/s      00:00
```

   Note that these numbers are for example only and may not match your installation.

7. Installing the execution hosts.

   You must log into each execution host and run the **`install_execd`** script under the SGE_ROOT.

   In the case where there are a large number of nodes, automation procedures contained in the accompanying N1 Grid Engine documentation can be followed; however, it may be easier to do each node manually than to set up the automation. After installing the first execution host, it takes about 5 seconds to run through the rest.

   Below is the sequence of screen shots when running **`install_execd`** on the execution host named "node1."

   NOTE: Again, we're going to take all the default answers in this example.

   We will choose **`/var/spool/sge`** as the local spooling directory when prompted by the installation script. At completion, local spooling will take place in **`/var/spool/sge/$HOSTNAME`**.

   a. Log in as root on each execution host to be configured and change into the SGE_ROOT.

   b. Execute **`install_execd`** as shown below:

      ```
      [root@node1 sge]# ./install_execd
      ```

```
Welcome to the Grid Engine execution host installation
------------------------------------------------------

If you haven't installed the Grid Engine qmaster host yet, you must
execute this step (with >install_qmaster<) prior the execution host
installation.

For a sucessfull installation you need a running Grid Engine qmaster. It
is also neccesary that this host is an administrative host.

You can verify your current list of administrative hosts with the command:

   # qconf -sh

You can add an administrative host with the command:

   # qconf -ah <hostname>

The execution host installation will take approximately 5 minutes.

Hit <RETURN> to continue >>
```

```
Checking $SGE_ROOT directory
----------------------------

The Grid Engine root directory is not set!
Please enter a correct path for SGE_ROOT.

If this directory is not correct (e.g. it may contain an automounter
prefix) enter the correct path to this directory or hit <RETURN> to use
default [/usr/local/sge] >>
Your $SGE_ROOT directory: /usr/local/sge

Hit <RETURN> to continue >>
```

```
Grid Engine cells
-----------------

Please enter cell name which you used for the qmaster
installation or press <RETURN> to use [default] >>

Using cell: >default<

Hit <RETURN> to continue >>
```

NOTE: If you changed your cell name, replace ‘default’ with the new cell name in all future references.

```
Checking hostname resolving
---------------------------

This hostname is known at qmaster as an administrative host.

Hit <RETURN> to continue >>
```

```
Local execd spool directory configuration
-----------------------------------------

During the qmaster installation you've already entered a global execd spool
directory. This is used, if no local spool directory is configured.

Now you can enter a local spool directory for this host.

Do you want to configure a local spool directory for this host (y/n) [n] >>
y

Please enter the local spool directory now! >> /var/spool/sge
Using local execd spool directory [/var/spool/sge]
Hit <RETURN> to continue >>
```

```
Creating local configuration
----------------------------
sgeadmin@node1 added "node1" to configuration list

Local configuration for host >node1< created.

Hit <RETURN> to continue >>
```

```
execd startup script
--------------------

We can install the startup script that will start execd at machine boot
(y/n) [y] >>

cp /usr/local/sge/default/common/sgeexecd /etc/init.d/sgeexecd
/usr/lib/lsb/install_initd /etc/init.d/sgeexecd

Hit <RETURN> to continue >>
```

```
Grid Engine execution daemon startup
------------------------------------

Starting execution daemon. Please wait ...
   starting sge_execd

Hit <RETURN> to continue >>
```

```
Adding a queue for this host
----------------------------

We can now add a queue instance for this host:

   -  it is added to the >allhosts< hostgroup
   -  the queue provides 4 slot(s) for jobs in all queues
      referencing the >allhosts< hostgroup

You do not need to add this host now, but before running jobs on this
host it must be added to at least one queue.

Do you want to add a default queue instance for this host (y/n) [y] >>

root@node1 modified "@allhosts" in host group list
root@node1 modified "all.q" in cluster queue list

Hit <RETURN> to continue >>
```

```
Using Grid Engine
-----------------

You should now enter the command:

   source /usr/local/sge/default/common/settings.csh

if you are a csh/tcsh user or

   # . /usr/local/sge/default/common/settings.sh

if you are a sh/ksh user.

This will set or expand the following environment variables:

 - $SGE_ROOT              (always necessary)
 - $SGE_CELL              (if you are using a cell other than >default<)
 - $SGE_QMASTER_PORT      (if you haven't added the service >sge_qmaster<)
 - $SGE_EXECD_PORT        (if you haven't added the service >sge_execd<)
 - $PATH/$path            (to find the Grid Engine binaries)
 - $MANPATH               (to access the manual pages)

Hit <RETURN> to see where Grid Engine logs messages >>
```

```
Grid Engine messages
--------------------

Grid Engine messages can be found at:

   /tmp/qmaster_messages  (during qmaster startup)
   /tmp/execd_messages    (during execution daemon startup)

After startup the daemons log their messages in their spool directories.

   Qmaster:               /usr/local/sge/default/spool/qmaster/messages
   Exec daemon:           <execd_spool_dir>/<hostname>/messages


Grid Engine startup scripts
---------------------------

Grid Engine startup scripts can be found at:

   /usr/local/sge/default/common/sgemaster (qmaster and scheduler)
   /usr/local/sge/default/common/sgeexecd (execd)

Do you want to see previous screen about using Grid Engine again (y/n) [n]
>>

Your execution daemon installation is now completed.
```

Testing the Configuration

Once you have configured the cluster with Grid Engine, you should test the configuration. This simple test illustrates basic job submission and how work is divided up on the cluster.

Test Program and Run Script

For this test, the program below is called **run.c**. It submits the same job 20 times.

```
#include <sys/types.h>
#include <unistd.h>

main(int argc, char* argv[])
{
  char name[20], str[100];
  int length;
  pid_t pid;

  gethostname(name, length);
  printf("\"%s\" with process-id %d executed by: %s\n", argv[0], getpid(),
    name) ;

}
```

To run the program, we use a simple one line job script called **a.sh** shown below. Run this as sgeadmin:

```
/home/sgeadmin/run >> out
```

Test Results

As shown in the results below, submitting the same job 20 times results in a total of twenty processes created across the node with each node doing a piece of the work.

In order for the job to be run, the program must be "visible" to each node. In this case, the specified program was copied to the same path on each node: **/home/sgeadmin**.

We see that the output file **out** contains 10 entries on the node named node1, 6 entries on the node named node2, and 4 entries on the node named node3.

```
[root@node1 sgeadmin]# cat out
"/home/sgeadmin/run" with process-id 5372 executed by: node1
"/home/sgeadmin/run" with process-id 5408 executed by: node1
"/home/sgeadmin/run" with process-id 5420 executed by: node1
"/home/sgeadmin/run" with process-id 5444 executed by: node1
"/home/sgeadmin/run" with process-id 5487 executed by: node1
"/home/sgeadmin/run" with process-id 5508 executed by: node1
"/home/sgeadmin/run" with process-id 5516 executed by: node1
"/home/sgeadmin/run" with process-id 5540 executed by: node1
"/home/sgeadmin/run" with process-id 5570 executed by: node1
"/home/sgeadmin/run" with process-id 5594 executed by: node1

[root@node2 sgeadmin]# cat out
"/home/sgeadmin/run" with process-id 4326 executed by: node2
"/home/sgeadmin/run" with process-id 4347 executed by: node2
"/home/sgeadmin/run" with process-id 4386 executed by: node2
"/home/sgeadmin/run" with process-id 4395 executed by: node2
"/home/sgeadmin/run" with process-id 4423 executed by: node2
```

```
"/home/sgeadmin/run" with process-id 4447 executed by: node2

[root@node3 sgeadmin]# cat out
"/home/sgeadmin/run" with process-id 3352 executed by: node3
"/home/sgeadmin/run" with process-id 3353 executed by: node3
"/home/sgeadmin/run" with process-id 3381 executed by: node3
"/home/sgeadmin/run" with process-id 3452 executed by: node3
```

If **/home/sgeadmin** had been commonly mounted via nfs, there would have been no need to copy the program to each node first, and the output would have been interspersed into a single file called **out** with twenty lines.

Using QMON

Use the following procedure to run the test using QMON.

1. Build the sample program **run.c** shown above and create the wrapper script **a.sh** as shown above. Copy them to the sgeadmin home on each node.

2. You must configure a “submit host.” In this case the master host “master” was also the “submit host.”

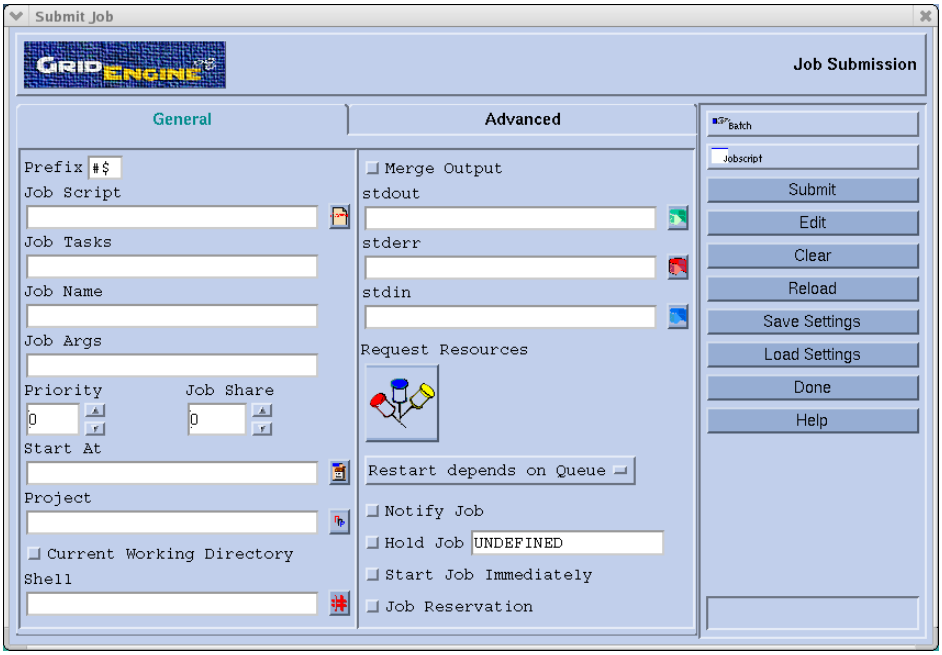
   Example:

   ```
   [sgeadmin@master sgeadmin]$ qconf -as master
   master added to submit host list
   ```

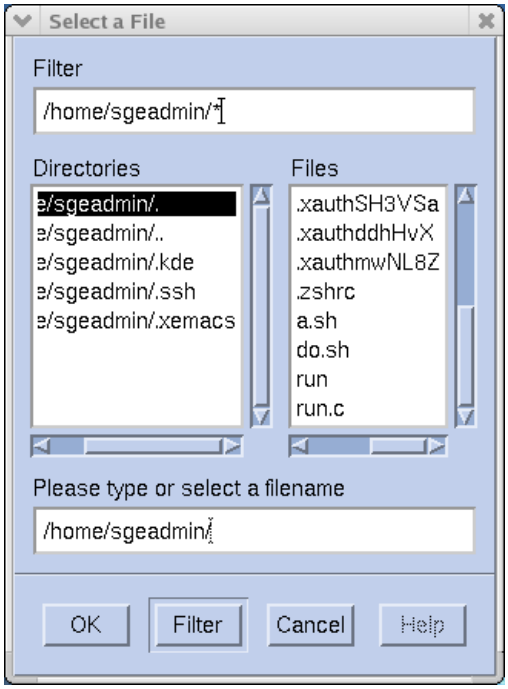
3. Start QMON on the submit host by executing the following command, which displays the QMON Main Control panel.

   ```
   qmon &
   ```

4. Click the Submit Jobs button on the QMON main control panel. Button labels display when the mouse pointer rolls over the button – Submit Jobs is the top row, 3rd from the left. The Job Submission GUI displays.



5. Click on the yellow button to the right of the Job Script field. This displays another GUI from which to select the job to be submitted.

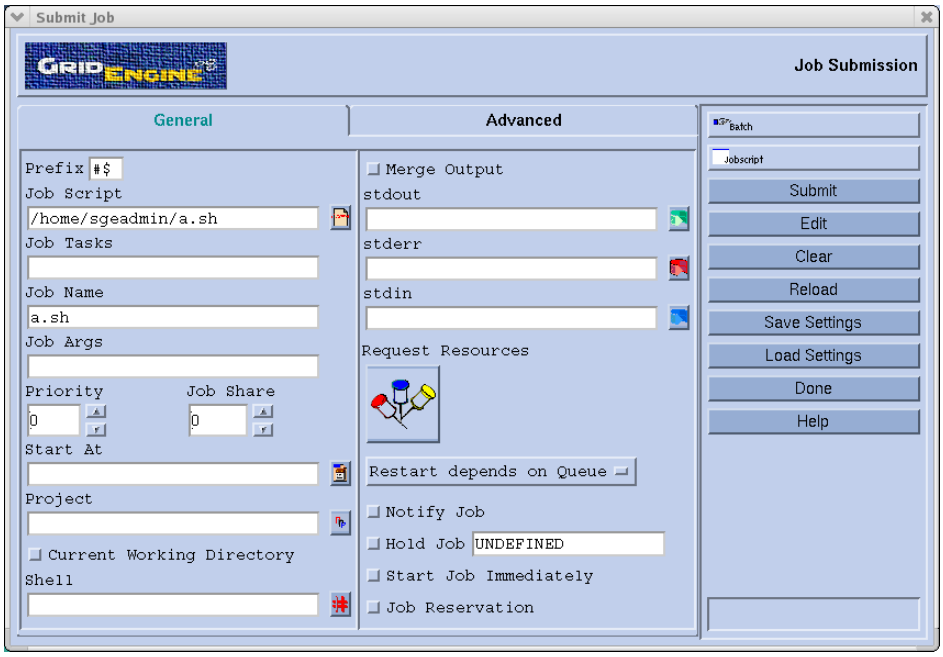


6. In the Filter field, enter the following:

   `/home/sgeadmin/*`

   In the Files field, select **`a.sh`** and click OK.

   The Job Submission GUI will now contain the job to submit.



7. Click the Submit button on the Job Submission GUI 20 times.

8. Select the Job Control button on the QMON main control panel to display the Job Control GUI.

You will be able to see the status of the jobs as they move from Pending Jobs to Running Jobs to Finished Jobs by selecting the appropriate tab at the top and periodically hitting the Refresh button. In this screen shot, all pending jobs have run and would be listed under Running Jobs or Finished Jobs.

The output should appear on each node in the file **`~sgeadmin/out`**.

You may wish to further experiment with QMON using the Grid Engine documentation as a guide.

A
Node Information Worksheet

Use this worksheet to record information about the master and nodes that will compose your cluster. Refer to the section “Configuring Network Services” in Chapter 1 for more information.

Master Host

Cluster subnet	. . .
Cluster netmask	. . .
Broadcast address	. . .
Routers	. . .
Domain name servers	. . .
Domain name	

Cluster Nodes

Hostname	MAC Address	IP Address	Node Type
	: : : : :	. . .	
	: : : : :	. . .	
	: : : : :	. . .	
	: : : : :	. . .	
	: : : : :	. . .	
	: : : : :	. . .	
	: : : : :	. . .	
	: : : : :	. . .	
	: : : : :	. . .	

Index