Sun N1 System Manager 1.3 Grid Engine Provisioning and Monitoring Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819–5138
April, 2006

060428@14774

Contents

Preface

The Sun N1 System Manager for N1 Grid Engine Provisioning and Monitoring Guide helps system administrators to understand and administer Sun N1™ Grid Engine (GE) using . This book provides detailed examples and procedures to explain how you can use the Sun N1 System Manager (N1SM) for Grid Engine module to manage provision and monitor grids built with N1 Grid Engine.

Who Should Use This Book

This guide is intended for system administrators who are responsible for managing servers running the Sun N1 System Manager software. These system administrators are expected to have the following background:

- Knowledge of the Solaris™, Red Hat Linux, and Microsoft Windows operating systems as well as the network administration tools provided by each operating system.
- Knowledge of using the N1 System Manager product.
- Knowledge of using N1 Grid Engine.

Before You Read This Book

Familiarity with the following documents will help you use this manual.

- *Sun N1 System Manager 1.3 Introduction*
- *Sun N1 System Manager 1.3 Installation and Configuration Guide*
- *Sun N1 System Manager 1.3 Operating System Provisioning Guide*
- *Sun N1 System Manager 1.3 Discovery and Administration Guide*

How This Book Is Organized

This book is divided into the following sections

Chapter Chapter 1, tells you how to launch the N1SM for Grid Engine module and what to do if the module is not enabled.

Chapter Chapter 2, describes how to use the N1SM Command Line Interface (CLI) to create N1 Grid Engine versions and installation templates as well as how to provision the N1 Grid Engine versions onto managed servers.

Chapter Chapter 3 explains how to provision servers and monitor grids when using a system external to the grid network.

Chapter Chapter 4, tells you how to use the N1SM GE Graphical User Interface (GUI) to monitor a grid with emphasis on getting a quick overview of grid performance.

Chapter Chapter 5, shows you how to use the different GUI Job views to analyze Job status, resource usage, and scheduling.

Chapter Chapter 6, describes how to analyze N1 Grid Engine Queue status and details.

Chapter Chapter 7, describes how to analyze N1 Grid Engine Host status and details.

Chapter Chapter 8, tells you how to find grid problems using the N1 Grid engine daemon logs, as well as Job, Queue, and Host Alerts.

Related Books

The following books are useful for installing and using the N1SM GE module. All theses documents are available from Sun's documentation web site, docs.sun.com (`http://docs.sun.com`).

- *Sun N1 System Manager 1.3 Command Line Reference Manual*
- *Sun N1 System Manager 1.3 Release Notes*
- *Sun N1 System Manager 1.3 Troubleshooting Guide*
- *Sun N1 System Manager 1.3 Site Preparation Guide*
- *N1 Grid Engine 6 Installation Guide*
- *N1 Grid Engine 6 User's Guide*
- *N1 Grid Engine 6 Administration Guide*
- *N1 Grid Engine Management Module User's Guide*

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.sun.com/training/`)

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX® system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

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

CHAPTER 1

Getting Started With N1 Grid Engine Provisioning and Monitoring

The N1 System Manager (N1SM) for N1 Grid Engine (N1GE) module enables you to both provision (install) N1 Grid Engine software onto servers managed by N1SM and to use N1SM to monitor the performance of the resulting grid. The module consists of two parts:

- N1GE Commands in N1SM CLI

  You do provisioning and software management tasks using the N1SM CLI with N1GE specific commands. These commands are documented in the Provisioning Grid Engine Onto Managed Servers chapter.

- N1GE Monitor GUI

  You use the N1GE Monitor GUI to do the N1 Grid Engine monitoring tasks. The Monitor GUI is described in the Monitoring N1 Grid Engine chapter.

Enabling the N1GE Module

While the N1GE module is included as a standard part of N1SM 1.3, the module is not enabled by default. Use the following steps to enable the module before you attempt to launch it. Otherwise, you receive an error message.

▼ To Enable the N1GE Module

Before You Begin

This procedure requires that you have root privileges on the N1SM management server. Be aware that there is a difference in the output format of the `ifconfig -a` depending on whether you are running your management station on a Linux or on a Solaris machine.

In either case, you need to pick the MAC address for the port that is associated with the hostname. In other words, use the IP address that is on the same line as the hostname in the `/etc/hosts`. For example, if the `/etc/hosts` file contains the line, *129.144.3.100 myhost*, and the hostname command displays *myhost*, you need to select the MAC address associated with the IP address *129.144.3.100*.

Note – In either situation, you must enter the MAC address in the case in which it appears in the command output (including lowercase or uppercase).

---

1 **Run an `ifconfig -a` command on the management server and find the correct MAC address.**

Linux Example:

```
[root@hdco09 lib]# ifconfig -a
eth0 Link encap:Ethernet HWaddr 00:09:3D:00:23:8D
inet addr:10.0.0.109 Bcast:10.0.0.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:19915156 errors:0 dropped:0 overruns:0 frame:0
TX packets:4652765 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:1492354783 (1423.2 Mb) TX bytes:947655171 (903.7 Mb)
Interrupt:25
eth1 Link encap:Ethernet HWaddr 00:09:3D:00:25:81
inet addr:172.20.48.109 Bcast:172.20.48.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:47450642 errors:0 dropped:0 overruns:0 frame:0
TX packets:5943396 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:3061524439 (2919.6 Mb) TX bytes:1133911299 (1081.3 Mb)
Interrupt:26
```

In this example, the *eth0* entry is the correct interface and **00:09:3D:00:23:8D** is the MAC address. This address will function as the license key.

Solaris Example:

```
# ifconfig -a
 lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
         inet 127.0.0.1 netmask ff000000
 bge0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
         inet 10.0.0.114 netmask ffffff00 broadcast 10.0.0.255
         ether 0:9:3d:0:66:8f
 bge1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
         inet 10.10.4.114 netmask ffffff00 broadcast 10.10.4.255
         ether 0:9:3d:0:66:90
```

In this example, the *bge1* entry is the correct interface and the corresponding MAC address for this entry is **0:9:3d:0:66:90**. This address will function as the license key.

2 **From the CLI on the management server, run a command similar to the following; substitute your MAC address for the one in the example.**

For Linux use:

```
n1-ok> set module n1ge enabled true licensekey 00:09:3D:00:23:8D
```

or for Solaris

```
n1-ok> set module n1ge enabled true licensekey 0:9:3d:0:66:90
```

3 **Use the following command to verify that the N1GE module is enabled**

```
n1-ok> show module all

Name                          Version     Installed      Enabled
Core                          1.0         true           true
Drivers                       1.0         true           false
n1ge                          1.0        true           true
-------------------------------------------------------------------
```

The N1GE module should be in the enabled state.

Accessing the N1SM CLI

Use the following steps to access the N1SM CLI.

▼ To Access the N1SM CLI

You access the N1SM CLI from either a terminal window on the management server or the CLI pane of the N1SM GUI. You can get the instructions on how to use the CLI from the "To Access the N1 System Manager Command Line" in *Sun N1 System Manager 1.3 Discovery and Administration Guide* section of the *Sun N1 System Manager 1.3 Discovery and Administration Guide*.

- **If you are using the browser interface, enter your commands in the CLI pane. If you are use a terminal window on the management server, as root, type:**

```
# n1sh
```

You then see the N1 command prompt:

```
N1-ok>
```

Accessing the N1GE Monitor GUI

This section describes how you access the N1GE Graphical User Interface (GUI).

▼ To Access the N1GE Monitor GUI

- **Click on the N1 System Manager for Grid Engine link on the N1SM launch page as shown in the following figure.**

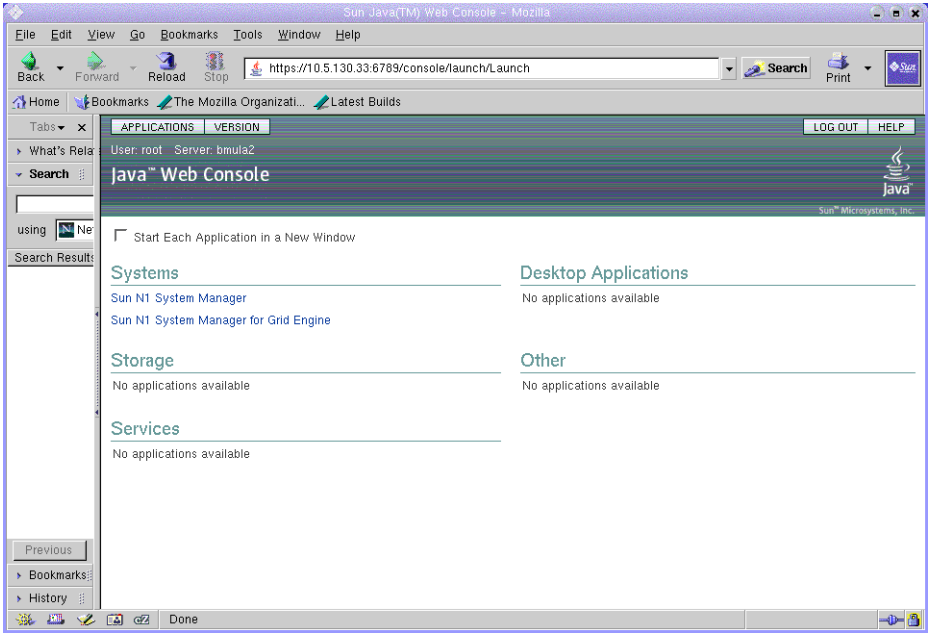


FIGURE 1–1 N1SM for Grid Engine Launch Page Link

If you receive an error message when you click on the Grid Engine link, this module has probably not been enabled. Use the instructions in the previous section to enable the N1 Grid Engine module.

CHAPTER 2

Provisioning N1 Grid Engine Onto Managed Servers

This chapter describes how you use the N1SM CLI commands to provision (install) and manage N1 Grid Engine . Using the N1SM CLI you can perform the following tasks:

- Create and manage N1GE versions
- Create and manage N1GE application profiles
- Create and manage N1GE settings
- Install and remove N1GE versions from managed servers

Most of the functionality provided by the Sun Control Station Grid Engine Management Module (GEMM) is replicated by N1SM CLI commands. These functions include:

- Creating an N1GE version and adding files to it: `create application` command
- Creating N1GE installation settings: `create applicationprofile` command
- Installing a master host: `load server` command
- Installing compute and submit hosts. `Load server`, `load group` commands
- Uninstalling N1GE from a master host. `unload server` command
- Uninstalling N1GE from compute and submit hosts. `unload server`, `unload group` commands
- View N1GE settings: `show application`, `show applicationprofile` commands
- Modifying N1GE settings: modify the application profile

Using N1SM to Install N1 Grid Engine

The general task flow to install N1GE onto a managed server is the following:

1. Use the or use some other media as the source to copy the N1GE software onto the N1SM management server as shown in the "To Download the N1 Grid Engine Software Onto a Management Server" on page 17 section.
2. Create an N1GE version using these files. See the "To Create an N1GE Version" on page 17 section.

3. Create an N1GE application profile associated with the version as shown in the "To Create an N1GE Application Profile" on page 19 section.
4. Load the application profile onto the managed servers while defining each server's N1GE role as shown in the "Installing N1 Grid Engine onto Servers" on page 25 section.

Creating and Managing N1 Grid Engine Versions

This section describes the various commands that you use to do the following tasks:

- Download N1GE software files onto N1 System Manager.
- Create an N1GE version from these files.
- Create an application profile to associate with the version.
- List the available N1GE versions.
- View the details of a particular version.

What is an N1GE Version?

The term N1GE version specifically means the combination of an N1GE OS-specific `tar.gz` file and the `n1ge-6_0u4–common.tar.gz` file. For example, you would want to have separate N1GE versions for Solaris, Linux, and MS-Windows specific servers. The following table lists some of the OS-specific N1GE versions available from the Sun Download Center (SDLC). The versions in this table are the ones supported by N1SM. N1GE versions for other operating systems are also available and may function with N1SM but they are not officially supported.

N1GE Platform-Specific File	*Platform*
`Solaris_sparc/tar/n1ge-6_0u4-bin-solaris.tar.gz`	Solaris (SPARC platform) 32-bit binaries for Solaris 7, Solaris 8, and Solaris 9 Operating Systems. Note that N1SM does not support Solaris 7 and Solaris 8.
`Solaris_sparc/tar/n1ge-6_0u4-bin-solaris-sparcv9.tar.gz`	Solaris (SPARC platform) 64-bit binaries for Solaris 7, Solaris 8, and Solaris 9 Operating Systems. Note that N1SM does not support Solaris 7 and Solaris 8.
`Solaris_x86/tar/n1ge-6_0u4-bin-solaris-i586.tar.gz`	Solaris (x86 platform) binaries for Solaris 8, and Solaris 9 Operating Systems
`Solaris_x64/tar/n1ge-6_0u4-bin-solaris-x64.tar.gz`	Solaris (x64 platform) 64-bit binaries for Solaris 10

`Windows/tar/n1ge-6_0u4-bin-win32-x86.tar.gz`	Microsoft Windows (x86 platform 32-bit binaries for Windows 2000, XP and Windows Server 2003. Note that N1SM does not support Windows 2000.
`Linux24_i586/tar/n1ge-6_0u4-bin-linux24-i586.tar.gz`	Linux (x86 platform) binaries for the 2.4 kernel
`Linux24_amd64/tar/n1ge-6_0u4-bin-linux24-amd64.tar.gz`	Linux (AMD platform) binaries for the 2.4 kernel

You must copy an OS-specific N1GE `tar` file for each OS you plan to support as well as the `n1ge-6_0u4–common.tar.gz` file.

Note – The N1GE module can only use file in the `tar.gz` format.

▼ To Download the N1 Grid Engine Software Onto a Management Server

Before you can create an N1GE version, you must make the N1GE application files accessible to the management server that you will use provision the version out to managed servers. The previous table lists the available tar files.

◗ **Copy the desired tar files from the source location onto your N1SM management server.**

▼ To Create an N1GE Version

Before You Begin Once you have the N1GE `tar` files available on your management server, you can use them to create an N1SM N1GE version.

1. **Access the N1SM CLI (see "Accessing the N1SM CLI" on page 13).**

2. **You create versions of N1 Grid Engine software using the `create application` command. The syntax for this command is:**

   `create application` *application* `file [`*file*`,` *file*`...] type GridEngine`

application	A unique name for the N1GE version. For example, *N1GE6_U4*
file	A fully qualified path to the N1GE file to be copied. You can specify `*.tar.gz` installation files for the N1GE application, and each N1GE application requires the `n1ge-6_0u4–common.tar.gz` file.
type	The type of application; in this case, GridEngine.

Note – Unlike the behavior for OS profiles, a default application profile is not automatically created when you copy an N1GE to the N1 System Manager. You must create this profile yourself using the `create applicationprofile` command.

Example 2–1 Create an N1 Grid Engine Version

If your grid consists of Solaris 9 SPARC hosts , then you must include in the version these files:

```
N1-ok>create application N1GE6_U4 file Solaris_sparc/tar/n1ge-6_0u4-bin-solaris.tar.gz,n1ge-6_0u4–n1ge-6_0u4–common.tar
```

▼ To View Available N1GE Versions

You use the `show application` command to list all the available N1GE versions or detailed information about a specific version like the file list.

1. **Access the N1SM CLI (see "Accessing the N1SM CLI" on page 13).**
2. **To list all the available N1GE versions use this command:**

   ```
   show application all type GridEngine
   ```

3. **To list detailed information about a specific N1GE version use this command:**

   `show application` *application* `type GridEngine`

all	List all the available N1GE versions.
application	The name of an N1GE versions.
type	The type of application; in this case, GridEngine.

Example 2–2 Show N1GE Versions

```
N1-ok>show application all type GridEngine

N1-ok>show application N1GE6_U4 type GridEngine
```

▼ To Delete an N1GE Version

Before You Begin You cannot delete an N1GE version if it is currently deployed on a server. To undeploy it, use the `unload groupor` `unload server` commands to remove the application profile first.

`unload group` *group* `applicationprofile` *applicationprofile* `type GridEngine`

```
unload server server applicationprofile applicationprofile type GridEngine
```

1. **Access the N1SM CLI (see "Accessing the N1SM CLI" on page 13).**

2. **If you want to delete an N1GE version from N1 System Manager, use this command.**

   `delete application` *application* `type GridEngine`

   application The name of the N1GE version that you specified with the `.Create Application` command.

   type The type of application that the profile belongs to; in this case, GridEngine.

Example 2–3 Delete N1GE Version

```
N1-ok>delete application N1GE6_U4 type GridEngine
```

What's Next?

After you create an N1GE version, you must create an application profile and associate it with a particular version before your can provision the version to servers. The next section describes how to do this task.

Creating and Managing N1GE Application Profiles

This section describes how you create and manage an N1GE application profile. An application profile describes the deployment and functional attributes for an N1GE version. This section lists the following topics:

- Creating an application profile
- Listing the available application profiles
- Viewing the details of a particular application profile
- Deleting an application profile

▼ To Create an N1GE Application Profile

Before You Begin After you create the N1GE version, you create an application profile and associate it with the version. This profile is sort of like a configuration file for an N1GE version (although it is actually a set of database values). The profile specifies attributes like which TCP port to use for the N1GE `execd` daemon or the threshold values that will provoke a warning when exceeded.

You can have several application profiles associated with a version but only one profile can be active for a grid at any particular time. It is the application profile that you specify when you deploy N1GE onto a managed server.

Tip – This functionality is similar to that provided by the Settings menu choice of the GEMM application.

Note – The actual role that a server plays in a grid (master host and so forth) is not an application profile attribute. You define that role when you load the application profile onto the targeted server.

---

1. **Access the N1SM CLI (see "Accessing the N1SM CLI" on page 13).**

2. **Use the following command to create an application profile. If you are satisfied with the default N1GE attributes, you do not have to specify them explicitly. The syntax for this command is:**

   `create applicationprofile` *applicationprofile* `application` *application* `type GridEngine [`*N1GE-Attribute attributevalue*`,` *N1GE-Attribute attributevalue*`, ...]`

applicationprofile	A unique name for the application profile that will be used to provision the various N1GE servers.
application	The name of the particular N1GE version to associate with this application profile. This value is the name you specified with the `create application` command.
type	The type of application that the profile belongs to; in this case, GridEngine.
N1GE-Attribute	The specific N1GE attribute you want to define.

   N1GE ATTRIBUTES — These attributes define how an application version will be deployed and function when the profile they belong to becomes active. You can have several application profiles but only one profile can be active for a grid at any particular time.

   - `adminhomedir` – The home directory of the N1GE admin user. Default value is `/gridware/sge`.
   - `adminuid` – The UID of the N1GE admin user. Default value is `218`.
   - `adminusername` – The user name of the N1GE admin user. Default value is `sgeadmin`.
   - `execdport` – The TCP port to use for the N1GE `execd` daemon. Default value is `837`.
   - `instversion` – The version of N1GE that will be deployed on the compute and submit hosts. There is no default value.
   - `lnxnfsmtopts` – The options used when mounting the common directory onto a Linux compute or submit host. The value in this field is inserted into the Linux `/etc/fstab` file on each host as: *nfsservername*`:`*nfsmountpoint nfsmountpoint* **`nfs`** *lnxnfsmtopts* **`0 0`**. Default value is `intr,softload`. This value cannot contain spaces.

- `loadcritical` – Use this parameter to specify the load critical threshold. If this threshold is exceeded, a load critical alert appears in the Monitor. Similar to the Load Warning parameter, you set this parameter in terms of the system load scaled by number of CPUs. Default value is `3.00`.
- `loadwarning` – Use this parameter to specify the load warning threshold. If this threshold is exceeded, a load warning alert appears in the Monitor. The value is in terms of system load, as reported by the OS, divided by the number of CPUs. Default value is `1.00`.
- `masterport` – The CP port to use for the N1GE `qmaster` daemon. Default value is `836`.
- `maxpendtime` – Use this parameter to specify the amount of time that a job spends pending after which a Job Pending alert appears in the Monitor. You set the value in hours. Default value is `24`.
- `memcritical` – Use this parameter to set the memory critical threshold. If the value drops below this threshold, a memory critical alert appears in the Monitor. You set the value in terms of megabytes of free virtual memory. Default value is `10`.
- `memwarning` – Use this parameter to set the memory warning threshold. If the value drops below this threshold, a memory warning alert appears in the Monitor. You set the parameter value in terms of megabytes of free virtual memory. Default value is `100`.
- `nfsmountpoint` – The directory that is mounted from the NFS server for the N1GE common directory. When deploying the master host using N1GE, this value is set automatically to *sgeroot*/*sgecell*/`common`. Once you deploy the master host, you cannot edit this value and it remains in effect for all further deployments of compute and submit hosts. You can edit this setting again only if you uninstall the master host. Default value is `/gridware/sge/default/common`.
- `nfsservername` – The name of the NFS server from which all compute and submit hosts will mount the N1GE "common" directory. When you deploy the master host using N1GE, this parameter is set automatically to the master host. Once you deploy the master host, you cannot edit this value and it remains in effect for all further deployments of compute and submit hosts. You can edit this setting again only if you uninstall the master host. There is no default value.
- `proxyhost` – Indicates the host on which monitoring commands are executed. If the master host has been previously deployed using N1GE, then the proxy host is set to this host and cannot be changed until the master is uninstalled. The host you chose must be an N1GE admin host; otherwise, installation and uninstallation of other hosts, as well as monitoring, could fail. There is no default value.
- `sgecell` – The N1GE cell name used for the deployment. Default value is `default`.
- `sgeroot` – The root directory under which the N1GE files will be installed. The files will be installed on all hosts in this directory. Default value is `/gridware/sge`.
- `solnfsmtopts` – The options used when mounting the "common" directory onto a Solaris compute or submit host. The value in this field is inserted into the Solaris `/etc/vfstab` file on each host as: *nfsservername*`:`*nfsmountpoint nfsmountpoint* **`nfs -yes`** *solnfsmtopts*. There is no default value. This value cannot contain spaces.

Example 2–4 Create an Application Profile

```
N1-ok>create applicationprofile N1GE6_U4_Profile application GE6U4 type GridEngine
```

▼ To View Available N1GE Application Profiles

Use the `show applicationprofile` command to list all available application profiles or detailed information about a specific application profile.

1. **Access the N1SM CLI (see "Accessing the N1SM CLI" on page 13).**

2. **To list all the available N1GE application profiles use:**

   ```
   show applicationprofile all type GridEngine
   ```

3. **To list detailed information about a specific N1GE application profile use:**

   ```
   show applicationprofile applicationprofile type GridEngine
   ```

all	List all the available N1GE application profiles.
applicationprofile	The name of a particular N1GE application profile.
type	The type of application that the profile belongs to ; in this case, GridEngine.

Example 2–5 Show an Application Profile

```
N1-ok>show applicationprofile [all] type GridEngine
```

```
N1-ok>show applicationprofile N1GE6_U4_Profile type GridEngine
```

The following is an example of a typical application profile produced by a `show applicationprofile`command.

```
Name:             p1
Application Name:
Type:             GridEngine
Active:           false
adminhomedir:     /gridware/sge
adminuid:         218
adminusername:    sgeadmin
execdport:        837
instversion:
lnxnfsmtopts:     defaults
loadcritical:     3
loadwarning:      1
masterhost:
masterport:       836
```

```
masterready:
maxpendtime:      24
memcritical:      10
memwarning:       100
nfsmountpoint:    /gridware/sge/default/common
nfsservername:
proxyhost:
proxyisadmin:
sgecell:          default
sgeroot:          /gridware/sge
solnfsmtopts:
```

▼ To Delete an N1GE Application Profile

Before You Begin You cannot delete an N1GE application profile if a master host installed with that profile has not been uninstalled first. To remove N1GE from the Master Host, use the `unload server` command.

1. **Access the N1SM CLI (see "Accessing the N1SM CLI" on page 13).**

2. **Use the `delete applicationprofile` command to delete an N1GE application profile. The command syntax is:**

   `delete applicationprofile` *applicationprofile* `type GridEngine`

applicationprofile	The name of the N1GE application profile that you want to delete.
type	The type of application that the profile belongs to; in this case, GridEngine.

Example 2–6 Delete an N1GE Application Profile

`N1-ok>delete applicationprofile` *N1GE6_U4_Profile* `type GridEngine`

What's Next?

After you have created an application profile for an N1GE version, you can use the profile to provision a grid engine system with it. See "Installing N1 Grid Engine onto Servers" on page 25.

Managing N1GE Settings

N1GE settings are global values reflecting the attributes of a particular application profile. You can have several application profiles but only one profile can be active for a grid at any particular time. The settings are described in the attributes section of the `create applicationprofile` command. To see the settings for a particular profile, use this command:

```
show applicationprofile applicationprofile type GridEngine
```

where *applicationprofile* is the name of the profile whose settings you want to see.

The following is an example of a typical application profile produced by a `show applicationprofile`command.

```
Name:             p1
Application Name:
Type:             GridEngine
Active:           false
adminhomedir:     /gridware/sge
adminuid:         218
adminusername:    sgeadmin
execdport:        837
instversion:
lnxnfsmtopts:     defaults
loadcritical:     3
loadwarning:      1
masterhost:
masterport:       836
masterready:
maxpendtime:      24
memcritical:      10
memwarning:       100
nfsmountpoint:    /gridware/sge/default/common
nfsservername:
proxyhost:
proxyisadmin:
sgecell:          default
sgeroot:          /gridware/sge
solnfsmtopts:
```

Changing Application Profile Settings

The active application profile is the one that was used when the master host was installed. You can change some of these global settings when the application profile is active and they are applied to the grid as a whole. However, you cannot change the settings of a particular server. For an inactive profile, you can change any of the settings.

You can only change the following settings for an active profile when the master host is managed by N1SM:

- *loadcritical*
- *loadwarning*
- *maxpendtime*

- *memcritical*
- *memwarning*

You can only change the following settings for an active profile when the Master host is an external host (proxy host is being used):

- *proxyhost*

To Change an Application Profile Setting

1. **Access the N1SM CLI (see "Accessing the N1SM CLI" on page 13).**
2. **If the profile you want to change is currently active, unload the profile (see "To Unload N1GE From a Managed Server" on page 27).**
3. **Edit the profile to make the desired changes.**
4. **Reload the profile (see "To Load N1GE Onto Managed Servers" on page 26).**

Installing N1 Grid Engine onto Servers

You can install N1GE versions onto managed server groups or onto individual servers. The method of installation is to load an application profile onto a server while specifying the server's N1GE role.

Note – You cannot install the master host with the `load group` command. To create an N1GE master host, use the `load server` command.

To Load N1GE Onto a Managed Server Group

Before You Begin To deploy N1GE, you must previously have created an application (specifying a particular N1GE version) and an associated application profile (specifying the installation parameters).

1. **Access the N1SM CLI (see "Accessing the N1SM CLI" on page 13).**
2. **Use the `load group` command to install an N1GE version onto a group of servers. This is the command syntax:**

   `load group` *group* `applicationprofile` *applicationprofile*
   `type GridEngine hosttype [`*hosttype*`]`

   group The name of a server group. To create a server group, use the N1SM `create group` command.

applicationprofile	The name of the N1GE application profile that you want to load.
type	The type of application that the profile belongs to; in this case, GridEngine.
hosttype	The type of N1 Grid Engine host to install. Valid values are compute (also known as an execution host) and submit (also known as an access host).

Example 2–7 Loading N1GE onto a Server Group

```
N1-ok>load group MyComputeServers applicationprofile N1GE6_U4_profile type GridEngine hosttype compute
```

▼ To Load N1GE Onto Managed Servers

1. **Access the N1SM CLI (see "Accessing the N1SM CLI" on page 13).**

2. **Use the `load server` command to install N1GE on one or several managed servers. This is the command syntax:**

```
load server server[,server...] applicationprofile applicationprofile type GridEngine hosttype [hosttype]
```

server	The management name of a server.
applicationprofile	The name of the N1GE application profile that you want to load.
type	The type of application that the profile belongs to; in this case, GridEngine.
hosttype	The type of N1 Grid Engine host to install. Valid values are compute (also known as execution host), submit (also known as an access host), and master.

Example 2–8 Loading N1GE on a Master Host

```
N1-ok>load server MyMasterHost applicationprofile N1GE6_U4_profile
type GridEngine hosttype master
```

▼ To Unload N1GE From Managed Server Group

Before You Begin You cannot use the `unload group` command to uninstall a N1GE master host; you must use the `unload server` command.

1. **Access the N1SM CLI (see "Accessing the N1SM CLI" on page 13).**

2. **Use the `unload group` command to uninstall N1GE from a group of servers. This is the command syntax:**

```
unload group group applicationprofile applicationprofile type GridEngine
```

group	The name of a server group. To create a server group, see the N1SM `create group` command.
applicationprofile	The name of the N1GE application profile that you want to unload.
type	The type of application that the profile belongs to; in this case, GridEngine.

Example 2–9 Unloading a N1GE From a Managed Server Group

```
N1-ok>unload group MyComputeServers applicationprofile N1GE6_U4_profile type GridEngine
```

▼ To Unload N1GE From a Managed Server

1. **Access the N1SM CLI (see "Accessing the N1SM CLI" on page 13).**

2. **Use the `unload server` command to uninstall N1GE from one or more servers. The command syntax is:**

```
unload server server[,server...] applicationprofile applicationprofile type GridEngine
```

server	The management name of a server.
applicationprofile	The name of the N1GE application profile that you want to unload.
type	The type of application that the profile belongs to; in this case, GridEngine.

Example 2–10 Unloading a Profile from a Managed Server

```
N1-ok>unload server MyMasterHost applicationprofile N1GE6_U4_profile type GridEngine
```

CHAPTER 3

Setting Up a Grid Using a Proxy Host

Under certain circumstances, you may want to use a proxy host to manage a grid rather than a master host that is part of the local network. To do so, use the information documented in this chapter.

Steps Outside of N1SM

It is assumed that you have set up a grid using the N1 Grid Engine software. This task is done outside of the scope of N1SM.

To Define the Proxy Host

1. **If you have not done so already, log into the master host as the root user.**

2. **Source the settings script. The script you use depends on which shell you are using. The settings are in `/gridware/sge/default/common/settings.[sh|csh]`. The actual path to these settings depends on where you installed the N1 Grid Engine files.**

3. **Run the following command:**

   `gconf -ah` *proxy_host_name*

   where *proxy_host_name* is the name of the N1SM managed server that you want to be the proxy to this Grid. The *proxy_host_name* must be known by the master host using DNS or the `/etc/hosts` file.

Steps Using N1SM

You perform the following steps using the N1SM CLI and N1GE module GUI.

▼ To Use N1SM to Discover the Proxy Host

◗ **Discover the hosts to be added to the grid. For more information, go to Chapter Chapter 4, "Discovering Manageable Servers," in** *Sun N1 System Manager 1.3 Discovery and Administration Guide*, **of the** *Sun N1 System Manager 1.3 Discovery and Administration Guide*.

▼ To Set Up the Proxy Host

1. **Load an OS on these hosts if you not done so already. For more information, go to the** *Sun N1 System Manager 1.3 Operating System Provisioning Guide*

2. **Set up** `/etc/hosts` **or DNS on each of the hosts so that they know the location of the master host.**

3. **Create an application (see** "To Create an N1GE Version" on page 17**) and application profile (see** "Creating and Managing N1GE Application Profiles" on page 19**.**

4. **Add the basemanagement or osmonitoring feature to the hosts that need to be managed. For more information, see the** *Sun N1 System Manager 1.3 Discovery and Administration Guide*.

5. **Set the application profile settings to:**
   - *proxyhost*: The name of the N1SM managed host that you want to be the proxy.
   - *nfsservername*: The name of the master host on the external grid.
   - *nfsmountpoint*: The directory on the NFS server that contains the N1GE common directory.
   - *instversion*: The application that you are using.

6. **Run the** `load server`**command with** *hosttype compute* **or** *hosttype submit* **for the host to set it up as a proxy.**

▼ To Set Up Compute and Submit Hosts

◗ **After the proxy host has been set up, install other compute or submit hosts by running the** `load server` **or** `load group` **command.**

CHAPTER 4

Monitoring N1 Grid Engine

This chapter tells you how to get a snapshot of a grid's performance, and how to view details about cluster queues and different types of N1 Grid Engine alerts. All these features are available from the N1 Grid Engine Monitor GUI.

Note – To actually manage applications using N1GE, you must use the various tools and commands available from N1GE itself. For example, you can use N1GE Monitor GUI to view the status of a submitted job but you cannot actually submit a job from this GUI.

Quickly Viewing Grid Performance

You use the Overview tab to view a quick picture of the health of your grid. This tab displays the Monitoring Overview page which shows three tables that have Summary status, Cluster queue information, and aggregated Alerts for Queues, Hosts, and Jobs.

Note – You should reload this page to get the freshest data.

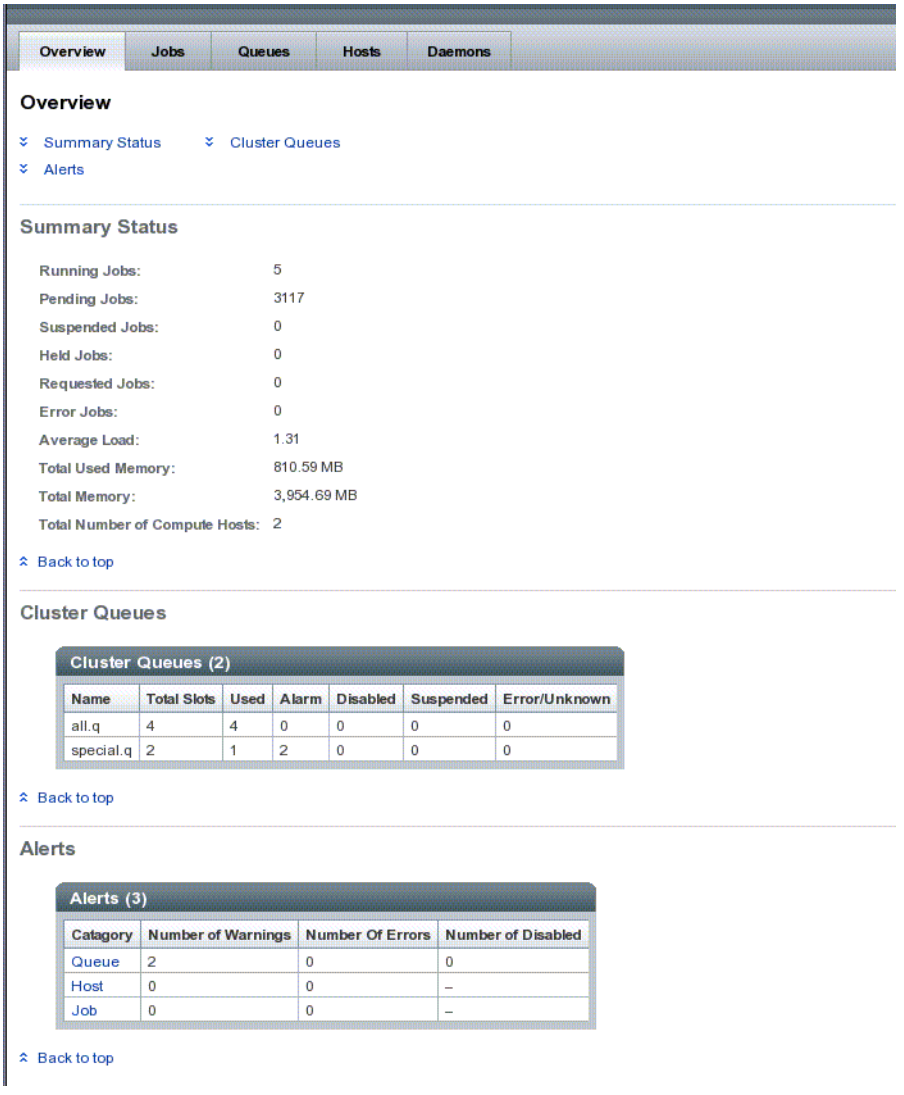


FIGURE 4–1 N1 Grid Engine Monitoring Overview Page

Summary Status Table

The Summary Status table shows the total number of jobs in the grid in various states: pending, running, suspended, and so forth). It also shows the load averaged across all compute hosts and the total amount of used and installed memory summed over all compute hosts.

- **Running Jobs** – The number of all the jobs currently running in the grid.

- **Pending Jobs** – The number of jobs waiting to be dispatched by the scheduler.
- **Suspended Jobs** – The number of jobs that are temporarily suspended.
- **Held Jobs** – The number of jobs explicitly held in the pending state.
- **Requeued Jobs** – The number of jobs that were formerly running but that have been placed back in the pending state.
- **Error Jobs** – The number of jobs no longer running or that never were run due to error conditions like invalid requests.
- **Avg Load** – The amount of CPU cycles being used by all the running jobs divided by the number of compute hosts being used by the grid.
- **Total Used Memory** – The amount of total memory being used by all the running jobs in the grid.
- **Total Memory** – The total amount of memory available across all compute hosts.
- **Total Number of Compute Hosts** – The number of hosts available to execute job tasks.

Cluster Queues Table

Throughout its duration, a running job is associated with its queue. Queues provide a way to define various job execution parameters that apply to multiple hosts. You can think of an N1GE queue as a container, or description, for a class of jobs. Queues that span multiple execution hosts are sometimes referred to as cluster queues.

The Cluster Queues table shows a summary of the state of all the cluster queues configured on the grid. The slots are indicative of general performance. The states indicate which queues are running various potential error states. The fields include:

- **Cluster Queue** — The name given to a queue.
- **Total Slots** — The total number of slots configured for this queue. Slots are the maximum number of jobs that a queue can run simultaneously.
- **Used** — The number of total slots currently being used by the queue. Queues should be using all of the total slots, although in some cases, enough free resources might not be available to accommodate every slot.
- **Alarm** — When present, indicates that at least one of the load thresholds defined in the *load_thresholds* list of the queue configuration is currently exceeded. This state prevents N1GE from scheduling further jobs to that queue. For more information, see the `queue_conf(5))` man page.
- **Disabled** — The number of slots that are not running because the queue or host has been disabled either manually or automatically. All jobs associated with that queue are also disabled. You assign and release this state to a queues using the `qmod(1)` command. New jobs are also not accepted by these slots, although jobs running continue to run.
- **Suspended** — The number of slots that are not running because the queue or host has been suspended either manually or automatically. All jobs associated with these slots are also suspended, and no new jobs are accepted by these slots.

- **Error/Unknown** — the number of slots that are in the error state, due either to a problem experienced by a previous job in this slot or else due to a host being unreachable.

For information on cluster queues, see the Monitoring and Controlling Queues section in the *N1GE 6 User's Guide* and the `qmon` man page. For more information on queue states see the Queue Alerts.

Alerts Table

The Alerts table displays a quick look at potential or actual problems with the grid. You receive alerts when any of these categories generates a warning, an error, or becomes disabled. Clicking on a category displays the Alert page for that category which contains a table of alerts with additional information. Categories include:

- Queue Alerts
- Host Alerts
- Job Alerts

Sorting and Pagination Controls

Items display ten rows at a time. You can see the entire list by using the pagination controls at the bottom of the table. By default, rows are displayed numerically by job ID, but you can use any column to change the ordering of the rows. Clicking on a column header sorts the rows according to the values in that column. Clicking on the column header again reverses the sort. The sorting is preserved across pages if you click on a pagination button.

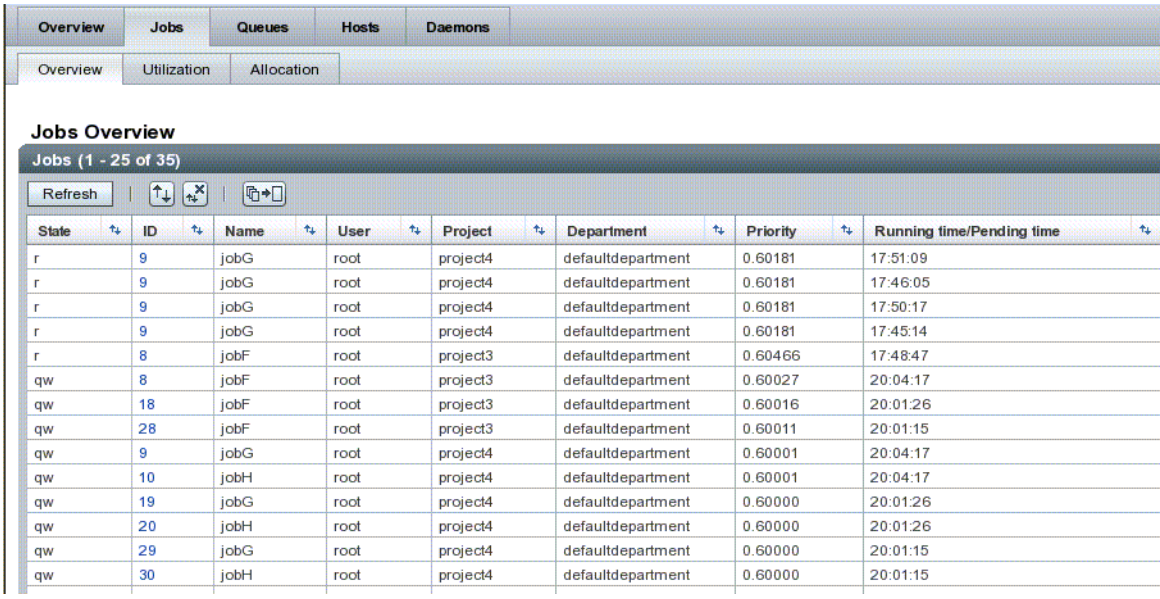
CHAPTER 5

Working With N1 Grid Engine Jobs

Each application running on the grid is considered a job. The following sections describe how you can check a job's state as well as it's utilization of resources and it's scheduling policy. This information is displayed in different views of a jobs data including and overview, a utilization view, and an allocation view. You can also see fine-grained information about each job including details about each job's composite tasks.

Checking a Job's State

Use the Jobs Overview tab as a quick way to check a job's State and see some of the factors that might affect its performance. Clicking a job ID displays a Job Details page that provides very detailed information.



State	ID	Name	User	Project	Department	Priority	Running time/Pending time
r	9	jobG	root	project4	defaultdepartment	0.60181	17:51:09
r	9	jobG	root	project4	defaultdepartment	0.60181	17:46:05
r	9	jobG	root	project4	defaultdepartment	0.60181	17:50:17
r	9	jobG	root	project4	defaultdepartment	0.60181	17:45:14
r	8	jobF	root	project3	defaultdepartment	0.60466	17:48:47
qw	8	jobF	root	project3	defaultdepartment	0.60027	20:04:17
qw	18	jobF	root	project3	defaultdepartment	0.60016	20:01:26
qw	28	jobF	root	project3	defaultdepartment	0.60011	20:01:15
qw	9	jobG	root	project4	defaultdepartment	0.60001	20:04:17
qw	10	jobH	root	project4	defaultdepartment	0.60001	20:04:17
qw	19	jobG	root	project4	defaultdepartment	0.60000	20:01:26
qw	20	jobH	root	project4	defaultdepartment	0.60000	20:01:26
qw	29	jobG	root	project4	defaultdepartment	0.60000	20:01:15
qw	30	jobH	root	project4	defaultdepartment	0.60000	20:01:15

FIGURE 5–1 Jobs Overview Tab

The fields on the Job Overview tab include:

- **State** – The Job state is indicated by the following letters:
  - **d** (deletion) — Indicates that a job has been deleted (using `qdel(1)`).
  - **r** (running) — Indicates that a job is about to be executed or is already executing
  - **R** (restarted) — Indicates that the job was restarted. This state can be caused by a job migration or because of one of the reasons described in the *-r* section of the `qsub` man page.
  - **s** (suspended) — Shows that an already running job has been suspended (using `qmod(1)`).
  - **S** (suspended) — Show that an already running job has been suspended because the queue that it belongs to has been suspended.
  - **t** (transferring) — Indicates that a job is about to be executed or is already executing.
  - **T** (threshold) — Show that an already running job has been suspended because at least one suspend threshold of the corresponding queue was exceeded (for more information, see the `queue_conf` man page) and that the job has been suspended as a consequence.
  - **w** (waiting) — Indicates that the job is suspended pending the availability of a critical resource or specified condition.

  See the `qstat(1)` man page for a detailed explanation about these state conditions. For more information, you can also see Monitoring and Controlling Jobs and Queues in the *N1 Grid Engine User* manual.
- **ID** – The job ID provides a unique identity for the job and also a method of accessing the Job Details page.

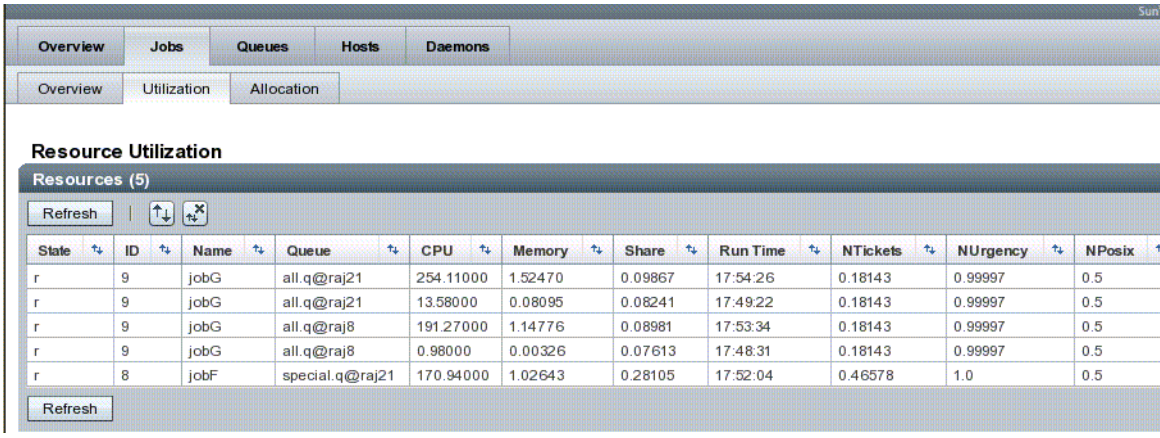
- **Name** – The name of the job. Assigning names to jobs makes them more comprehensible and easier to track than just relying on job IDs.
- **User** – The name of the user who submitted the job.
- **Project** – The name of the project to which the job is assigned as specified in the *qsub(1) -P* option or by the default project of the submitting user.
- **Department** – The name of the department to which the user belongs. Use the *-sul* and *-su* options of *qconf* command to display the current department definitions).
- **Priority** – The dispatch priority of the job determining its position in the pending jobs list. The dispatch priority is a decimal number with higher values denoting higher priority. The priority value is determined dynamically based on the ticket and urgency policy setup.
- **Running Time/Pending Time** – The time that has elapsed since the job started running or, for the case jobs that are still in the queue, how long the job has been waiting to run.
- **Task** – The currently executing task. Some jobs consist of a single task (the task ID is always 1.). However, parallel jobs and array jobs each consist of more than one task. The tasks are usually numbered in ascending order starting with 1. Depending upon how the job was submitted, sometimes the numbers might skip, 1,3,5. On running jobs, each task runs distinctly and so has its own configuration information, environment, and trace. For details about the task, click the task number to display the Task Details page.

The Job User, Project, and Department are elements that you can use in an Entitlement policy (also known as a Ticket policy) to affect a job's dispatch priority. For example, jobs from one Department can always be entitled to have a higher dispatch priority than those from another Department.

Dispatch Priority is computed from three top-level scheduling policies: Entitlement, Urgency, and Custom (also known as POSIX) . For more detailed information on N1GE scheduling policies and dispatch priority, see the `sge_priority` man page and *Scheduler Policies for Job Prioritization in the Sun N1 Grid Engine 6 System* (www.sun.com/blueprints/1005/819-4325.html (`http://www.sun.com/blurprints/1005/819-4325.html`)).

Checking Grid Resources

Use the Job Utilization View tab to display information that is relevant to a job's consumption of a grid computing resources as well as other elements that factor into a job's dispatch priority. Unlike the Overview view, only running and suspended jobs appear. In the Utilization view, the columns are as follows:



State	ID	Name	Queue	CPU	Memory	Share	Run Time	NTickets	NUrgency	NPosix
r	9	jobG	all.q@raj21	254.11000	1.52470	0.09867	17:54:26	0.18143	0.99997	0.5
r	9	jobG	all.q@raj21	13.58000	0.08095	0.08241	17:49:22	0.18143	0.99997	0.5
r	9	jobG	all.q@raj8	191.27000	1.14776	0.08981	17:53:34	0.18143	0.99997	0.5
r	9	jobG	all.q@raj8	0.98000	0.00326	0.07613	17:48:31	0.18143	0.99997	0.5
r	8	jobF	special.q@raj21	170.94000	1.02643	0.28105	17:52:04	0.46578	1.0	0.5

FIGURE 5–2 Job Utilization View Tab

- **State** – The Job State is indicated by the following letters:
  - **d** (deletion) – Indicates that a job has been deleted (using `qdel`).
  - **r** (running) – Indicates that a job is about to be executed or is already executing
  - **R** (restarted) – Indicates that the job was restarted. This can be caused by a job migration or because of one of the reasons described in the *-r* section of the `qsub(1)` command.
  - **s** (suspended) – Shows that an already running job has been suspended (using `qmod(1))`..
  - **S** (suspended) – Show that an already running job has been suspended because the queue that it belongs to has been suspended.
  - **t** (transferring) – Indicates that a job is about to be executed or is already executing.
  - **T** (threshold) – Show that an already running job has been suspended because at least one suspend threshold of the corresponding queue was exceeded (see `queue_conf(5))` and that the job has been suspended as a consequence.
  - **w** (waiting) – Indicates that the job is suspended pending the availability of a critical resource or specified condition.

  See the `qstat`man page for a detailed explanation about these state conditions. For more information, you can also see Monitoring and Controlling Jobs and Queues in the *N1 Grid Engine User* manual.
- **ID** – The job ID provides a unique identity and also a method of accessing the Job Details page.
- **Name** – The name of the job. Assigning names to jobs makes them more comprehensible and easier to track than just relying on job IDs.
- **Queue** – The queue instance which this the job belongs to.
- **CPU** – The amount of CPU time that the job has consumed.
- **Memory** – The amount of memory that the job is using.
- **Share** – The calculated share of the total system to which the job is entitled currently.

- **Run time** – The length of time the job has been running since it was dispatched.
- **NTickets** – The normalized Ticket priority. You can use the Override component of the ticket policy to increase the entitlement of a specific User, Project, or Department. By assigning Override Tickets, you can modify the entitlement without affecting any prioritization assignments of the Urgency policy.
- **NUrgency** – The normalized Urgency priority. Three factors contribute to this priority: the deadline contribution, the wait-time contribution, and the resource requirement contribution.
- **NPOSIX** – The normalized POSIX priority. An administrator can use this value to arbitrarily increase the priority of certain jobs.
- **Task** – The currently executing task. Some jobs consist of a single task, in which case, the task ID is always 1. However, parallel jobs and array jobs each consist of more than one task. The tasks are usually numbered in ascending order starting with 1. Depending upon how the job was submitted, sometimes the numbers might skip, (1,3,5,). On running jobs, each task runs distinctly and so has its own configuration information, environment, and trace. For details about the task, click the task number to display the Task Details page.

Note – If the CPU usage or memory usage values are blank, the usage information for that job has not yet been reported. Check back at a later time to see if the usage is then reported.

For more information on the meaning of each column, see the `QMON` man page.

Normalized Priorities

The normalized ticket, urgency, and POSIX priorities are the three top level policies used by the N1GE Scheduler to determine a job's dispatch priority. Each calculate a factor that contributes to the overall priority. In order for these three policy contributions to be added together in a meaningful way, they are each normalized to a number between 0 and 1.

Checking Scheduling Policies

With the Job Allocation View tab, you can see information about the factors that constitute scheduling policies that contribute to the dispatch priority that a job enjoys. You can use this view to determine whether your priority policies are actually in effect and to troubleshoot the components that determine an job's overall priority in the queue.

A job's priority is determined based on three policies:

- Ticket policy
- Custom (or POSIX) policy
- Urgency policy

The first part of the equation, Tickets, tells you the calculations that the scheduler is making in order to implement the entitlement-oriented scheduling policy that has been configured. Tickets provide a

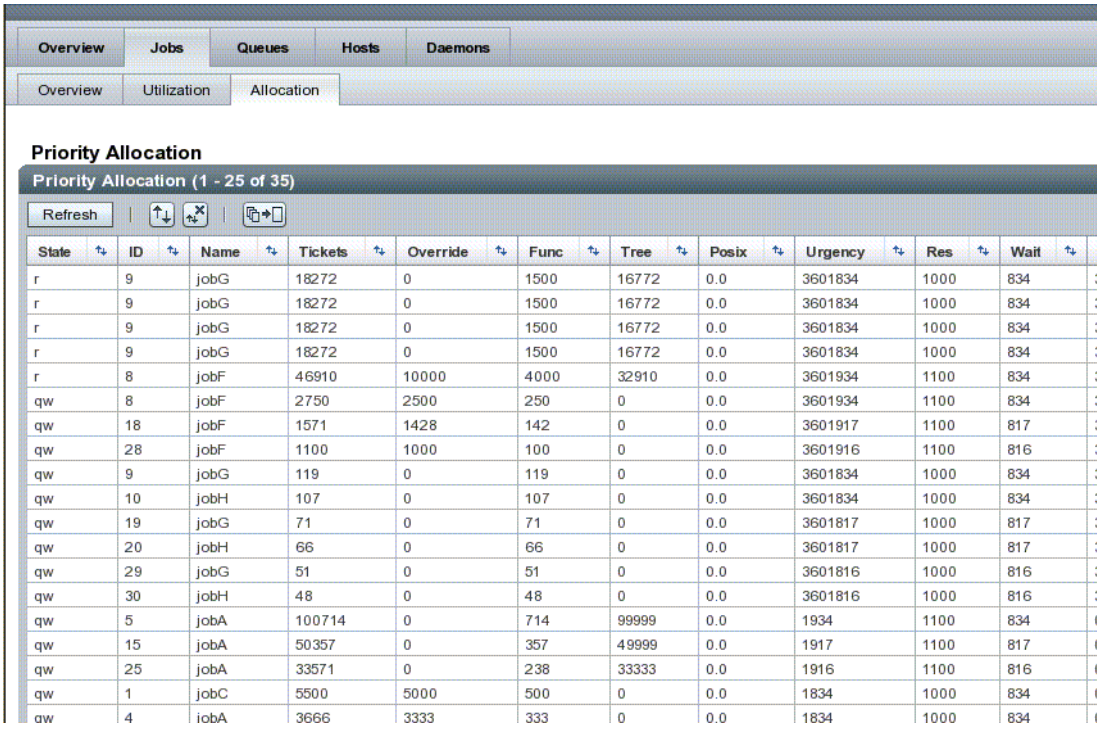
window into the inner logical workings of the scheduler. This feature helps you to verify that whatever policy you wanted is in fact being obeyed. It also provides you with a means for diagnosing any problems or unexpected behavior you might be seeing.

From a high level, the number of tickets assigned to a job is directly proportional to the job's entitlement. The higher the number, the greater the entitlement. Jobs with a large entitlement often have a high priority, however, the overall priority is affected by the other two aspects as well unless you have deliberately turned off the urgency and custom policies In that case, only the entitlement ("tickets") policy is active.

The second part of the priority equation is Custom (also called POSIX) priority. An administrator can use this value to arbitrarily increase the priority of certain jobs.

The third part of the priority equation, Urgency, accounts for only the job's individual characteristics, not its owner. The urgency value is derived from the sum of three contributions: the deadline contribution, the wait-time contribution, and the resource requirement contribution.

For more detailed information on N1GE scheduling policies and dispatch priority, see the `sge_priority` man page and *Scheduler Policies for Job Prioritization in the Sun N1 Grid Engine 6 System* (www.sun.com/blueprints/1005/819-4325.html (`http://www.sun.com/blurprints/1005/819-4325.html`)).



State	ID	Name	Tickets	Override	Func	Tree	Posix	Urgency	Res	Wait
r	9	jobG	18272	0	1500	16772	0.0	3601834	1000	834
r	9	jobG	18272	0	1500	16772	0.0	3601834	1000	834
r	9	jobG	18272	0	1500	16772	0.0	3601834	1000	834
r	9	jobG	18272	0	1500	16772	0.0	3601834	1000	834
r	8	jobF	46910	10000	4000	32910	0.0	3601934	1100	834
qw	8	jobF	2750	2500	250	0	0.0	3601934	1100	834
qw	18	jobF	1571	1428	142	0	0.0	3601917	1100	817
qw	28	jobF	1100	1000	100	0	0.0	3601916	1100	816
qw	9	jobG	119	0	119	0	0.0	3601834	1000	834
qw	10	jobH	107	0	107	0	0.0	3601834	1000	834
qw	19	jobG	71	0	71	0	0.0	3601817	1000	817
qw	20	jobH	66	0	66	0	0.0	3601817	1000	817
qw	29	jobG	51	0	51	0	0.0	3601816	1000	816
qw	30	jobH	48	0	48	0	0.0	3601816	1000	816
qw	5	jobA	100714	0	714	99999	0.0	1934	1100	834
qw	15	jobA	50357	0	357	49999	0.0	1917	1100	817
qw	25	jobA	33571	0	238	33333	0.0	1916	1100	816
qw	1	jobC	5500	5000	500	0	0.0	1834	1000	834
qw	4	jobA	3666	3333	333	0	0.0	1834	1000	834

FIGURE 5–3 Job Allocation View Tab

The Job Allocation View page displays the following information:

- **State** – The Job State is indicated by letters, specifically:
  - **d** (deletion) – Indicates that a job has been deleted (using`qdel(1)`).
  - **r** (running) – Indicates that a job is about to be executed or is already executing
  - **R** (restarted) – Indicates that the job was restarted. This can be caused by a job migration or because of one of the reasons described in the *-r* section of the `qsub(1)` command.
  - **s** (suspended) – Shows that an already running job has been suspended (using `qmod(1)`).
  - **S** (suspended) – Show that an already running job has been suspended because the queue that it belongs to has been suspended.
  - **t** (transferring) – Indicates that a job is about to be executed or is already executing.
  - **T** (threshold) – Show that an already running job has been suspended because at least one suspend threshold of the corresponding queue was exceeded (see `queue_conf(5)`) and that the job has been suspended as a consequence.
  - **w** (waiting) – Indicates that the job is suspended pending the availability of a critical resource or specified condition.

See the `qstat`man page for a detailed explanation about these state conditions. For more information, you can also see Monitoring and Controlling Jobs and Queues in the *N1 Grid Engine User* manual.

- **ID** – The job ID provides a unique identity and also a method of accessing the Job Details page.
- **Name** – The name of the job. Assigning names to jobs makes them more comprehensible and easier to track than just relying on job IDs.
- **Tickets** – T he total number of tickets for the job. The more tickets a job has assigned to it, the higher that job's priority. This value is the "raw" number before it is normalized.
- **Override** – The number of Override tickets. By assigning Override tickets, you can modify the entitlement without affecting any prioritization assignments of the Urgency policy.
- **Func** – The number of functional tickets.
- **Tree** – The number of share tree tickets. The share tree defines the long-term resource entitlements of users/projects and of a hierarchy of arbitrary groups made up of them.
- **Posix** – The POSIX priority. This feature provides a way to increase a job's priority. This is the "raw" number before it is normalized.
- **Urgency** – The total urgency for the job made up of the deadline contribution, the wait-time contribution, and the resource requirement contribution. This is the "raw" number before it is normalized.
- **Res** – The resource contribution to the urgency
- **Wait** – The waiting time contribution to the urgency.
- **Ddln** – The deadline contribution to the urgency.
- **Task** – The currently executing task. Some jobs consist of a single task in which case, the task ID is always 1. However, parallel jobs and array jobs each consist of more than one task. The tasks are usually numbered in ascending order starting with 1. Depending upon how the job was submitted, sometimes the numbers might skip like 1,3,5. On running jobs, each task runs distinctly and so has its own configuration information, environment, and trace. For details about the task, click the task number to display the Task Details page.

Note – You can see the normalized values for Tickets, POSIX, and Urgency using the Job Utilization View tab.

For more information on the meaning of each column, see the `qmon` man page.

Seeing Detailed Job Information

You can see complete details about a job by selecting the job ID on any of the job views tabs. The Job Details page that appears presents this information in three tables: General, Usage Details, and Schedule Details.

The General table provides details including various properties related to the jobs environment, resource requests, submit options, and so forth.

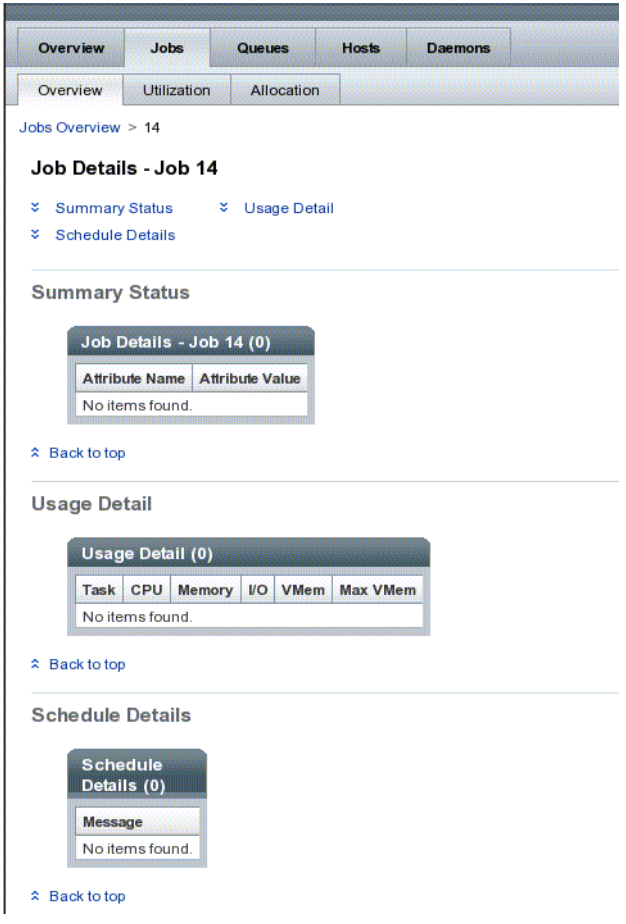


FIGURE 5–4 Job Details Page

The Usage Details table shows the current resource utilization for that job. If this information is not available, for example, because the job started too recently or the job is still pending, then this table is empty. For jobs with multiple tasks, the usage of each task appears on a separate line.

The Schedule Details table shows the scheduling information for that job.

Most of the fields on this page are self-explanatory. For more information, see the `qstat` man page.

Seeing Detailed Task Information

The Task Details page contains four tables that provide detailed information about the selected task. This one details page contains information for each task that appears in the three job views tabs. All the information on this page is useful for diagnosing jobs that might be experiencing some kind of problem or issue.

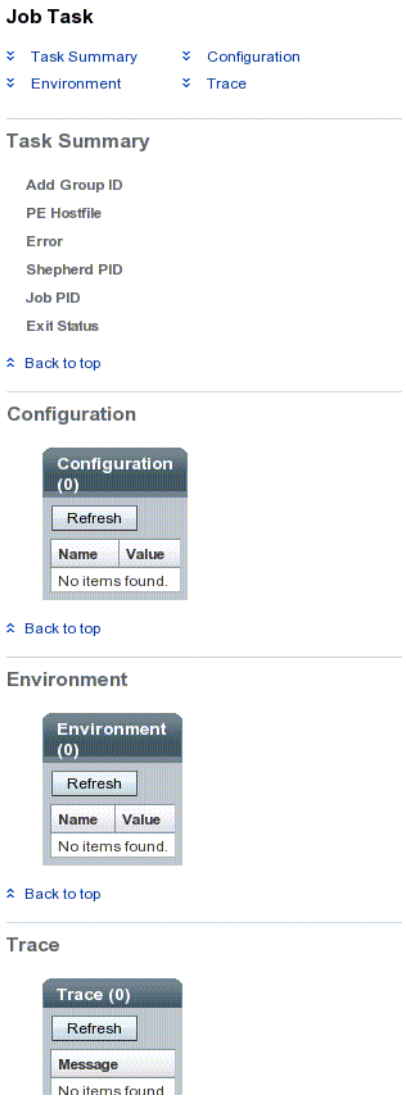


FIGURE 5–5 Task Details Page

This Task Details page contains tables of information that correspond to a different file from the job spool directory. For more information on the information in the job spool directory, see the N1 Grid Engine 6 Administration manual. The tables are:

- Task Summary
- Configuration
- Environment

- Trace

Task Summary Table

The Task Summary table tells you basic information about the job task.

- **Add Group ID** — Contains one line with the additional group ID used to control and monitor the job.
- **PE Hostfile** — A file describing the host setup of a parallel job which contains each involved host, the queues the job was spooled into, and the number of reserved slots (tasks) per host.
- **Error** — Contains an error message in the case of severe errors during the startup of a job. For example, `Execd cannot start shepherd`.
- **Shepherd PID** — The process ID of the shepherd.
- **Job PID** — The process ID of the job (the shepherd's child process).
- **Exit Status** — The numeric exit code of the job in a single line.

CHAPTER 6

Working With N1 Grid Engine Queues

This chapter describes how to access information about a grid's queues. You can see a general picture of the performance health of all the queues and view details about a particular queue.

Monitoring Queues

Queue information is available from the Queue Summary tab. You use this page to see whether a queue is functioning and how efficiently it is performing. From this page you can also view extensive details on any queue.

A *queue* in the N1GE environment is a means of defining a job's execution environment. This context includes features like:

- job runtime limits (memory, stack, and CPU time)
- control action methods (how to suspend and resume the job)
- virtual job container (Solaris, Linux, or MS–Windows resource pools)

A queue instance is the portion of the queue that exists on a single host.

The information in this tab is presented in a table of queue instances, that is, the portion of the queue that runs on a particular host. Every queue instance that exists in the grid is listed.

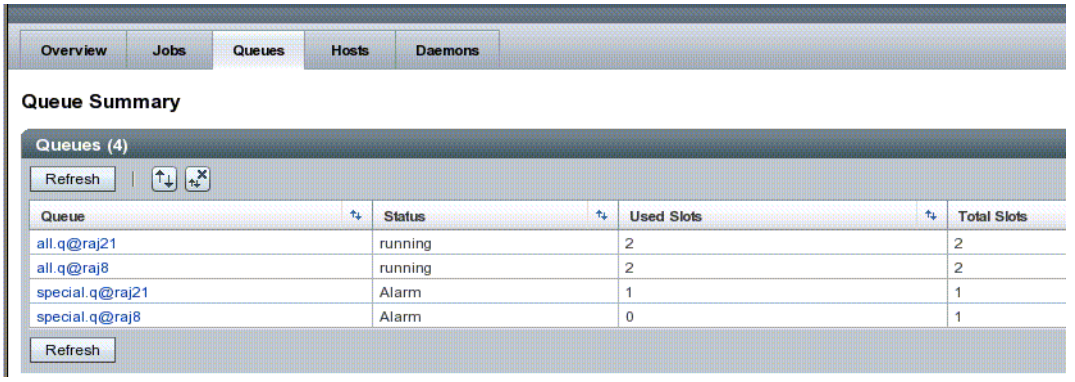


FIGURE 6–1 Queue Summary Page

The Queue Summary page show the following information:

- **Queue** – The queue name. To see more detailed information on any queue, click the queue instance name.
- **Status** – Describes whether this queue instance is running, suspended (manually or automatically in the case of an error), or waiting for a required resource to become available or a condition be met. If a queue instance is suspended or waiting, you may want to see more queue details.
- **Used Slots** – The number of total slots this queue instance is consuming
- **Total Slots** – The number of slots defined for this queue instance. Slots are the maximum number of jobs that a queue can run simultaneously.

Note – You do not prioritize jobs using an N1GE queue. You define priorities using the extended policy system of the Sun N1 Grid Engine software. For information on job priorities, see the `sge_priority(5)` man page and *Scheduler Policies for Job Prioritization in the Sun N1 Grid Engine 6 System* (`www.sun.com/blueprints/1005/819-4325.html`).

For information on cluster queues, see the Monitoring and Controlling Queues section in the *N1GE 6 User's Guide* and the `qmon` man page. For more information on queue states, see the Queue Alerts page.

Viewing Complete Queue Information

The Queue Details page contains complete information for the queue instance that you selected on the Queue Summary page.

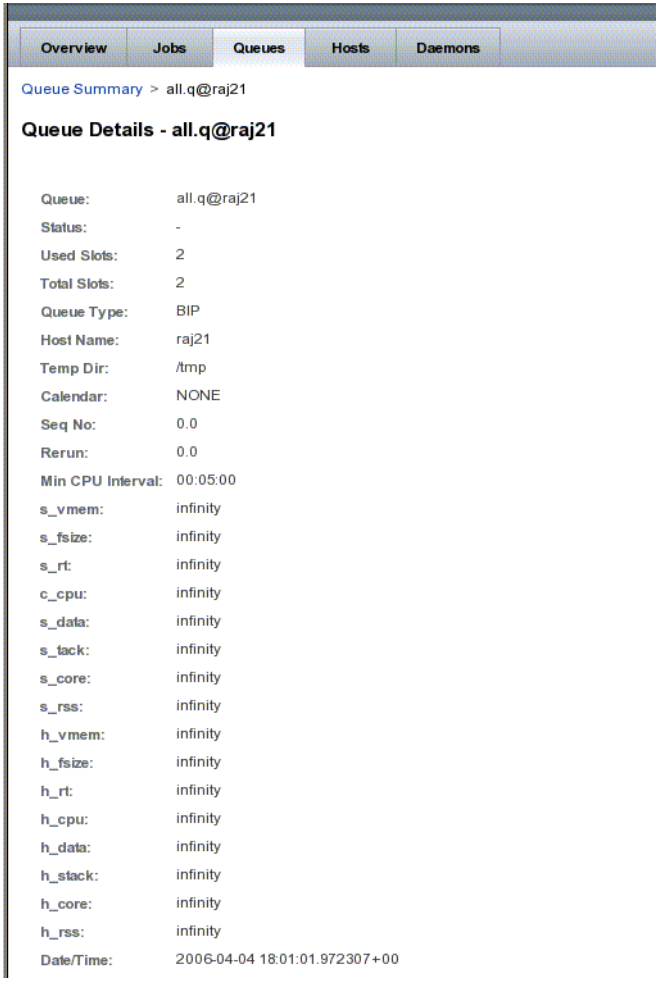


FIGURE 6–2 Queue Details Page

The Queue Details page shows the following information:

- **Queue** – The queue instance name.
- **Status** – Describes whether this queue instance is running, suspended (manually or automatically in the case of an error), or waiting for a required resource to become available or condition be met. See the Queue Alerts page for more information.
- **Used Slots** – The number concurrently executing in the queue instance. The type is number
- **Total Slots** – The maximum number of concurrently executing jobs allowed in the queue instance. The type is number.

- **Queue Type** – The type of queue. Currently one of batch, interactive, parallel, or checkpointing or any combination in a comma separated list. The type is string; the default is batch interactive parallel.
- **Hostname** – The fully qualified host name of the node (type string; template default: *host.dom.dom.dom*).
- **Calendar** – Specifies the valid calendar for this queue instance or contains NONE (the default). A calendar defines the availability of a queue instance depending on time of day, week, and year. Refer to the `calendar_conf` man page for details on the N1 Grid Engine calendar facility.
- **Seq No** – The sequence number. This parameter combined with the host's load situation specifies this queue's position within the suitable queue scheduling order. A job is dispatched under consideration of the *queue_sort_method* (see the `sched_conf` man page). Regardless of the *queue_sort_method* setting, `qstat` reports queue information in the order defined by the value of the *seq_no*. Set this parameter to a monotonically increasing sequence. The type is number and the default is 0.
- **Rerun** – Defines a default behavior for jobs which are aborted by system crashes or manual violent shutdown (using `kill`) of the complete Sun N1 Grid Engine system on the queue host (including the `sge_shepherd` of the jobs and their process hierarchy). As soon as the `sge_execd` daemon restarts and detects that a job has been aborted for such reasons, it can be restarted if the jobs are restartable. A job may not be restartable, for example, if it updates databases (first reads then writes to the same record of a database/file) because the cancellation of the job may have left the database in an inconsistent state. The type of this parameter is Boolean, so you can specify either TRUE or FALSE. The default is FALSE, that is, do not restart jobs automatically. To overrule the default behavior for the jobs in the queue, the owner of the job can use the `-r` option of the `qsub` command.
- **Min Cpu Interval** – The time between two automatic checkpoints in case of transparently checkpointing jobs. The maximum of the time requested by the user (using `qsub`) and the time defined by the queue configuration is used as checkpoint interval. The checkpoint files may be quite large and writing them to the file system may become expensive. So, users and administrators are advised to choose sufficiently large time intervals. The type of *min_cpu_interval* is time and the default is 5 minutes which usually is suitable for test purposes only.
- **s_rt** (soft real time) and **h_rt** (hard real time) resource limit parameters define the real time (also called elapsed or wall clock time) passed since the start of the job. If *h_rt* is exceeded by a job running in the queue, it is stopped using the `SIGKILL` signal (see the `kill` command. If the *s_rt* is exceeded, the job is first warned by the `SIGUSR1` signal which can be caught by the job and finally stopped after the notification time defined in the queue configuration *notify* parameter has passed.
- **s_cpu** (soft cpu) and **h_cpu** (hard cpu — the per-job CPU time limit in seconds) resource limit parameters impose a limit on the amount of combined CPU time consumed by all the processes in the job. If *h_cpu* is exceeded by a job running in the queue, it is stopped by a `SIGKILL` signal (see the `kill` command). If *s_cpu* is exceeded, the job is sent a `SIGXCPU` signal which can be caught by the job. To warn a job so it can exit gracefully before it is killed, set the *s_cpu* limit to a lower value than *h_cpu*. For parallel processes, the limit is applied per slot. The limit is multiplied by the number of slots being used by the job before being applied.

- **s_vmem** (soft virtual memory) – The same as *s_data*. If both are set the minimum is used and **h_vmem** (hard virtual memory — This is the same as *h_data*. If both are set the minimum is used and resource limit parameters impose a limit on the amount of combined virtual memory consumed by all the processes in the job. If *h_vmem* is exceeded by a job running in the queue, it is topped by a `SIGKILL` signal. If *s_vmem* is exceeded, the job is sent a `SIGXCPU` signal which can be caught by the job. To warn a job so it can exit gracefully before it is killed, Set the *s_vmem* limit to a lower value than *h_vmem*. For parallel processes, the limit is applied per slot. The limit is multiplied by the number of slots being used by the job before being applied.
- **s_core** (soft core) - The per-process maximum core file size in bytes
- **s_data** (soft data) – The per-process maximum memory limit in bytes.
- **h_data** (hard data) – The per-job maximum memory limit in bytes.
- **h_fsize** (hard file size) – The total number of disk blocks that this job can create.

These parameters specify per job soft and hard resource limits as implemented by the `setrlimit(2)` system call. By default, each limit field is set to infinity which means *RLIM_INFINITY* as described in the `setrlimit` man page. The value type for the CPU-time limits *s_cpu* and *h_cpu* is time. The value type for the other limits is memory.

Note – Not all systems support the `setrlimit` command. Also, *s_vmem* and *h_vmem* are only available on systems supporting *RLIMIT_VMEM* (see the `setrlimit(2)` man page on system hosting the queue).

For more information, see the `complex` man page.

CHAPTER 7

Working With N1 Grid Engine Hosts

This chapter describes how you can see information about the hosts constituting a grid. Information is available about all the hosts performance as well detailed information about any particular host.

Viewing Host Resources

You use the Hosts page both as a means of checking on how efficiently the hosts's resources are being used and as a way to access more details about the host itself.

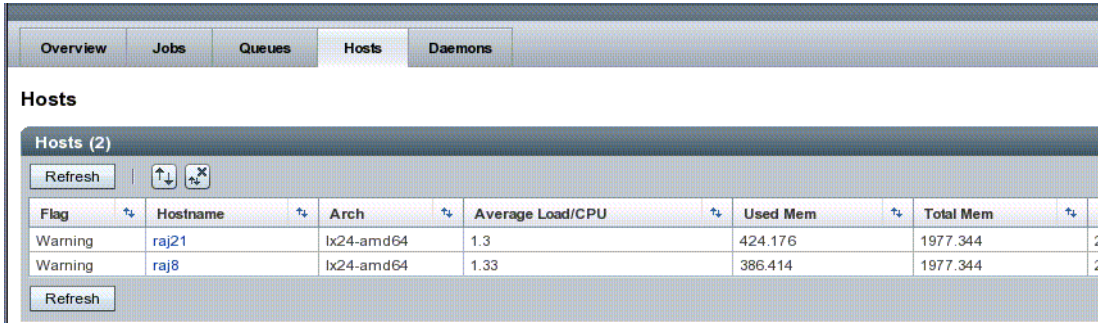


FIGURE 7–1 Hosts Page

The fields in the Host table have the following meaning:

- **Hostname** – The name you assign to this host. Clicking the Hostname displays the very detailed Host Details page.
- **Arch** – The host's processor architecture like win32-x86 or sol-sparc64. For a complete list of supported architectures, see the Host Details page.
- **Average Load/CPU** – Shows how efficiently the Host's CPU is being used. This parameter can be any positive decimal number but is usually between zero and 2 or 3. Ideally, this number should be close to 1. A smaller number could mean the host is under-utilized, and a larger number could mean that the host is overutilized. The ideal value depends on the workload that is being run. Only the local administrator can really know the implications of the workload.

- **Used Mem** – The percentage of total memory currently being used to execute jobs. If this value is too close to the total memory, then the host is possibly in trouble. However, if the workloads are tuned to fit in the server, then it could be perfectly fine that the used memory is just under the total memory. In fact, this is tunable: you can set the value at which the difference between these two parameters triggers an alarm. So, in one case, a difference of less than 100 MB triggers a warning, while in another case the value could be set at 25 MB.
- **Total Mem** – The total amount of memory on this host.
- **Free Swap** – The amount of free swap space left on this host measured in MBs. In a well-architected grid, the free swap space should never drop very far below its initial value. It is possible that temporary drops in this value can be tolerated, again, depending on how the grid is architected. If this value goes close to zero, the host is in danger of failing completely.

Viewing Complete Host Information

The Host Details page contains detailed information about the host system that is helping to execute a job and is hosting a queue.

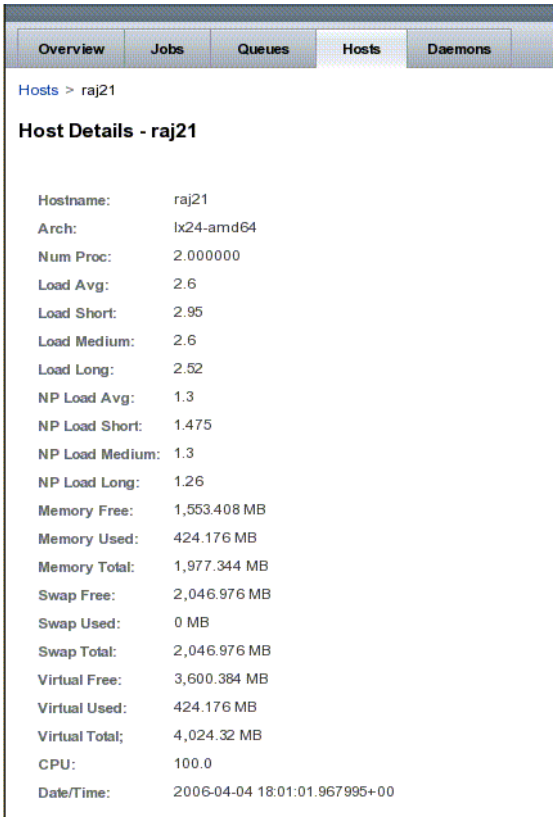


FIGURE 7–2 Host Details Page

The Host Details page contains the following information:

- **Hostname** – The name assigned to this host.
- **Arch** – An architecture string compiled into the `cod_execd` describing the operating system architecture for which the execd is targeted. Possible values are:
  - sol-sparc for Sun Solaris (Sparc) 7 and higher, 32–bit kernel
  - sol-sparc64 for Sun Solaris (Sparc) 7 and higher, 64–bit kerne
  - sol-x86 for Sun Solaris (x86) 8 and higher
  - x24-amd64 for Linux 2.4.*x* (AMD64) glibc 2.2+ based
  - lx24-x86 for Linux 2.4.*x* (x86) glibc 2.2+ based
  - win-x86 for MS-Windows NT

Note – An `sge_execd` daemon for a particular architecture may run on multiple OS versions but the architecture string does not reveal this level of detail.

- **Num Proc** – The number of processors provided by the execution host. The host is in this case defined by a single Internet address, for example, rack-mounted multihost systems are counted as a cluster rather than a single multiheaded machine.
- **Load Avg** – The same as Load Medium.
- **Load Short** – The short time average OS run queue length. This value is the first of the values triple reported by the `uptime` command. Many implementations provide a one minute average.
- **Load Medium** – The medium time average OS run queue length. This value is the second of the values triple reported by the `uptime` command. Many implementations provide a 5 minute average with this value.
- **Load Long** — The long time average OS run queue length. This value is the third of the values triple reported by the `uptime` command. Many implementations provide a 10 or 15 minutes.
- **NP Load Avg** – The same as Load Medium.
- **NP Load Short** – The same as Load Short but divided by the number of processors. This value allows you to compare the load of single and multiheaded hosts.
- **NP Load Medium** – The same as Load Medium but divided by the number of processors. This value allows you to compare the load of single and multiheaded hosts.
- **NP Load Long** – The same as Load Long but divided by the number of processors. This value allows you to compare the load of single and multiheaded hosts.
- **Memory Free** – The amount of free memory.
- **Memory Used** – The amount of used memory.
- **Memory Total** – The total amount of memory (free plus used).
- **Swap Free** – The amount of free swap memory.
- **Swap Used** – The amount of used swap space.
- **Swap Total** – The total amount of swap space (free plus used).
- **Virtual Free** – The sum of Mem Free and Swap Free.
- **Virtual Used** – The sum of Mem Used and SwapUsed.
- **Virtual Total** – The sum of Mem Total and Swap Total.
- **CPU** – The percentage of CPU of cpu busy time when the data was gathered.
- **Date/Time** – The timestamp for when the data was gathered.

For more information about configuring execution host parameters in the *Configuring Execution Hosts With QMON* chapter of the *N1 Grid Engine Administrators Guide* on `docs.sun.com`.

CHAPTER 8

Troubleshooting N1 Grid Engine

This chapter tells you how to use the various alerts and the N1 Grid Engine daemon logs to troubleshoot a grid.

Using N1 Grid Engine Daemon Logs

You use the N1 Grid Engine Daemon Logs page to see a historical view of all the messages logged by the various N1 Grid Engine daemons. To see the log file for a particular host, click its host name. To see the log files for the system hosting the queue, click on a name in the QMASTER column.

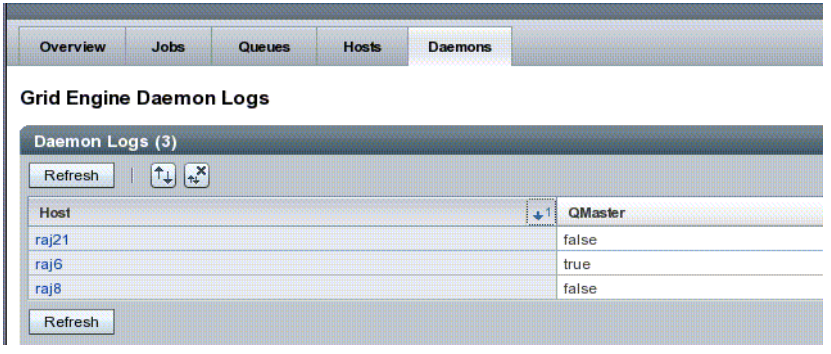


FIGURE 8–1 Daemon Logs List Page

The log file for a particular host contains fields for a Flag, a Time Stamp, and a Message. The flag tells you what kind of message was logged. Flags exist for the following message types:

- **N** (notice) – for informational purposes
- **I** (info) – for informational purposes
- **W** (warning)
- **E** (error) – An error condition has been detected

- **C** (critical) – Which can lead to a program abort

Use the *loglevel* parameter in the cluster configuration to specify on a global basis or a local basis what message types you want to log.

Troubleshooting Queues

You can use the information on the Queue Alerts page to troubleshoot any queue problems. You access this page from the Alerts table on the Overview page. Queue alerts are generated when the Queue Resource Limit parameters defined using the `queue_conf` command are exceeded.

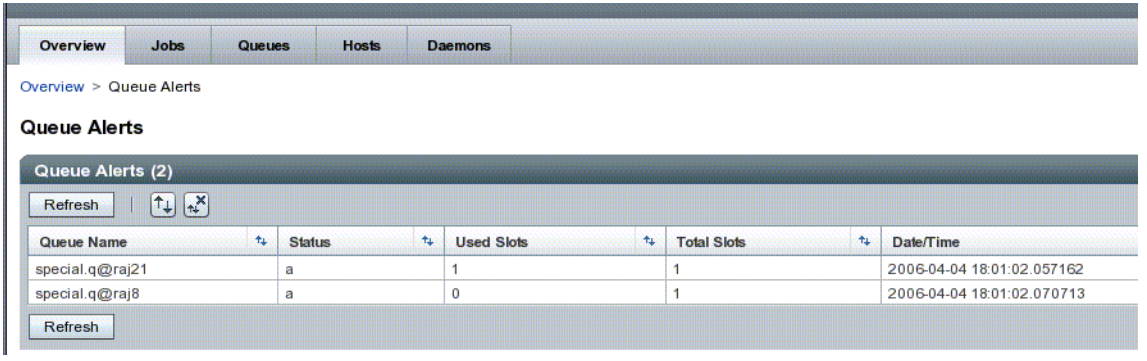


FIGURE 8–2 Queue Alerts List Page

The three types of queue alerts are:

- **Warnings** – When resource limits are exceeded, a warning can be generated before a queue is disabled.
- **Errors** – Errors are generated when a queue makes an invalid request.
- **Disabled** – After receiving a set number of warnings, queues are aborted after the notification time defined in the queue configuration parameter notify has passed.

The Queue states are:

- **a** (alarm) – At least one of the load thresholds defined in the *load_thresholds* list of the queue configuration is currently exceeded. This state prevents N1GE from scheduling further jobs to that queue. For more information, see the `queue_conf`) man page.
- **A** (Alarm) – At least one of the suspend thresholds of the queue is currently exceeded. This state causes jobs running in that queue to be successively suspended until no threshold is violated. For more information, see the `queue_conf` man page.
- **c** (configuration ambiguous) – The queue instance configuration specified using `sge_conf` is ambiguous. The state resolves when the configuration becomes unambiguous again. This state prevents you from scheduling further jobs to that queue instance. You can find detailed reasons why a queue instance entered this state in the `sge_qmaster` messages file. You can also see the

reasons using the `qstat` command with `-explain`. For queue instances in this state, the cluster queue's default settings are used for the ambiguous attribute.

- **C** (Calendar suspended) – The queue has been disabled or suspended automatically using the N1GE calendar facility. See the `calendar_conf` man page for more information.
- **d** (disabled) – This setting is assigned to queues and released using the `qmod` command. Suspending a queue will suspend all jobs executing in that queue.
- **D** (Disabled) – The queue has been disabled or suspended automatically using the N1GE calendar facility. See the `calendar_conf` man page for more information.
- **E** (Error) – This setting appears when the N1GE daemon (`sge_execd`) on that host was unable to locate the `sge_shepherd` executable on that host in order to start a job. Check that daemon's error log for information how to resolve the problem. Enable the queue afterwards using the `qmod` command with the `-c` option.
- **o** (orphaned) – The current cluster queue's configuration and host group configuration no longer needs this queue instance. The queue instance is kept because unfinished jobs are still associated with it. The orphaned state prevents you from scheduling further jobs to that queue instance. It disappears from `qstat` output when these jobs finish. To help resolve an orphaned queue instance associated with a job, use the `qdel` command. You can revive an orphaned queue instance by changing the cluster queue configuration so that the configuration covers that queue instance.
- **s** (suspended) – Assigned to queues and released using the `qmod` command. Suspending a queue suspends all jobs executing in that queue.
- **S** (Subordinate) – The queue has been suspend due to subordination to another queue. See `queue_conf` for details. When suspending a queue, regardless of the cause, all jobs executing in that queue are suspended too.
- **u** (unknown) – The corresponding `sge_execd(8)` cannot be contacted.

Troubleshooting Hosts

You can see potential host problems from the Host Alerts page. This page is available from the Alerts table on the Overview page.

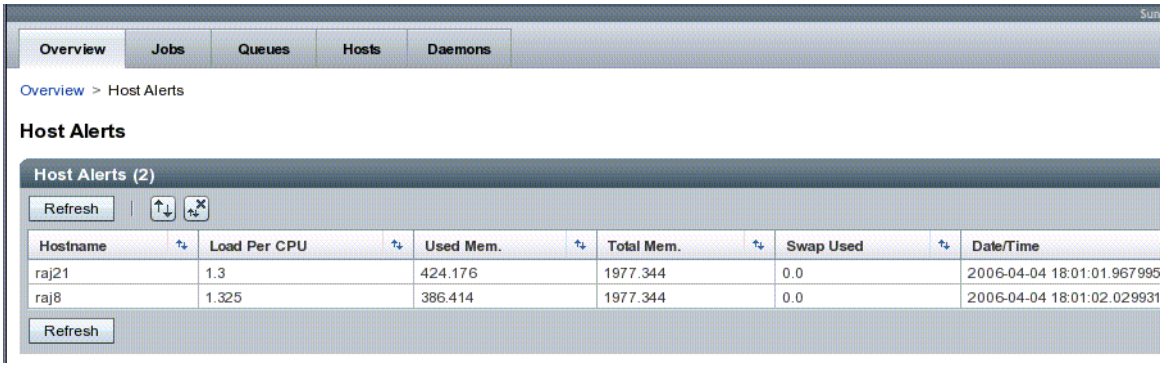


FIGURE 8–3 Hosts Alerts List Page

The following host alert parameters can all be alarmed so that if they pass a specified threshhold, an alert will be generated and appear on the Overview Alerts table.

- **Load Per CPU** – Shows how efficiently the Host's CPU is being used. This parameter can be any positive decimal number but is usually between zero and 2 or 3. Ideally, this number should be close to 1. A smaller number could mean the host is under utilized, and a larger number could mean the host is overutilized. The ideal value depends on the workload that is being run. Only the local administrator can really know the implications of the workload.
- **Used Mem.** – The percentage of total memory currently being used to execute jobs. If the used memory is too close to the total memory, then the host could be in trouble. However, if the workloads are tuned to fit in the server, then it could be perfectly fine that the used memory is just under the total memory. In fact, this is tunable. You can set the value at which the difference between these two parameters triggers an alarm. So, in one case, a difference of less than 100 MB triggers a warning, while in another case it could be at 25 MB.
- **Total Mem.** – The total amount of memory on this host.
- **Swap Used** – The amount of free swap space left on this host measured in MBs. In a well-architected grid, the free swap space should never drop very far below its initial value. It is possible that temporary drops in this value can be tolerated depending on how the grid is architected. If this value goes close to zero, then the host is in danger of failing completely.
- **Date/Time** – The timestamp for when the alert was generated.

Troubleshooting Jobs

You can view potential job problems from the Job Alerts page. This page is available from the Alerts table on the Overview page. The Pending Time and Deadline job alert parameters can be alarmed so that if the values pass a specified threshold, an alert will be generated and appear on the Overview Alerts table.

FIGURE 8–4 Job Alerts List Page

The Job Alerts page shows the following information:

- **Job ID** – The unique identifier for the job. Clicking on the Job ID brings you to the Job Details page.
- **Task** – The currently executing task. Some jobs consist of a single task (in which case, the task ID is always 1.) However, parallel jobs and array jobs each consist of more than one task. The tasks are usually numbered in ascending order starting with 1. Depending upon how the job was submitted, sometimes the numbers might skip as in 1,3,5. On running jobs, each task runs distinctly and so has its own configuration information, environment, and trace. For details about the task, click the task number to display the Task Details page.
- **Job Name** – The name assigned to the job.
- **Pending time** – How long the job has been waiting to be assigned to a queue.
- **Deadline** – The time specified by which a job must start or generate an alarm.

See the `qstat` man page for more information about alarms and thresh holds.

Index

U

V