N1 Grid Engine 6 Release Notes

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

Part No: 817–5678–10
June 2004

040609@9061

Contents

N1™

Grid Engine 6 Software Release Notes

Read this document carefully before you install the accompanying software. This document includes the following main sections:

Accessing Documentation

The distribution CD includes full documentation for a networked set of computer hosts that run N1™ Grid Engine 6 software (*grid engine system*):

- `N1GE6_User_Guide.pdf` – *N1 Grid Engine 6 User's Guide*
- `N1GE6_Administration_Guide.pdf` – *N1 Grid Engine 6 Administration Guide*
- `N1GE6_Installation_Guide.pdf` – *N1 Grid Engine 6 Installation Guide*

You can access these files directly from the CD, in either PDF or HTML formats. The files are in the *cdmountpoint*`/N1_Grid_Engine_6/Docs` directory.

Contents of This Software Package

The N1 Grid Engine 6 software (*grid engine software*) distribution is made up of the following components:

- The grid engine software binary packages, including all daemons, client programs, and libraries. You must load and install one binary package for each of the distinct operating system architectures you intend to use.
- The grid engine software common package, containing install scripts, and other architecture-independent utilities.
- The grid engine software documentation package, containing these release notes, the installation guide, user guide and administration guide in PDF and HTML formats.
- The optional Accounting and Reporting Console (ARCo) software, which is made up of three separate packages:
  - The Sun Web Console package. You must select the package appropriate for the operating system architecture on which you plan to run the web console server.
  - The `dbwriter` package, written in Java and therefore available in only one version.
  - The ARCo module package, usable across different supported architectures.

Note – In addition to installing these three packages, you must set up a PostgreSQL or an Oracle database server in order to operate ARCo. PostgreSQL and Oracle are not included in the N1 Grid Engine 6 software distribution. For more information, see Chapter 8, "Installing the Accounting and Reporting Console," in *N1 Grid Engine 6 Installation Guide*.

The N1 Grid Engine 6 software distribution kit contains the following top-level directory hierarchy:

- `3rd_party` – Contains information about freeware, public domain, and public license software used
- `bin` – Grid engine software executables
- `catman` – Online manual pages organized into admin and user commands
- `ckpt` – Sample checkpointing configurations
- `dbwriter` – DbWriter software used by the accounting and reporting console
- `doc` – Documentation in PDF and HTML formats
- `examples` – Sample script files, configuration files, and application programs
- `dbwriter` – DbWriter software used by the accounting and reporting console
- `include` – DRMAA header file
- `lib` – Required shared libraries
- `man` – Online manual pages in `nroff` format
- `mpi` – A sample parallel environment interface for the MPI message-passing system
- `pvm` – A sample parallel environment interface for the PVM message-passing system

- `qmon` – Pixmaps, resource, and help files for `QMON`, the graphical user interface
- `reporting` – Accounting and reporting console software
- `util` – Some utility shell procedures used for installation tasks and some template grid engine system shutdown and boot scripts
- `utilbin` – Some utility programs that are mainly required during the installation

Installing the N1 Grid Engine 6 Software

See `N1GE6_Installation_Guide.pdf`, the *N1 Grid Engine 6 Installation Guide* that is included on the distribution CD.

New Features in N1 Grid Engine 6 Software

N1 Grid Engine 6 provides the following new features.

Accounting and Reporting Console (ARCo)

The optional ARCo enables you to gather live accounting and reporting data from a grid engine system and store the data in a standard SQL database. ARCo also provides a web-based tool for generating information queries on that database and for retrieving the results in tabular or graphical form. ARCo enables you to store queries for later use, to run predefined queries, and to run queries in batch mode, for example, overnight.

For details, see Chapter 5, "Accounting and Reporting," in *N1 Grid Engine 6 User's Guide*, and Chapter 8, "Installing the Accounting and Reporting Console," in *N1 Grid Engine 6 Installation Guide*.

Resource Reservation

The grid engine system scheduler supports a highly flexible resource reservation scheme. Jobs can reserve resources depending on criteria such as resource requirements, priority, waiting time, resource sharing entitlements, and so forth. The

scheduler enforces reservations in such a way that jobs with highest urgency receive the earliest possible resource assignment. Resource reservation completely avoids well-known problems such as job starvation.

The grid engine system resource reservation scheme enables you to determine the urgency of a job based on such different factors as resource requirements, priority, waiting time, resource sharing entitlements, and so forth. With respect to resource requirements, a job's importance can be defined on a per resource basis for arbitrary resources, as well as for administrator-defined resources such as third party licenses or network bandwidth. Reservations can be assigned across the full hierarchy of grid engine system resource containers: global, host, or queue.

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

Cluster Queues

N1 Grid Engine 6 software provides a new administrative concept for managing queues. It enables easier administration while maintaining the flexibility of the Sun Grid Engine 5.3 queue concept.

A *cluster queue* can extend across multiple hosts. Those hosts can be specified as a list of individual hosts, as a host group, or as a list of individual hosts and host groups. By adding a host to a cluster queue, the host receives an instance of that cluster queue. A *queue instance* corresponds to a queue in Sun Grid Engine 5.3.

When you modify a cluster queue, all of its queue instances are modified simultaneously. Even within a single cluster queue, you can specify differences in the configuration of queue instances, depending on individual hosts or host groups. Therefore, a typical N1 Grid Engine 6 software setup will have only a few cluster queues, and the queue instances controlled by those cluster queues remain largely in the background.

For further details, see the `queue_conf`(5) man page.

DRMAA

N1 Grid Engine 6 software includes a standard-compliant implementation of the Distributed Resource Management Application API (DRMAA), version 1.0. DRMAA 1.0 is a standard draft for review at Global Grid Forum. It provides a standard API for the integration of applications with Distributed Resource Management System, such as N1 Grid Engine 6 software, with external applications like ISV codes or graphical interfaces. Major functions provided by DRMAA include job submission, job monitoring, and job control. N1 Grid Engine 6 software includes an implementation for the C-language binding of DRMAA. Details are available in the `drmaa_*`(3) man pages and on the DRMAA home page http://www.drmaa.org/.

Scalability

N1 Grid Engine 6 software implements a number of architectural changes from previous releases in order to support increased scalability:

- Spooling of persistent status information for the `sge_qmaster` can now be done using the high-performance Berkeley DB database instead of the previous file-based spooling.
- The `sge_qmaster` is multithreaded to support concurrent execution of tasks on multi-CPU systems.
- The Sun Grid Engine 5.3 communication system has been replaced. The communication system is now multithreaded and no longer requires a separate communication daemon.

Scheduler Enhancements

Different scheduling profiles can be selected for setups ranging from high throughput and low scheduling overhead to full policy control. The setups can be selected during the `sge_qmaster` installation procedure. In addition, a series of enhancements has improved scheduler performance greatly.

Automated Installation and Backup

The N1 Grid Engine 6 software installation procedure can be completely automated to facilitate installation on large numbers of execution hosts, frequently recurring reinstallation of hosts, or integration of the installation process into system management frameworks. For more information, see the file `doc/README-Autoinstall.txt`.

N1 Grid Engine 6 software also includes an automatic backup script that backs up all cluster configuration files.

`qping` Utility

A new `qping` utility enables you to query the status of the `sge_qmaster` and `sge_execd` daemons.

Starting Binaries Directly

The `qsub` command now supports the `-shell {y | n}` option, which is used with the `-b y` option, to start a submitted binary directly without an intermediate shell.

Resource Requests for Individual `make` Rules

In dynamic allocation mode, the `qmake` command can now specify resource requests for individual `make` rules.

Grid Engine System Binary Directory

The environment variable `SGE_BINARY_PATH` is set in the job environment. This variable points to the directory where the grid engine system binaries are installed.

Known Limitations and Workarounds

The following sections contain information about product irregularities discovered during testing, but too late to fix or document.

Known Limitations of N1 Grid Engine 6 Software

This N1 Grid Engine 6 software release has the following limitations:

- When using the automatic installation procedure, the `sgeadmin` user is not considered.
- The backup and restore function is not implemented for classic spooling and Berkeley DB RPC server installations.
- The stack size for `sge_qmaster` should be set to 16 MBytes. `sge_qmaster` might not run with the default values for stack size on the following architectures: IBM AIX 4.3 and 5.1, and HP UX 11.
- You should set a high file descriptor limit in the kernel configuration on hosts that are designated to run the `sge_qmaster` daemon. You might want to set a high file descriptor limit on the shadow master hosts as well. A large number of available file descriptors enables the communication system to keep connections open instead of having to constantly close and reopen them. If you have many execution hosts, a high file descriptor limit significantly improves performance. Set the file descriptor limit to a number that is higher than the number of intended execution hosts. You should also make room for concurrent client requests, in particular for jobs submitted with `qsub -sync` or when you are running DRMAA sessions that maintain a steady communication connection with the master daemon. Refer to you operating system documentation for information about how to set the file descriptor limit.
- The number of concurrent dynamic event clients is limited by the number of file descriptors. The default is 99. Dynamic event clients are jobs submitted with the `qsub -sync` command and a DRMAA session. You can limit the number of

dynamic event clients with the `qmaster_params` global cluster configuration setting. Set this parameter to `MAX_DYN_EC=`*n*. See the `sge_conf`(5) man page for more information.

- The ARCo module is available only for the Solaris Sparc, Solaris x86, and Linux x86 platforms.
- ARCo currently supports only the following database servers: PostgreSQL 7.3.2, 7.4.1, 7.4.2, and Oracle 9i. An integration with MySQL will be provided once MySQL supports views.
- Only a limited set of predefined queries is currently shipped with ARCo. Later releases will include more comprehensive sets of predefined queries.
- Error diagnosis is known to be challenging in a framework combined of the ARCo module, the Sun Web Console, and the database server. Appropriate diagnostic messages are not always displayed. Nor do the messages always indicate the real cause of a problem. Also, the integrated ARCo framework currently shows unexpected effects in case of error conditions.
- Jobs requesting the amount `INFINITY` for resources are not handled correctly with respect to resource reservation. `INFINITY` might be requested by default in case no explicit request for a certain resource has been made. Therefore it is important to request that all resources be explicitly taken into account for resource reservation.
- Resource reservation currently takes only pending jobs into account. Consequently, jobs that are in a hold state due to the submit options `-a` *time* and `-hold_jid` *joblist*, and are thus not pending, do not get reservations. Such jobs are treated as if the `-R n` submit option were specified for them.
- The scheduler does not look ahead to consider queue calendar state transitions. As a result, it can happen that jobs get dispatched to queues that soon become unavailable due to a suspended or disabled calendar. This behavior can also be caused if not enough information about the job's run length is provided with the `qsub -l h_rt` and `qsub -l s_rt` commands. For similar reasons, queues that will soon become available after a calendar is suspended or disabled are not taken in account when resource reservation is enabled with the `max_reservation` parameter of `sched_conf`(5). As a result, jobs of lower priority can get global resources or host-based resources that otherwise would be reserved for high priority jobs.
- Berkeley DB requires that the database files reside on the local disk. If the `sge_qmaster` cannot be run on the file server intended to store the spooling data (for example, if you want to use the shadow master facility), a Berkeley DB RPC server can be used. The RPC server runs on the file server and connects with the Berkeley DB `sge_qmaster` instance. However, Berkeley DB's RPC server uses an insecure protocol for this communication and thus presents a security problem. Do *not* use the RPC server method if you are concerned about security at your site. Use `sge_qmaster` local disks for spooling instead and, for fail-over, a high availability solution such as Sun Cluster, which maintains host local file access in the fail-over case.

- In cases where many active jobs are passed to the grid engine system through DRMAA, the call `drmaa_control(DRMAA_CONTROL_TERMINATE)` can cause a kind of a DoS attack to `sge_qmaster`. Workaround: Don't leave too many DRMAA jobs active at the same time.
- Busy `QMON` with large array task numbers. If large array task numbers are used, you should use "compact job array display" in the `QMON` Job Control dialog box customization. Otherwise the `QMON` GUI will cause high CPU load and show poor performance.
- The automatic installation option is known to provide limited diagnostic information in case of failure. If the installation process aborts, check for the presence and the contents of an installation log file `/tmp/install.`*pid*.
- `qrsh` jobs are not scheduled to queues that are interactive only.
- On IBM AIX 4.3 and 5.1, HP/UX 11, and SGI IRIX 6.5 systems, two different binaries are provided for `sge_qmaster`, `spooldefaults`, and `spoolinit`. One of these binaries is for the Berkeley DB spooling method, the other binary is for the classic spooling method. The names of these binaries are `binary.spool_db` and `binary.spool_classic`.

  To change to the desired spooling method, modify three symbolic links before you install the master host. Do the following:

  ```
  # cd sge-root/bin/arch
  # rm sge_qmaster
  # ln -s sge_qmaster.spool_classic sge_qmaster

  # cd sge-root/utilbin/arch
  # rm spooldefaults spoolinit
  # ln -s spooldefaults.spool_classic spooldefaults
  #ln -s spoolinit.spool_classic spoolinit
  ```

- Gathering of online usage statistics for running jobs, and dynamic reprioritization for such jobs, do not work for the following operating systems:
  - IBM AIX
  - HP/UX
  - Mac OS X

  For a workaround, see the `sge_conf`(5) man page for information about how to adjust the execution host parameters `ACCT_RESERVED_USAGE` and `SHARETREE_RESERVED_USAGE`.