

Grid Engine Users Guide



6.1.1 Edition



Grid Engine Users Guide :

6.1.1 Edition

Published Apr 14 2014

Copyright © 2014 University of California and Scalable Systems

This document is subject to the Rocks® License (see Appendix A: Rocks Copyright).

Table of Contents

Preface	v
1. Overview	1
2. Installing	2
2.1. On a New Server	2
2.2. On an Existing Server.....	2
3. Using	3
3.1. How to use SGE	3
3.2. Submitting Batch Jobs to SGE	3
3.3. Monitoring SGE Jobs	4
3.4. Managing SGE queues.....	5
A. Rocks® Copyright	8
B. Third Party Copyrights and Licenses	9
B.1. Sun Grid Engine	9

List of Tables

1-1. Summary.....	1
1-2. Compatibility.....	1

Preface

The SGE Roll installs and configures the SUN Grid Engine scheduler.

Please visit the SGE site¹ to learn more about their release and the individual software components.

Notes

1. <http://gridengine.sunsource.net/>

Chapter 1. Overview

Table 1-1. Summary

Name	sgc
Version	6.1.1
Maintained By	Rocks Group
Architecture	i386, x86_64
Compatible with Rocks®	6.1.1

The sgc roll has the following requirements of other rolls. Compatibility with all known rolls is assured, and all known conflicts are listed. There is no assurance of compatibility with third-party rolls.

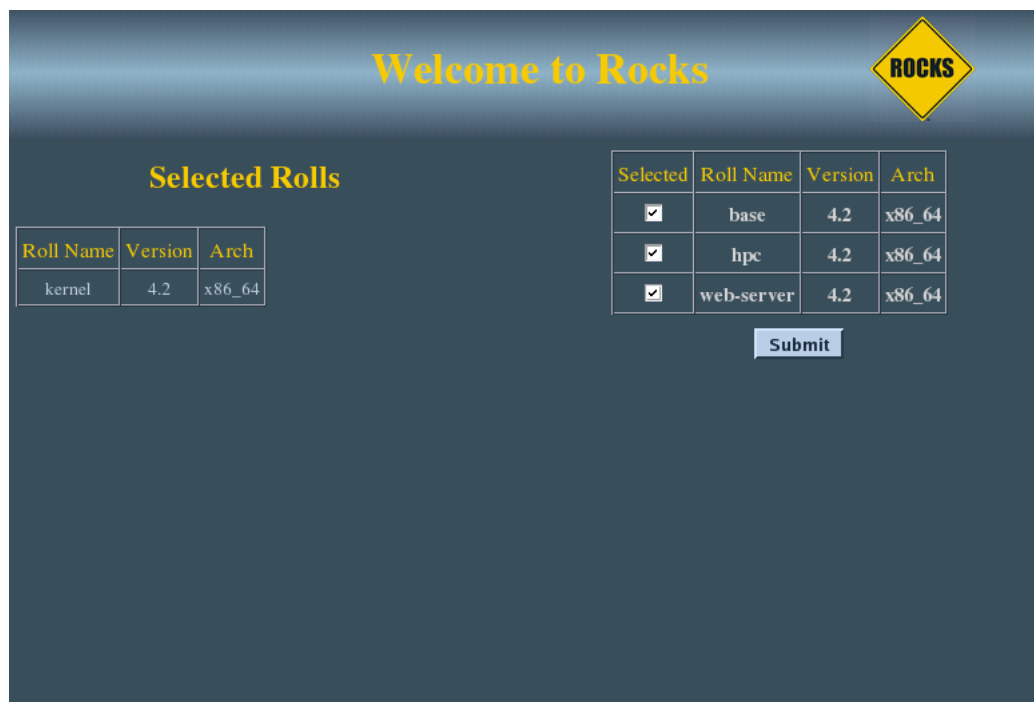
Table 1-2. Compatibility

Requires	Conflicts
Base	
Kernel	
OS	
	PBS

Chapter 2. Installing

2.1. On a New Server

The sge roll should be installed during the initial installation of your server (or cluster). This procedure is documented in section 1.2 of the Rocks® usersguide. You should select the sge roll from the list of available rolls when you see a screen that is similar to the one below.



2.2. On an Existing Server

The sge Roll may not be installed on an already existing server. The only supported method of installation is to install the Roll at the time of the server installation.

Chapter 3. Using

3.1. How to use SGE

This section tells you how to get started using Sun Grid Engine (SGE). SGE is a distributed resource management software and it allows the resources within the cluster (cpu time, software, licenses etc) to be utilized effectively. Also, the SGE Roll sets up Sun Grid Engine such that NFS is not needed for its operation. This provides a more scalable setup but it does mean that we will lose the high availability benefits that a SGE with NFS setup offers. Another thing that the Roll does is that generic queues are setup automatically the moment new nodes are being integrated within the Rocks cluster and booted up.

3.2. Submitting Batch Jobs to SGE

Batch jobs are submitted to SGE via scripts. Here is an example of a serial job script, `sleep.sh`¹. It basically executes the `sleep` command.

```
[sysadm1@frontend-0 sysadm1]$ cat sleep.sh
#!/bin/bash
#
#$ -cwd
#$ -j y
#$ -S /bin/bash
#
date
sleep 10
date
```



Entries which start with `#$` will be treated as SGE options.

- `-cwd` means to execute the job for the current working directory.
- `-j y` means to merge the standard error stream into the standard output stream instead of having two separate error and output streams.
- `-S /bin/bash` specifies the interpreting shell for this job to be the Bash shell.

To submit this serial job script, you should use the `qsub` command.

```
[sysadm1@frontend-0 sysadm1]$ qsub sleep.sh
your job 16 ("sleep.sh") has been submitted
```

Next, we'll submit a parallel job. First, let's get and compile a test MPI program. As a non-root user, execute:

```
$ cd $HOME
$ mkdir test
$ cd test
$ cp /opt/mpi-tests/src/*.c .
$ cp /opt/mpi-tests/src/Makefile .
$ make
```


Now we'll create an SGE submission script for *mpi-ring*. The program *mpi-ring* sends a 1 MB message in a ring between all the processes of an MPI job. Process 0 sends a 1 MB message to process 1, then process 1 send a 1 MB message to process 2, etc. Create a file named `$HOME/test/mpi-ring.qsub` and put the following in it:

```
#!/bin/bash
#
#$ -cwd
#$ -j y
#$ -S /bin/bash
#

/opt/openmpi/bin/mpirun $HOME/test/mpi-ring
```

The command to submit a MPI parallel job script is similar to submitting a serial job script but you will need to use the `-pe orte N.N` refers to the number of processes that you want to allocate to the MPI program. Here's an example of submitting a job that will use 2 processors:

```
$ qsub -pe orte 2 mpi-ring.qsub
```

When the job completes, the job's output will be in the file `mpi-ring.qsub.o*`. Error messages pertaining to the job will be in `mpi-ring.qsub.po*`.

To run the job on more processors, just change the number supplied to the `-pe orte` flag. Here's how to run the job on 16 processors:

```
$ qsub -pe orte 16 mpi-ring.qsub
```

If you need to delete an already submitted job, you can use **qdel** given it's job id. Here's an example of deleting a fluent job under SGE:

```
[sysadm1@frontend-0 sysadm1]$ qsub fluent.sh
your job 31 ("fluent.sh") has been submitted
$ qstat
job-ID prior name      user      state submit/start at      queue      master ja-task-ID
-----
      31     0 fluent.sh  sysadm1   t      12/24/2003 01:10:28 comp-pvfs- MASTER
$ qdel 31
sysadm1 has registered the job 31 for deletion
$ qstat
$
```

Although the example job scripts are bash scripts, SGE can also accept other types of shell scripts. It is trivial to wrap serial programs into a SGE job script. Similarly, for MPI parallel jobs, you just need to use the correct **mpirun** launcher within the job script. For other parallel jobs other than MPI, a Parallel Environment or PE needs to be defined. This is covered within the SGE documentation found on Sun's web site.

3.3. Monitoring SGE Jobs

To monitor jobs under SGE, use the **qstat** command. When executed with no arguments, it will display a summarized list of jobs

```
[sysadm1@frontend-0 sysadm1]$ qstat
job-ID prior name      user      state submit/start at      queue      master ja-task-ID
-----
      20     0 sleep.sh  sysadm1   t      12/23/2003 23:22:09 frontend-0 MASTER
```

```

21      0 sleep.sh  sysadm1    t      12/23/2003 23:22:09 frontend-0 MASTER
22      0 sleep.sh  sysadm1    qw     12/23/2003 23:22:06

```

Use **qstat -f** to display a more detailed list of jobs within SGE.

```

[sysadm1@frontend-0 sysadm1]$ qstat -f
queuename          qtype used/tot. load_avg arch      states
-----
comp-pvfs-0-0.q    BIP   0/2         0.18   glinux
-----
comp-pvfs-0-1.q    BIP   0/2         0.00   glinux
-----
comp-pvfs-0-2.q    BIP   0/2         0.05   glinux
-----
frontend-0.q       BIP   2/2         0.00   glinux
  23      0 sleep.sh  sysadm1    t      12/23/2003 23:23:40 MASTER
  24      0 sleep.sh  sysadm1    t      12/23/2003 23:23:40 MASTER

#####
- PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS
#####
  25      0 linpack.sh sysadm1    qw     12/23/2003 23:23:32

```

You can also use **qstat** to query the status of a job, given its job id. For this, you would use the **-j N** option where **N** would be the job id.

```

[sysadm1@frontend-0 sysadm1]$ qsub -pe mpich 1 single-xhpl.sh
your job 28 ("single-xhpl.sh") has been submitted
[sysadm1@frontend-0 sysadm1]$ qstat -j 28
job_number:          28
exec_file:           job_scripts/28
submission_time:     Wed Dec 24 01:00:59 2003
owner:               sysadm1
uid:                 502
group:               sysadm1
gid:                 502
sge_o_home:          /home/sysadm1
sge_o_log_name:      sysadm1
sge_o_path:           /opt/sge/bin/glinux:/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/u
sge_o_mail:          /var/spool/mail/sysadm1
sge_o_shell:         /bin/bash
sge_o_workdir:       /home/sysadm1
sge_o_host:          frontend-0
account:             sge
cwd:                 /home/sysadm1
path_aliases:        /tmp_mnt/ * * /
merge:               y
mail_list:           sysadm1@frontend-0.public
notify:              FALSE
job_name:            single-xhpl.sh
shell_list:          /bin/bash
script_file:         single-xhpl.sh
parallel environment: mpich range: 1
scheduling info:     queue "comp-pvfs-0-1.q" dropped because it is temporarily not availab
queue "comp-pvfs-0-2.q" dropped because it is temporarily not availab
queue "comp-pvfs-0-0.q" dropped because it is temporarily not availab

```

3.4. Managing SGE queues

To display a list of queues within the Rocks cluster, use **qconf -sql**.

```
[sysadm1@frontend-0 sysadm1]$ qconf -sql
comp-pvfs-0-0.q
comp-pvfs-0-1.q
comp-pvfs-0-2.q
frontend-0.q
```

If there is a need to disable a particular queue for some reason, e.g scheduling that node for maintenance, use **qmod -d Q** where **Q** is the queue name. You will need to be a SGE manager in order to disable a queue like the root account. You can also use wildcards to select a particular range of queues.

```
[sysadm1@frontend-0 sysadm1]$ qstat -f
queuename          qtype used/tot. load_avg arch      states
-----
comp-pvfs-0-0.q    BIP   0/2         0.10   glinux
comp-pvfs-0-1.q    BIP   0/2         0.58   glinux
comp-pvfs-0-2.q    BIP   0/2         0.02   glinux
frontend-0.q       BIP   0/2         0.01   glinux
[sysadm1@frontend-0 sysadm1]$ su -
Password:
[root@frontend-0 root]# qmod -d comp-pvfs-0-0.q
Queue "comp-pvfs-0-0.q" has been disabled by root@frontend-0.local
[root@frontend-0 root]# qstat -f
queuename          qtype used/tot. load_avg arch      states
-----
comp-pvfs-0-0.q    BIP   0/2         0.10   glinux   d
comp-pvfs-0-1.q    BIP   0/2         0.58   glinux
comp-pvfs-0-2.q    BIP   0/2         0.02   glinux
frontend-0.q       BIP   0/2         0.01   glinux
```

To enable back the queue, you can use **qmod -e Q**. Here is an example of **Q** being specified as range of queues via wildcards.

```
[root@frontend-0 root]# qmod -e comp-pvfs-*
Queue "comp-pvfs-0-0.q" has been enabled by root@frontend-0.local
root - queue "comp-pvfs-0-1.q" is already enabled
root - queue "comp-pvfs-0-2.q" is already enabled
[root@frontend-0 root]# qstat -f
queuename          qtype used/tot. load_avg arch      states
-----
comp-pvfs-0-0.q    BIP   0/2         0.10   glinux
comp-pvfs-0-1.q    BIP   0/2         0.58   glinux
comp-pvfs-0-2.q    BIP   0/2         0.02   glinux
frontend-0.q       BIP   0/2         0.01   glinux
```

For more information in using SGE, please refer to the SGE documentation and the man pages.

Notes

1. `examples/sleep.sh`

Appendix A. Rocks® Copyright

Rocks(r)
www.rocksclusters.org
version 6.1.1 (Sand Boa)

Copyright (c) 2000 - 2014 The Regents of the University of California.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice unmodified and in its entirety, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising and press materials, printed or electronic, mentioning features or use of this software must display the following acknowledgement:

"This product includes software developed by the Rocks(r)
Cluster Group at the San Diego Supercomputer Center at the
University of California, San Diego and its contributors."

4. Except as permitted for the purposes of acknowledgment in paragraph 3, neither the name or logo of this software nor the names of its authors may be used to endorse or promote products derived from this software without specific prior written permission. The name of the software includes the following terms, and any derivatives thereof: "Rocks", "Rocks Clusters", and "Avalanche Installer". For licensing of the associated name, interested parties should contact Technology Transfer & Intellectual Property Services, University of California, San Diego, 9500 Gilman Drive, Mail Code 0910, La Jolla, CA 92093-0910, Ph: (858) 534-5815, FAX: (858) 534-7345, E-MAIL:invent@ucsd.edu

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Appendix B. Third Party Copyrights and Licenses

This section enumerates the licenses from all the third party software components of this Roll. A "best effort" attempt has been made to insure the complete and current licenses are listed. In the case of errors or omissions please contact the maintainer of this Roll. For more information on the licenses of any components please consult with the original author(s) or see the Rocks CVS repository¹.

B.1. Sun Grid Engine

Sun Industry Standards Source License Version 1.2

=====

The contents of this file are subject to the Sun Industry Standards Source License Version 1.2 (the "License"); You may not use this file except in compliance with the License. You may obtain a copy of the License at http://gridengine.sunsource.net/Gridengine_SISSL_license.html

Software provided under this License is provided on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE SOFTWARE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. See the License for the specific provisions governing your rights and obligations concerning the Software.

The Initial Developer of the Original Code is: Sun Microsystems, Inc.

Copyright: 2001 by Sun Microsystems, Inc.

Notes

1. <http://cvs.rocksclusters.org>