SGE Roll: Users Guide

Version 4.3 Edition

SGE Roll: Users Guide :
Version 4.3 Edition
Published Jul 2007
Copyright © 2007 University of California and Scalable Systems

Table of Contents

List of Tables

Preface

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

Please visit the SGE site[1] 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	sge
Version	4.3
Maintained By	Rocks Group
Architecture	i386, x86_64, ia64
Compatible with Rocks™	4.3

Table 1-2. Roll Compatibility

| Roll	Requires a	Optional b	Conflicts
alpha		X	
area51		X	
base	X		
bio		X	
condor		X	
ganglia		X	
grid		X	
hpc	X		
java		X	
kernel	X		
os (disk 1)	X		
os (disk 2)	X		
os (disk 3)		X	
os (disk 4)		X	
pbs			X
service-pack		X	
sge	X		
viz		X	
web-server		X	
Notes:			
a. You may also substitute your own OS CDs for the Rocks™ OS Roll CDs. In this case you must use all the CDs from your distribution and not use any of the Rocks™ OS Roll CDs.			
b. Only Rolls that have been verified as compatible with this Roll are listed. Other Rolls will likely work, but have not been tested by the maintainer of this Roll.			

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 similar to the one below.
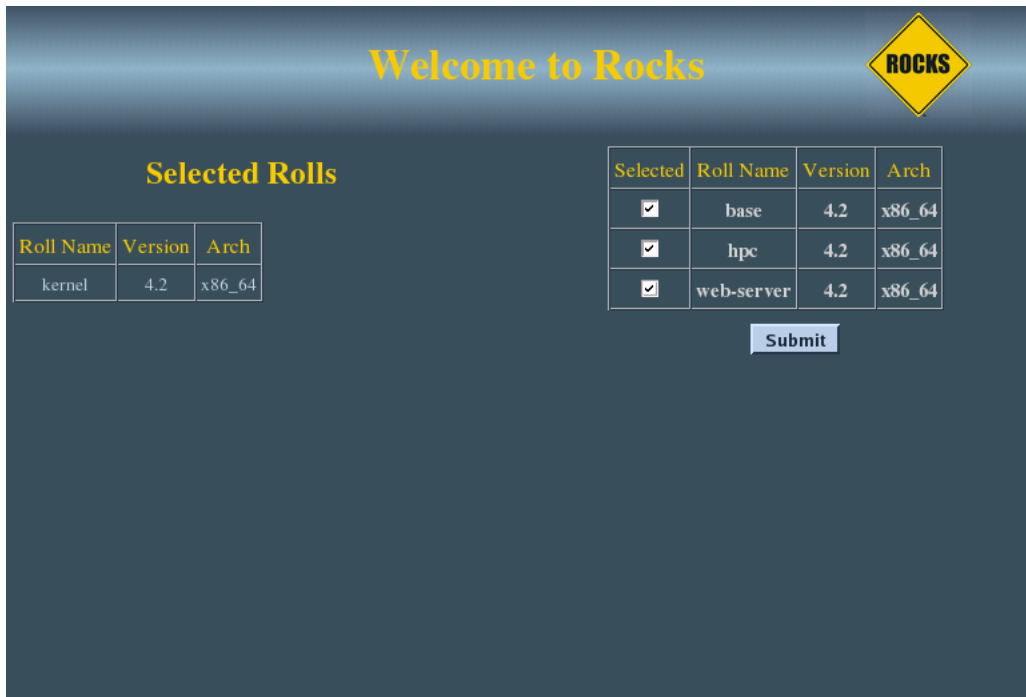
2.2. On an Existing Server

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 it's 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 that generic queues are setup automatically the moment new nodes are being integrated within the Rocks cluster and booted up.

3.2. Setting the SGE environment

When you log into the cluster, the SGE environment would have already been set up for you. The SGE commands should have been automatically added into your $PATH.

```
[sysadm1@frontend-0 sysadm1]$ echo $SGE_ROOT
/opt/gridengine
[sysadm1@frontend-0 sysadm1]$ which qsub
/opt/gridengine/bin/glinux/qsub
```

3.3. 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
```

For a parallel MPI job script, take a look at this script, linpack.sh[2]. Note that you need to put in two SGE variables, `$NSLOTS` and `$TMP/machines` within the job script.

```
[sysadm1@frontend-0 sysadm1]$ cat linpack.sh
#!/bin/bash
#
#$ -cwd
#$ -j y
#$ -S /bin/bash
#
MPI_DIR=/opt/mpich/gnu/
HPL_DIR=/opt/hpl/mpich-hpl/


# OpenMPI part. Uncomment the following code and comment the above code
# to use OpemMPI rather than MPICH

# MPI_DIR=/opt/openmpi/
# HPL_DIR=/opt/hpl/openmpi-hpl/

$MPI_DIR/bin/mpirun -np $NSLOTS -machinefile $TMP/machines \
        $HPL_DIR/bin/xhpl
```

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 mpich N`. N refers to the number of processes that you want to allocate to the MPI program. Here's an example of submitting a 2 processes linpack program using this HPL.dat[3] file:

```
[sysadm1@frontend-0 sysadm1]$ qsub -pe mpich 2 linpack.sh
your job 17 ("linpack.sh") has been submitted
```

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
[sysadm1@frontend-0 sysadm1]$ 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
[sysadm1@frontend-0 sysadm1]$ qdel 31
sysadm1 has registered the job 31 for deletion
[sysadm1@frontend-0 sysadm1]$ qstat
[sysadm1@frontend-0 sysadm1]$
```

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 and to also add in the two SGE variables, $NSLOTS and $TMP/machines within the job script. For other parallel jobs other than MPI, a Parallel Environment or PE needs to be defined. This is covered withn the SGE documentation.

3.4. 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 it's 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:/usr/
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 available
                           queue "comp-pvfs-0-2.q" dropped because it is temporarily not available
                           queue "comp-pvfs-0-0.q" dropped because it is temporarily not available
```

3.5. 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

2. examples/linpack.sh

3. examples/HPL.dat

Appendix A. Rocks Copyright

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 ommisions 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 repository[1].

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,
MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE, OR NON-INFRINGING.
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