GridEngine

Install procedure has been moved to Installing GridEngine


- Using
- Wrapper Script
- tuning parameters
- Array Jobs
- Job Dependency
- Interactive login
- Queue states
- References

Using

qsub <job> - submit job to queue
qsub -e ./ -o ./ -S /bin/bash -N label <script> - a typical set of qsub parameters
qsub -b yes <program> - when you want to run a program.
qstat -f - show which nodes are running jobs
qstat -u "*" - show user information for all jobs in the queue
qstat -f -u "*" - show user information for all nodes in the cluster
qstat -j <job_number> - show detailed information about a particular job
qconf -sql - list all queues
qconf -sq all.q - show settings for all.q queue
qrsh - interactive login to a free node.

Wrapper Script

Typically for programs that are used frequently a wrapper script is created that include all the qsub parameters for that program.

Included examples for such scripts are found in $SGE_ROOT/examples/jobs/

Here is a generic script which can be used as a template to create custom scripts. It will be named qsub_runme.sh and will submit a program called runme to the queue.
#! /bin/bash

# execute script in current directory
#$ -cwd
# want any .e/.o stuff to show up here too
#$ -e ./
#$ -o ./
# shell for qsub to use:
#$ -S /bin/bash
# label for the job; displayed by qstat
#$ -N runme_job

# if you normally run the program like this:
# runme param1 param2 param3 ...
# then run this script like this:
# qsub qsub_runme.sh "param1 param2 param3 ..."
# it's important to contain all parameters in quotes

PROGRAM=~/runme

# $1 are all the parameters passed above

$PROGRAM $1

A more complex example:
#!/bin/bash

# execute script in current directory
#$ -cwd
# want any .e/.o stuff to show up here too
#$ -e ./
#$ -o ./
# shell for qsub to use:
#$ -S /bin/bash
# label for the job; displayed by qstat
#$ -N runsmp_job
#$ -l mem_free=9G
#$ -pe smp 4

# if you normally run the program like this:
# runme param1 param2 param3 ...
# then run this script like this:
# qsub qsub_runme.sh "param1 param2 param3 ..."
# it's important to contain all parameters in quotes

PROGRAM=~/runsmp

START=$(date +%s)
NSTART=$(date +%s.%N)

# uncomment the following if you want a delay before
# the script is run -- helpful when debugging.
#time=20
#sleep $time

# $1 are all the parameters passed above

$PROGRAM $1

END=$(date +%s)
NEND=$(date +%s.%N)

let TOTAL="${END} - ${START}"
DIFF=$(echo "${NEND} - ${NSTART}" | bc)

echo "Script took ${TOTAL} seconds"
echo "Script took ${DIFF} Nanoseconds"

tuning parameters

-1 mem_free=9G
-1 mem_free=17G

Array Jobs

Array jobs are the preferred method of submitting many jobs which only differ due to an iterative process. For example: a large data set that has been split into smaller sets but are all being run through the same analysis.

Here is an example:
#! /bin/bash

# execute script in current directory
#$ -cwd
# want any .e/.o stuff to show up here too
#$ -e ./
#$ -o ./
# shell for qsub to use:
#$ -S /bin/bash
# label for the job; displayed by qstat
#$ -N array_job
#$ -t 1-200

# if you normally run the program like this:
# runme param1 param2 param3 ...
# then run this script like this:
# qsub qsub_runme.sh "param1 param2 param3 ...
# it's important to contain all parameters in quotes

PROGRAM=~/runmeagain

#$1 are all the parameters passed above

$PROGRAM --input ~/data/input.$SGE_TASK_ID  --output ~/results/output.$SGE_TASK_ID

Where there exist 200 input files and $SGE_TASK_ID iterates between 1 and 200

More examples here:
http://www.machlea.com/bohemian/?p=4

Job Dependency

Don't run job2 until job1 has completed.

Example:
#!/usr/bin/env bash

######################################################
# qsub_dependency.sh
# Function: Report various environmental variables
# Usage: qsub -N <job_name> qsub_dependency.sh
######################################################
# execute script in current directory
#$ -cwd
# want any .e/.o stuff to show up here too
#$ -e ./
#$ -o ./
# shell for qsub to use:
#$ -S /bin/bash
# label for the job; displayed by qstat
## #$ -N dependency
######################################################
time=30
sleep ${time}
echo "Hostname: ${HOSTNAME}" 
echo "N Hosts: ${NHOSTS}" 
echo "N Slots: ${NSLOTS}" 
echo "N Queues: ${NQUEUES}" 
echo "Job ID: ${JOB_ID}" 
echo "Job Name: ${JOB_NAME}" 

qsub -N job1 qsub_dependency.sh
qsub -N job2 -hold_jid job1 qsub_dependency.sh

http://arc.leeds.ac.uk/using-the-systems/why-have-a-scheduler/advanced-sge-job-dependencies/

Interactive login

qrsh

Queue states

https://gist.github.com/cmaureir/4fa2d34bc9a1bd194af1

References

Note: after Oracle purchased Sun much of the SGE documentation has been pulled from public use.

- http://docs.sun.com/app/docs/coll/1017.4
- http://docs.sun.com/app/docs/doc/820-0697
- http://himiko.dnsalias.net/twiki/bin/view/Main/SunGridEngineGuide#Sun_Grid_Engine_on_Linux_Howto
- http://gridengine.sunsource.net/howto/howto.html
- http://docs.sun.com/app/docs/doc/817-6118?q=N1GE
- http://www.perceus.org/portal/node/81
- https://wiki.duke.edu/display/SCSC/SGE+Queueing+System
- http://wiki2.bx.psu.edu/Admin/Config/Performance/Cluster
- http://wiki.hpc.ufl.edu/index.php/Getting_Started
- http://naf.desy.de/general_naf_docu/e66164/e66272/index_eng.html
- https://wiki.duke.edu/display/SCSC/SGE+Queueing+System
- https://wiki.duke.edu/display/SCSC/SGE+Queueing+System
- http://help.igb.uiuc.edu/Computation_Cluster
- http://www-it.desy.de/systems/services/batch/sge/ - seems good
- http://www.macresearch.org/lab_journal_installing_sun_grid_engine_on_xserve_cluster- os x
- http://www.biostat.jhsph.edu/bit/cluster-usage.html#MemSpec
- http://www.ace-net.ca/wiki/FAQ
• http://www.biostat.jhsph.edu/bit/cluster-usage.html
• https://cgs.wustl.edu/howto/gridengine/
• https://wiki.zserv.tuwien.ac.at/doku.php?id=doku:vsc2

• http://gridscheduler.sourceforge.net/howto/GridEngineHowto.html
• http://gridscheduler.sourceforge.net/security.html


• http://bioinformatics.mdc-berlin.de/intro2UnixandSGE/sun_grid_engine_for_beginners/README.html