Grid Engine Quick Intro

chris@bioteam.net

First things first

- You'll need a cluster account
- Run jobs from your cwd directory

- Watch out for …
  - Location of your output and error files
  - Default SGE location is your cwd directory

chris@bioteam.net

What is Grid Engine?

- "Distributed Resource Management"
- Comparable to Platform LSF or Slurm
- What it does (vastly simplified):
  - Queues & schedules jobs
  - Matches jobs to most suitable execution host.
  - Manages resources (memory, vCPU, etc.)
  - Enforces resource allocation.

chris@bioteam.net

What does it do for me?

- Allows multiple users, groups & projects to work together on shared infrastructure
- Scientific/Research priorities can be reflected in how the system is used
- Treats you fairly
- Stays out of your way

chris@bioteam.net

Grid Engine does the following:

- Accept work requests (jobs) from users
- Puts jobs in a pending area
- Sends jobs from the pending area to the best available machine
- Manages the job while it runs
- Returns results, logs accounting data when the job is finished

chris@bioteam.net

“The Contract”

- Your responsibility:
  - Describe what resources are necessary to ensure success for your job(s)
- Grid Engine’s responsibility:
  - Match job to available resources, get results back to you

chris@bioteam.net

Key Message

1. Don’t worry about queues or specific machines.
2. All you need to do when submitting a job is describe the resources your job will need to run
3. The ‘default’ settings are good enough for most cases

chris@bioteam.net

Getting work done

chris@bioteam.net

Submitting jobs

- Jobs are submitted via the ‘qsub’ command
- Many factors affect how/when a job gets dispatched for execution
  - Job resource requirements
  - Availability of eligible execution hosts
  - Various job slot limits
  - Job dependency conditions
  - Load conditions

chris@bioteam.net

Submitting Jobs

- Important to note that jobs are not necessarily dispatched in the order received
- Your cluster is currently running a scheduling policy

chris@bioteam.net

Most useful SGE commands

- `qsub / qdel`
  - Submit jobs & delete jobs
- `qstat`
  - Status info for jobs
- `qacct`
  - Summary info and reports on completed job

chris@bioteam.net

qsub

General format:

```
$ qsub <qsub options> program <prog. options>
```

The simplest possible SGE submit syntax would be of this form:

```
$ qsub ./myjob.sh
```

chris@bioteam.net

Example: sleeper.sh

```
#!/bin/sh!
#!
# Usage: sleeper.sh [time]]!
#        default for time is 60 seconds!

# -- our name ---!
#$ -N Sleeper!
#$ -S /bin/sh!

/bin/echo I am running on host `hostname`. !
/bin/echo Sleeping now at: `date` !

time=60!
if [ $# -ge 1 ]; then!
   time=$1!
fi!
sleep $time!

echo Now it is: `date`!
```

chris@bioteam.net

SGE embedded in jobscripts

```
#!/bin/sh!
#!
# Usage: sleeper.sh [time]]!
#        default for time is 60 seconds!

# -- SGE ARGUMENTS --!
#$ -N Sleeper!
#$ -S /bin/sh!

/bin/echo I am running on host `hostname`. !
/bin/echo Sleeping now at: `date` !

time=60!
if [ $# -ge 1 ]; then!
   time=$1!
fi!
sleep $time!

echo Now it is: `date`!
```

chris@bioteam.net

Real world example

```
#!/bin/sh!

# Batch-submission script for SGE (Sun GridEngine)
system!

# Do we need to re-source our grid engine environment?!
source /common/sge/default/common/settings.sh!

## -- Chris Dagdigian; BioTeam Inc.!
## -- Embedded grid engine directives follow!
#$ -N %NAME%!
#$ -o %DIR%/.%JOBID%.qlog.out!
#$ -e %DIR%/.%JOBID%.qlog.err!
#$ -P glide!
#$ -hard -l glideL-impact-main=1!
#$ -hard -l glideL-impact-glide=4!

## -- ok back to work (Glide stuff below) ...!
```

chris@bioteam.net

Requesting Resources

- Soft resource requests:
  - Optional, SGE will try to find the resource but may dispatch without it
  - qsub -l matLabLicense ./my-job-script.sh
- Hard request
  - Non-negotiable, job will not run until resource is available
  - qsub -hard -l matLabLicene=true ./my-job-script.sh
  - qsub -l h_vmem=8G ./my-job-script.sh
- Remember:
  - This is only the tip of the iceberg; resource framework is very powerful
  - Requests can be embedded in scripts so they don't have to be typed all the time

chris@bioteam.net

Jobs: Binaries vs. Scripts

- Grid Engine assumes script submission:
  - “qsub ./my-job-script.sh”
  - Directly submitting a binary will not work
- To override & submit a binary:
  - Use qsub “-b” switch
  - “qsub -b y /stf/bin/blastall … “

chris@bioteam.net

Jobs: Parallel jobs

- SGE vocabulary:
  - “Parallel Environment” or “PE”
- Example:
  - qsub -pe pthreads 200 ./my-200proc-job.sh
- Example using CPU ranges:
  - qsub -pe pthreads 100-200 ./my-mpi-job.sh

chris@bioteam.net

Job Control & Status Checking

- Job Deletion
  - Use the ‘`qdel`’ command
- Status of active jobs
  - Use the ‘`qstat`’ command
- Data regarding completed jobs:
  - Use the ‘`qacct`’ command

chris@bioteam.net

qstat simple usage

- `qstat -help`
  - More usage info
- `qstat`
  - Displays current jobs in the system
- `qstat -j [job ID or joblist]`
  - Shows config and scheduler info for job
- `qstat -u <user>`
  - Show only jobs from that user (or all users with -u “*”)
- `qstat -t`
  - Information on array jobs

chris@bioteam.net

qstat simple usage continued

- `qstat -explain`
  - More info about the reason queue(s) in alarm state
- `qstat -f !`
  - Full queue summary
- `qstat -f -ne`
  - Queue summary with empty queues ignored

chris@bioteam.net

Possible job states reported by qstat

- ‘t’ -- Transferring
- ‘r’ -- Running
- ‘R’ -- Restarted
- ‘s’ -- Suspended
- ‘S’ -- Suspended by the queue
- ‘T’ -- Suspend queue threshold reached
- ‘w’ -- Waiting
- ‘h’ -- Hold
- ‘e’ – Error
- ‘q’ -- queued

chris@bioteam.net

Possible queue states reported by qstat

- ‘u’ -- Unknown (sge_execd or server down?)
- ‘a’ -- Alarm (load threshold reached)
- ‘A’ -- Alarm (suspend threshold reached)
- ‘s’ -- Suspended (by user or admin)
- ‘d’ -- Disabled (by user or admin)
- ‘C’ -- Suspended (by calendar)
- ‘D’ -- Disabled (by calendar)
- ‘S’ -- Suspended (by subordination)
- ‘E’ -- Error (sge_execd can’t reach shepherd)

chris@bioteam.net

Wrapping Up

- ‘qsub’, ‘qstat’ & ‘qdel’ will get you started
  - Each of these programs is quite powerful
  - We’ve only covered the absolute minimum
  - Read the docs or manpages for more details

chris@bioteam.net