Submit a Job

• Want to run a batch script:

```bash
#!/bin/sh
echo Starting job
date
/usr/bin/time ./hello
date
echo Ending job
```

• Have to ask scheduler to do it.

```
#!/bin/sh
#$ -N ht3d-hyb
#$ -cwd
#$ -o $JOB_NAME.o$JOB_ID
#$ -j y
#$ -A C-RANGER
#$ -q development
#$ -pe 4way 16
#$ -V
#$ -l h_rt=00:10:00
echo Starting job
date
echo Ending job
```

```bash
qsub -A 20090528HPC job.sge
```
Two Time Commands

- Used to see how long your program runs and estimate if it’s having gross difficulties
- `/usr/bin/time` generally more information

```bash
login4% time ./hello
Hello world!
0.000u 0.030s 0:00.06 50.0% 0+0k 0+0io 2pf+0w

login4% /usr/bin/time ./hello
Hello world!
0.00user 0.01system 0:00.03elapsed 32%CPU (0avgtext+0avgdata 0maxresident)k
0inputs+0outputs (0major+213minor)pagefaults 0swaps
```
How Are the Queues?

- List available queue: `qconf --sql`
- `showq` or “`showq --u`”
- Delete job: `qdel` or `qdel -f`
$HOME
$WORK
$SCRATCH
Queue Examples

login3% qconf -sql clean development large long normal request reservation serial syslog debug systest vis

login3% qconf -sq development
qname qtype pe_list slots	 tmpdir shell prolog epilog shell_start_mode s_rt h_rt
development
BATCH INTERACTIVE
16way 15way 14way 12way 8way 4way 2way 1way 16
/tmp /bin/csh
/share/sge/default/pe_scripts/prologWrapper
/share/sge/default/pe_scripts/tacc_epilog
unix_behavior
07:58:00
08:00:00

Slots = number of cores, 16 per node
pe = wayness, how many cores per node
Job is killed if over time limit.

Why 15way?
Four States

• Unscheduled – Likely not good
• DepWait – You can ask that one job run after another finishes.
• Waiting – Queued, waiting for resources to run.
• Running – As far as SGE is concerned, it’s going.
Un-TAR Job to Submit

- TAR = Tape archive.
- Just concatenates files.
- `tar <switches> <files>`
- `z` = compress or decompress
- `x` = extract
- `c` = create
- `v` = verbose
- `t` = list files
- `f` = next argument is the file to read or write
- `~userid` is the home directory of that user
- For example, to create a tar: `tar cvf myfiles.tar dir1 dir2 README`
Active Jobs

<table>
<thead>
<tr>
<th>JOBID</th>
<th>JOBNAME</th>
<th>USERNAME</th>
<th>STATE</th>
<th>CORE</th>
<th>REMAINING</th>
<th>STARTTIME</th>
</tr>
</thead>
</table>

378 active jobs: 3629 of 3852 hosts (94.21%)

Waiting Jobs

<table>
<thead>
<tr>
<th>JOBID</th>
<th>JOBNAME</th>
<th>USERNAME</th>
<th>STATE</th>
<th>CORE</th>
<th>WCLIMIT</th>
<th>QUEUETIME</th>
</tr>
</thead>
</table>

Waiting Jobs with Job Dependencies

<table>
<thead>
<tr>
<th>JOBID</th>
<th>JOBNAME</th>
<th>USERNAME</th>
<th>STATE</th>
<th>CORE</th>
<th>WCLIMIT</th>
<th>QUEUETIME</th>
</tr>
</thead>
</table>

Unscheduled Jobs

<table>
<thead>
<tr>
<th>JOBID</th>
<th>JOBNAME</th>
<th>USERNAME</th>
<th>STATE</th>
<th>CORE</th>
<th>WCLIMIT</th>
<th>QUEUETIME</th>
</tr>
</thead>
</table>

Total jobs: 963  Active Jobs: 378  Waiting Jobs: 469  Dep/Unsched Jobs: 116
Four States

• Unscheduled – Likely not good
• DepWait – You can ask that one job run after another finishes.
• Waiting – Queued, waiting for resources to run.
• Running – As far as SGE is concerned, it’s going.
Un-TAR Job to Submit

- TAR = Tape archive.
- Just concatenates files.
- `tar <switches> <files>`
  - `z` = compress or decompress
  - `x` = extract
  - `c` = create
  - `v` = verbose
  - `t` = list files
  - `f` = next argument is the file to read or write
  - `~userid` is the home directory of that user
- For example, to create a tar: `tar cvf myfiles.tar dir1 dir2 README`
Submit a Job Example

- less job.sge # examine the script
- ./job.sge # To run the job on current node.
- qsub -A 20090528HPC job.sge # Submit the job
Running and Output

- `showq -u` # Watch it run.
- `less hello.oXXX` # Look at the output file when it’s done.
- Try comparing the environment variables on login with batch.
  - `env | sort > z.txt`
  - `diff z.txt hello.oXXX | less`
Environment Variables in Batch

- `ENVIRONMENT=BATCH`
- `HOSTNAME=i182-401.ranger.tacc.utexas.edu`
- `JOB_ID=743637`
- `JOB_NAME=hello`
- `JOB_SCRIPT=/share/sge/execd_spool//i182-401/job_scripts/743637`
- `NHOSTS=1`
- `NQUEUES=1`
- `NSLOTS=16`
- `PE=1way`
- `PE_HOSTFILE=/share/sge/execd_spool//i182-401/active_jobs/743637.1/pe_hostfile`
- `QUEUE=development`
- `SGE_ACCOUNT=20090528HPC`
- `SGE_CWD_PATH=/share/home/0002/train200/submit`
- `SGE_O_SHELL=/bin/csh`
- `SGE_O_WORKDIR=/share/home/0002/train200/submit`
- `SGE_STDERR_PATH=/share/home/0002/train200/submit/hello.o743637`
- `SGE_STDOUT_PATH=/share/home/0002/train200/submit/hello.o743637`
Mount the Drive

• Only for Cornell to CAC’s V4
• One filesystem for Linux and Windows.
• Linux-based filesystem.
• From Windows, mount \\
storage01.cac.cornell.edu[<userid>]
  – May need to add cac.cornell.edu to your DNS.
• From Mac, mount smb://storage01.cac.cornell.edu[/<userid>]
• From Linux,
  mount –o user=<userid> -t cifs \\
  /storage01.cac.cornell.edu/<userid> \
  /mount/point
TeraGrid!

kilo 1000
mega 1000,000
giga 1000,000,000
tera 1000,000,000,000
peta 1000,000,000,000,000