SGI® Altix™
Running Batch Jobs With PBSPro

Reiner Vogelsang
SGI GmbH
reiner@sgi.com
January 19, 2005
Module Objectives

• After completion of this module you should be able to
  – Submit batch jobs
  – Create job chains
  – Monitor your jobs
Submitting a Request

- The following example submits a batch script named myjob to PBS

  ```bash
  linux$ qsub myjob
  13.privateer.americas.sgi.com
  ```

- Each job submitted is assigned a unique request ID

- Use the request ID number to obtain more information on the request

- Use `qstat -a` to see the queue which received the request
A PBS Job Sample

#! /bin/sh
#PBS -N code2
#PBS -S /bin/sh
#PBS -j oe
#PBS -l pcput=2599
#PBS -l cput=2599
#PBS -l pmem=100mb
#PBS -l mem=100mb
#PBS -l ncpus=1
set -x
time /home/daw/bin/prog.exe \
    >output

Name of your job
The top level job shell
Join standard and error output
Per process CPU time limit
Aggregate CPU time limit
Per process memory limit
Aggregate memory limit
Number of CPUS
Common qsub Arguments

- a  submit at date and time
- j  join stdout and stderr
- l  specify resource requirements and limits
- N  job name
- u  user list
- h  hold job
- m  mail options
- o  stdout redirection
- p  Priority
- q  Queue to submit to
- r y|n  Rerunnable
- W depend=..  Job dependency
Viewing Job Status

```
linux$ qstat -a
privateeer.americas.sgi.com:

<table>
<thead>
<tr>
<th>Job ID</th>
<th>Username</th>
<th>Queue</th>
<th>Jobname</th>
<th>SessID</th>
<th>NDS</th>
<th>TSK</th>
<th>Memory</th>
<th>Time</th>
<th>S Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.privateer.am daw</td>
<td>ace</td>
<td>code2</td>
<td></td>
<td>22712</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>R 00:00</td>
</tr>
<tr>
<td>16.privateer.am daw</td>
<td>ace</td>
<td>code2</td>
<td></td>
<td>1902</td>
<td>--</td>
<td>--</td>
<td>100mb</td>
<td>00:43</td>
<td>R 00:00</td>
</tr>
</tbody>
</table>
```

- or use xpbs
  - You only see your own jobs by default
  - Controlled by a server attribute

```
qmgr -c"set server query_other_jobs=true"
```
What Does Queue Status Mean?

```
linux$ qstat -a
privateer.americas.sgi.com:
```

<table>
<thead>
<tr>
<th>Job ID</th>
<th>Username</th>
<th>Queue</th>
<th>Jobname</th>
<th>SessID</th>
<th>NDS</th>
<th>TSK</th>
<th>Memory</th>
<th>Time</th>
<th>S Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>70.privateer.am daw</td>
<td></td>
<td>small</td>
<td>code1</td>
<td>9418</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>00:00</td>
<td>R 00:00</td>
</tr>
<tr>
<td>71.privateer.am daw</td>
<td></td>
<td>small</td>
<td>code1</td>
<td>32267</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>00:00</td>
<td>R 00:00</td>
</tr>
<tr>
<td>72.privateer.am daw</td>
<td></td>
<td>medium</td>
<td>code2</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>00:05</td>
<td>Q</td>
</tr>
<tr>
<td>73.privateer.am daw</td>
<td></td>
<td>small</td>
<td>code3</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>00:00</td>
<td>Q</td>
</tr>
<tr>
<td>74.privateer.am daw</td>
<td></td>
<td>huge</td>
<td>code4</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>08:20</td>
<td>Q</td>
</tr>
</tbody>
</table>

- Run qstat -s

- Have to run qstat -f jid and look at comment field

- Also check ps and top in case PBS server hasn’t updated the comment field or qstat information, just to be sure
## Getting Job Status Summary

```bash
linux$ qstat -s
privateer.americas.sgi.com:
```

<table>
<thead>
<tr>
<th>Job ID</th>
<th>Username</th>
<th>Queue</th>
<th>Jobname</th>
<th>SessID</th>
<th>NDS</th>
<th>TSK</th>
<th>Memory</th>
<th>Time</th>
<th>S Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.privateer.a daw ace code2 22848 -- -- 100mb 00:09 R 00:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job run on node ramjet - started on Mon Jan 31 at 09:57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>194.privateer.a daw ace code4 -- -- -- 500mb 08:20 Q --</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Running: All timesharing nodes are too loaded to run job</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>196.privateer.a daw ace code2 22862 -- -- 100mb 00:09 R 00:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job run on node ramjet - started on Mon Jan 31 at 09:57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>198.privateer.a daw ace code4 -- -- -- 500mb 08:20 Q --</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Running: All timesharing nodes are too loaded to run job</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200.privateer.a daw ace code2 22876 -- -- 100mb 00:09 R 00:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job run on node ramjet - started on Mon Jan 31 at 09:57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>202.privateer.a daw ace code4 -- -- -- 500mb 08:20 Q --</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Running: All timesharing nodes are too loaded to run job</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>204.privateer.a daw ace code2 8554 -- -- 100mb 00:09 R 00:02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job run on node privateer - started on Mon Jan 31 at 09:57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>206.privateer.a daw ace code4 -- -- -- 500mb 08:20 Q --</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Running: All timesharing nodes are too loaded to run job</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>207.privateer.a daw ace code1 22890 -- -- 100mb 00:00 R 00:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job run on node ramjet - started on Mon Jan 31 at 09:57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>208.privateer.a daw ace code2 22922 -- -- 100mb 00:09 R 00:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job run on node ramjet - started on Mon Jan 31 at 09:57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>210.privateer.a daw ace code4 -- -- -- 500mb 08:20 Q --</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Running: All timesharing nodes are too loaded to run job</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>211.privateer.a daw ace code1 22904 -- -- 100mb 00:00 R 00:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job run on node ramjet - started on Mon Jan 31 at 09:57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>213.privateer.a daw ace code4 -- -- -- 500mb 08:20 Q --</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Running: All timesharing nodes are too loaded to run job</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Getting Full Job Status

```
linux$ qstat -f 74
Job Id: 74.privateer.americas.sgi.com
  Job_Name = code4
  Job_Owner = daw@privateer.americas.sgi.com
  job_state = Q
  queue = huge
  server = privateer.americas.sgi.com
  Checkpoint = u
  ctime = Wed Jan 12 14:44:51 2000
  Error_Path = privateer.americas.sgi.com:/home/tulip26/daw/realwl/code4.e74
  Hold_Types = n
  Join_Path = oe
  Keep_Files = n
  Mail_Points = a
  mtime = Wed Jan 12 14:45:25 2000
  Output_Path = privateer.americas.sgi.com:/home/tulip26/daw/realwl/code4.o74
  Priority = 0
  qtime = Wed Jan 12 14:44:51 2000
  Rerunnable = True
  Resource_List.cput = 08:20:00
  Resource_List.nodes = 1
  Resource_List.pcpu = 08:20:00
  Resource_List.vm = 300mb
  Resource_List.vmem = 300mb
  Shell_Path_List = /bin/sh
  Variable_List = PBS_O_HOME=/home/tulip26/daw,PBS_O_LOGNAME=daw,
    PBS_O_PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin,
    PBS_O_MAIL=/var/spool/mail/daw,PBS_O_SHELL=/bin/csh,
    PBS_O_HOST=privateer.americas.sgi.com,
    PBS_O_WORKDIR=/home/tulip26/daw/realwl,PBS_O_QUEUE=batch
  comment = Not Running: Not enough of the right type of nodes are available
  etime = Wed Jan 12 14:44:51 2000
```
Job State

- E Exiting PBS
- H Operator hold with qhold, user hold or system hold
- Q Queued up for reason in comment field
- R Executing or running in the cluster
- S Suspended
- T Transiting
- W Waiting for qsub -a start time
Deleting a PBS Job

```
linux$ qstat -a
```

```
<table>
<thead>
<tr>
<th>Job ID</th>
<th>Username</th>
<th>Queue</th>
<th>Jobname</th>
<th>SessID</th>
<th>NDS</th>
<th>TSK</th>
<th>Memory</th>
<th>Time</th>
<th>S Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>80.privatee.am daw</td>
<td>medium</td>
<td>code2</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>00:05 Q</td>
<td>--</td>
</tr>
<tr>
<td>81.privatee.am daw</td>
<td>small</td>
<td>code3</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>00:00 Q</td>
<td>--</td>
</tr>
<tr>
<td>82.privatee.am daw</td>
<td>huge</td>
<td>code4</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>08:20 Q</td>
<td>--</td>
</tr>
</tbody>
</table>
```

```
linux$ qdel 82
```

```
linux$ qstat -a
```

```
privateer.americas.sgi.com:
```

```
<table>
<thead>
<tr>
<th>Job ID</th>
<th>Username</th>
<th>Queue</th>
<th>Jobname</th>
<th>SessID</th>
<th>NDS</th>
<th>TSK</th>
<th>Memory</th>
<th>Time</th>
<th>S Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>79.privatee.am daw</td>
<td>small</td>
<td>code1</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>00:00 Q</td>
<td>--</td>
</tr>
<tr>
<td>80.privatee.am daw</td>
<td>medium</td>
<td>code2</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>00:05 Q</td>
<td>--</td>
</tr>
<tr>
<td>81.privatee.am daw</td>
<td>small</td>
<td>code3</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>00:00 Q</td>
<td>--</td>
</tr>
</tbody>
</table>
```

- Sends SIGTERM then after a grace period a SIGKILL
  - Each queue has a default setting with attribute **kill_delay**
  - `qdel -w` can override the default
Signaling a PBS Job

- `qsig -s SIGUSR1 77.job`

- Program has to handle signals
  
  ```c
  signal(int SIGUSR1, void(*signal_handler));
  ```

- See the `signal(2)` man page or `kill -l`

- Signals may be used to:
  
  - checkpoint or application save,
  
  - PBS does not use the OS Checkpoint on LINUX

  - change application
    
    » output device
    
    » number of iterations
### xpbs

#### Hosts

<table>
<thead>
<tr>
<th>Server</th>
<th>Max</th>
<th>Tot</th>
<th>Que</th>
<th>Run</th>
<th>Hld</th>
<th>Wet</th>
<th>Trn</th>
<th>Ext Status</th>
<th>PEsInUse</th>
<th>Select All</th>
</tr>
</thead>
<tbody>
<tr>
<td>privateer.americas.sgi.com</td>
<td>0</td>
<td>15</td>
<td>13</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0 Active</td>
<td>-/-</td>
<td></td>
</tr>
</tbody>
</table>

#### Queues Listed By Host(s):

<table>
<thead>
<tr>
<th>Queue</th>
<th>Max</th>
<th>Tot</th>
<th>Enq</th>
<th>Str</th>
<th>Que</th>
<th>Run</th>
<th>Hld</th>
<th>Wet</th>
<th>Trn</th>
<th>Ext Type</th>
<th>Server</th>
<th>Select All</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch</td>
<td>0</td>
<td>0</td>
<td>yes</td>
<td>yes</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0 Route</td>
<td>privateer.americas.sgi.com</td>
<td></td>
</tr>
<tr>
<td>small</td>
<td>10</td>
<td>6</td>
<td>yes</td>
<td>yes</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0 Execution</td>
<td>privateer.americas.sgi.com</td>
<td></td>
</tr>
<tr>
<td>medium</td>
<td>10</td>
<td>0</td>
<td>yes</td>
<td>yes</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0 Execution</td>
<td>privateer.americas.sgi.com</td>
<td></td>
</tr>
<tr>
<td>large</td>
<td>1</td>
<td>6</td>
<td>yes</td>
<td>yes</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0 Execution</td>
<td>privateer.americas.sgi.com</td>
<td></td>
</tr>
<tr>
<td>huge</td>
<td>1</td>
<td>3</td>
<td>yes</td>
<td>yes</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0 Execution</td>
<td>privateer.americas.sgi.com</td>
<td></td>
</tr>
</tbody>
</table>

#### Jobs Listed By Queue(s):

<table>
<thead>
<tr>
<th>Job id</th>
<th>Name</th>
<th>User</th>
<th>PEs</th>
<th>CputUse</th>
<th>WalltUse</th>
<th>Queue</th>
<th>Select All</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>privateer.americas code1</td>
<td>daw</td>
<td>-</td>
<td>0:00:00</td>
<td>0:00:00</td>
<td>R <a href="mailto:small@privateer.americas.sgi.com">small@privateer.americas.sgi.com</a></td>
<td>detail</td>
</tr>
<tr>
<td>44</td>
<td>privateer.americas code3</td>
<td>daw</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>R <a href="mailto:small@privateer.americas.sgi.com">small@privateer.americas.sgi.com</a></td>
<td>modify...</td>
</tr>
<tr>
<td>46</td>
<td>privateer.americas code1</td>
<td>daw</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>R <a href="mailto:small@privateer.americas.sgi.com">small@privateer.americas.sgi.com</a></td>
<td>delete...</td>
</tr>
<tr>
<td>48</td>
<td>privateer.americas code3</td>
<td>daw</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>R <a href="mailto:small@privateer.americas.sgi.com">small@privateer.americas.sgi.com</a></td>
<td>hold...</td>
</tr>
<tr>
<td>50</td>
<td>privateer.americas code1</td>
<td>daw</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>R <a href="mailto:small@privateer.americas.sgi.com">small@privateer.americas.sgi.com</a></td>
<td>release...</td>
</tr>
<tr>
<td>52</td>
<td>privateer.americas code3</td>
<td>daw</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>R <a href="mailto:small@privateer.americas.sgi.com">small@privateer.americas.sgi.com</a></td>
<td>signal...</td>
</tr>
</tbody>
</table>

#### INFO

01/12/00 10:39:381 /usr/local/lib/xpbs/bin/xpbs_datadump -t 30 -u daw privateer.americas.sgi.com

---

January 19, 2005
Page 13
Finding the Job Output

- Check `qstat -a` for the job first, `qstat -f` shows where the output should return to

- Check job `qsub` arguments for `-o` `-e` `-k` specified

- Check your email (caution about `.forward`)

- When the job is done the output will be returned to:
  - directory request was submitted from
  - home directory on the execution server
    
    `/usr/spool/pbs/undelivered directory`
    `/usr/spool/pbs` can fill up or `/home` can fill up

- Need permissions to send output back to submit directory

- PBS uses `cp`, `rcp` or `scp` to stage files
  - No error recovery like File Transfer Agent
Request Output

– Returned to directory job was submitted from

– Affected by:

  #PBS-o Put stdout at this path
  #PBS-e Put stderr at this path
  #PBS-j Join the stdout and stderr
  #PBS-k Keep job output files on execution server

– Comes back as jobname.oJID and jobname.eJID unless -j used

– Use set -x(sh) or set echo(csh) for a verbose trace of commands executed
PBS Access Control

– Server host ACL

– Operator/manager ACL

– Queue user/group ACL

– Queue host ACL

– Job access (root, manager, operator)

– Job status access (root, managers, operators,)

  • query_other_jobs controls user access to other users qstat information

– root can’t submit jobs by default

  • set server acl_roots=root@privateer.americas.sgi.com
User Authentication

• Host access with /etc/hosts.equiv or .rhosts
  – NIS or flat UID’s preferred but not required
• pbs_iff provides a modular user authentication
  – runs as setuid as root
  – started by pbs_connect from client side
  – can be replaced by kerberos
• PBS is using its own connect mechanism.
  – Does not use rsh or ssh.
  – pps_server sends job script to pps_mom which finally forks().
Are the proper Resources Declared?

```bash
linux$ qsub code4.pbs
qsub: Job rejected by all possible destinations
• Look for `-l nodes=` mismatch and server neednodes attribute

linux$ qsub code2.pbs
qsub: Job exceeds queue resource limits

• Check `qsub -l attributes compared to qstat -q`
  – pcpus
  – cput
  – pmem
  – vmem
  – pvmem
  – nodes
```
Resource Limit Errors

• Not enough CPU time
  
  `linux$ more code2.o41`
  
  `....`
  
  Command terminated by signal 9

• `kill -l will list the signal number`

• Not enough memory

  `linux$ more code4.o40`
  
  `....`
  
  `code4:ERROR:malloc: Cannot allocate memory`
linux$ qstat -q

server: privateer.americas.sgi.com

<table>
<thead>
<tr>
<th>Queue</th>
<th>Memory</th>
<th>CPU</th>
<th>Time</th>
<th>Walltime</th>
<th>Node</th>
<th>Run</th>
<th>Que</th>
<th>Lm</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>--</td>
<td>E</td>
</tr>
<tr>
<td>small</td>
<td>--</td>
<td>00:02:00</td>
<td>--</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>ER</td>
<td></td>
</tr>
<tr>
<td>medium</td>
<td>--</td>
<td>00:08:00</td>
<td>--</td>
<td>--</td>
<td>1</td>
<td>0</td>
<td>10</td>
<td>ER</td>
<td></td>
</tr>
<tr>
<td>large</td>
<td>--</td>
<td>00:24:00</td>
<td>--</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>ER</td>
<td></td>
</tr>
<tr>
<td>huge</td>
<td>--</td>
<td>64:00:00</td>
<td>--</td>
<td>--</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>ER</td>
<td></td>
</tr>
</tbody>
</table>

linux$ qstat -Q

<table>
<thead>
<tr>
<th>Queue</th>
<th>Max</th>
<th>Tot</th>
<th>Ena</th>
<th>Str</th>
<th>Que</th>
<th>Run</th>
<th>Hld</th>
<th>Wat</th>
<th>Trn</th>
<th>Ext</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch</td>
<td>0</td>
<td>0</td>
<td>yes</td>
<td>yes</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Route</td>
</tr>
<tr>
<td>small</td>
<td>10</td>
<td>0</td>
<td>yes</td>
<td>yes</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Execution</td>
</tr>
<tr>
<td>medium</td>
<td>10</td>
<td>1</td>
<td>yes</td>
<td>yes</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Execution</td>
</tr>
<tr>
<td>large</td>
<td>1</td>
<td>0</td>
<td>yes</td>
<td>yes</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Execution</td>
</tr>
<tr>
<td>huge</td>
<td>1</td>
<td>0</td>
<td>yes</td>
<td>yes</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Execution</td>
</tr>
</tbody>
</table>
Important User Environment Variables

- PBS_O_HOST: Originating host
- PBS_O_QUEUE: Originating queue
- PBS_O_WORKDIR: Originating directory of qsub
- PBS_ENVIRONMENT: Indicates this is a PBS batch job
- PBS_JOBID: Job ID
- PBS_JOBNAME: Job name
- PBS_NODEFILE: node file path
- PBS_QUEUE: Executing queue
- PBS_DEFAULT: Default server name
User .profile/.login Setup

if ( ! $?PBS_ENVIRONMENT ) then
    do terminal stuff like stty
else
    date
    set echo
endif
Why a TMPDIR

• Job runs in $HOME by default
  – Contention on that file system with the interactive user is bad
• Needs unique directory to avoid file name collision if user runs more than one identical job
• Removal of scratch files automatically saves disk space
  – i.e., out-of-core-solvers
• Put job in appropriate bandwidth/latency file system
  – large sequential
  – small random
• Use PBS_QUEUE in .profile to switch to the right type of file system
Job Dependency Example

#PBS -N code2
#PBS -W depend=synccount:2
/home/daw/bin/prog.exe >output

#PBS -N code2a
#PBS -W depend=syncwith:JID
/home/daw/bin/prog.exe >output2
Job Dependency -W depend Options

- synccount  Master job (1st) requires this
- syncwith   start at same time as master
- after      start after master job has started
- afterok    start after master job finished no errors
- afternotok start after master job finished with errors
- afterany   start after master job finished
- before     start specified job before this one
- beforeok   start specified job before this one, no errs
- beforenotok start specified job before this one with errs
- beforeany  start specified job before this one
- on         start after count dependencies
File Staging

#PBS -N code2
#PBS -l pcput=2599
#PBS -l cput=2599
#PBS -l pmem=100mb
#PBS -l mem=100mb
#PBS -W stagein=/tmp/file@myhost:origfile
#PBS -W stageout=output@myhost:output
/home/daw/bin/prog.exe  <file >output
Exercises

• Edit your $HOME/.profile or .login to handle batch different using ENVIRONMENT

• Create, submit and monitor a PBS job call myenv that does the following
  – displays date and time
  – displays your current working directory
  – displays your environment
  – displays who is on the system
  – Enter qstat and examine the queues on your system.
    Resubmit your myenv request and use qstat to monitor its progress
Lab

• Write job examples for codes provided by instructor
  – code1.pbs
  – code2.pbs
  – code3.pbs
  – code4.pbs
• Submit and test the jobs with qsub, qstat
  – Run qstat right after submitting the job
  – Find job output when the job is done
• Submit and test the jobs with xpbs
Declaring Nodes

#PBS -N code2
#PBS -l pcput=2599
#PBS -l cput=2599
#PBS -l pmem=100mb
#PBS -l mem=100mb
#PBS -l nodes=6+3:fat+2:hippi+disk
/home/daw/bin/prog.exe >output
rcp output myworkstation:data/output$$