Open Computing ‘‘Hands-On’’ Tutorial: April 1994: Listings

Listing 1. A sample batch job script.

```
# This job could be submitted with the command qsub filename
#Set up the user’s environment
#Execute this job with the Bourne shell
#$ -s /bin/sh
#First set the working directory for the job to the current
#working environment with -cwd
#$ -cwd
#Send all mail from the job to myself
#$ -mu sjh
#Send mail at the beginning and the end of the job
#$ -mb
#$ -me
#Put the output from stdout and stderr in the batchjob.out file
#$ -eo batchjob.out
#Now ask for resources - Submitted to the sun group - a single
#queue job. - Multiple queues may be used for PVM jobs.
#$ -G sun 1
#Begin the commands to be executed by the remote system
cd /usr/local/src/DQS-2.1.5/SUN3
echo cleaning the area
pwd
make clean
echo make the batch system
echo The job number of the job is $JOB_ID.
echo The architecture of this machine is $ARC
#Do the actual compilation
make
echo .
```

Listing 2. Submitting a batch job, examining and manipulating the queues.

A. Submitting the batch-job script depicted in Listing 1.

```
pogo% qsub buildDQS.script
your job buildDQS.12 has been accepted
pogo% []
```

B. Submitting two jobs, checking the status of the queue after each job submission. Using the -l option shows more information.

```
pogo% qstat -G sun
QUEUE           GROUP        ARCHITECTURE  STATUS     OWNER    REQUEST  JOB_ID

pogo            sun          SUN3          idle
churchy         sun          SUN3          idle
pogo% qsub buildGCC.script
```

```
your job buildGC.13 has been accepted
pogo% qstat -G sun
QUEUE           GROUP       ARCHITECTURE  STATUS     OWNER    REQUEST   JOB_ID

churchy         sun         SUN3          idle
pogo            sun         SUN3          running    sjh      buildGC   13
pogo% qsub buildDQS.script
your job buildDQ.14 has been accepted
pogo% qstat -G sun -l

*******************************************************************
QUEUE           GROUP       ARCHITECTURE  LOAD_AV    STATUS
pogo            sun         SUN3          0.38       running

OWNER           REQUEST     JOB_ID        REQ/ALLOC  HARD/SOFT
sjh             buildGC     13            running


*******************************************************************
QUEUE           GROUP       ARCHITECTURE  LOAD_AV    STATUS
churchy         sun         SUN3          0.38       running

OWNER           REQUEST     JOB_ID        REQ/ALLOC  HARD/SOFT
sjh             buildDQ     14            running
pogo% []
```

C. Suspending and restarting the queue on `pogo` using `qmod`.

```
pogo% qmod -s pogo
pogo% qstat -G sun
QUEUE           GROUP       ARCHITECTURE  STATUS     OWNER    REQUEST   JOB_ID

pogo            sun         SUN3          suspended  sjh      buildGC   13

churchy         sun         SUN3          running    sjh      buildDQ   14
pogo% qmod -e pogo
pogo% qstat -G sun
QUEUE           GROUP       ARCHITECTURE  STATUS     OWNER    REQUEST   JOB_ID

pogo            sun         SUN3          running    sjh      buildGC   13
churchy         sun         SUN3          running    sjh      buildDQ   14
pogo% []
```

D. Deleting the GCC building job on `pogo`, which kills the job.

```
pogo% qdel 13
deleted job 13
pogo% qstat -G sun
QUEUE           GROUP       ARCHITECTURE  STATUS     OWNER    REQUEST   JOB_ID

pogo            sun         SUN3          idle
churchy         sun         SUN3          running    sjh      buildDQ   14
pogo% []
```

E. Looking at accounting for use of the whole cluster by the user sjh.

```
pogo% qacct -u sjh
USER      REAL      USER      SYSTEM
========================================
```

```
sjh       871       31       58
pogo% []
```

Table 1. Options to `qconf` that can modify the behavior of the queue and limit access to queues.

-ah *hostname*
: adds the host to the trusted host list (adds it to the cluster)

-am *user*
: adds the user to the DQS manager list

-ao *user*
: adds the user to the DQS operator list

-aq
: allows the user to edit the default queue configuration; it opens up a `vi` session (or uses whatever is defined as the `EDITOR` variable).

-au *user* [*queue*]
: add the user to the access list for the cluster or the queue

-dh *hostname*
: remove the host from the cluster

-dm *user*
: delete a manager from the cluster

-do *user*
: delete an operator from the cluster

-dq *queuename*
: remove the queue from the configuration. Active jobs will run to completion

-du *user* [*queue*]
: remove the user from the access list for the system or queue

-mc
: retrieve the current global configuration from the cluster and let the user change it.

-ql
: returns a list of users with entries in the DQS user-access database.

-qmon
: show a brief version of the system status and statistics as would be shown by the `qmon` command

-sh
: show the trusted host list

-sm
: show the manager list

-so
: show the operator list

-su *user*
: show the user's access permissions

Go To Content

Search

Last Modified: Thursday, 30-Nov-95 16:53:00 PST