Running On Frost

Contents

Cobalt is being used to manage Frost. It handles partition sizes of 4096, 2048, 1024, 512, 256, 128, 64, and 32 nodes. The job queuing commands are located under `/usr/bin` on the login node.

Job Status with Cobalt

Use the **cqstat** command to see what jobs are queued or running. `WallTime` is in `hours:minutes:seconds`.

```
$ cqstat
JobID   User      WallTime  Nodes  State    Location
============================================================
188178  luke      24:00:00  64     running  64_R001_J207_NA
188313  luke      24:00:00  64     running  64_R001_J203_N8
188345  luke      24:00:00  64     running  64_R001_J106_N2
188348  luke      24:00:00  64     running  64_R001_J102_N0
188400  hsolo     06:00:00  8      running  32_R001_J113_N5
188403  hsolo     04:00:00  64     queued   None
188404  hsolo     04:00:00  64     queued   None
188408  hsolo     04:00:00  64     queued   None
188409  hsolo     04:00:00  64     queued   None
188428  hsolo     06:00:00  8      running  32_R001_J117_N7
188445  hsolo     04:00:00  64     queued   None
188450  yoda      24:00:00  128    running  128_R000_J102_N0
188451  yoda      24:00:00  128    running  128_R000_J111_N4
188452  yoda      24:00:00  128    running  128_R000_J210_NC
188453  yoda      24:00:00  128    running  128_R000_J203_N8
```

The `-f` flag gives more info:

```
$ cqstat -f
JobID   JobName  User      WallTime  QueuedTime  RunTime   Nodes  State    Location          Mode  Procs
Queue    Kernel   StartTime          Index
=============================================================================================================
====================================
188178  -        luke      24:00:00  00:03:15    23:04:06  64     running  64_R001_J207_NA   vn    127
default  ZeptoOS  04/22/08 13:09:01  None
188313  -        luke      24:00:00  00:00:21    14:41:22  64     running  64_R001_J203_N8   vn    127
default  default  04/22/08 21:31:45  None
188345  -        luke      24:00:00  00:00:25    11:15:10  64     running  64_R001_J106_N2   vn    127
default  default  04/23/08 00:57:56  None
188348  -        luke      24:00:00  00:00:28    09:57:37  64     running  64_R001_J102_N0   vn    127
default  default  04/23/08 02:15:30  None
188400  -        hsolo     06:00:00  00:00:15    02:55:05  8      running  32_R001_J113_N5   vn    16
default  default  04/23/08 09:18:02  None
188403  -        hsolo     04:00:00  02:55:16    N/A       64     queued   None              vn    128
default  ZeptoOS  N/A                None
188404  -        hsolo     04:00:00  02:55:15    N/A       64     queued   None              vn    128
default  default  N/A                None
188408  -        hsolo     04:00:00  02:35:07    N/A       64     queued   None              vn    128
default  default  N/A                None
188409  -        hsolo     04:00:00  02:15:01    N/A       64     queued   None              vn    128
default  default  N/A                None
188428  -        hsolo     06:00:00  00:04:38    01:50:18  8      running  32_R001_J117_N7   vn    16
default  default  04/23/08 10:22:49  None
188445  -        hsolo     04:00:00  01:24:50    N/A       64     queued   None              vn    128
default  default  N/A                None
188450  -        yoda      24:00:00  00:00:18    00:07:01  128    running  128_R000_J102_N0  vn    256
default  default  04/23/08 12:06:06  None
188451  -        yoda      24:00:00  00:00:20    00:04:26  128    running  128_R000_J111_N4  vn    256
default  default  04/23/08 12:08:40  None
188452  -        yoda      24:00:00  00:00:22    00:02:04  128    running  128_R000_J210_NC  vn    256
default  default  04/23/08 12:11:02  None
188453  -        yoda      24:00:00  00:00:21    00:01:16  128    running  128_R000_J203_N8  vn    256
default  default  04/23/08 12:11:51  None
```

Queues with Cobalt

There are 3 primary queues in use on Frost.

- `default` for long running jobs
- `debug` for small, short jobs
- `JumboFridays` for half or full rack jobs during the big run window (usually 8-10am on Fridays.)
  Use the **cqstat -q** command to show queue state and restrictions. The restrictions are detailed in the cqstat man page (`man cqstat`). You can check which queues a partition is a member of with the partlist command.

```
$ cqstat -q debug default JumboFridays
Name          State    Users  MinTime  MaxTime   MaxRunning  MaxQueued  MaxUserNodes  TotalNodes  Priority
=============================================================================================================
JumboFridays  running  None   None     03:00:00  4           None       None          None        0
debug         running  None   None     02:00:00  4           50         None          None        0
default       running  None   None     24:00:00  4           None       None          None        0
```

Submitting a Job with Cobalt

Use the **cqsub** command to submit a job to the queue.

```
cqsub <required flags> <optional flags> executable
```

Required flags

Required Flag	Description
`-n NP`	where `NP` is the number of nodes
`-t TIME`	where `TIME` is how much time your job will take to run, in `hours:minutes:seconds` format (though the `seconds` field is ignored.)

Example: Simple submission

```
$ cqsub -n 32 -t 00:10:00 example.rts
submitting walltime=10.0 minutes
162
```

In this example STDOUT is stored in `162.output`, and STDERR is stored in `162.error` in the current directory.

Optional flags

Some optional flags (see **man cqstat** for a full descrption of all flags):

Flag	Description
`-O OUTPUT_PREFIX`	where `OUTPUT_PREFIX` is the name of the output prefix, which means the output files will be named `OUTPUT_PREFIX.output` and `OUTPUT_PREFIX.error` for STDOUT and STDERR, respectively. If `OUTPUT_PREFIX` is not specified, the output will be placed in `<jobid>.output` and `<jobid>.error`
`-i INPUT_FILE`	where `INPUT_FILE` is the name of the file to be read from STDIN
`-C CWD`	where `CWD` is the working directory for the code to run in (not necessarily where the executable resides). The output files and any other files that are opened without specifying a path will be stored in the directory specified by CWD.
`-m MODE`	where `MODE` is `co` (coprocessor mode) or `vn` (virtual-node mode)
`-c COUNT`	where `COUNT` is the number of processors to use. By default this is equal to the number of nodes in coprocessor mode, and twice the number of nodes in virtual-node mode. This option is generally used in conjunction with `-m vn` to specify an odd number of processes in virtual-node mode.
`-N email address`	sends an email message at the start and stop of the job to the specified email address. Multiple email addresses, separated by colons, can be specified.
`--dependencies <jobid1>:<jobid2>`	forces the job to wait to run until the listed jobids have finished running

Example: Specify 55 processes in virtual-node mode

```
$ cqsub -n 28 -c 55 -m vn -t 00:10:00 example.rts
```

Managing Jobs with Cobalt

Delete a job

Use the **cqdel** command to cancel a job that has been submitted to the queue.

```
$ cqdel 162
      Deleted Jobs
JobID   User
===============
  162  voran
```

It may take a bit for the job to be deleted if it is running, but you can check the `.error` file to see if the job is being deleted

Move a job

Use the **qmove** command to move a job to another queue, after the job has been submitted.

```
$ qmove debug 118399
    Moved Jobs to queue: debug
      118399
```

Hold and release jobs

Use the **qhold** and **qrls** commands to hold and release your jobs that are in the queue. When a job is in a hold state, it will not be scheduled to run by the scheduler.

```
$ qhold 118399
   Placed user hold on jobs:
      188499
$ qrls 118399
   Removed user hold on jobs:
      188499
```

Alter jobs

The **qalter** command allows you to change many attributes of a job that you previously submitted to a queue, including walltime, number of nodes

```
$ qalter -t 60 -n 32 --mode vn 118399
```

See **man qalter** for more information.

Wait on a job

The **cqwait** command can be used to wait for a job to either start running, or complete. If cqwait is used with no arguments, cqwait will not return until the job has finished running, or has left the queue because of a deletion. If cqwait is used with the --start option, cqwait will return when the job starts running in the queue.

```
$ cqwait 182949
```

System Availability Commands

Use the **partlist** command to see what partitions are available and which are in use.

```
$ partlist
Name              Queue                                          State                        Backfill
=============================================================================================================
NCAR_R00          default:debug:teragrid:JumboFridays:admin  blocked (128_R000_J102_N0)  -
NCAR_R000         default:debug:teragrid:JumboFridays:admin  blocked (128_R000_J102_N0)  -
NCAR_R001         default:debug:teragrid:JumboFridays:admin  blocked (128_R001_J102_N0)  -
256_R000_J102_N0  default:teragrid:debug:JumboFridays:admin  blocked (128_R000_J102_N0)  -
256_R000_J203_N8  default:teragrid:debug:JumboFridays:admin  blocked (128_R000_J203_N8)  -
256_R001_J102_N0  debug:JumboFridays:admin                   blocked (128_R001_J102_N0)  -
256_R001_J203_N8  debug:JumboFridays:admin                   blocked (64_R001_J203_N8)   -
128_R000_J102_N0  debug:default:teragrid:admin               busy                        -
128_R000_J111_N4  debug:default:teragrid:admin               idle                        4:23
128_R000_J203_N8  debug:default:teragrid:admin               busy                        -
128_R000_J210_NC  debug:default:teragrid:admin               busy                        -
128_R001_J102_N0  debug:default:teragrid:admin               busy                        -
128_R001_J111_N4  debug:default:teragrid:admin               blocked (64_R001_J111_N4)   -
128_R001_J203_N8  debug:default:teragrid:admin               blocked (64_R001_J203_N8)   -
128_R001_J210_NC  debug:default:teragrid:admin               blocked (64_R001_J210_NC)   -
...
```

This example shows that three of the four 128-node partitions in midplane R000 are in use by jobs ('busy'), which consequently block the full rack partition ('NCAR_R00'), and the 512-node and 256-node partitions on midplane R000 ('NCAR_R000, 256_R000_J102_N0, and 256_R000_J203_N8, respectively). Note that partition 128_R000_J111_N4 is idle, and the scheduler has calculated a 4h23m backfill window for it. This means that if a user submits a 128-node job with a requested walltime of less than 4h23m, the job can be run right away.

You may also use the **nodes** helper script which will display only the currently free partitions, along with their associated backfill windows. **nodes -v** lists all of the partitions (from the **partlist -l** output) and indents the output to show the partition hierarchy.

Use the **showres -l** command to see what system reservations are in place.

When your job isn't running...

If your job is sitting 'queued' and it seems like it should be running, there are a few things to check before sending a message to frost-help@ucar.edu:

- Check the output from **partlist** or **nodes**. That will tell you which partitions are idle (available for jobs to run)
- If there are reservations in **showres** output that start before your job's walltime would end, then your job will not run
- If your job is listed in the **maxrun_hold** state, then your other jobs in the queue have hit the max running limit for that queue. See queue limits with the **cqstat -q** command.
- If your job is listed in the **hold** state, then your job has been held by an administrator. Contact frost-help@ucar.edu to see why.
- Your job may be limited by queue restrictions. Check the queue restrictions with **cqstat -q**.