Slurm and Supercomputing Scheduling

Shelley Knuth
Research Computing
University of Colorado-Boulder

Outline

- What is job scheduling?
- Software
  - Moab/Torque
  - Slurm
- Examples

Janus Supercomputer

What is Job Scheduling

- Supercomputers usually consist of many nodes
- Users submit jobs that may run on one or multiple nodes
- Sometimes these jobs are very large; sometimes there are many small jobs
- Need software that will distribute the jobs appropriately
  - Make sure the job requirements are met
    - Reserve nodes until enough are available to run a job
    - Account for offline nodes
- Also need software to manage the resources
- Integrated with scheduler
  - http://www.glue.umd.edu/hpcc/help/slurm-vs-moab.html

Job Scheduling

- On a supercomputer, jobs are scheduled rather than just run instantly at the command line
  - People “buy” time to use the resources
  - Shared system
  - Request the amount of resources needed and for how long
  - Jobs are put in a queue until resources are available
  - Once the job is run they are “charged” for the time they used

Job Scheduling - Priority

- What jobs receive priority?
  - Can depend on the center
  - Can arrange for certain people who “pay more” receive priority
  - Generally though based on job size and time of entry
- Might have different queues based on different job needs
- Can receive priority on a job by creating a reservation

Job Schedulers

- Jobs on supercomputers are managed and run by different software
  - Previously, jobs on RC resources were submitted using Torque and scheduled with Moab
  - Licensing, performance, and functionality issues have caused us to change to Slurm
  - SLURM = Simple Linux Utility for Resource Management
    - Open source
    - Increasingly popular at other sites
    - Stampede uses Slurm

Running Jobs

What is a “job”?

- Interactive jobs
  - Work interactively at the command line of a compute node
  - Slurm command:
  - RC: salloc --qos=janus-debug
  - Stampede: srun -p development -t 0:30:00 -n 32 --pty /bin/bash -l

- Batch jobs
  - Submit job that will be executed when resources are available
  - Create a text file containing information about the job
  - Submit the job file to a queue
  - Slurm command: sbatch --qos=<queue> jobfile

Torque/PBS

- Torque is a software package commonly used on clusters to manage jobs and compute resources (nodes)
- Called a “resource manager”
- Keeps track of what nodes are busy/available, and what jobs are queued or running
- Provides a user interface for submitting or deleting jobs
- Uses information about each job’s requirements as provided by the user through PBS directives PBS=Portable Batch System
    - Directives used to request resources for a job and to define other aspect’s of the job’s behavior

Moab/Maui

- Scheduling software is needed to tell the resource manager when to run each job
- The Moab software package is commonly used on clusters to schedule jobs
  - Receives info from the resource manager about available resources and job requirements
  - Can handle job prioritization and reservations well
- Maui is the open-source predecessor of Moab

Slurm

- Simple Linux Utility for Resource Management
- Slurm is a resource manager much like Torque
- Also includes a sophisticated scheduler so Moab is not needed
- Open source

- Other scheduling software you may encounter:
  - LSF
  - LoadLeveler
  - GridEngine (SGE, UGE)

Queues

- In Slurm, there are several ways to define a “queue”
- Clusters may have different queues set up to run different types of jobs
  - Certain queues might exist on certain clusters/resources
  - Other queues might be limited by maximum wall time
- On Janus, we use a “quality of service” for each queue
  - aka “QOS”
- On Stampede, a “partition” (or set of nodes) corresponds to a queue

Moab/Torque and Slurm Commands

- Moab/Torque

```
module load torque
module load moab

qsub –q janus-debug test.sh

qstat –u $USER
```

- Slurm

```
module unload torque
module unload moab
module load slurm

sbatch –qos=janus-debug test.sh

squeue –u $USER
```

More at https://www.rc.colorado.edu/support/examples/slurmtestjob

Moab/Torque and Slurm Directives

- Moab/Torque

```
#PBS –l nodes=1:ppn=1, walltime=00:10:00

#PBS -q janus-debug

#PBS –o testjob.out

#PBS -N matlab_test_serial

#PBS -m be

#PBS –M ralphie@colorado.edu
```

- Slurm

```
#SBATCH –N 1
#SBATCH --time=0:10:00

#SBATCH --qos=janus-debug

#SBATCH -o testjob.out

#SBATCH –J matlab_test_serial

#SBATCH --mail-type begin, end

#SBATCH --mail-user ralphie@colorado.edu
```

More at https://www.rc.colorado.edu/support/examples/slurmtestjob

Other Handy Job Features

- Job arrays – manage a collection of jobs that all have the same options
- Job dependencies – one job can start running only after another job has finished successfully
- File staging – copying input or output files to or from a scratch disk space when a job starts or stops

EXAMPLES

Submit Batch Job example

- **Batch Script:**

```
#!/bin/bash
#SBATCH –N 2                       # Number of requested nodes
#SBATCH --ntasks-per-node=12       # number of cores per node
#SBATCH --time=1:00:00             # Max walltime
#SBATCH --job-name=SLURMDemo       # Job submission name
#SBATCH --output=SLURMDemo.out     # Output file name
###SBATCH -A <account>             # Allocation
###SBATCH --mail-type=end          # Send Email on completion
###SBATCH --mail-user=<your@email> # Email address
module load openmpi/openmpi-1.8.0_intel-13.0.0
mpirun ./hello
```

- **Submit the job:**
  - `sbatch --qos janus-debug slurmSub.sh`

- **Check job status:**
  - `squeue –q janus-debug`
  - `cat SLURMDemo.out`

Questions?

- More examples to come after presentations!

Peter.Ruprecht@colorado.edu

Shelley.Knuth@colorado.edu