Day 2

- Brief Introduction
- Bad News
- Good News
- Software Environment
- PBS Directives
- Client Commands
- Job Priority
- MPI
Brief Introduction

• CRI
• CRI Infrastructure
• My Team
• Me
• Yesterday’s session
Bad News

• Users not understanding the software environment (later!)
• Scheduling slow for clinical work
• Resource manager problems
• Scheduler crash
• Scratch space issues
• Job Array Problems
• Job Geometry Changing
Software Environment Confusion

• Problem:
  – Users seem to be confused with how the software environment (Lmod) works

• Solution:
  – Cover it again in today’s session
  – User Guide
Clinical Scheduling

• Problem
  – Some clinical work that influences patient care has been slow to run
  – Had purchased exclusive nodes on Tarbell
  – Feel users can take over large memory and mid-tier nodes
  – Giving them exclusive nodes on Gardner would let them

• Solution
  – Partitions
  – Reservations
Proposed Limits

• Overall
  – Processor Cores: 500
  – Memory: 2 TB

• Standard Nodes
  – Processor Cores: 500

• Mid-Tier Nodes
  – Processor Cores: 100 cores

• High-Tier Nodes
  – Processor Cores: 14 cores
Torque Issues

• Problem:
  – Error message: “server is shutting down”
  – Problem with the init script used to start/stop Torque

• Solution:
  – Wrote new init script
  – Fixed in the latest version of Torque (released 3/31/17)
Torque Issues

• Problem:
  – PBS Server crashed (three times!!!)
  – This is an issue with validating PBS directives
    • #PBS –t 1-
    • #PBS walltime=1;00:00

• Solution:
  – Hotfix (fixed some problems)
  – Fixed in the latest Torque Release
Moab Crash

• Problem:
  – Moab crashed hard and was down for close to 8 hours
  – This was due to a corrupt job that needed to be tracked down

• Solution:
  – Hotfix (included in next release as well)
  – Still have yet to determine what caused the corrupt job
Scratch Space Issues

• Problem:
  – Two OS Hard Drives were lost on our Scratch Servers
  – Replacements causing problems too

• Solution:
  – Servers have been removed for maintenance
  – Storage at half capacity
Scratch Space Issues

• Problem:
  – Scratch Space can be slow at times
    • Only have two scratch servers instead of four
    • Negative cache

• Solution:
  – Regain full strength in the scratch environment
  – Lengthen time for cache
  – Make sure cache is always up-to-date
Job Array Problems

• Problem:
  – Job Arrays tend to get stuck
  – Will show up in Torque as running but not Moab

• Solution:
  – Restart pbs_server
  – Fixed in the next release (maybe)
Job Geometry Changes

• Problem:
  – Jobs changed from 4 cores to 1 core
  – Only happened to two users

• Solution:
  – Can be fixed on the sysadmin level
  – Working with Adaptive to handle the issue
Good News

- 113 users are now on Gardner
- 74% of CPU hours for March were on Gardner
- Amount of FLOPS completed in 3 months on Gardner is equivalent to 8 months on Tarbell
- Half of Tarbell is decommissioned
- Most of the positive feedback on Gardner has been computational speed
- New contract with Adaptive Computing (3 years)
- Tony
Tarbell Plans

• Half the cluster decommissioned on 3/31/17
• Who should be using Tarbell
  – Have a Gardner account and a qualifying analysis between the two systems
  – Those who have applied and are waiting for a Gardner account
  – Those who have purchased exclusive nodes on Tarbell
  – Graham School biomedical informatics class
• June 2017 – Rest of Tarbell will be decommissioned
Software Environment

• Tarbell -> Environment Modules
  – Flat module system
  – Modules written in TCL
  – Last Update: December 2012

• Gardner -> Lmod
  – Hierarchical module system
  – Modules written in Lua
  – Last Update: August 2016
Lmod Basics

• See which modules are available to be loaded
  – module avail

• Load packages
  – module load <package1> <package2>

• See which packages are loaded
  – module list

• Unload a package
  – module unload <package>
Lmod Basics

• Swap compilers
  – `module swap gcc/5.4.0 gcc/6.1.0`

• Find a module by keyword
  – `module keyword alignment`

• List all possible versions of a module
  – `module spider bwa`

• Print detailed information for a specific module
  – `module spider bwa/0.7.5`
Lmod Basics

• Save your loaded modules as the default
  – module save

• Restore your default modules
  – module restore
Lmod Basics

• Clean up environment
  – module purge

• Save a named collection
  – module save <collection>

• Restore a named collection
  – module restore <collection>

• List all the modules in a collection
  – module describe <collection>
Lmod Basics

• Print help message
  - module help <package>

• Print description
  - module whatis <package>

• Get help on Lmod
  - module help
Safety Features of Lmod

- Users can only load one version of a module at a time
  - For example, only one BWA module can be loaded
- Can only load one module from a family at a time
  - Compilers
  - MPI
- Conflict
- Prereq
- Prereq_any
How do I find?

• Perl Modules
  – instmodsh

• Python packages
  – pip freeze

• R packages
  – installed.packages()
Elog Example

• Where do we need help?
PBS Directives

- procs vs. ppn
  - Use procs only for MPI
- mem vs. pmem
- Accounting - #PBS -A
- gpus
- features
- mail - #PBS –M, #PBS –m
- Priority - #PBS –p
- Environment variables - #PBS –V, #PBS -v
Client Commands

- Job monitoring (all jobs)
  - qstat
  - showq

- Job monitoring (individual jobs)
  - qstat -f <jobid>
  - checkjob <jobid>

- Partition monitoring (immediate available resources)
  - showbf

- When will my job start?
  - showstart <jobid>
  - Evaluates historical data, reservations, priority backlogs
mjobctl

• Cancel Job:  -c <jobid>
• Hold Job:   -h user <jobid>
• Rerun Job: -e <jobid>
Reservations

• When to request them?
  – Deadline
  – Large batch of jobs

• showres
  – Shows all reservations that apply to you
Job Arrays

• qsub –t <range>%<limit>

• $PBS_ARRAYID

• How to track your job arrays?
  – showq
  – checkjob –v
  – qstat -t
Job Dependencies

- `#PBS -W depend=type:jobid[:jobid[:jobid...]]`

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>after</td>
<td>This job may be scheduled after jobs jobid have started</td>
</tr>
<tr>
<td>afterok</td>
<td>This job may be scheduled after jobs jobid have completed with no errors</td>
</tr>
<tr>
<td>afternotok</td>
<td>This job may be scheduled after jobs jobid have completed with errors</td>
</tr>
<tr>
<td>afterany</td>
<td>This job may be scheduled after jobs jobid have completed with or without errors</td>
</tr>
<tr>
<td>before</td>
<td>After this job begins, jobs jobid may be scheduled</td>
</tr>
<tr>
<td>beforeok</td>
<td>After this job completes without errors, jobs jobid may be scheduled</td>
</tr>
<tr>
<td>beforenotok</td>
<td>After this job completes with errors, jobs jobid may be scheduled</td>
</tr>
<tr>
<td>beforeany</td>
<td>After this job completes with or without errors, jobs jobid may be scheduled</td>
</tr>
</tbody>
</table>
Quality of Service

- `#PBS -l qos=<qos_name>`
- Standard = 1000
- Premium = 2000
- Biocore = 3000
- Deadline = 4000
- VIP = 5000
Job Priority

• How does that work.

• Starts at 1000

• Add one every scheduler iteration
MPI Example

• Hello World
• Calculating the volume of a molecule
Upcoming Work

- Trickle – qsub replacement
- Dbuilder
- Data staging
- Software statistics (Tony)
- Viewpoint
- Remote Visualization (?)
- User Guide
- Elog
- Future training (?)