PACE

Instructional Cluster Environment (ICE) Orientation

Emanuele Di Lorenzo
Giovanni Liguori

www.pace.gatech.edu

What is PACE

A **P**artnership for an **A**dvanced **C**omputing **E**nvironment

- Provides faculty and researchers vital tools to accomplish the Institute's vision to define the technological research university of the 21st century.
- A strong HPC environment through a tight partnership with our world-class students, researchers and innovators to help them make the greatest impact with their work.

ICE Accounts

- Automated based on class enrollments
- Departments manage access groups without PACE's involvement
- COC has their own resources (coc-ice), other departments share PACE resources (pace-ice)

3-Tier Help Structure

1. Students: First, reach out to Instructors/TAs (no direct tickets to PACE)

2. Instructors/TAs can contact their departmental contact:

   - COC : David Mercer
   - ECE : David Webb
   - COS : Mark Keever & Mack Jenkins

3. Instructors, TAs and dpt. contacts can open tickets:

   pace-support@oit.gatech.edu

   (Please mention that it's about the educational cluster)

Accessing Clusters

- You will need an SSH Client (a.k.a. terminal):
  - Windows: MobaXterm , PuTTy, Xming (free), X-win32 (software.oit.gatech.edu)
  - MacOSX: iTERM2, Terminal, XQuartz
  - Linux: System-default terminal (gnome/KDE)

Needed for GUI access

- SSH access:

```
ssh -X <GT_user_ID>@pace-ice.pace.gatech.edu
```

- You need to be on campus, or connect via VPN.

  For information on VPN access, see:

  http://faq.oit.gatech.edu/search/node/vpn

COC-ICE Queues

login node: **pace-ice.pace.gatech.edu**

One queue for Ocean Modeling:

1. **eas-pace-ice:** Everyone

COC-ICE Resources

You can check the available resources and their current status using "pace-check-queue coc-ice"

```
$ pace-check-queue eas-pace-ice

=== coc-ice Queue Summary: ====
       Last Update                              : 02/02/2018 14:15:02
       Number of Nodes (Accepting Jobs/Total) : 22/22 (100.00%)
       Number of Cores (Used/Total)           : 0/220 ( 0.00%)
       Amount of Memory (Used/Total) (MB)     : 25846/2753414 ( 0.94%)
================================================================================
  Hostname       tasks/np Cpu%  loadav%  used/totmem(MB)   Mem%   Accepting Jobs?
================================================================================
  rich133-s30-10     0/12    0.0      1.8      1266/131128      1.0    Yes (free)
  rich133-s30-11     0/12    0.0      0.2      1251/131128      1.0    Yes (free)
  rich133-s30-12     0/12    0.0      1.7      1231/131128      0.9    Yes (free)
  rich133-s30-13     0/12    0.0      0.8      1243/131128      0.9    Yes (free)
  rich133-s30-14     0/12    0.0      4.9      1243/131128      0.9    Yes (free)
  rich133-s30-15     0/12    0.0      0.3      1239/131128      0.9    Yes (free)
  rich133-s30-16     0/12    0.0      0.0      1232/131128      0.9    Yes (free)
  rich133-s30-17     0/12    0.0      1.3      1233/131128      0.9    Yes (free)
  rich133-s30-18     0/12    0.0      1.2      1235/130991      0.9    Yes (free)
  rich133-s30-19     0/12    0.0      0.0      1235/130991      0.9    Yes (free)
  rich133-s30-20     0/12    0.0      5.1      1240/131128      0.9    Yes (free)
  rich133-s30-21     0/12    0.0      0.7      1234/131128      0.9    Yes (free)
  rich133-s30-22     0/12    0.0      0.0      1257/131128      1.0    Yes (free)
  rich133-s42-21     0/8     0.0      0.0      1281/131128      1.0    Yes (free)
  rich133-s42-22     0/8     0.0      1.5      1202/131128      0.9    Yes (free)
  rich133-s42-23     0/8     0.0      0.0      1202/131128      0.9    Yes (free)
  rich133-s42-24     0/8     0.0      0.2      1205/131128      0.9    Yes (free)
  rich133-s42-25     0/8     0.0      0.8      1205/131128      0.9    Yes (free)
  rich133-s42-26     0/8     0.0      1.4      1202/131128      0.9    Yes (free)
  rich133-s42-27     0/8     0.0      0.0      1206/131128      0.9    Yes (free)
  rich133-s42-28     0/8     0.0      0.0      1204/131128      0.9    Yes (free)
```

- 12x cores
- 128GB RAM/node
- 2x Tesla K40m GPUs

- 8x cores
- 128GB RAM/node
- 1x Tesla P100 GPU

PACE-ICE Queues

login node: **pace-ice.pace.gatech.edu**

- Pace-ice queues (sharing the same cluster nodes):

2. **eas-pace-ice :** Only TAs, admins and Faculty

➡ 48 hrs walltime, 98 cores max

- Walltime can be adjusted per request.
- 'pace-check-queue' shows all 7 nodes and 196 cores, but submissions with 98+ cores will hang:

```
$ qsub -q eas-pace-ice -I -l nodes=5:ppn=28
qsub: waiting for job 111.pace-ice-sched.pace.gatech.edu to start

each node has 28 cpus (or cores)
```

PACE-ICE Resources

You can check the available resources and their current status using:

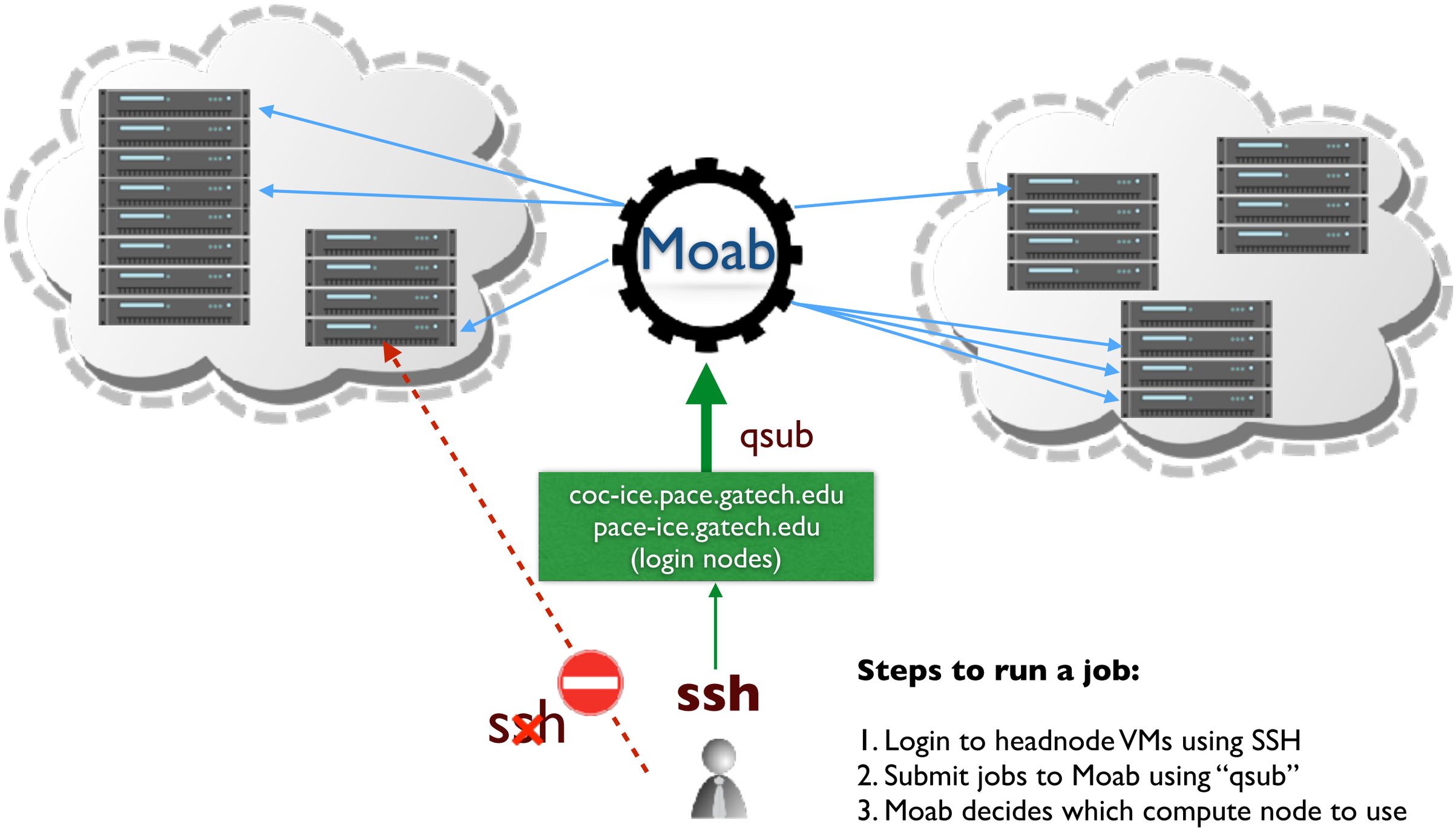
pace-check-queue <queue_name>

```
$ pace-check-queue phys-pace-ice

=== phys-pace-ice Queue Summary: ====
     Last Update                              : 08/14/2018 11:00:02
     Number of Nodes (Accepting Jobs/Total)   : 7/7 (100.00%)
     Number of Cores (Used/Total)             : 0/196 ( 0.00%)
     Amount of Memory (Used/Total) (MB)       : 18839/917745 ( 2.05%)
================================================================================
  Hostname       tasks/np Cpu%  loadav%  used/totmem(MB)   Mem%   Accepting Jobs?
================================================================================
rich133-c32-10-r   0/28    0.0     0.4      2700/131126     2.1    Yes (free)
rich133-c32-11-l   0/28    0.0     0.1      2698/131126     2.1    Yes (free)
rich133-c32-11-r   0/28    0.0     0.0      2619/131126     2.0    Yes (free)
rich133-c32-12-l   0/28    0.0     1.5      2791/130989     2.1    Yes (free)
rich133-c32-12-r   0/28    0.0     0.6      2588/131126     2.0    Yes (free)
rich133-c32-13-l   0/28    0.0     0.1      2721/131126     2.1    Yes (free)
rich133-c32-13-r   0/28    0.0     1.0      2722/131126     2.1    Yes (free)
```

- 28x cores
- 128GB RAM/node
- No GPUs (for now)

Summary



Steps to run a job:

1. Login to headnode VMs using SSH
2. Submit jobs to Moab using "qsub"
3. Moab decides which compute node to use
4. Your job starts on the node that Moab picks

Login nodes vs. Compute Nodes

- **Login Nodes:** For logging in and submitting jobs
  - Shared by all users
  - Good for compiling, editing, debugging, etc.
  - Not good for actual computations or visualization

- **Compute Nodes:** For running codes/simulations
  - No direct access by users
  - Allocated per-job by the scheduler

ICE Storage

- COC has 41TB total capacity owned by the department, 10GB per student.
- PACE-ICE queues come with 10GB per student (provided by PACE)
- All data are accessible from all nodes (login and compute nodes)
- Complete separation from the rest of PACE (you can have accounts on both)
- **Mounter** applications mount remote storage so you can drag/drop or edit in place as if the files are on your local machine
    - Windows : webdrive (free via software.oit.gatech.edu)
    - OSX : macfusion (open source)
    - Linux : SSHFS, autofs (open source, no GUI)
- Any **SFTP** client will work with PACE. FileZilla is a free FTP tool for Windows, OSX and Linux.
- Use the login nodes as the server for configuring any of these clients.

Running Jobs: Overview

- Users make requests to **Moab scheduler** specifying the requirements of the code:
    - The number of Nodes and/or Cores per node.
    - The total Memory or Memory-per-core.
    - An estimated Runtime (walltime, not CPU time)
    - Specific hardware resources, e.g. GPU

- Allocated resources can only be used by the user for the duration of requested walltime. This is the only time users can directly login to compute nodes.

Operation Modes

Two modes of operation:

- **Batch:** Submit & forget. Job waits in the queue until resources become available, runs, emails user on exit.
- **Interactive:** Allows interactive use, no different than remotely using any workstation

(required for using GUI, such as MATLAB, R, COMSOL, ANYSYS, visualization, etc.)

Submitting Batch Jobs

- Everything needs to be scripted. Not for codes that require user interaction (e.g. press ‘y’ to continue).
- A ‘PBS script’ that includes resource requirements, environmental settings, and tasks.
- Use ‘qsub’ to submit the job.

```
qsub example_PBS_Script.pbs
```

- The output and error logs are printed on files, as they would appear on the screen.

PBS Script Example

comment

command

```
# This is an example PBS script
#PBS -N hello
#PBS -l nodes=2:ppn=4
#PBS -l pmem=2gb
#PBS -l walltime=15:00:00
#PBS -q eas-pace-ice
#PBS -j oe
#PBS -o myjob.out
#PBS -m abe
#PBS -M youremail@gatech.edu

cd ~/test_directory
echo "Started on `/bin/hostname`"
module load gcc mvapich2/2.0ga
mpirun -np 8 ./hello
```

PBS Script Example

```
# This is an example PBS script
#PBS -N hello
#PBS -l nodes=2:ppn=4
#PBS -l pmem=2gb
#PBS -l walltime=15:00:00
#PBS -q eas-pace-ice
#PBS -j oe
#PBS -o myjob.out
#PBS -m abe
#PBS -M youremail@gatech.edu

cd ~/test_directory
echo "Started on `/bin/hostname`"
module load gcc mvapich2/2.0ga
mpirun -np 8 ./hello
```

A name for this run, can be anything

PBS Script Example

```
# This is an example PBS script
#PBS -N hello
#PBS -l nodes=2:ppn=4
#PBS -l pmem=2gb
#PBS -l walltime=15:00:00
#PBS -q eas-pace-ice
#PBS -j oe
#PBS -o myjob.out
#PBS -m abe
#PBS -M youremail@gatech.edu

cd ~/test_directory
echo "Started on `/bin/hostname`"
module load gcc mvapich2/2.0ga
mpirun -np 8 ./hello
```

2 nodes, 4 cores in each

PBS Script Example

```
# This is an example PBS script
#PBS -N hello
#PBS -l nodes=2:ppn=4
#PBS -l pmem=2gb
#PBS -l walltime=15:00:00
#PBS -q eas-pace-ice
#PBS -j oe
#PBS -o myjob.out
#PBS -m abe
#PBS -M youremail@gatech.edu

cd ~/test_directory
echo "Started on `/bin/hostname`"
module load gcc mvapich2/2.0ga
mpirun -np 8 ./hello
```

2GB memory per core (16GB total)

PBS Script Example

```
# This is an example PBS script
#PBS -N hello
#PBS -l nodes=2:ppn=4
#PBS -l pmem=2gb
#PBS -l walltime=15:00:00
#PBS -q eas-pace-ice
#PBS -j oe
#PBS -o myjob.out
#PBS -m abe
#PBS -M youremail@gatech.edu

cd ~/test_directory
echo "Started on `/bin/hostname`"
module load gcc mvapich2/2.0ga
mpirun -np 8 ./hello
```

15 hrs "max", after which job is killed!!

PBS Script Example

```
# This is an example PBS script
#PBS -N hello
#PBS -l nodes=2:ppn=4
#PBS -l pmem=2gb
#PBS -l walltime=15:00:00
#PBS -q eas-pace-ice
#PBS -j oe
#PBS -o myjob.out
#PBS -m abe
#PBS -M youremail@gatech.edu

cd ~/test_directory
echo "Started on `/bin/hostname`"
module load gcc mvapich2/2.0ga
mpirun -np 8 ./hello
```

queue name (replace with yours)

PBS Script Example

```
# This is an example PBS script
#PBS -N hello
#PBS -l nodes=2:ppn=4
#PBS -l pmem=2gb
#PBS -l walltime=15:00:00
#PBS -q eas-pace-ice
#PBS -j oe
#PBS -o myjob.out
#PBS -m abe
#PBS -M youremail@gatech.edu

cd ~/test_directory
echo "Started on `/bin/hostname`"
module load gcc mvapich2/2.0ga
mpirun -np 8 ./hello
```

Put **o**utput and **e**rror files in specified format

PBS Script Example

```
# This is an example PBS script
#PBS -N hello
#PBS -l nodes=2:ppn=4
#PBS -l pmem=2gb
#PBS -l walltime=15:00:00
#PBS -q eas-pace-ice
#PBS -j oe
#PBS -o myjob.out
#PBS -m abe
#PBS -M youremail@gatech.edu

cd ~/test_directory
echo "Started on `/bin/hostname`"
module load gcc mvapich2/2.0ga
mpirun -np 8 ./hello
```

Notify on start, finish and error, via email

PBS Script Example

```
# This is an example PBS script
#PBS -N hello
#PBS -l nodes=2:ppn=4
#PBS -l pmem=2gb
#PBS -l walltime=15:00:00
#PBS -q eas-pace-ice
#PBS -j oe
#PBS -o myjob.out
#PBS -m abe
#PBS -M youremail@gatech.edu

cd ~/test_directory
echo "Started on `/bin/hostname`"
module load gcc mvapich2/2.0ga
mpirun -np 8 ./hello
```

Actual Computation

Requesting GPU nodes

- Currently COC-ICE only (PACE-ICE GPUs are on the way!)
- Add your request to PBS scripts as follows:

```
#PBS -q coc-ice
#PBS -l nodes=1:ppn=4:gpus=2:exclusive_process
```

- Alternatively, you can request a particular model (teslak40 and teslap100):

```
#PBS -l nodes=1:ppn=4:gpus=2:teslak40:exclusive_process

or

#PBS -l nodes=1:ppn=4:gpus=1:teslap100:exclusive_process
```

Interactive Jobs

- Same PBS commands, but this time on the command line:

Allows GUI

commas to bind multiple values for a parameter (-l)

```
qsub -I -X -q coc-ice -l nodes=2:ppn=4,walltime=15:00:00,pmem=2gb
```

- The scheduler logs the user in a compute node when resources become available.
- Session is terminated when:
  - The user exits
  - The terminal is closed
  - Walltime is exceeded

Monitoring Jobs

qstat lists your queued jobs and their state.

```
qstat -u <username> -n
```

```
Job ID                    Username    Queue    Jobname          SessID  NDS   TSK   Memory   Time    S   Time
------------------------ ----------- -------- ---------------- ------ ----- ------ ------ --------- - ---------
7693767.shared-sched.p  mbelgin3       force-6 Testrun_3        45994   32     64  1900m 120:00:00 R  23:55:47
   iw-h29-16+iw-h29-12+iw-h29-11
7693771.shared-sched.p  mbelgin3       cygnus64 Testrun_1       8552    16     32   48gb 120:00:00 R  23:55:17
   iw-h29-15+iw-h29-7
7693775.shared-sched.p  mbelgin3       force-6 Testrun_2        64492   16     64  1900m 120:00:00 R  23:51:47
   iw-h29-10+iw-h29-7+iw-h29-13+iw-h29-17
7693778.shared-sched.p  mbelgin3       force-6 Testrun_3L       1006    32     64  1900m 120:00:00 R  23:46:00
   iw-h29-10+iw-h29-11+iw-h29-15
7693780.shared-sched.p  mbelgin3       force-6 Testrun_3L       13369   32    128  1900m 120:00:00 R  23:45:09
   iw-h29-12+iw-h29-10+iw-h29-13+iw-h29-16+iw-h29-15+iw-h29-17+iw-h29-9
   +iw-k30-15
7695869.shared-sched.p  mbelgin3       force-6 L6.246full       38241   16    128  1900m 120:00:00 R  09:17:47
   iw-h29-16+iw-h29-11+iw-h29-7+iw-h29-13+iw-h29-8+iw-h29-9+iw-k30-21+iw-k30-22
   +iw-k30-19+iw-k30-17+iw-k30-16+iw-h31-19
```

More on PBS Jobs

- Cancelling a submitted job:

```
qdel <jobID>
```

- Querying for specific users/queues

```
showq -w class=eas-pace-ice,user=<UserID>
```
=> All jobs for the queue & user

```
showq -r -w user=<UserID>
```
=> All "running" jobs for the given user

```
showq -b -w class=<UserID>
```
=> All "blocked" jobs for the given queue

PACE Software Stack

Everything is in "`/usr/local/pacerepov1`"

- Licensed software packages:
  - Common license: Matlab, Fluent, Mathematica, Abaqus, Comsol...
  - Individual license: Vasp, Gaussian, ...
- Open source packages and HPC libraries:
  - BLAS, PETSc, NAMD, NetCDF, FFTW, BLAST, LAMMPS...
- Compilers:
  - C/C++ & Fortran: GNU, Intel, PGI, NAG
  - Parallel Compilers: OpenMP, MPICH, MPICH2, MVAPICH
  - GPU compilers: CUDA, PGI
- Scripting Languages: Python, Perl, R, ...

Modules (RHEL6 only)

- Painless configuration for software environment and switching between different versions: No more editing `PATH`, `LD_LIBRARY_PATH`, etc!
- Main Commands:
  - `module avail` : Lists all available modules that can be loaded
  - `module list` : Displays all the modules that are currently loaded
  - `module load` : Loads a module to the environment.
  - `module rm` : Removes a module from the environment
  - `module purge` : Removes all loaded modules (buggy)
- Modules may depend on, or conflict with, each other

```
$ module load matlab/r2009b

matlab/r2009b(7):ERROR:150: Module 'matlab/r2009b' conflicts with the currently loaded module(s) 'matlab/r2010a'
```

- **Must-Read:** PACE-specific use cases, examples, and gotchas

  http://www.pace.gatech.edu/using-software-modules

THANK YOU!