Programming Environment on LONI HPC Clusters

Le Yan
HPC consultant
User services group
Louisiana Optical Network Initiative
Goal of Training

• Learn how to manage software environment on LONI clusters
• Learn how to compile serial and parallel programs
• Learn to manage jobs through the queuing system
Outline

• Overview

• Hardware

• Software
  – User environment
  – Compilers
  – Application software

• Job management
Outline

• Overview
• Hardware
• Software
  – User environment
  – Compilers
  – Application software
• Job management
Two Major Types of Clusters

• Linux clusters
  – Vendor: Dell
  – OS: Linux (Red hat)
  – Processor: Intel

• AIX clusters
  – Vendor: IBM
  – OS: AIX
  – Processor: IBM
Current deployment status - Dell Linux clusters

<table>
<thead>
<tr>
<th>Name</th>
<th>Peak TeraFLOPS/s</th>
<th>Location</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queen Bee</td>
<td>50.7</td>
<td>ISB</td>
<td>Available</td>
</tr>
<tr>
<td>Eric</td>
<td>4.7</td>
<td>LSU</td>
<td>Available</td>
</tr>
<tr>
<td>Oliver</td>
<td>4.7</td>
<td>ULL</td>
<td>Available</td>
</tr>
<tr>
<td>Louie</td>
<td>4.7</td>
<td>Tulane</td>
<td>Available</td>
</tr>
<tr>
<td>Poseidon</td>
<td>4.7</td>
<td>UNO</td>
<td>Available</td>
</tr>
<tr>
<td>Painter</td>
<td>4.7</td>
<td>LaTech</td>
<td>Available</td>
</tr>
<tr>
<td>Satellite</td>
<td>4.7</td>
<td>Southern</td>
<td>To be deployed</td>
</tr>
</tbody>
</table>

Manage your account: https://allocations.loni.org/balances.php
Current deployment status - IBM AIX clusters

<table>
<thead>
<tr>
<th>Name</th>
<th>Peak TeraFLOPS/s</th>
<th>Location</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bluedawg</td>
<td>0.85</td>
<td>LaTech</td>
<td>Available</td>
</tr>
<tr>
<td>Ducky</td>
<td>0.85</td>
<td>Tulane</td>
<td>Available</td>
</tr>
<tr>
<td>Zeke</td>
<td>0.85</td>
<td>ULL</td>
<td>Available</td>
</tr>
<tr>
<td>Neptune</td>
<td>0.85</td>
<td>UNO</td>
<td>Available</td>
</tr>
<tr>
<td>Lacumba</td>
<td>0.85</td>
<td>Southern</td>
<td>Available</td>
</tr>
</tbody>
</table>

Manage your account:
https://allocations.loni.org/balances.php
Outline

• Overview

• Hardware

• Software
  – User environment
  – Compilers
  – Application software

• Job management
Generic Cluster Architecture

Definition of Cluster (from Wikipedia): A group of linked computers working together closely.
Hardware (Linux)

• Queen Bee
  – 668 nodes with each node having: 8 Intel “Cloverton” Xeons cores @ 2.33 GHz, 8 GB RAM, 36 GB HD
  – 192 TB storage

• Other LONI Linux clusters
  – 128 nodes with each node having: 4 Intel “Woodcrest” Xeons cores @ 2.33 Ghz, 4 GB RAM, 80 GB HD
  – 9 TB storage

• LONI AIX clusters
  – 14 power5 nodes with each node having: 8 IBM Power5 processors @ 1.9 GHz, 16 GB RAM
  – 280 GB storage
More on Hardware

• Technical details are usually not of interest to normal users

• A couple of things to keep in mind
  – Max usable amount of memory per node
    ▪ Linux clusters: ~6 GB for Queen Bee, ~3 GB for others
    ▪ AIX clusters: ~26 GB for Power5+ nodes (Pelican), ~13 GB for others
  – Which ARCHITECTURE to choose when trying to download/install/use software
    ▪ Linux clusters: EM64T, AMD64, X86_64
    ▪ AIX clusters: PowerPC, Power5
Outline

• Overview

• Hardware

• Software
  – User environment
  – Compilers
  – Application software

• Job management
Initial Login

- Log in via ssh
  - example: ssh <your_user_name>@oliver.loni.org
- Linux clusters
  - When you first login you'll see something like this:
    Generating public/private dsa key pair.
    Enter file in which to save the key (/home1/me/.ssh/id_dsa):
    Enter passphrase (empty for no passphrase):
    Enter same passphrase again:
    Your identification has been saved in /home1/me/.ssh/id_dsa.
    Your public key has been saved in /home1/me/.ssh/id_dsa.pub.
    The key fingerprint is:
  - What you need to do: press <enter> all the way down
  - Do not enter a phassphrase !!!!!!!!
Login Shell

• The default Login shell is bash
• Supported shells: bash, tcsh, ksh, csh & sh
• View your shell by “echo $SHELL”
• Change your shell at the profile page
  – LONI: allocations.loni.org
### File Systems

<table>
<thead>
<tr>
<th></th>
<th>Distributed file system</th>
<th>Throughput</th>
<th>File life time</th>
<th>Typically used for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>Yes</td>
<td>Low</td>
<td>Unlimited</td>
<td>Code in development, compiled executables</td>
</tr>
<tr>
<td>Scratch</td>
<td>Yes</td>
<td>High</td>
<td>30 days</td>
<td>Job input/output</td>
</tr>
<tr>
<td>Local Scratch</td>
<td>No</td>
<td></td>
<td>Job duration</td>
<td>Temporary files needed by running jobs</td>
</tr>
</tbody>
</table>

- **Never ever let your job write output to your home directory**
- **The “scratch” space is not for long-term storage**
## Disk Quota

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Home</th>
<th>Scratch</th>
<th>Local scratch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Access point</td>
<td>Quota</td>
<td>Access point</td>
</tr>
<tr>
<td>LONI Linux</td>
<td>/home/$USER</td>
<td>5 GB</td>
<td>/scratch/$USER</td>
</tr>
<tr>
<td>LONI AIX</td>
<td>/home/$USER</td>
<td>5 GB</td>
<td>/work/default/$USER</td>
</tr>
</tbody>
</table>
Exercise 1: Now it's time to log in

• Log in any cluster
• Check your disk quota
  – Linux clusters: use “showquota” command
    • Your scratch directory will be created within an hour of the first login
  – AIX clusters: use “quota” command
• Locate the directory /home/lyan1/traininglab/environment
  – There are files that you will need for following exercises
Manage the environment

• Environment variables
  – PATH: where to look for executables
  – LD_LIBRARY_PATH: where to look for shared libraries
  – Other custom environment variables needed by various software
• SOFTENV is a software that is used to set up these environment variables on all the clusters
  – More convenient than setting numerous environment variables in .bashrc or .cshrc
SOFTENV

• Command “softenv” lists all packages that are managed by SOFTENV

[lyan1@tezpur2 ~]$ softenv
...
These are the macros available:
*   @default
*   @globus-4.0
*   @intel-compilers

globus client
compiler: 'Intel Compilers', version: Latest.
A pointer to the latest installed intel compilers.

These are the keywords explicitly available:
+Mesa-6.4.2
+R-2.8.0-gcc-3.4.6
+ansys-lsdyna-11.0

No description yet for Mesa-6.4.2.
application: 'R', version 2.8.0
application: 'ANSYS LS-DYNA', version: 11.0
ANSYS LS-DYNA is a premier software package
for explicit nonlinear structural simulation with finite element pre- and post-processor. docs =>
http://www1.ansys.com/customer/

Softenv key
...

LONI High Performance Computing Workshop – Tulane University
April 13, 2009
• **Set up the environment variables to use a certain software**

  – **First add the key to** $HOME/.soft

```
[lyan1@tezpur2 ~]$ cat .soft
#
# This is the .soft file.
# It is used to customize your environment by setting up environment
# variables such as PATH and MANPATH.
# To learn what can be in this file, use 'man softenv'.
+fds
+smv
+matlab-r2007b
```

  – **Then execute** `resoft` **at the command line**

```
[lyan1@tezpur2 ~]$ resoft
```
SOFTENV

- Command "soft-dbq" shows which variables are set by a certain SOFTENV key

[lyan1@tezpur2 ~]$ soft-dbq +gcc-4.3.0
This is all the information associated with the key or macro +gcc-4.3.0.

Name: +gcc-4.3.0
Description: GNU gcc compiler, version 4.3.0
Flags: none
Groups: none
Exists on: Linux

On the Linux architecture, the following will be done to the environment:
The following environment changes will be made:
LD_LIBRARY_PATH = ${LD_LIBRARY_PATH}:/usr/local/compilers/GNU/gcc-4.3.0/lib64
PATH = ${PATH}:/usr/local/compilers/GNU/gcc-4.3.0/bin
Exercise 2: Use Softenv

- Find the key for VISIT (a visualization package)
- Check what variables are set through the key
- Set up your environment to use VISIT
- Check if the variables are correctly set by "which visit"
Exercise 2: Use Softenv

• Find the key for VISIT (a visualization package)
  – Use `softenv`
  – Or `softenv | grep -i visit` in case that the list is too long

• Check what variables are set through the key
  – Use `soft-dbq +visit`

• Set up your environment to use VISIT
  – Add “+visit” to your .soft file and `resoft`

• Check if the variables are correctly set by “which visit”
  – The output should be the path to the executable `visit`
Compilers

<table>
<thead>
<tr>
<th>Language</th>
<th>Linux clusters</th>
<th>AIX clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Intel</td>
<td>GNU</td>
</tr>
<tr>
<td>Fortran</td>
<td>ifort</td>
<td>g77</td>
</tr>
<tr>
<td>C</td>
<td>icc</td>
<td>gcc</td>
</tr>
<tr>
<td>C++</td>
<td>icpc</td>
<td>g++</td>
</tr>
</tbody>
</table>

- **Usage:** `<compiler> <options> <your_code>`
  - **Example:** `icc -O3 -o myexec mycode.c`

- **Some compilers options are architecture specific**
  - Linux: EM64T, AMD64 or X86_64
  - AIX: power5 or powerpc
Compilers for MPI code

<table>
<thead>
<tr>
<th>Language</th>
<th>Linux clusters</th>
<th>AIX clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fortran</td>
<td>mpif77,mpif90</td>
<td>mpxlf,mpxlfr,mpxlfr90,mpxlfr90r</td>
</tr>
<tr>
<td>C</td>
<td>mpicc</td>
<td>mpcc,mpccr</td>
</tr>
<tr>
<td>C++</td>
<td>mpiCC</td>
<td>mpCC,mpCCr</td>
</tr>
</tbody>
</table>

- Usage: similar to what we have seen
  - Example: `mpif90 -O2 -o myexec mycode.f90`

- On Linux clusters
  - We don't differentiate between different vendors, i.e. We don't have things like `intel_mpicc` and `pg_mpicc`
## Compilers for MPI code

<table>
<thead>
<tr>
<th>Language</th>
<th>Linux clusters</th>
<th>AIX clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fortran</td>
<td>mpif77, mpif90</td>
<td>mpixlf, mpixlf_r, mpixlf90, mpixlf90_r</td>
</tr>
<tr>
<td>C</td>
<td>Mpicc</td>
<td>mpcc, mpcc_r</td>
</tr>
<tr>
<td>C++</td>
<td>MpiCC</td>
<td>mpCC, mpCC_r</td>
</tr>
</tbody>
</table>

- These MPI compilers are actually **wrappers**
  - They still use the same compilers we've seen on the previous slides
  - They take care of everything we need to run MPI codes
  - What they actually do can be reveal by the `-show` option

```
[lyan1@tezpur2 ~]$ mpicc -show
icc -DUSE_STDARG -DHAVE_STDLIB_H=1 -DHAVE_STRING_H=1 -DHAVE_UNISTD_H=1
-DHAVE_STDARG_H=1 -DUSE_STDARG=1 -DMALLOC_RET_VOID=1
-L/usr/local/packages/mvapich-1.0-intel10.1/lib -lmproc
-L/usr/local/soft/lib64 -Wl,-rpath=/usr/local/soft/lib64 -libverbs
-libumad -lpthread -lpthread -lrt
```
Be careful on Linux clusters...

```bash
[lyan1@qb2 ~]$ ls -ld /usr/local/packages/mvapich*
drwxr-xr-x 12 root root 4096 Oct 18 13:25 /usr/local/packages/mvapich-0.98-gcc
drwxr-xr-x 12 root root 4096 Jan 23 11:35 /usr/local/packages/mvapich-0.98-intel10.1
drwxr-xr-x 12 root root 4096 Oct 18 13:25 /usr/local/packages/mvapich-0.98-intel19.1
```

- We have many different versions of MPI compilers
- So it is extremely important to compile and run you code with the same version of MPI compiler and mpirun!!!
Application Packages

- Installed under /usr/local/packages
- Most of them are managed by SOFTENV
  - Libraries
    - FFTW, HDF5, NetCDF, PETSc, MKL
  - Chemistry
    - Amber, Gaussian, CPMD, NWChem, NAMD
  - Profiling/debugging tools
    - TAU, Totalview
  - ...
Exercise 3: Compile a code

• Serial code
  – Copy hello.f90 from /home/lyan1/traininglab/environment
  – Compile it with a compiler of your choice
  – Run the executable from the command line

• MPI code
  – Copy hello_mpi.f90 from /home/lyan1/traininglab/environment
  – Compile it with a serial compiler and see what happens
  – Compile it with an MPI compiler
  – We will run it later
Exercise 3: Compile a code

• Serial code
  - Linux
    - cp /home/lyan1/traininglab/environment/*.f90 .
    - icc -o hello_ser hello.f90
    - ./hello_ser
  - AIX
    - cp /home/lyan1/traininglab/environment/*.f90 .
    - xlf90_r -o hello_ser hello.f90
    - ./hello_ser

• MPI code
  - mpif90 -o hello hello_mpi.f90
  - mpxlf90_r -o hello hello_mpi.f90
Outline

• Overview

• Hardware

• Software
  – User environment
  – Compilers
  – Application software

• Job management
Batch Queuing System

- A software suite that schedules job execution on (the computation nodes of) a cluster
  - Linux clusters: Torque/Moab
  - AIX clusters: Loadleveler

- Jobs are scheduled for execution in a number of queues, each of which has different
  - Number of available nodes
  - Max running jobs per user
  - Max run time
  - ...
## Queue Characteristics - Queen Bee

<table>
<thead>
<tr>
<th>Queue</th>
<th>Max Runtime</th>
<th>Total number of available nodes</th>
<th>Max running jobs per user</th>
<th>Max nodes per job</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workq</td>
<td>2 days</td>
<td>530</td>
<td>8</td>
<td>128</td>
<td>Unpreemptable (default)</td>
</tr>
<tr>
<td>Checkpt</td>
<td>668</td>
<td></td>
<td></td>
<td>256</td>
<td>Preemptable jobs</td>
</tr>
<tr>
<td>Preempt</td>
<td>668</td>
<td></td>
<td>NA</td>
<td></td>
<td>Require permission</td>
</tr>
<tr>
<td>Priority</td>
<td>668</td>
<td></td>
<td>NA</td>
<td></td>
<td>Require permission</td>
</tr>
</tbody>
</table>
Queue Characteristics - Other LONI Linux Clusters

<table>
<thead>
<tr>
<th>Queue</th>
<th>Max Runtime</th>
<th>Total number of available nodes</th>
<th>Max running jobs per user</th>
<th>Max nodes per job</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>14 days</td>
<td>16</td>
<td>64</td>
<td>1</td>
<td>Single processor jobs</td>
</tr>
<tr>
<td>Workq</td>
<td>3 days</td>
<td>64</td>
<td>8</td>
<td>40</td>
<td>Unpreemptable (default)</td>
</tr>
<tr>
<td>Checkpt</td>
<td>3 days</td>
<td>128</td>
<td>8</td>
<td>64</td>
<td>Preemptable jobs</td>
</tr>
<tr>
<td>Preempt</td>
<td>64</td>
<td>NA</td>
<td>NA</td>
<td></td>
<td>Require permission</td>
</tr>
<tr>
<td>Priority</td>
<td>64</td>
<td>NA</td>
<td>NA</td>
<td></td>
<td>Require permission</td>
</tr>
</tbody>
</table>
# Queue Characteristics - LONI AIX Clusters

<table>
<thead>
<tr>
<th>Queue</th>
<th>Max Runtime</th>
<th>Total number of available nodes</th>
<th>Max running jobs per user</th>
<th>Max nodes per job</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>14 days</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>Single processor jobs</td>
</tr>
<tr>
<td>Workq</td>
<td>5 days</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>Unpreemptable (default)</td>
</tr>
<tr>
<td>Checkpt</td>
<td></td>
<td>14</td>
<td></td>
<td>14</td>
<td>Preemptable jobs</td>
</tr>
<tr>
<td>Preempt</td>
<td></td>
<td>6</td>
<td>NA</td>
<td></td>
<td>Require permission</td>
</tr>
<tr>
<td>Priority</td>
<td></td>
<td>6</td>
<td>NA</td>
<td></td>
<td>Require permission</td>
</tr>
</tbody>
</table>
Job management

- Queue querying
  - Check free nodes and processors in each queue

- Job submission
  - Linux clusters: `qsub <job_script>`
  - AIX clusters: `llsubmit <job_script>`

- Job monitoring
  - Check the status of submitted jobs

- Job manipulation
  - Cancel/hold/release jobs
Queue Querying – Linux Clusters

• **Command:** `qfree`

```
[lyan1@qb4 ~]$ qfree
PBS total nodes: 668, free: 11, busy: 649, down: 8, use: 97%
PBS workq nodes: 530, free: 2, busy: 175, queued: 39
PBS checkpt nodes: 659, free: 2, busy: 474, queued: 250
(Highest priority job 94315 on queue checkpt will start in 2:01:35)
```

`qfree` shows the number of queued nodes, not queued jobs
Queue Querying – Linux Clusters

- **Command:** `qfreelon`
  - Help you to choose a cluster to submit your jobs

```
[lyanl@qb4 ~]$ qfreelon
-------- qb --------
PBS total nodes: 668, free: 11, busy: 649, down: 8, use: 97%
PBS workq nodes: 530, free: 2, busy: 175, queued: 39
PBS checkpt nodes: 659, free: 2, busy: 474, queued: 250
...
-------- poseidon --------
PBS total nodes: 128, free: 1, busy: 126, down: 1, use: 98%
PBS workq nodes: 96, free: 1, busy: 0, queued: 0
PBS checkpt nodes: 128, free: 1, busy: 126, queued: 180
PBS single nodes: 17, free: 0, busy: 0, queued: 0
-------- painter --------
PBS total nodes: 128, free: 8, busy: 120, down: 0, use: 93%
PBS workq nodes: 96, free: 8, busy: 0, queued: 0
PBS checkpt nodes: 128, free: 8, busy: 120, queued: 8
PBS single nodes: 16, free: 4 *16, busy: 0, queued: 0
```
Queue Querying – AIX Clusters

- **Command - llclass**

```bash
lyanl@l2f1n03$ llclass
Name                  MaxJobCPU      MaxProcCPU  Free  Max  Description                                      
                     d+hh:mm:ss    d+hh:mm:ss   Slots  Slots                                      
---- ---------------- -------------- ------------- ------- ------- ---------------------------------------------
interactive         undefined       undefined     8       8      Interactive Parallel jobs running          
on interactive node
single               unlimited        unlimited     4       8      One node queue (14 days) for               
serial and up to 8-processors parallel jobs
workq                unlimited        unlimited     51      56      Default queue (5 days), up to 56       
processors
priority             unlimited        unlimited     40      40      priority queue reservered for on-     
demand jobs (5 days), up to 48 processors
preempt              unlimited        unlimited     40      40      preemption queue reservered for on-      
demand jobs (5 days), up to 48 processors
checkpt              unlimited        unlimited     91      96      queue for checkpointing jobs (5 days), up to 104 processors, Job running on this queue can be preempted for on-demand job
```

---

LONI High Performance Computing Workshop – Tulane University
April 13, 2009
Job submission script – Linux clusters

#!/bin/bash

#SBATCH -l nodes=4:ppn=4  Number of nodes and processor
#SBATCH -l walltime=24:00:00  Maximum wall time
#SBATCH -N myjob  Job name
#SBATCH -o pbsout  Output file name (stdout)
#SBATCH -j oe  Join stdout and stderr
#SBATCH -q checkpt  Submission queue
#SBATCH -A loni_allocation  Account (allocation name)
#SBATCH -m e  Send mail when job ends
#SBATCH -M user@lsu.edu  Send mail to this address

<shell commands>
mpirun -machinefile $PBS_NODEFILE -np 16 <path_of_your_executable>
<shell commands>
Job submission script – AIX clusters

#!/bin/sh

#@ environment = COPY_ALL
#@ job_type = parallel
#@ output = /work/default/username/$(jobid).out
#@ error = /work/default/username/$(jobid).err
#@ notify_user = youremail@domain
#@ notification = error
#@ class = checkpoint
#@ wall_clock_limit = 24:00:00
#@ node_usage = shared
#@ node = 2,2
#@ total_tasks = 16
#@ initialdir = /work/default/username
#@ queue

<shell commands>
/usr/bin/poe <path_of_your_executable>
<shell commands>
Job Monitoring – Linux Clusters

**Command:** `qstat <options> <job_id>`

- All jobs are displayed if `<job_id>` is omitted
- Display a full status display: `qstat -f <job_id>`
- Display in the alternative format: `qstat -a <job_id>`

```
[lyan1@qb2 ~]$ qstat -a
qb2:

<table>
<thead>
<tr>
<th>Job ID</th>
<th>Username</th>
<th>Queue</th>
<th>Jobname</th>
<th>SessID</th>
<th>NDS</th>
<th>TSK</th>
<th>Memory</th>
<th>Req'd Time</th>
<th>Req'd S</th>
<th>Elap S Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2063.qb2</td>
<td>skeasler</td>
<td>checkpoint</td>
<td>nh4claa1</td>
<td>22534</td>
<td>12</td>
<td>1</td>
<td></td>
<td>48:00</td>
<td>R</td>
<td>00:00</td>
</tr>
<tr>
<td>2064.qb2</td>
<td>skeasler</td>
<td>checkpoint</td>
<td>nh4claa2</td>
<td>20625</td>
<td>12</td>
<td>1</td>
<td></td>
<td>48:00</td>
<td>R</td>
<td>00:00</td>
</tr>
<tr>
<td>2065.qb2</td>
<td>skeasler</td>
<td>checkpoint</td>
<td>nh4no3hs1</td>
<td>29016</td>
<td>12</td>
<td>1</td>
<td></td>
<td>48:00</td>
<td>R</td>
<td>00:00</td>
</tr>
<tr>
<td>2079.qb2</td>
<td>ade</td>
<td>checkpoint</td>
<td>F3ran_dlv</td>
<td>19851</td>
<td>10</td>
<td>1</td>
<td></td>
<td>48:00</td>
<td>R 36:26</td>
<td></td>
</tr>
<tr>
<td>2080.qb2</td>
<td>cott</td>
<td>checkpoint</td>
<td>D0HR7</td>
<td>23738</td>
<td>32</td>
<td>1</td>
<td></td>
<td>48:00</td>
<td>R 36:25</td>
<td></td>
</tr>
<tr>
<td>2081.qb2</td>
<td>pakya</td>
<td>workq</td>
<td>blade</td>
<td>24485</td>
<td>20</td>
<td>1</td>
<td></td>
<td>48:00</td>
<td>R 36:19</td>
<td></td>
</tr>
<tr>
<td>2099.qb2</td>
<td>ade</td>
<td>checkpoint</td>
<td>sp10</td>
<td>1531</td>
<td>10</td>
<td>1</td>
<td></td>
<td>48:00</td>
<td>R 31:04</td>
<td></td>
</tr>
<tr>
<td>2100.qb2</td>
<td>ade</td>
<td>checkpoint</td>
<td>F3ran2_dlv</td>
<td>3359</td>
<td>10</td>
<td>1</td>
<td></td>
<td>48:00</td>
<td>R 31:00</td>
<td></td>
</tr>
<tr>
<td>2106.qb2</td>
<td>ade</td>
<td>checkpoint</td>
<td>PLdt4_rani</td>
<td>25354</td>
<td>10</td>
<td>1</td>
<td></td>
<td>48:00</td>
<td>R 28:58</td>
<td></td>
</tr>
</tbody>
</table>
```
Job Monitoring – Linux Clusters

• **Command:** `showq`

```bash
[lyan1@oliver2 ~]$ showq
active jobs---------------------
JOBID  USERNAME  STATE  PROCS  REMAINING  STARTTIME
87809  pradeepv  Running  16   2:22:00:29  Fri Feb 27 10:36:41
87805  bnovak1  Running  32   2:20:54:58  Fri Feb 27 09:31:10
...
87810  rama     Running  1    4:07:44   Fri Feb 27 10:43:56

13 active jobs
437 of 504 processors in use by local jobs (86.71%)
110 of 126 nodes active (87.30%)

eligible jobs---------------------
JOBID  USERNAME  STATE  PROCS  WCLIMIT  QUEUETIME
0 eligible jobs

blocked jobs---------------------
JOBID  USERNAME  STATE  PROCS  WCLIMIT  QUEUETIME
0 blocked jobs
Total jobs: 13
```
When Will A Job Start – Linux Clusters

- **Command:** `showstart <job_id>
- Things that can change the start time
  - Higher priority jobs are submitted
  - Nodes come in or out of service
  - System has trouble starting your job

```
[lyan1@poseidon2 ~]$ showstart 30857
job 30857 requires 80 procs for 3:00:00:00

Estimated Rsv based start in 1:05:53:27 on Tue Apr 14 02:17:30
Estimated Rsv based completion in 4:05:53:27 on Fri Apr 17 02:17:30

Best Partition: base
```
Job Status Querying – Linux Clusters

- **Command:** `qshow <job_id>`

```bash
[lyan1@oliver2 ~]$ qshow 95024
PBS job: 95024, nodes: 16
Hostname  Days  Load  CPU  U# (User:Process:VirtualMemory:Memory:Hours)
oliver032  216  4.90  400  10  motl:hydro:371M:143M:12.1
oliver033  216  5.01  404  6   motl:hydro:328M:99M:12.1
oliver045  54   4.87  400  6   motl:hydro:323M:102M:12.1
oliver053  216  4.90  400  6   motl:hydro:324M:103M:12.1
oliver054  216  4.91  400  6   motl:hydro:324M:102M:12.1
oliver062  34   4.71  400  6   motl:hydro:321M:100M:12.1
```

...
Job Monitoring – AIX Clusters

- **Command:** `llq <options> <job_id>`
  - All jobs are displayed if `<job_id>` is omitted
  - Display detailed information: `llq -l <job_id>`
  - Display jobs from a certain user: `llq -u <username>`

```bash
lyan1@l2f1n03$ llq
Id        Owner   Submitted   ST  PRI  Class       Running On
----------  --------  -----------  --  ---  ----------       ----------
12f1n03.3697.0  collin    1/22 16:59 R  50    single     l2f1n14
12f1n03.3730.0  jheiko    1/28 13:30 R  50     workq    l2f1n10
12f1n03.3726.0  collin    1/26 08:21 R  50    single     l2f1n14
12f1n03.3698.0  collin    1/22 17:00 R  50    single     l2f1n14
12f1n03.3727.0  collin    1/26 08:21 R  50    single     l2f1n14

5 job step(s) in queue, 0 waiting, 0 pending, 5 running, 0 held, 0 preempted
```
Job Manipulation – Linux Clusters

- To kill a running or queued job (it could take a while to complete)
  - `qdel <job_id>`
  - `qdel -W force <job_id>`

- Put a queued job on hold
  - `qhold <job_id>`

- Resume a held job
  - `qrls <job_id>`
Job Manipulation – AIX Clusters

• Cancel a job
  – `llcancel <job_id>`

• Hold a job
  – `llhold <job_id>`

• Release a job
  – `llhold -r <job_id>`
Exercise 4: Run the MPI “hello world” program

• Run the parallel executable you compiled in Exercise 3 through the batch queuing system
  – On any cluster
  – In any queue
  – Recommended parameters
    ▪ Number of processors: 8
    ▪ Wall clock limit: 10 minutes
Exercise 4: Run the MPI “hello world” program

- Run the parallel executable you compiled in Exercise 3 through the batch queuing system
  - On any cluster
  - In any queue
  - Recommended parameters
    - Number of processors: 8
    - Wall clock limit: 10 minutes
  - There are two scripts in the directory where you copied the program from, which can be used as a template
    - Linux: qsub submit.linux
    - AIX: l1lsubmit submit.aix
When you have questions

- User's Guide
  - LONI: https://docs.loni.org/wiki/Main_Page
- User Support
  - LONI: sys-help@loni.org
- Live help (AIM, Yahoo Messenger, Google Talk)
  - Add “lsuhpchelp”