Programming Environment on LONI HPC Clusters

Le Yan
Scientific computing 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>LONI</td>
<td></td>
<td></td>
<td></td>
</tr>
<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>To be deployed</td>
</tr>
<tr>
<td>??</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
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
Hardware (AIX)

- 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 Access point</th>
<th>Home Quota</th>
<th>Scratch Access point</th>
<th>Scratch Quota</th>
<th>Local scratch Access point</th>
</tr>
</thead>
<tbody>
<tr>
<td>LONI Linux</td>
<td>/home/$USER</td>
<td>5 GB</td>
<td>/scratch/$USER</td>
<td>100 GB</td>
<td>/var/scratch</td>
</tr>
<tr>
<td>LONI AIX</td>
<td>/home/$USER</td>
<td>5 GB</td>
<td>/work/default/$USER</td>
<td>20 GB</td>
<td>/scratch/local</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

These are the keywords explicitly available:
+Mesa-6.4.2
+R-2.8.0-gcc-3.4.6
+ansys-1sdyana-11.0

Softenv key
```

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

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

• 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 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,mpxf90,mpxf90_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>

- 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>mpixf,mpixf_r,mpixf90,mpixf90_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 \(-\text{show}\) 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 -Lmpich
    -L/usr/local/ofed/lib64 -Wl,-rpath=/usr/local/ofed/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
-drwxr-xr-x 12 root root 4096 Feb 12 10:27 /usr/local/packages/mvapich-0.98-pgi6.1
-drwxr-xr-x 12 root root 4096 Nov 19 10:40 /usr/local/packages/mvapich-1.0-beta-intel10.0
-drwxr-xr-x 12 root root 4096 Nov 1 11:57 /usr/local/packages/mvapich-1.0-beta-intel-9.1
-drwxr-xr-x 12 root root 4096 Jan 24 16:38 /usr/local/packages/mvapich-1.0-intel10.1
-drwxr-xr-x 10 root root 4096 Oct 18 13:25 /usr/local/packages/mvapich2-0.98-gcc
-drwxr-xr-x 10 root root 4096 Jan 24 16:05 /usr/local/packages/mvapich2-0.98-intel10.1
-drwxr-xr-x 10 root root 4096 Oct 18 13:25 /usr/local/packages/mvapich2-0.98-intel19.1
-drwxr-xr-x 11 root root 4096 Nov 9 16:31 /usr/local/packages/mvapich2-1.01-intel10.0
-drwxr-xr-x  9 root root 4096 Jan 25 09:54 /usr/local/packages/mvapich2-1.0.1-intel10.1
-drwxr-xr-x 11 root root 4096 Nov 8 13:10 /usr/local/packages/mvapich2-1.0.1-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
  – ...

• We will provide tutorials on some of them as part of the HPC training series
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
    - mpxlf90_r -o hello hello_mpi.f90

- **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
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></td>
<td>668</td>
<td></td>
<td>256</td>
<td>Preemptable jobs</td>
</tr>
<tr>
<td>Preempt</td>
<td></td>
<td>668</td>
<td>NA</td>
<td></td>
<td>Require permission</td>
</tr>
<tr>
<td>Priority</td>
<td></td>
<td>668</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>3 days</td>
<td>64</td>
<td>NA</td>
<td>NA</td>
<td>Require permission</td>
</tr>
<tr>
<td>Priority</td>
<td>3 days</td>
<td>64</td>
<td>NA</td>
<td>NA</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:** `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
Queue Querying – AIX Clusters

- **Command - llclass**

```bash
lyan1@l2f1n03$ llclass
```

<table>
<thead>
<tr>
<th>Name</th>
<th>MaxJobCPU</th>
<th>MaxProcCPU</th>
<th>Free</th>
<th>Max</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>interactive</td>
<td>undefined</td>
<td>undefined</td>
<td>8</td>
<td>8</td>
<td>Interactive Parallel jobs running on interactive node</td>
</tr>
<tr>
<td>single</td>
<td>unlimited</td>
<td>unlimited</td>
<td>4</td>
<td>8</td>
<td>One node queue (14 days) for serial and up to 8-processor parallel jobs</td>
</tr>
<tr>
<td>workq</td>
<td>unlimited</td>
<td>unlimited</td>
<td>51</td>
<td>56</td>
<td>Default queue (5 days), up to 56 processors</td>
</tr>
<tr>
<td>priority</td>
<td>unlimited</td>
<td>unlimited</td>
<td>40</td>
<td>40</td>
<td>priority queue reserved for on-demand jobs (5 days), up to 48 processors</td>
</tr>
<tr>
<td>preempt</td>
<td>unlimited</td>
<td>unlimited</td>
<td>40</td>
<td>40</td>
<td>preemption queue reserved for on-demand jobs (5 days), up to 48 processors</td>
</tr>
<tr>
<td>checkpoint</td>
<td>unlimited</td>
<td>unlimited</td>
<td>91</td>
<td>96</td>
<td>queue for checkpointing jobs (5 days), up to 104 processors, Job running on this queue can be preempted for on-demand job</td>
</tr>
</tbody>
</table>
Job submission script – Linux clusters

#!/bin/bash
#PBS -l nodes=4:ppn=4  Number of nodes and processor
#PBS -l walltime=24:00:00  Maximum wall time
#PBS -N myjob  Job name
#PBS -o pbsout  Output file name (stdout)
#PBS -j oe  Join stdout and stderr
#PBS -q checkpt  Submission queue
#PBS -A loni_allocation  Account (allocation name)
#PBS -m e  Send mail when job ends
#PBS -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 = checkpt
#@ 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
```

<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>Elap S Time</th>
<th>Req'd Time</th>
<th>Req'd Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>2063.qb2</td>
<td>skeasler</td>
<td>checkpt</td>
<td>nh4claa1</td>
<td>22534</td>
<td>12</td>
<td>1</td>
<td>--</td>
<td>48:00</td>
<td>R 00:00</td>
<td></td>
</tr>
<tr>
<td>2064.qb2</td>
<td>skeasler</td>
<td>checkpt</td>
<td>nh4claa2</td>
<td>20625</td>
<td>12</td>
<td>1</td>
<td>--</td>
<td>48:00</td>
<td>R 00:00</td>
<td></td>
</tr>
<tr>
<td>2065.qb2</td>
<td>skeasler</td>
<td>checkpt</td>
<td>nh4no3hs1</td>
<td>29016</td>
<td>12</td>
<td>1</td>
<td>--</td>
<td>48:00</td>
<td>R 00:00</td>
<td></td>
</tr>
<tr>
<td>2079.qb2</td>
<td>ade</td>
<td>checkpt</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>checkpt</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>checkpt</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>checkpt</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>checkpt</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>

LONI High Performance Computing Workshop – Southern University
March 3, 2009
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>`

```
lyan1@12f1n03$ llq
Id    Owner  Submitted   ST  PRI  Class       Running On
---    ------  ----------  --  --   --------     -----------
12f1n03.3697.0  collin  1/22 16:59  R  50  single     12f1n14
12f1n03.3730.0  jheiko  1/28 13:30  R  50  workq      12f1n10
12f1n03.3726.0  collin  1/26 08:21  R  50  single     12f1n14
12f1n03.3698.0  collin  1/22 17:00  R  50  single     12f1n14
12f1n03.3727.0  collin  1/26 08:21  R  50  single     12f1n14

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: `llsubmit 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”