

# Programming Environment on LONI HPC Clusters

Le Yan

HPC consultant

User services group

Louisiana Optical Network Initiative



*LONI High Performance Computing Workshop – Tulane University  
April 13, 2009*



# 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

	Name	Peak TeraFLOPS/s	Location	Status
LONI	Queen Bee	50.7	ISB	Available
	Eric	4.7	LSU	Available
	Oliver	4.7	ULL	Available
	Louie	4.7	Tulane	Available
	Poseidon	4.7	UNO	Available
	Painter	4.7	LaTech	Available
	Satellite	4.7	Southern	To be deployed

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

# Current deployment status - IBM AIX clusters

	Name	Peak TeraFLOPS/s	Location	Status
LONI	Bluedawg	0.85	LaTech	Available
	Ducky	0.85	Tulane	Available
	Zeke	0.85	ULL	Available
	Neptune	0.85	UNO	Available
	Lacumba	0.85	Southern	Available

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



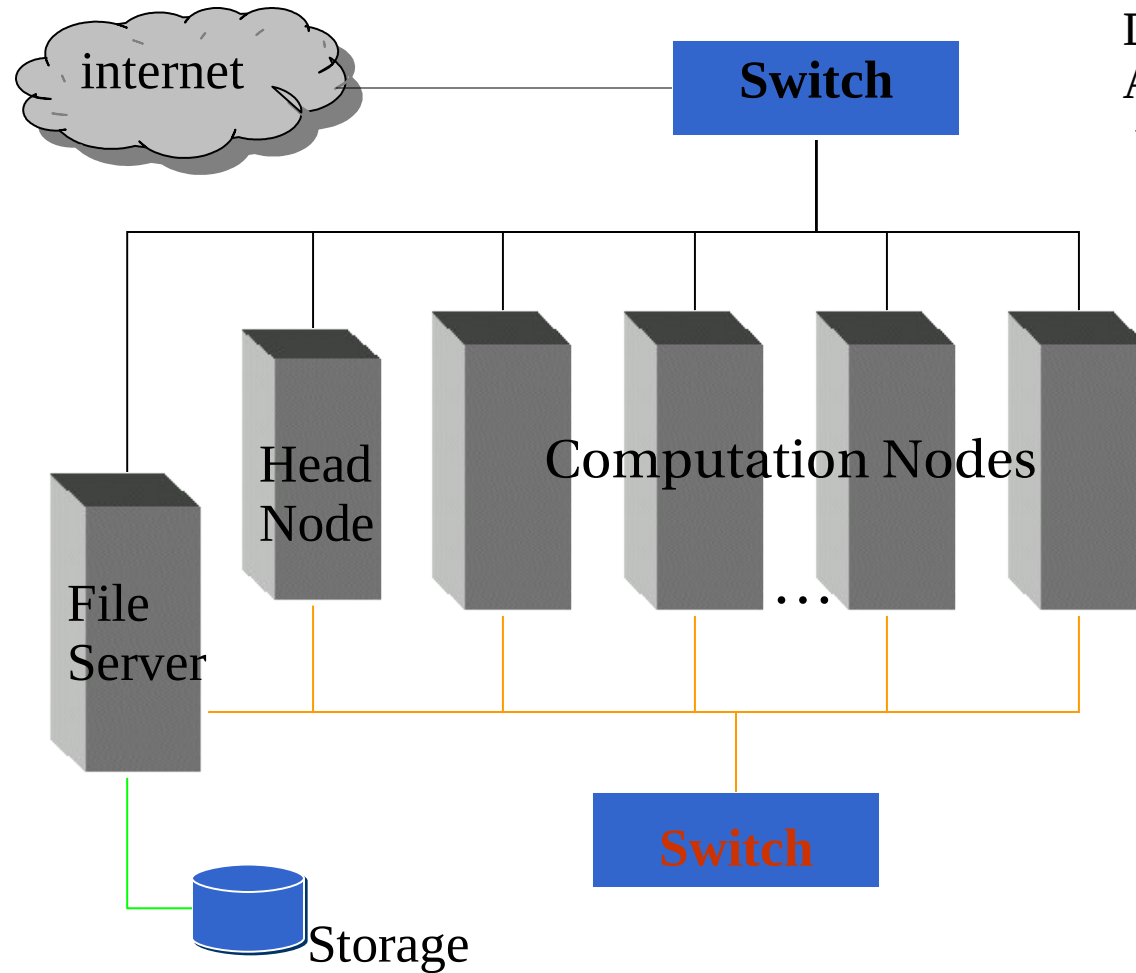
# Outline

- 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:  
b1:d4:d9:b4:90:8b:e1:10:e3:34:2c:75:57:b2:7d:83 me@oliver2.loni.org
```

- 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](http://allocations.loni.org)



# File Systems

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

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





# Disk Quota

Cluster	Home		Scratch		Local scratch
	Access point	Quota	Access point	Quota	Access point
LONI Linux	/home/\$USER	5 GB	/scratch/\$USER	100 GB	/var/scratch
LONI AIX	/home/\$USER	5 GB	/work/default/\$USER	20 GB	/scratch/local



# 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          globus client
* @intel-compilers    compiler: 'Intel Compilers', version: Latest.
                      A pointer to the latest installed intel
                      compilers.
```

```
These are the keywords explicitly available:
```

```
+Mesa-6.4.2          No description yet for Mesa-6.4.2.
+R-2.8.0-gcc-3.4.6   application: 'R', version 2.8.0
+ansys-lsdyna-11.0   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

```
...
```



# SOFTENV

- Set up the environment variables to use a certain software

- First add the key to `$HOME/.soft`

```
[lyan1@tezipur2 ~]$ 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@tezipur2 ~]$ 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-dbg +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

Language	Linux clusters			AIX clusters
	Intel	GNU	PGI	XL compilers
Fortran	ifort	g77	pgf77,pgf95	xlf,xlf_r,xlf90,xlf90_r
C	icc	gcc	pgcc	xlc,xlc_r
C++	icpc	g++	pgCC	xlC,xlC_r

- 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

Language	Linux clusters	AIX clusters
Fortran	mpif77,mpif90	mpxlf,mpxlf_r,mpxlf90,mpxlf90_r
C	mpicc	mpcc,mpcc_r
C++	mpiCC	mpCC,mpCC_r

- 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

Language	Linux clusters	AIX clusters
Fortran	mpif77,mpif90	mpxlf,mpxlf_r,mpxlf90,mpxlf90_r
C	Mpicc	mpcc,mpcc_r
C++	MpiCC	mpCC,mpCC_r

- 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 -lmpich
-L/usr/local/ofed/lib64 -Wl,-rpath=/usr/local/ofed/lib64 -libverbs
-libumad -lpthread -lpthread -lrt
```

# Be careful on Linux clusters...

```
[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-intel9.1  
drwxr-xr-x 12 root root 4096 Oct 18 13:25 /usr/local/packages/mvapich-0.98-intel9.1-LM  
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 Oct 18 13:25 /usr/local/packages/mvapich-0.98-pgi6.1-eric  
drwxr-xr-x 12 root root 4096 Nov 19 10:40 /usr/local/packages/mvapich-1.0beta-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-intel9.1  
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/mvapich2-1.01-intel9.1
```

- 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/environement`
  - 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`

- MPI

- AIX

- `cp /home/lyan1/traininglab/environment/*.f90 .`
    - `xlf90_r -o hello_ser hello.f90`
    - `./hello_ser`
    - `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

Queue	Max Runtime	Total number of available nodes	Max running jobs per user	Max nodes per job	Use
Workq	2 days	530	8	128	Unpreemptable (default)
Checkpt		668		256	Preemptable jobs
Preempt		668	NA		Require permission
Priority		668	NA		Require permission



# Queue Characteristics - Other LONI Linux Clusters

Queue	Max Runtime	Total number of available nodes	Max running jobs per user	Max nodes per job	Use
Single	14 days	16	64	1	Single processor jobs
Workq	3 days	64	8	40	Unpreemptable (default)
Checkpt		128		64	Preemptable jobs
Preempt		64	NA	Require permission	
Priority		64	NA	Require permission	



# Queue Characteristics - LONI AIX Clusters

Queue	Max Runtime	Total number of available nodes	Max running jobs per user	Max nodes per job	Use
Single	14 days	1	8	1	Single processor jobs
Workq	5 days	8		8	Unpreemptable (default)
Checkpt		14		14	Preemptable jobs
Preempt		6	NA	Require permission	
Priority		6	NA	Require permission	



# 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: `qfreeloni`
  - Help you to choose a cluster to submit your jobs

```
[lyan1@qb4 ~]$ qfreeloni
----- 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

```
lyan1@l2f1n03$ llclass
```

Name	MaxJobCPU d+hh:mm:ss	MaxProcCPU d+hh:mm:ss	Free Slots	Max Slots	Description
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-processor parallel jobs
workq	unlimited	unlimited	51	56	Default queue (5 days), up to 56 processors
priority	unlimited	unlimited	40	40	priority queue reserved for on-demand jobs (5 days), up to 48 processors
preempt	unlimited	unlimited	40	40	preemption queue reserved 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



# Job submission script – Linux clusters

```
#!/bin/bash
```

```
#PBS -l nodes=4:ppn=4
```

```
#PBS -l walltime=24:00:00
```

```
#PBS -N myjob
```

```
#PBS -o pbsout
```

```
#PBS -j oe
```

```
#PBS -q checkpt
```

```
#PBS -A loni_allocation
```

```
#PBS -m e
```

```
#PBS -M user@lsu.edu
```

Number of nodes and processor

Maximum wall time

Job name

Output file name (stdout)

Join stdout and stderr

Submission queue

Account (allocation name)

Send mail when job ends

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
qb2:
```

Job ID	Username	Queue	Jobname	SessID	NDS	TSK	Req'd Memory	Req'd Time	Elap S	Time
2063.qb2	skeasler	checkpt	nh4claa1	22534	12	1	--	48:00	R	00:00
2064.qb2	skeasler	checkpt	nh4claa2	20625	12	1	--	48:00	R	00:00
2065.qb2	skeasler	checkpt	nh4no3hs1	29016	12	1	--	48:00	R	00:00
2079.qb2	ade	checkpt	F3ran_dlv	19851	10	1	--	48:00	R	36:26
2080.qb2	cott	checkpt	D0HR7	23738	32	1	--	48:00	R	36:25
2081.qb2	pakya	workq	blade	24485	20	1	--	48:00	R	36:19
2099.qb2	ade	checkpt	sp10	1531	10	1	--	48:00	R	31:04
2100.qb2	ade	checkpt	F3ran2_dlv	3359	10	1	--	48:00	R	31:00
2106.qb2	ade	checkpt	PLdt4_rani	25354	10	1	--	48:00	R	28:58

# Job Monitoring – Linux Clusters

- Command: showq

```
[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           QUEUE TIME
0 eligible jobs

blocked jobs-----
JOBID                USERNAME            STATE  PROCS   WCLIMIT           QUEUE TIME
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>`

```
[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 motl:hydro:321M:100M:12.1 motl:hydro:  
328M:100M:12.1 motl:hydro:328M:100M:12.1 motl:pbs_demux:3M:0M motl:95024:52M:1M  
motl:mpirun:52M:5M motl:orted:50M:3M
```

```
oliver033 216 5.01 404 6 motl:hydro:328M:99M:12.1 motl:hydro:328M:100M:12.1 motl:hydro:  
331M:102M:12.1 motl:hydro:331M:103M:12.1 motl:orted:50M:3M
```

```
oliver045 54 4.87 400 6 motl:hydro:323M:102M:12.1 motl:hydro:322M:100M:12.1 motl:hydro:  
321M:100M:12.1 motl:hydro:324M:103M:12.1 motl:orted:50M:3M
```

```
oliver053 216 4.90 400 6 motl:hydro:324M:102M:12.1 motl:hydro:324M:103M:12.1 motl:hydro:  
322M:100M:12.1 motl:hydro:322M:100M:12.1 motl:orted:50M:3M
```

```
oliver054 216 4.91 400 6 motl:hydro:324M:102M:12.1 motl:hydro:322M:100M:12.1 motl:hydro:  
322M:100M:12.1 motl:hydro:324M:103M:12.1 motl:orted:50M:3M
```

```
...
```

```
oliver062 34 4.71 400 6 motl:hydro:321M:100M:12.1 motl:hydro:321M:101M:12.1 motl:hydro:  
321M:100M:12.1 motl:hydro:324M:103M:12.1 motl:orted:50M:3M
```



# 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@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: `llsubmit submit.aix`



# When you have questions

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

