User Environment on LONI and LSU HPC Clusters

Bhupender Thakur

HPC @ LSU

Outline

- Cluster Hardware
- Accessing Software
- Submitting and Monitoring Jobs

General Cluster Architecture

Login nodes get you access to the cluster. Individual nodes are not accessible.

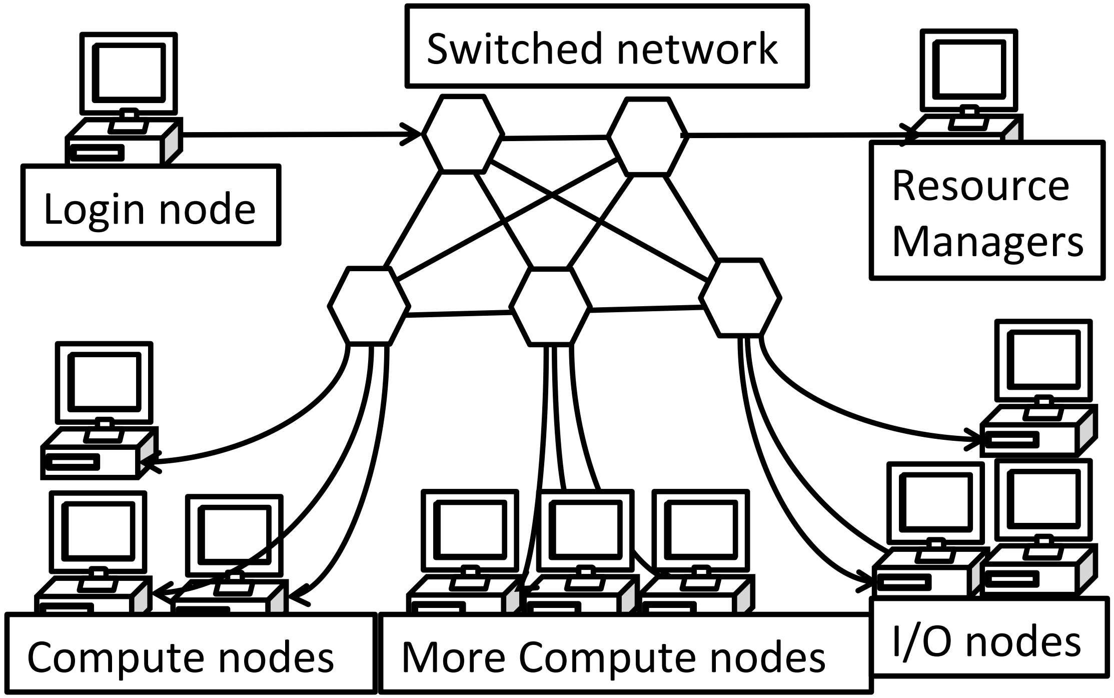
- Login via ssh
- Node are not meant to run jobs

Compute nodes are connected via a network of switches

- QDR switches on SM-II
- Latencies typically few microsecs
- Bandwidth 40Gbps

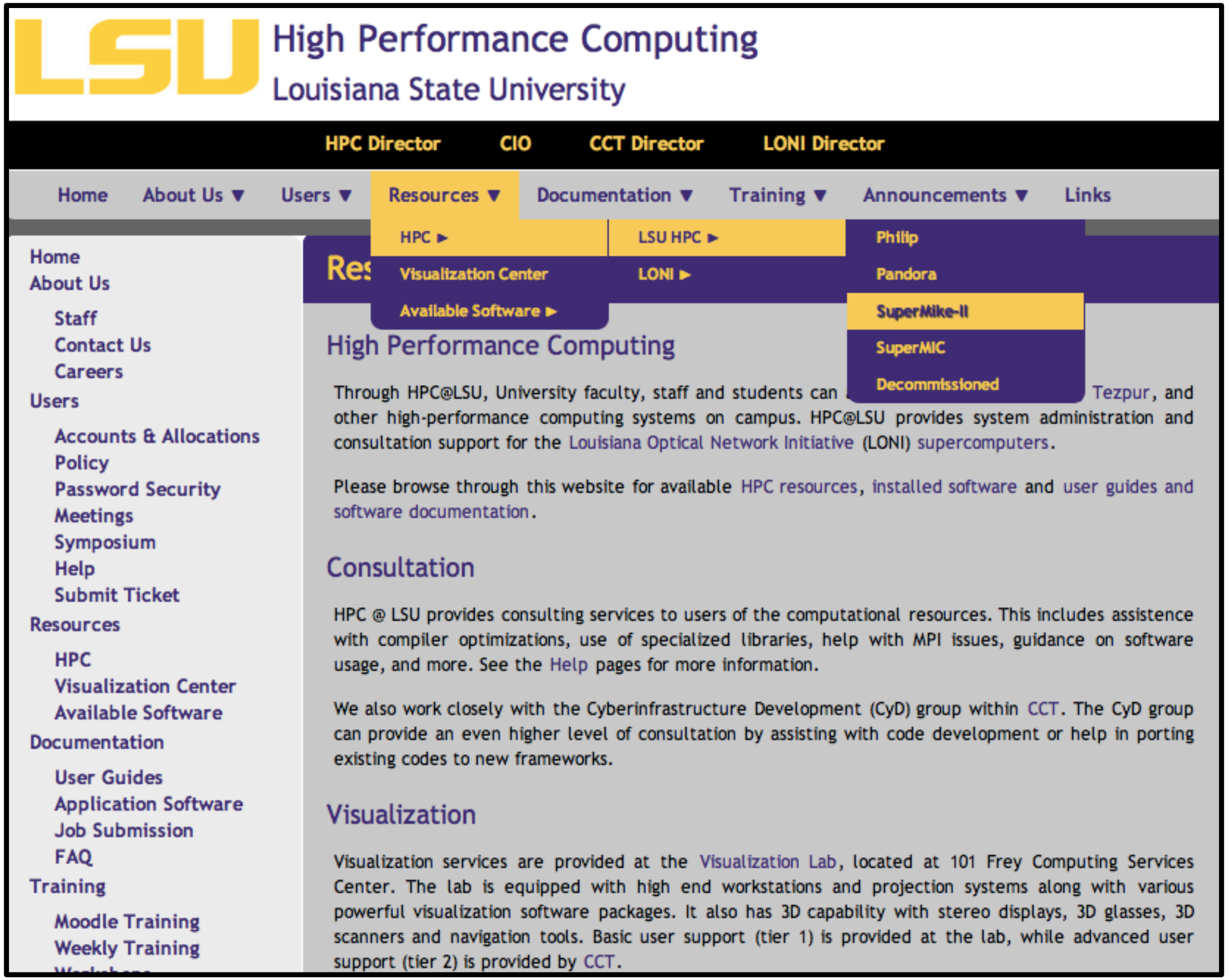
Resource managers give access to compute resource

- PBS/ loadleveler installed
- Run commands qsub, qstat, qdel



Available HPC resources

- Hardware resources consist of LONI and LSU HPC cluster systems
- LONI and LSU HPC maintain separate LDAP for authentication. In essence, You need separate accounts
- Both resources are managed centrally by a core team at LSU.
- To get help on either
  Docs: www.hpc.lsu.edu
  Help: sys-help@loni.org



High Performance Computing

Through HPC@LSU, University faculty, staff and students can Tezpur, and other high-performance computing systems on campus. HPC@LSU provides system administration and consultation support for the Louisiana Optical Network Initiative (LONI) supercomputers.

Please browse through this website for available HPC resources, installed software and user guides and software documentation.

Consultation

HPC @ LSU provides consulting services to users of the computational resources. This includes assistence with compiler optimizations, use of specialized libraries, help with MPI issues, guidance on software usage, and more. See the Help pages for more information.

We also work closely with the Cyberinfrastructure Development (CyD) group within CCT. The CyD group can provide an even higher level of consultation by assisting with code development or help in porting existing codes to new frameworks.

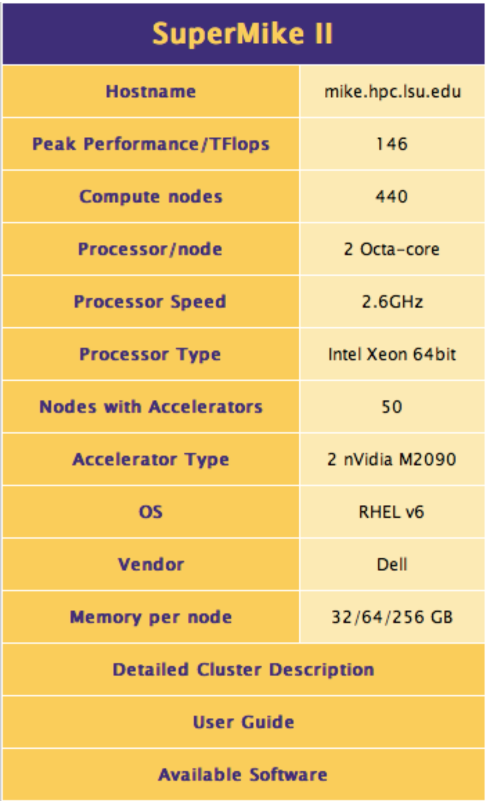
Visualization

Visualization services are provided at the Visualization Lab, located at 101 Frey Computing Services Center. The lab is equipped with high end workstations and projection systems along with various powerful visualization software packages. It also has 3D capability with stereo displays, 3D glasses, 3D scanners and navigation tools. Basic user support (tier 1) is provided at the lab, while advanced user support (tier 2) is provided by CCT.

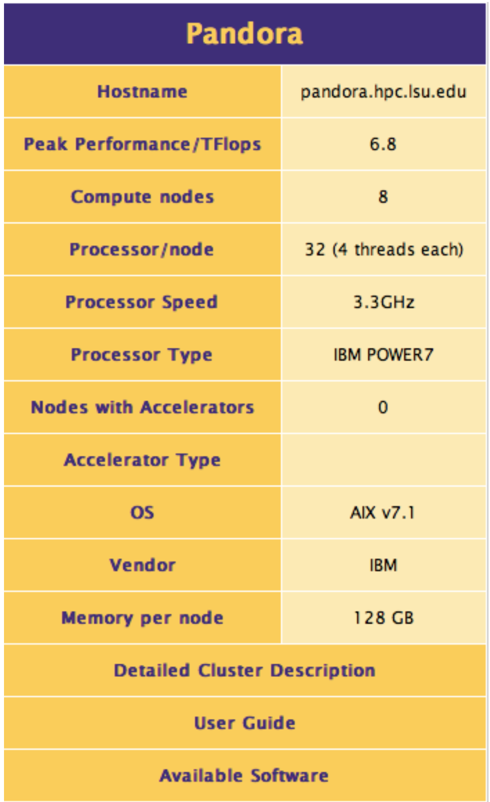
Available HPC resources

	Name	Peak Performance (TFLOPS)	Location	Vendor	Architecture	Status
LONI	Queen Bee	50.7	ISB	Dell	Linux x86_64	In production
	Eric	4.8	LSU	Dell	Linux x86_64	In production
	Oliver	4.8	ULL	Dell	Linux x86_64	In production
	Louie	4.8	Tulane	Dell	Linux x86_64	In production
	Poseidon	4.8	UNO	Dell	Linux x86_64	In production
	Painter	4.8	LaTech	Dell	Linux x86_64	In production
	Satellite	4.8	Southern	Dell	Linux x86_64	Being deployed?
LSU	~~Tezpur~~	~~15.3~~	~~LSU~~	~~Dell~~	~~Linux x86_64~~	~~In production~~
	Philip	3.5	LSU	Dell	Linux x86_64	In production
	Pandora	6.8	LSU	IBM	Power7	In production
	SuperMikeII	146(CPU)+66(GPU)	LSU	Dell	Linux x86_64	In production
	~~SuperMIC~~	~~~ 1000~~	~~LSU~~	~~Dell~~	~~Linux x86_64~~	~~Arriving~~

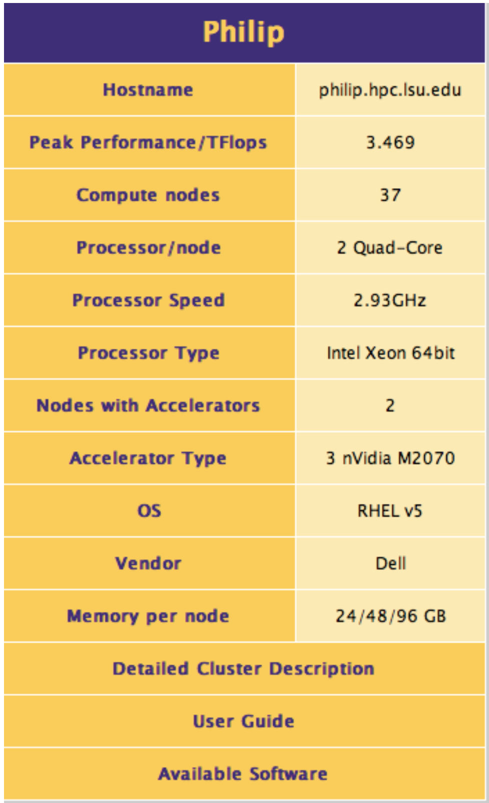
LSU HPC Resources



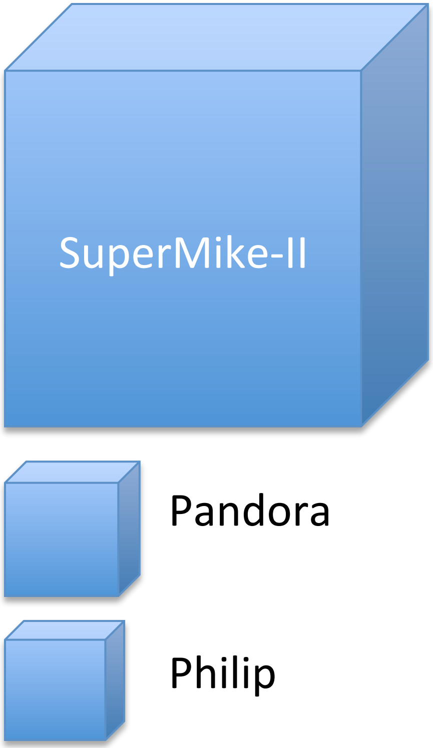
SuperMike II	
Hostname	mike.hpc.lsu.edu
Peak Performance/TFlops	146
Compute nodes	440
Processor/node	2 Octa-core
Processor Speed	2.6GHz
Processor Type	Intel Xeon 64bit
Nodes with Accelerators	50
Accelerator Type	2 nVidia M2090
OS	RHEL v6
Vendor	Dell
Memory per node	32/64/256 GB
Detailed Cluster Description	
User Guide	
Available Software	



Pandora	
Hostname	pandora.hpc.lsu.edu
Peak Performance/TFlops	6.8
Compute nodes	8
Processor/node	32 (4 threads each)
Processor Speed	3.3GHz
Processor Type	IBM POWER7
Nodes with Accelerators	0
Accelerator Type	
OS	AIX v7.1
Vendor	IBM
Memory per node	128 GB
Detailed Cluster Description	
User Guide	
Available Software	



Philip	
Hostname	philip.hpc.lsu.edu
Peak Performance/TFlops	3.469
Compute nodes	37
Processor/node	2 Quad-Core
Processor Speed	2.93GHz
Processor Type	Intel Xeon 64bit
Nodes with Accelerators	2
Accelerator Type	3 nVidia M2070
OS	RHEL v5
Vendor	Dell
Memory per node	24/48/96 GB
Detailed Cluster Description	
User Guide	
Available Software	



LSU HPC :What should I use?

Why would you use SuperMike II?

- You need many nodes with more cores
  - 16 cores, 32G / node
- You need special nodes
  - Memory > 200G
  - GPUs on the node
- You need special storage
  - /project

Why would you use Pandoa?

- You need an AIX cluster/IBM processors
- You need many cores/memory on one node. For threaded non-mpi jobs
  - 128G/node
  - 32 thds@3.3 GHz/ nodes

Why would you use Philip?

- You need medium memory, fast single core for serial jobs
  - 24-96G, 8 cores @2.93GHz / node
- You need shared storage with SuperMike-II
  - /project not shared with SM-II. Earlier with Tezpur

LSU HPC :Watch out for SuperMIC

360 Compute Nodes

- Two 2.8GHz 10-Core Ivy Bridge-EP E5-2680 Xeon 64-bit Processors
- Two Intel Xeon Phi 7120P Coprocessors
- 64GB DDR3 1866MHz Ram
- 500GB HD
- 56 Gigabit/sec Infiniband network interface

20 Hybrid Compute Nodes

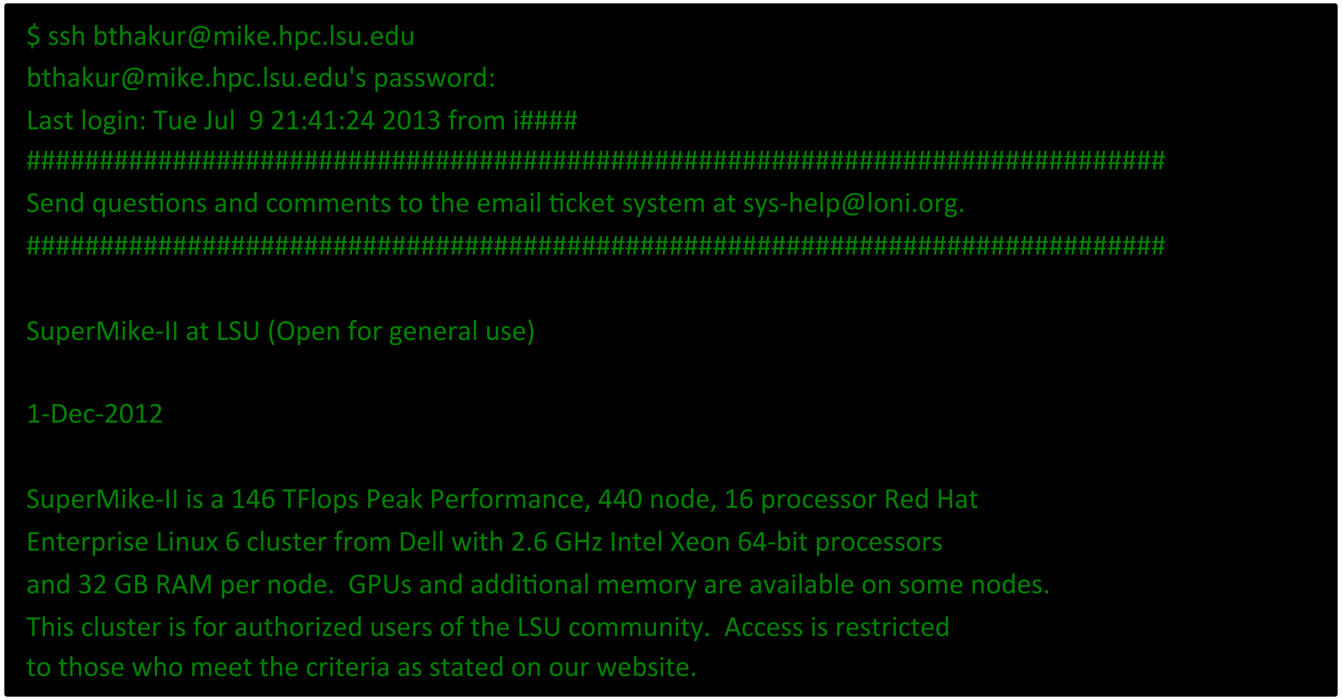
- Two 2.8GHz 10-Core Ivy Bridge-EP E5-2680 Xeon 64-bit Processors
- One Intel Xeon Phi 7120P Coprocessors
- One NVIDIA Tesla K20X 6GB GPU with GPUDirect Support
- 64GB DDR3 1866MHz Ram
- 500GB HD
- 56 Gigabit/sec Infiniband network interface

Cluster Storage

- 840TB Lustre High-Performance disk
- 5TB NFS-mounted /home disk storage

ssh: Accessing the cluster

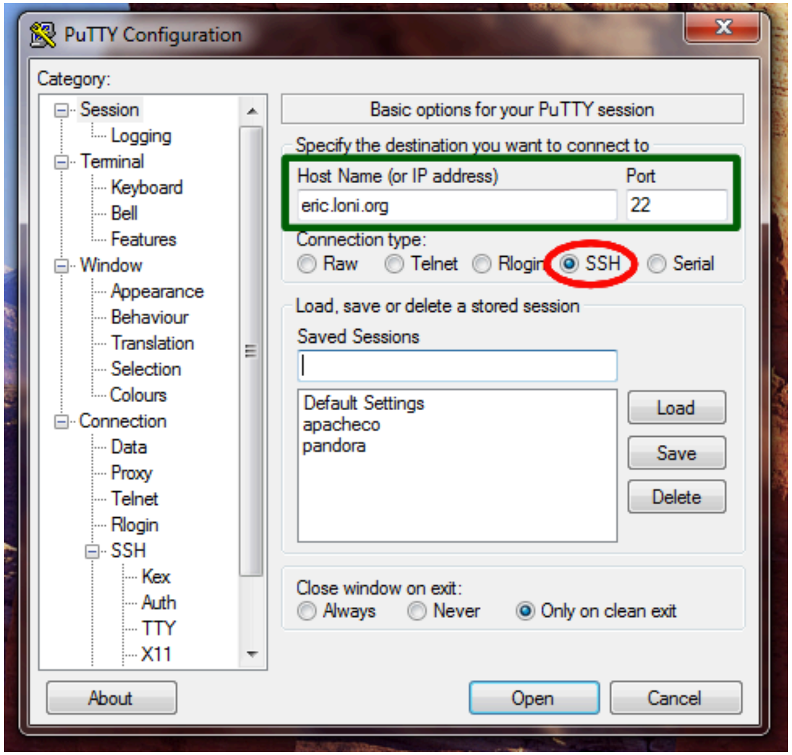
- Host name
  - LONI: *<cluster name>.loni.org* *e.g.: mike.hpc.lsu.edu*
  - LSU HPC: *<cluster name>.hpc.lsu.edu* *e.g.: qb.loni.org*
  - On Unix and Mac use ssh on a terminal to connect



```
$ ssh bthakur@mike.hpc.lsu.edu
bthakur@mike.hpc.lsu.edu's password:
Last login: Tue Jul  9 21:41:24 2013 from i####
##########################################################################################
Send questions and comments to the email ticket system at sys-help@loni.org.
##########################################################################################

SuperMike-II at LSU (Open for general use)

1-Dec-2012

SuperMike-II is a 146 TFlops Peak Performance, 440 node, 16 processor Red Hat
Enterprise Linux 6 cluster from Dell with 2.6 GHz Intel Xeon 64-bit processors
and 32 GB RAM per node.  GPUs and additional memory are available on some nodes.
This cluster is for authorized users of the LSU community.  Access is restricted
to those who meet the criteria as stated on our website.
```

Accessing the Clusters

- Host name
  - LONI: ssh *<cluster name>.loni.org* *e.g.: mike.hpc.lsu.edu*
  - LSU HPC: ssh *<cluster name>.hpc.lsu.edue.g.: qb.loni.org*
  - On Windows use putty



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): eric.loni.org

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

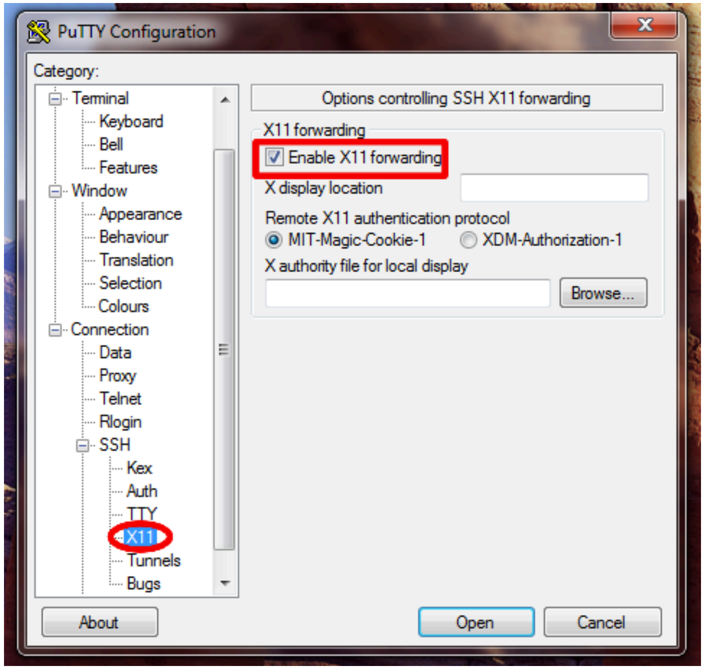
Saved Sessions

Default Settings
apacheco
pandora

Close window on exit: ○ Always ○ Never ● Only on clean exit

Connection with X11 Forwarding

- Some software packages have GUI, which requires X11 forwarding to be established with the ssh connection
- Unix/Linux users
  - Use the "-X" option of ssh
- Mac users
  - Use the "X11" application
- Windows users
  - Install X server (e.g. Xming)
  - Enable X11 forwarding in the client



Options controlling SSH X11 forwarding

X11 forwarding

☑ Enable X11 forwarding

X display location

Remote X11 authentication protocol

● MIT-Magic-Cookie-1 ○ XDM-Authorization-1

X authority file for local display

File Systems

	Distributed	Throughput	life	Best used for
Home	Yes	Low	Unlimited	Development/compilation
Work/ Scratch	Yes	High	30 days	Job input/output
Local scratch	No	Higher?	Job duration	Temporary files

- Tips
  - Never let your job write output to your home directory
  - Do not write temporary files to `/tmp`. Write to local scratch or work space
  - The work space is not for long-term storage. Files purged periodically
  - Use "`rmpurge`" to delete large amount of files

Disk Quota

Cluster	Home		Work		Local scratch
	Access point	Quota	Access Point	Quota	Access point
LONI Linux	/home/$USER	5 GB	/work/$USER	100 GB	/var/scratch
HPC Linux				NA	
HPC AIX				50 GB	/scratch/local

- No quota is enforced on the work space on Queen Bee, Tezpur, Philip and SuperMikeII
- On Linux clusters, the work directory is created within an hour after the first login
- Check current disk usage
  - Linux: `showquota`

Storage Allocation on /project

- One can apply for extra disk space on the /project volume if
  - your research requires some files to remain on the cluster for a fairly long period of time; **and**
  - their size exceeds the quota of the /home
- The unit is 100 GB
- Available on SuperMikeII and Queen Bee
- Storage allocations are good for 6 months, but can be extended based on the merit of the request
- Examples of valid requests
  - I am doing a 6-month data mining project on a large data set
  - The package I am running requires 10 GB of disk space to install
- Examples of invalid requests
  - I do not have time to transfer the data from my scratch space to my local storage and I need a temporary staging area

File Transfer

From/to a Unix/Linux/Mac machine

- Use scp or rsync

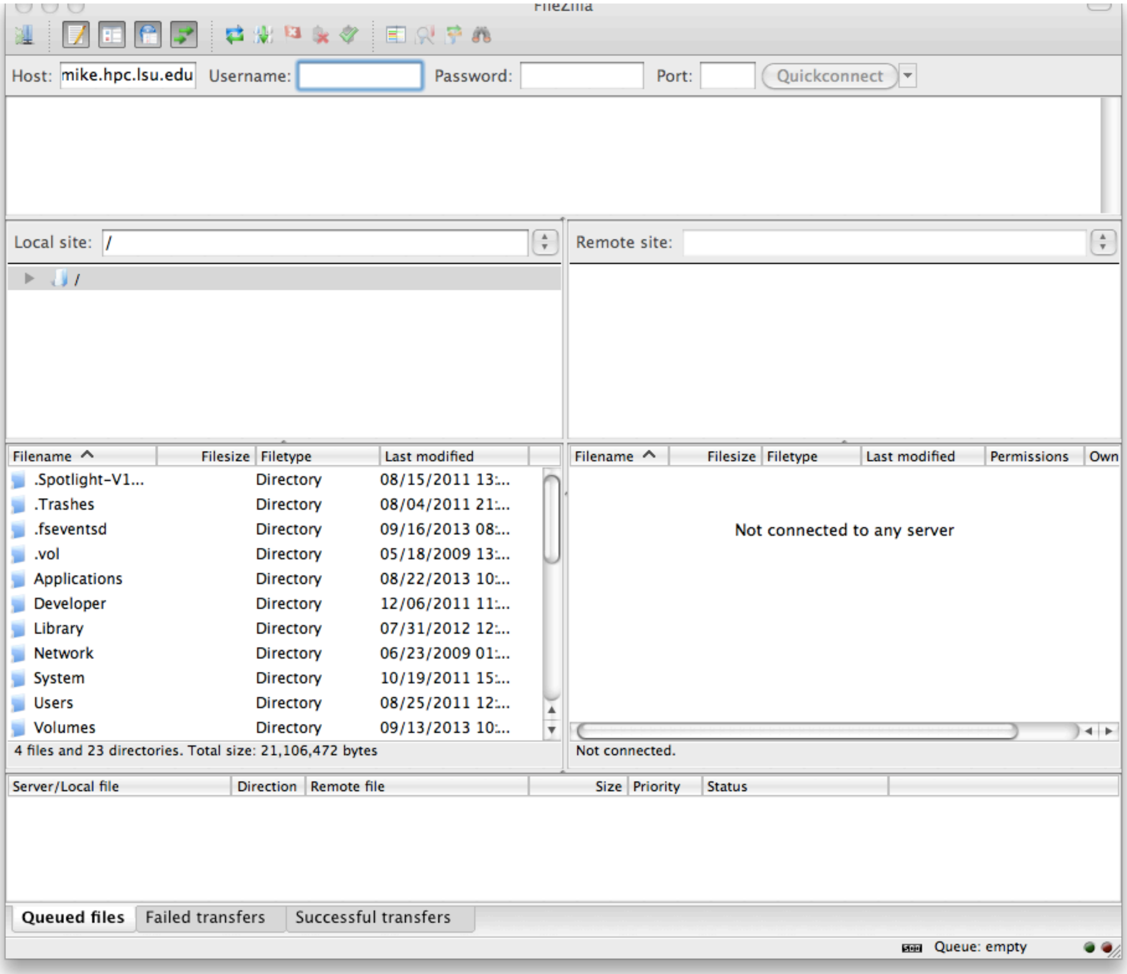
```
scp <options> <source> <destination>
rsync <options> <source> <destination>
```

```
$ scp
usage: scp [-1246BCEpqrv] [-c cipher] [-F ssh_config] [-i identity_file]
        [-l limit] [-o ssh_option] [-P port] [-S program]
        [[user@]host1:]file1 ... [[user@]host2:]file2
```

File Transfer

From a Windows machine

- Use a client that supports the scp protocol (e.g. WinSCP, Filezilla)



Host: mike.hpc.lsu.edu Username: Password: Port:

Local site: /

Remote site:

Filename	Filesize	Filetype	Last modified
.Spotlight-V1...		Directory	08/15/2011 13:...
.Trashes		Directory	08/04/2011 21:...
.fseventsd		Directory	09/16/2013 08:...
.vol		Directory	05/18/2009 13:...
Applications		Directory	08/22/2013 10:...
Developer		Directory	12/06/2011 11:...
Library		Directory	07/31/2012 12:...
Network		Directory	06/23/2009 01:...
System		Directory	10/19/2011 15:...
Users		Directory	08/25/2011 12:...
Volumes		Directory	09/13/2013 10:...

4 files and 23 directories. Total size: 21,106,472 bytes

Filename	Filesize	Filetype	Last modified	Permissions	Own

Not connected to any server

Not connected.

Server/Local file	Direction	Remote file	Size	Priority	Status

Queue: empty

Software

- Learn to use softenv
- Know your compilers
- Find your applications or port your stuff and setup your simulation

Using softenv

Environment variables

- PATH: where to look for executables
- LD_LIBRARY_PATH: where to look for shared libraries
- LD_INCLUDE_PATH: where to look for header and include files

Other environment variables sometimes needed by various software

- LIBRARY_PATH, C_LIBRARY_PATH
- LDFLAGS, LDLIBS

SOFTENV is a software that helps users set up environment variables properly to use other software package. Much more convenient than setting variables in .bashrc

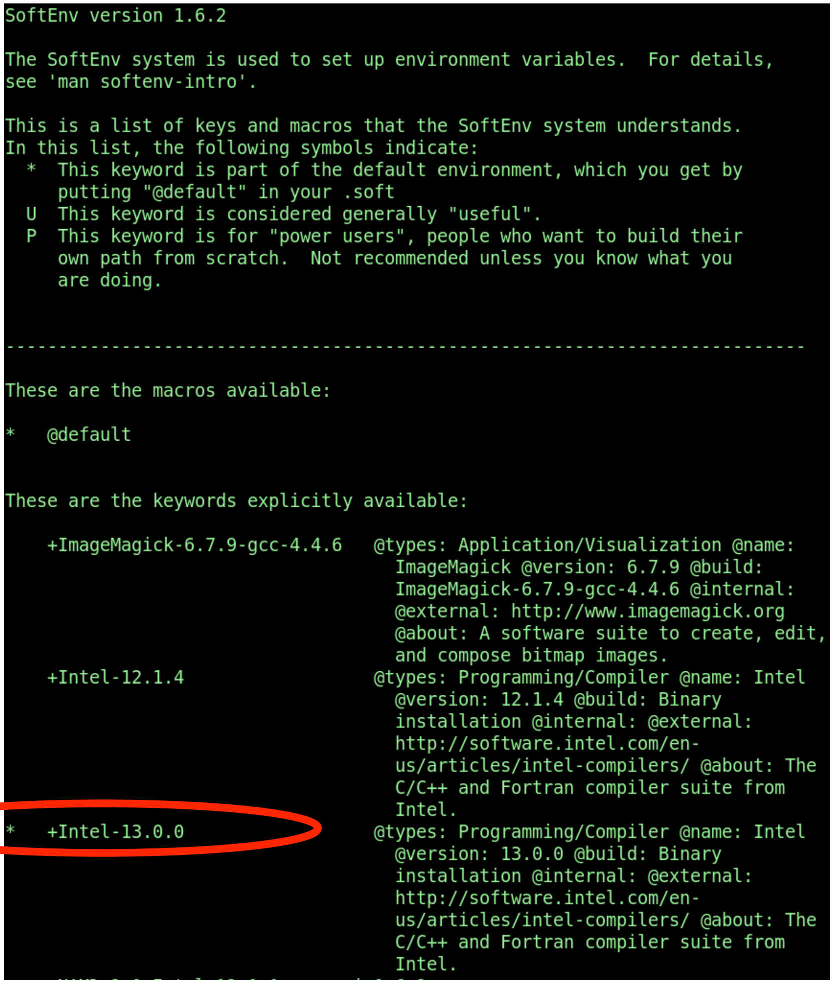
Modules is another software that helps users set up their environment. Most supercomputing sites have moved onto modules. We are also planning to move to modules with newer machines

Listing All Packages

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

Softenv on SuperMikeII shown here

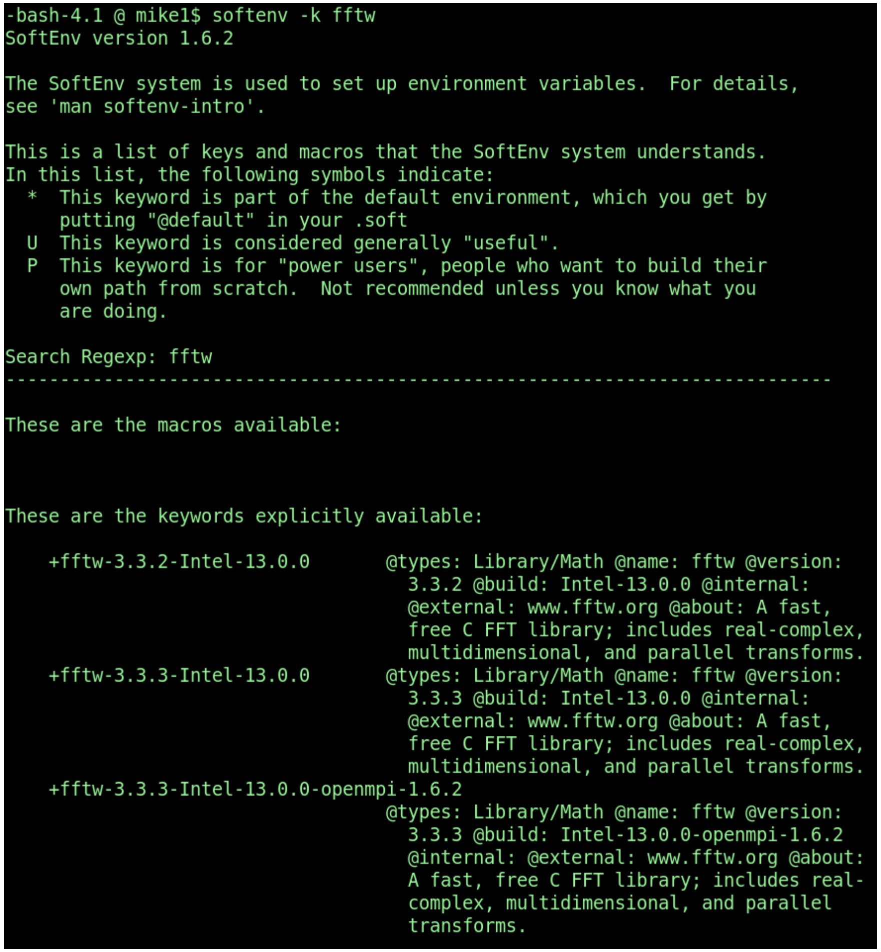
Softenv key



```
SoftEnv version 1.6.2

The SoftEnv system is used to set up environment variables.  For details,
see 'man softenv-intro'.

This is a list of keys and macros that the SoftEnv system understands.
In this list, the following symbols indicate:
  *  This keyword is part of the default environment, which you get by
     putting "@default" in your .soft
  U  This keyword is considered generally "useful".
  P  This keyword is for "power users", people who want to build their
     own path from scratch.  Not recommended unless you know what you
     are doing.


----------------------------------------------------------------------------

These are the macros available:

*   @default


These are the keywords explicitly available:

    +ImageMagick-6.7.9-gcc-4.4.6   @types: Application/Visualization @name:
                                     ImageMagick @version: 6.7.9 @build:
                                     ImageMagick-6.7.9-gcc-4.4.6 @internal:
                                     @external: http://www.imagemagick.org
                                     @about: A software suite to create, edit,
                                     and compose bitmap images.
    +Intel-12.1.4                  @types: Programming/Compiler @name: Intel
                                     @version: 12.1.4 @build: Binary
                                     installation @internal: @external:
                                     http://software.intel.com/en-
                                     us/articles/intel-compilers/ @about: The
                                     C/C++ and Fortran compiler suite from
                                     Intel.
*   +Intel-13.0.0                  @types: Programming/Compiler @name: Intel
                                     @version: 13.0.0 @build: Binary
                                     installation @internal: @external:
                                     http://software.intel.com/en-
                                     us/articles/intel-compilers/ @about: The
                                     C/C++ and Fortran compiler suite from
                                     Intel.
```

Searching A Specific Package

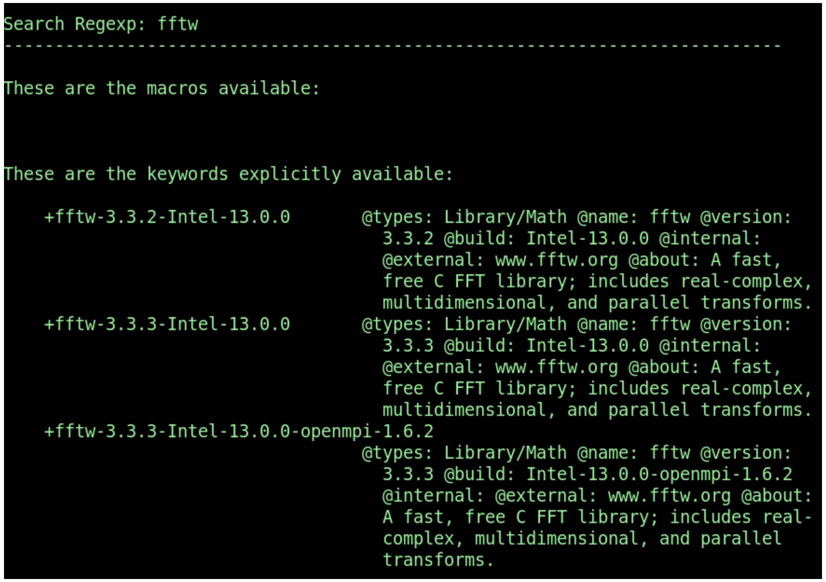
Use “–k” option with softenv command to search a particular key



```
-bash-4.1 @ mike1$ softenv -k fftw
SoftEnv version 1.6.2

The SoftEnv system is used to set up environment variables.  For details,
see 'man softenv-intro'.

This is a list of keys and macros that the SoftEnv system understands.
In this list, the following symbols indicate:
  *  This keyword is part of the default environment, which you get by
     putting "@default" in your .soft
  U  This keyword is considered generally "useful".
  P  This keyword is for "power users", people who want to build their
     own path from scratch.  Not recommended unless you know what you
     are doing.

Search Regexp: fftw
----------------------------------------------------------------------------

These are the macros available:



These are the keywords explicitly available:

    +fftw-3.3.2-Intel-13.0.0       @types: Library/Math @name: fftw @version:
                                     3.3.2 @build: Intel-13.0.0 @internal:
                                     @external: www.fftw.org @about: A fast,
                                     free C FFT library; includes real-complex,
                                     multidimensional, and parallel transforms.
    +fftw-3.3.3-Intel-13.0.0       @types: Library/Math @name: fftw @version:
                                     3.3.3 @build: Intel-13.0.0 @internal:
                                     @external: www.fftw.org @about: A fast,
                                     free C FFT library; includes real-complex,
                                     multidimensional, and parallel transforms.
    +fftw-3.3.3-Intel-13.0.0-openmpi-1.6.2
                                   @types: Library/Math @name: fftw @version:
                                     3.3.3 @build: Intel-13.0.0-openmpi-1.6.2
                                     @internal: @external: www.fftw.org @about:
                                     A fast, free C FFT library; includes real-
                                     complex, multidimensional, and parallel
                                     transforms.
```

Searching A Specific Package

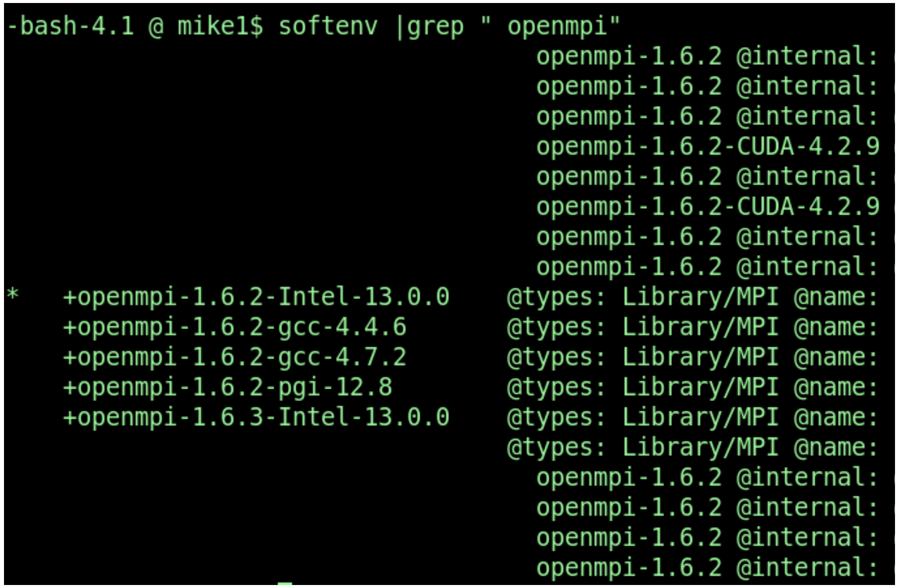
Use "–k" option with softenv command to search a key.



```
Search Regexp: fftw
----------------------------------------------------------------------------

These are the macros available:


These are the keywords explicitly available:

    +fftw-3.3.2-Intel-13.0.0       @types: Library/Math @name: fftw @version:
                                     3.3.2 @build: Intel-13.0.0 @internal:
                                     @external: www.fftw.org @about: A fast,
                                     free C FFT library; includes real-complex,
                                     multidimensional, and parallel transforms.
    +fftw-3.3.3-Intel-13.0.0       @types: Library/Math @name: fftw @version:
                                     3.3.3 @build: Intel-13.0.0 @internal:
                                     @external: www.fftw.org @about: A fast,
                                     free C FFT library; includes real-complex,
                                     multidimensional, and parallel transforms.
    +fftw-3.3.3-Intel-13.0.0-openmpi-1.6.2
                                   @types: Library/Math @name: fftw @version:
                                     3.3.3 @build: Intel-13.0.0-openmpi-1.6.2
                                     @internal: @external: www.fftw.org @about:
                                     A fast, free C FFT library; includes real-
                                     complex, multidimensional, and parallel
                                     transforms.
```

You can also grep

$ softenv |grep " openmpi"

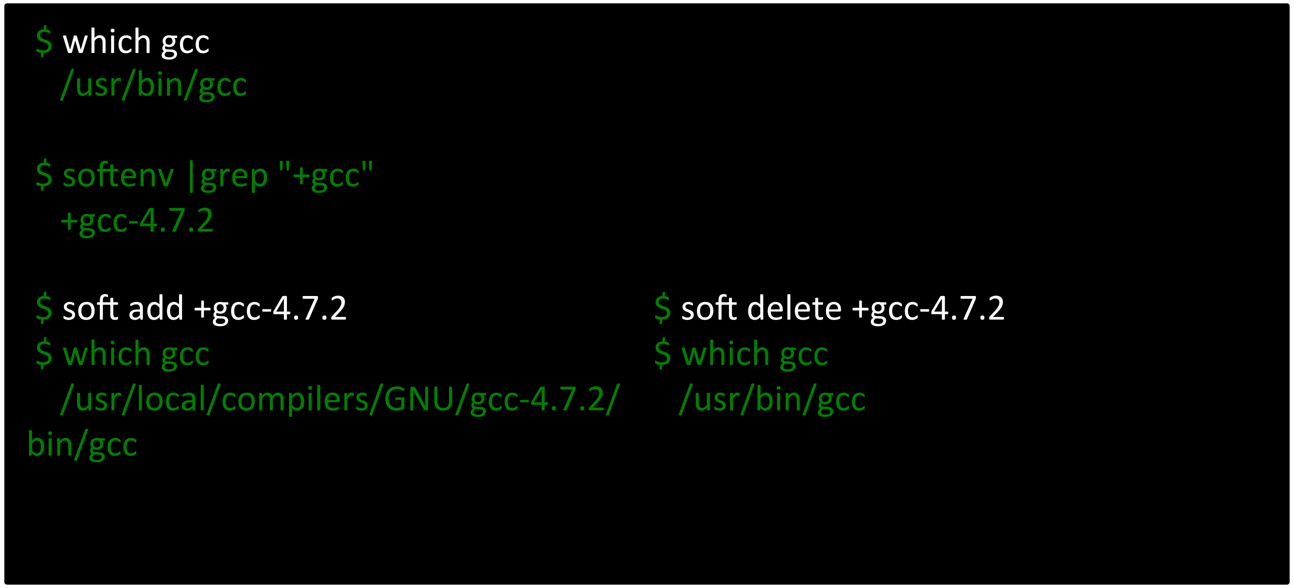


```
-bash-4.1 @ mike1$ softenv |grep " openmpi"
                                   openmpi-1.6.2 @internal:
                                   openmpi-1.6.2 @internal:
                                   openmpi-1.6.2 @internal:
                                   openmpi-1.6.2-CUDA-4.2.9
                                   openmpi-1.6.2 @internal:
                                   openmpi-1.6.2-CUDA-4.2.9
                                   openmpi-1.6.2 @internal:
                                   openmpi-1.6.2 @internal:
*   +openmpi-1.6.2-Intel-13.0.0    @types: Library/MPI @name:
    +openmpi-1.6.2-gcc-4.4.6       @types: Library/MPI @name:
    +openmpi-1.6.2-gcc-4.7.2       @types: Library/MPI @name:
    +openmpi-1.6.2-pgi-12.8        @types: Library/MPI @name:
    +openmpi-1.6.3-Intel-13.0.0    @types: Library/MPI @name:
                                 @types: Library/MPI @name:
                                   openmpi-1.6.2 @internal:
                                   openmpi-1.6.2 @internal:
                                   openmpi-1.6.2 @internal:
                                   openmpi-1.6.2 @internal:
```

Setting up Environment via Softenv : *One time change*

Set up the environment to use a package **in the current session only**

- Add a package: soft add <key>
- Remove a package: soft delete <key>

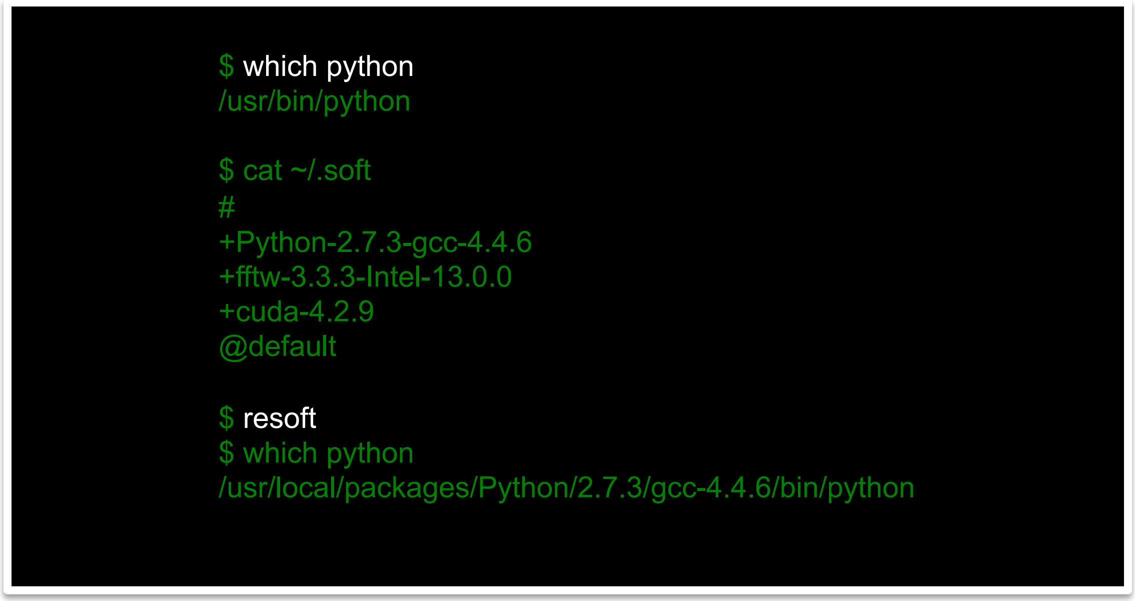


```
$ which gcc
  /usr/bin/gcc

$ softenv |grep "+gcc"
  +gcc-4.7.2

$ soft add +gcc-4.7.2                          $ soft delete +gcc-4.7.2
$ which gcc                                    $ which gcc
  /usr/local/compilers/GNU/gcc-4.7.2/            /usr/bin/gcc
bin/gcc
```

Setting up Environment via Softenv: *Permanent change*

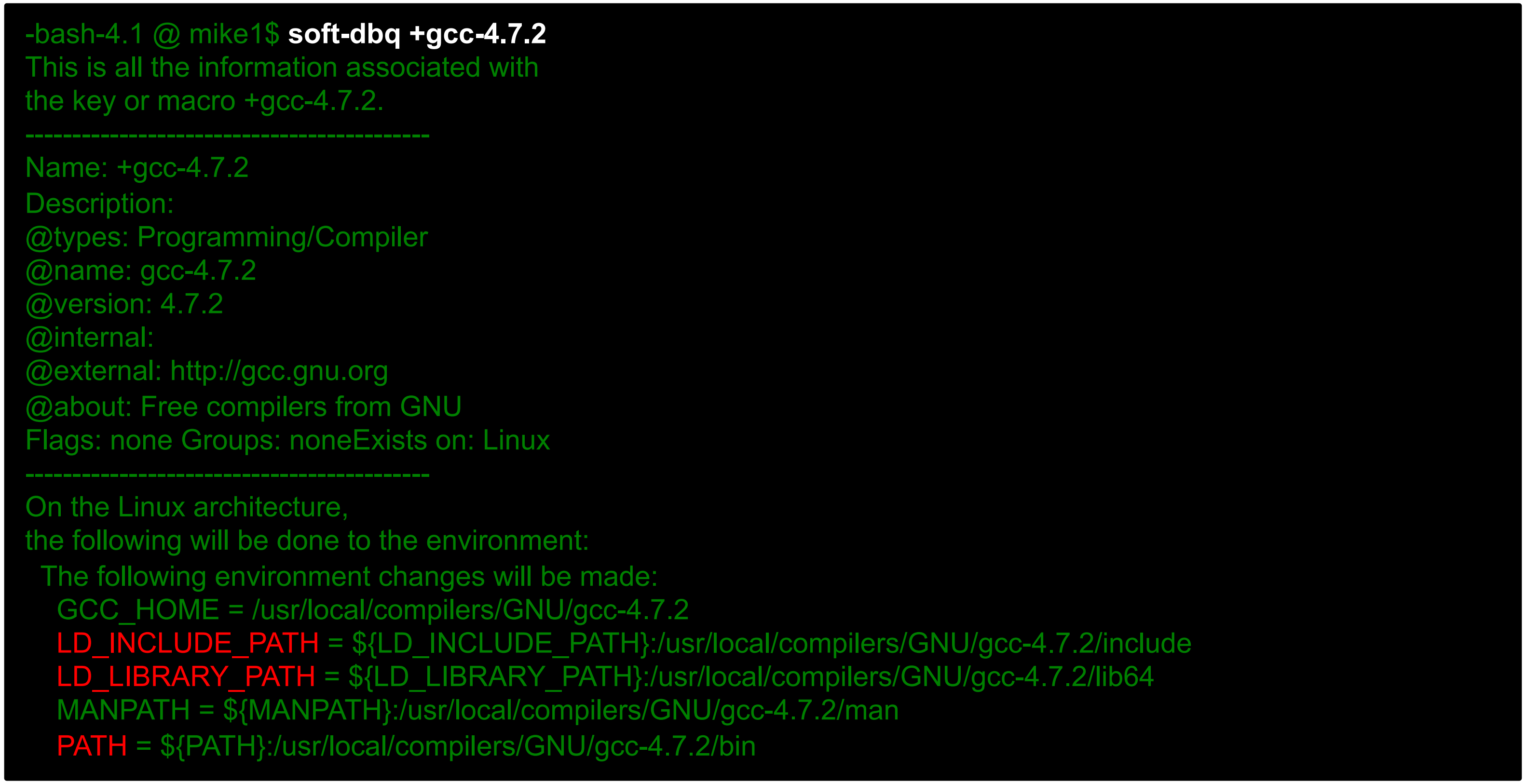
Set up the environment variables to use a certain software package

- First add the key to $HOME/.soft
- Then execute resoft at the command line
- The environment will be the same next time you log in



```
$ which python
/usr/bin/python

$ cat ~/.soft
#
+Python-2.7.3-gcc-4.4.6
+fftw-3.3.3-Intel-13.0.0
+cuda-4.2.9
@default

$ resoft
$ which python
/usr/local/packages/Python/2.7.3/gcc-4.4.6/bin/python
```

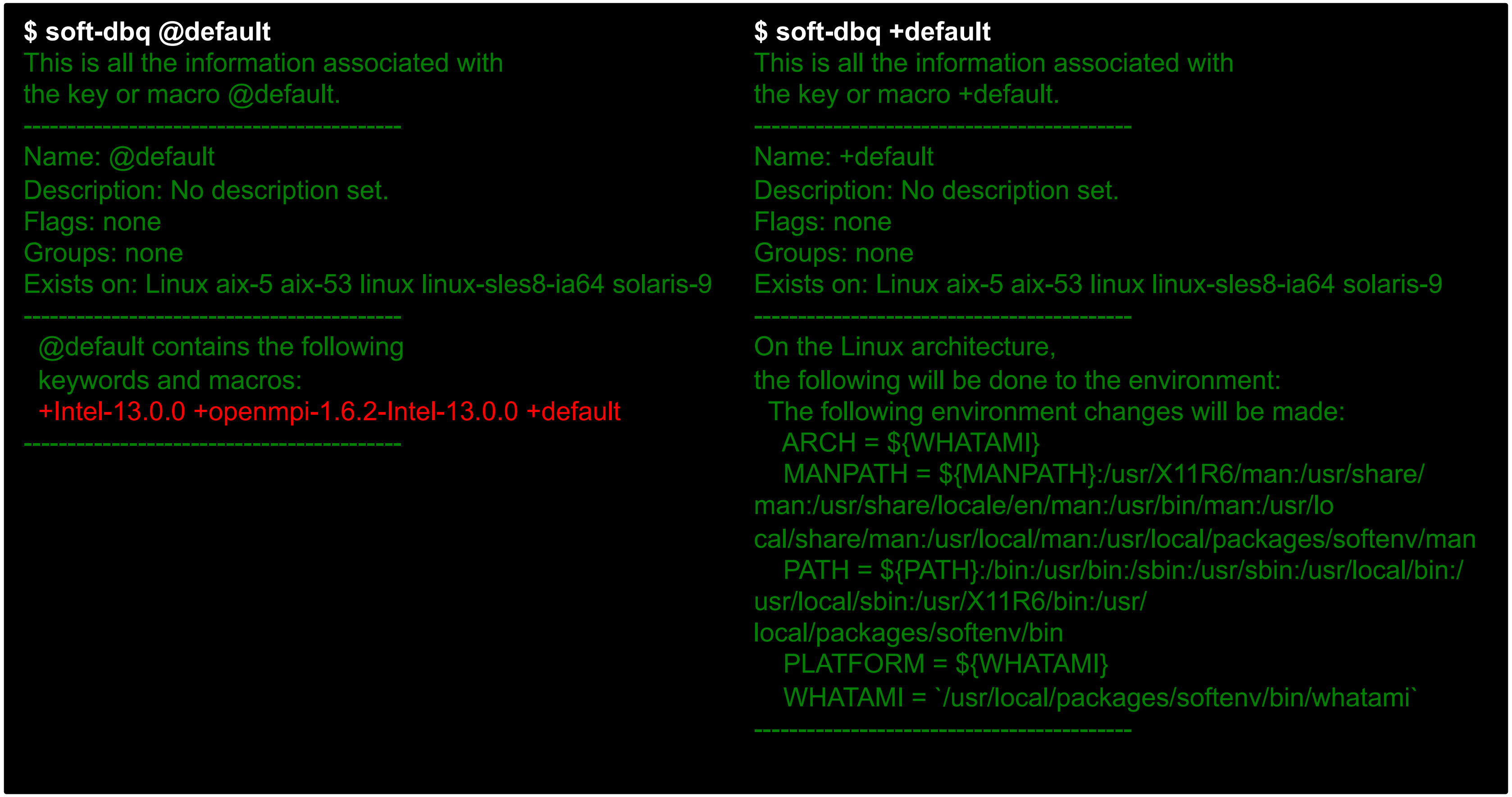
“soft-dbq” : Querying a Softenv key



```
-bash-4.1 @ mike1$ soft-dbq +gcc-4.7.2
This is all the information associated with
the key or macro +gcc-4.7.2.
-------------------------------------------
Name: +gcc-4.7.2
Description:
@types: Programming/Compiler
@name: gcc-4.7.2
@version: 4.7.2
@internal:
@external: http://gcc.gnu.org
@about: Free compilers from GNU
Flags: none Groups: noneExists on: Linux
-------------------------------------------
On the Linux architecture,
the following will be done to the environment:
  The following environment changes will be made:
    GCC_HOME = /usr/local/compilers/GNU/gcc-4.7.2
    LD_INCLUDE_PATH = ${LD_INCLUDE_PATH}:/usr/local/compilers/GNU/gcc-4.7.2/include
    LD_LIBRARY_PATH = ${LD_LIBRARY_PATH}:/usr/local/compilers/GNU/gcc-4.7.2/lib64
    MANPATH = ${MANPATH}:/usr/local/compilers/GNU/gcc-4.7.2/man
    PATH = ${PATH}:/usr/local/compilers/GNU/gcc-4.7.2/bin
```

“soft-dbq” : Querying a Softenv key

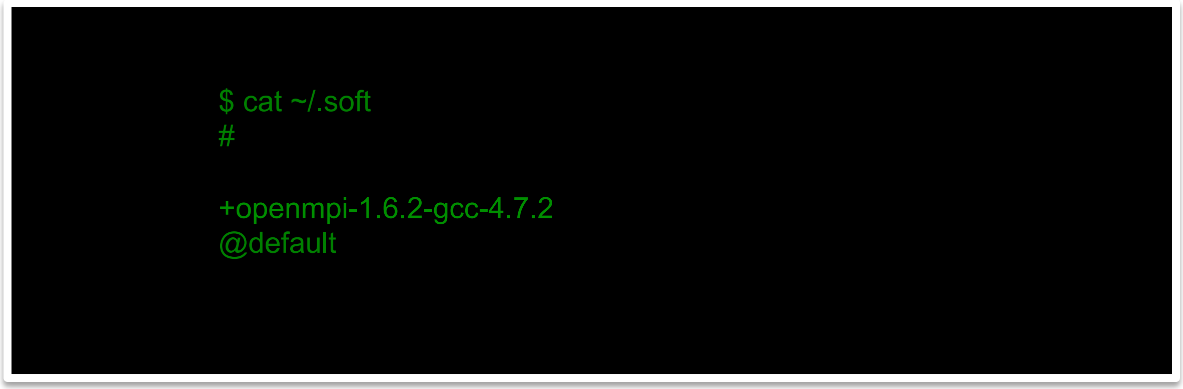
Do not remove the @default key



```
$ soft-dbq @default
This is all the information associated with
the key or macro @default.
-------------------------------------------
Name: @default
Description: No description set.
Flags: none
Groups: none
Exists on: Linux aix-5 aix-53 linux linux-sles8-ia64 solaris-9
-------------------------------------------
  @default contains the following
  keywords and macros:
  +Intel-13.0.0 +openmpi-1.6.2-Intel-13.0.0 +default
-------------------------------------------
```

```
$ soft-dbq +default
This is all the information associated with
the key or macro +default.
-------------------------------------------
Name: +default
Description: No description set.
Flags: none
Groups: none
Exists on: Linux aix-5 aix-53 linux linux-sles8-ia64 solaris-9
-------------------------------------------
On the Linux architecture,
the following will be done to the environment:
  The following environment changes will be made:
    ARCH = ${WHATAMI}
    MANPATH = ${MANPATH}:/usr/X11R6/man:/usr/share/
man:/usr/share/locale/en/man:/usr/bin/man:/usr/lo
cal/share/man:/usr/local/man:/usr/local/packages/softenv/man
    PATH = ${PATH}:/bin:/usr/bin:/sbin:/usr/sbin:/usr/local/bin:/
usr/local/sbin:/usr/X11R6/bin:/usr/
local/packages/softenv/bin
    PLATFORM = ${WHATAMI}
    WHATAMI = `/usr/local/packages/softenv/bin/whatami`
-------------------------------------------
```

Using softenv: *Quiz*

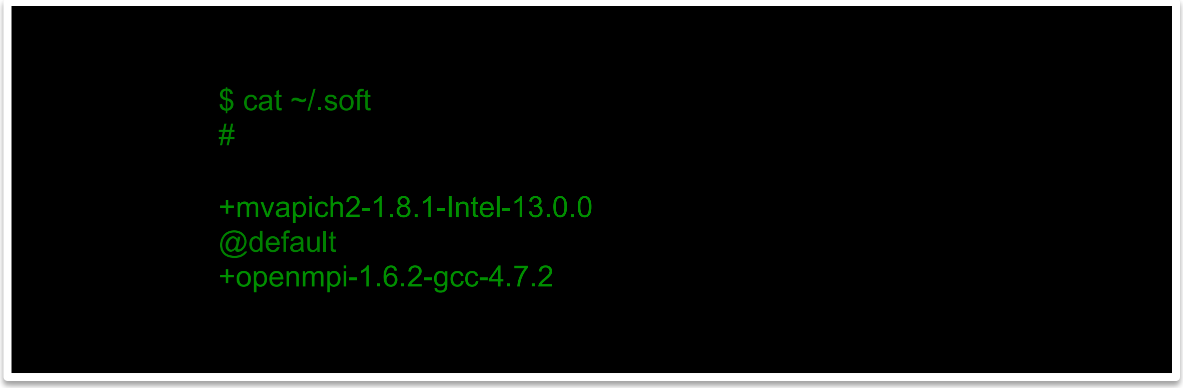


```
$ cat ~/.soft
#

+openmpi-1.6.2-gcc-4.7.2
@default
```

Which mpif90/mpirun will the system use if u just call mpif90/ mpirun?

Which compiler will be used?

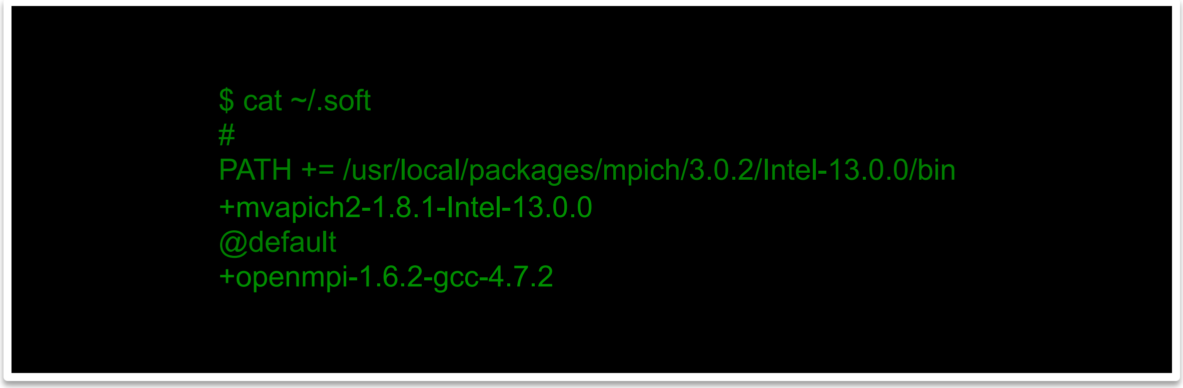
Using softenv: *Quiz*



```
$ cat ~/.soft
#

+mvapich2-1.8.1-Intel-13.0.0
@default
+openmpi-1.6.2-gcc-4.7.2
```

Which mpif90/mpirun will the system use if u just call mpirun?

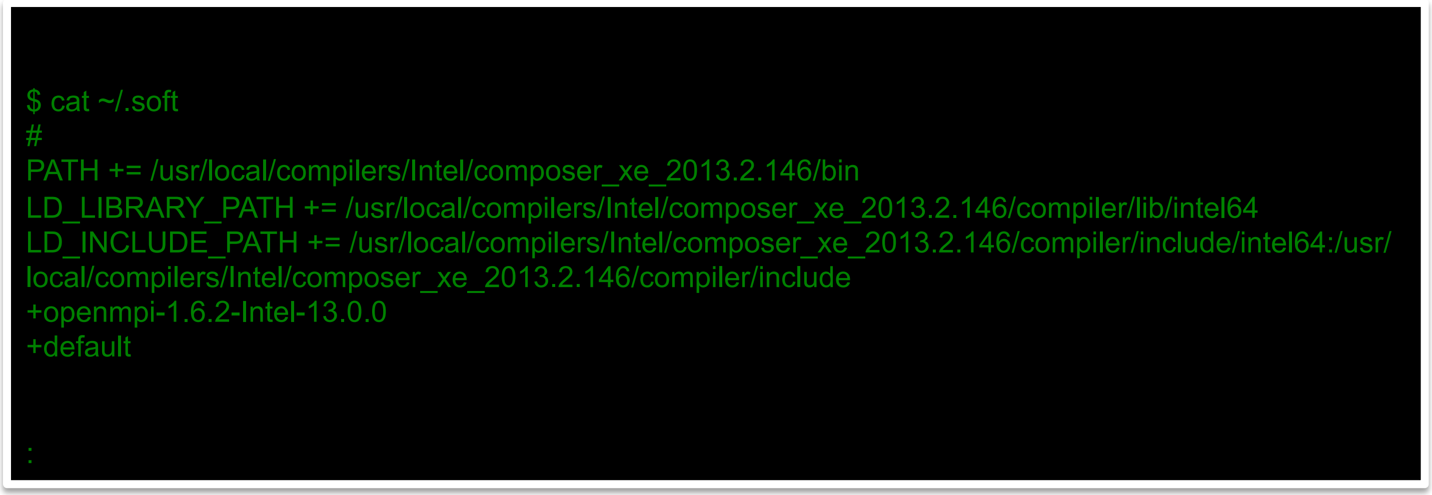
Using softenv: *Quiz*



```
$ cat ~/.soft
#
PATH += /usr/local/packages/mpich/3.0.2/Intel-13.0.0/bin
+mvapich2-1.8.1-Intel-13.0.0
@default
+openmpi-1.6.2-gcc-4.7.2
```

Which mpif90/mpirun will the system use if u just call mpirun?

Using softenv: *Quiz*



```
$ cat ~/.soft
#
PATH += /usr/local/compilers/Intel/composer_xe_2013.2.146/bin
LD_LIBRARY_PATH += /usr/local/compilers/Intel/composer_xe_2013.2.146/compiler/lib/intel64
LD_INCLUDE_PATH += /usr/local/compilers/Intel/composer_xe_2013.2.146/compiler/include/intel64:/usr/
local/compilers/Intel/composer_xe_2013.2.146/compiler/include
+openmpi-1.6.2-Intel-13.0.0
+default


:
```

Which version of intel fortran compiler will be displayed by the commands “mpif90 –version” ?

Exercise : Use Softenv

- Find the key for Python 2.7.3
- Check what variables are set through the key
- Set up your environment to use Python 2.7.3
- Check if the variables are correctly set by "which python"
- Check if you have access to ipython, scipy, numpy, matplotlib

Compilers

Language	Linux cluster			AIX clusters
	Intel	PGI	GNU	XL
Fortran	ifort	pgf77, pgf90	gfortran	xlf, xlf90
C	icc	pgcc	gcc	xlc
C++	icpc	pgCC	g++	xlC

Serial compilers

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

Parallel compilers

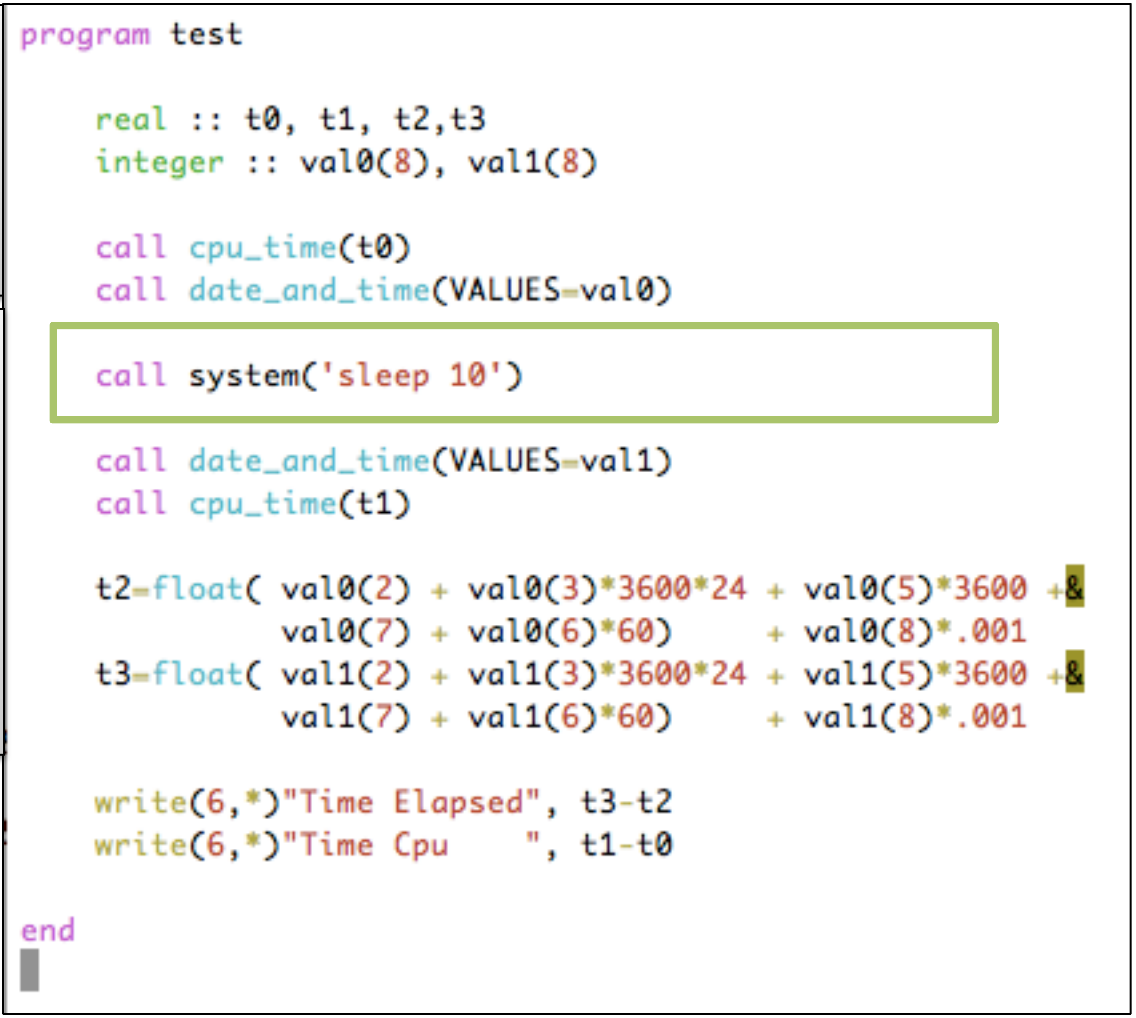
Compiling serial Fortran code

To compile the program, use any

```
$ ifort  test_hello2.f90
$ gfortran  test_hello2.f90
```

To verify which compiler was used

```
$ nm a.out |grep -i intel
… __intel_cpu_indicator

$ nm -s a.out |grep -i gfortran
… _gfortran_cpu_time_4@…
```



```
program test

    real :: t0, t1, t2,t3
    integer :: val0(8), val1(8)

    call cpu_time(t0)
    call date_and_time(VALUES=val0)

    call system('sleep 10')

    call date_and_time(VALUES=val1)
    call cpu_time(t1)

    t2=float( val0(2) + val0(3)*3600*24 + val0(5)*3600 +&
              val0(7) + val0(6)*60)      + val0(8)*.001
    t3=float( val1(2) + val1(3)*3600*24 + val1(5)*3600 +&
              val1(7) + val1(6)*60)      + val1(8)*.001

    write(6,*)"Time Elapsed", t3-t2
    write(6,*)"Time Cpu    ", t1-t0

end
```

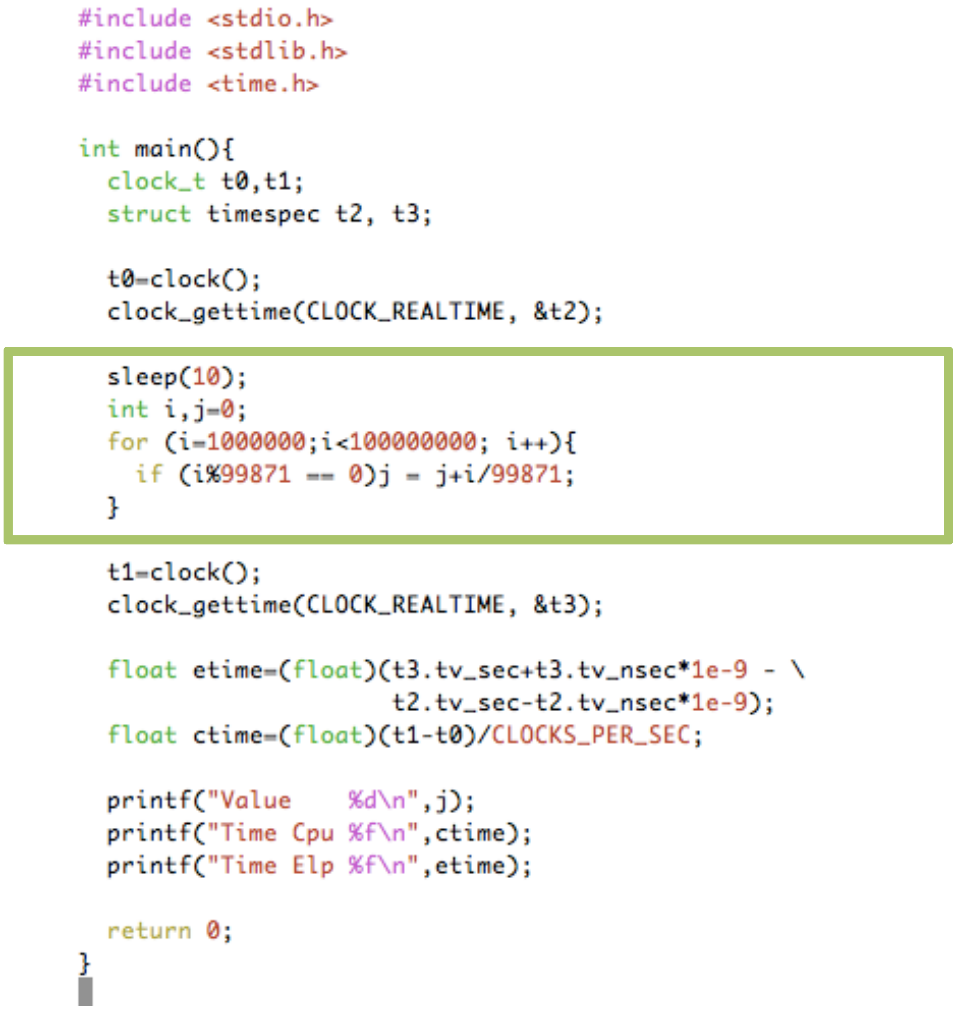
Compiling serial C code

To compile the program, use any

```
$ gcc test_hello2.c –lrt
$ icc test_hello2.c -lrt
```

Cpu vs Elapsed time

```
$ ./a.out
Value    501446
Time Cpu 0.320000
Time Elp 10.326020

$ ./a.out
Value    501446
Time Cpu 0.190000
Time Elp 10.198743
```



```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(){
  clock_t t0,t1;
  struct timespec t2, t3;

  t0=clock();
  clock_gettime(CLOCK_REALTIME, &t2);

  sleep(10);
  int i,j=0;
  for (i=1000000;i<100000000; i++){
    if (i%99871 == 0)j = j+i/99871;
  }

  t1=clock();
  clock_gettime(CLOCK_REALTIME, &t3);

  float etime=(float)(t3.tv_sec+t3.tv_nsec*1e-9 - \
                      t2.tv_sec-t2.tv_nsec*1e-9);
  float ctime=(float)(t1-t0)/CLOCKS_PER_SEC;

  printf("Value    %d\n",j);
  printf("Time Cpu %f\n",ctime);
  printf("Time Elp %f\n",etime);

  return 0;
}
```

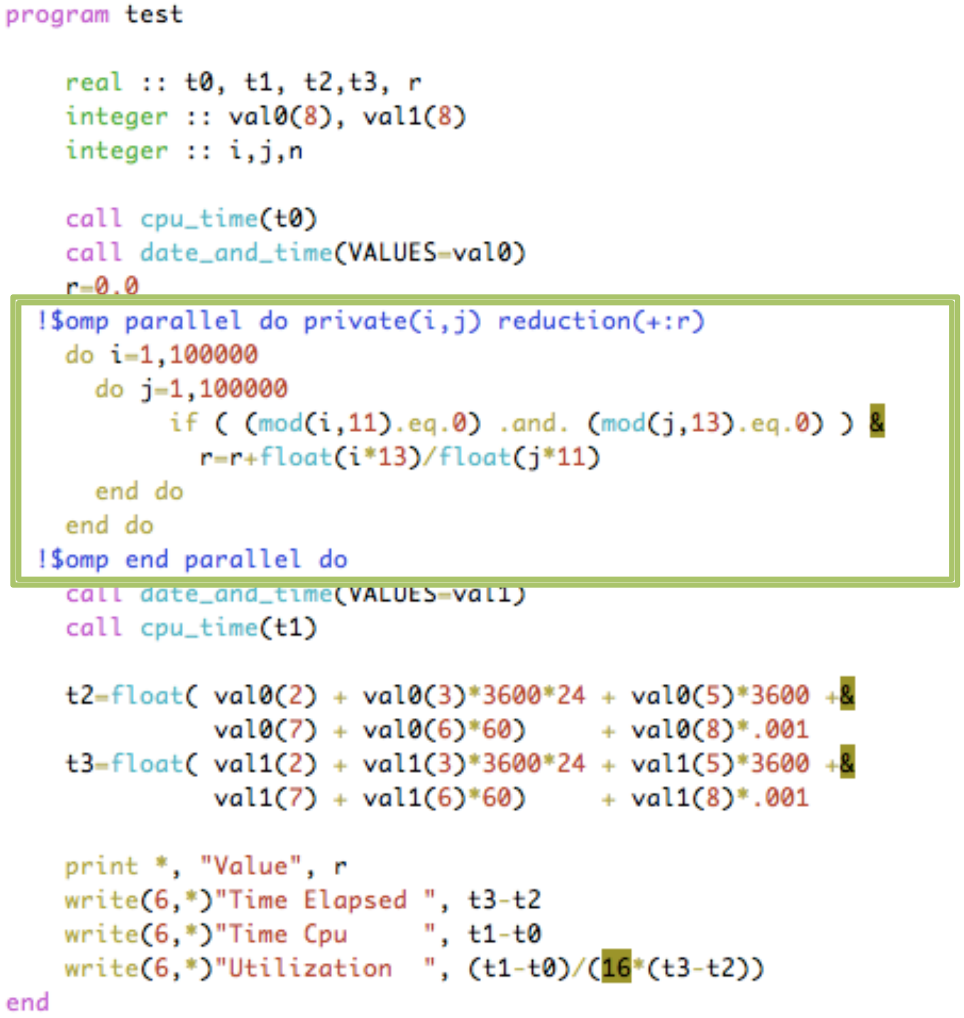
Compiling threaded Fortran code

To compile the program, use any

```
$ ifort  -openmp test_hello3.f90
$ gfortran  -fopenmp test_hello3.f90
```

Verify execution with intel

```
$ export OMP_NUM_THREADS=16
$ ./a.out
 Value  3.9361696E+08
 Time Elapsed   0.1250000
 Time Cpu       1.935706
 Utilization    0.9678530
```



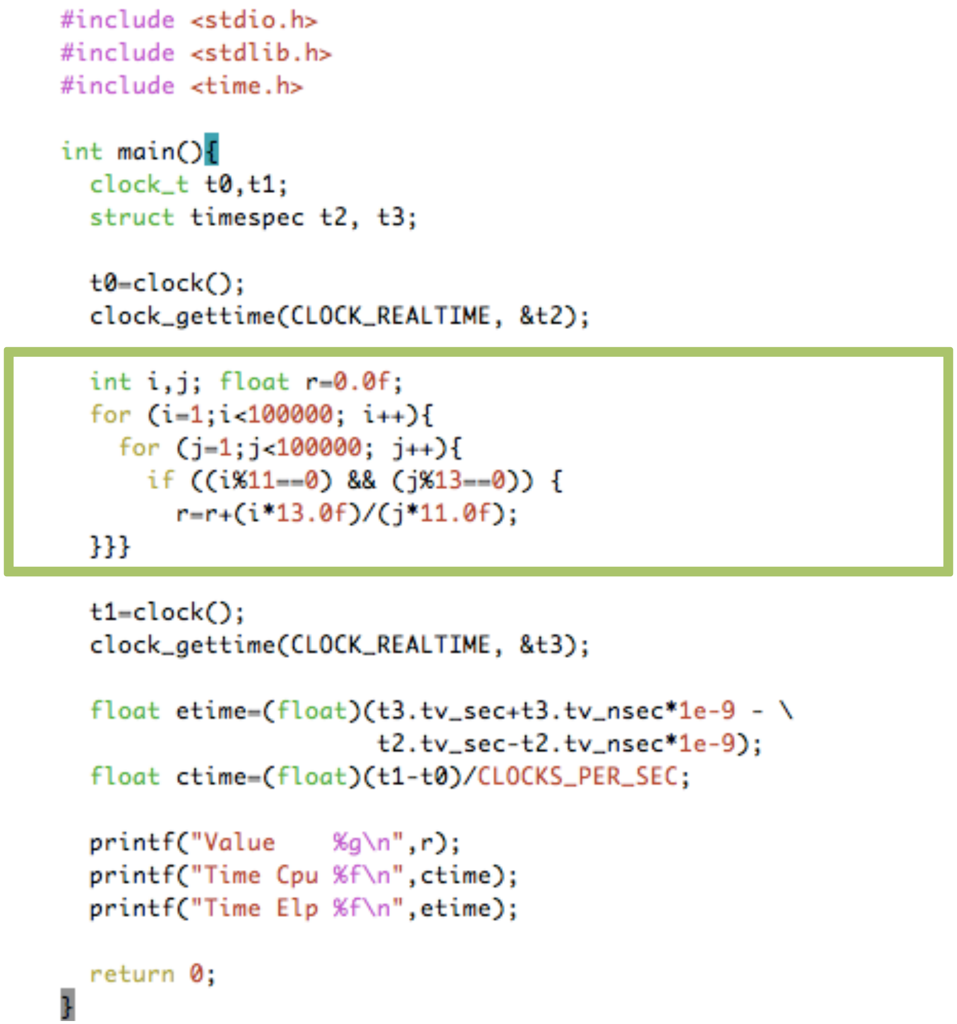
```
program test

    real :: t0, t1, t2,t3, r
    integer :: val0(8), val1(8)
    integer :: i,j,n

    call cpu_time(t0)
    call date_and_time(VALUES=val0)
    r=0.0
  !$omp parallel do private(i,j) reduction(+:r)
    do i=1,100000
      do j=1,100000
           if ( (mod(i,11).eq.0) .and. (mod(j,13).eq.0) ) &
             r=r+float(i*13)/float(j*11)
      end do
    end do
  !$omp end parallel do
    call date_and_time(VALUES=val1)
    call cpu_time(t1)

    t2=float( val0(2) + val0(3)*3600*24 + val0(5)*3600 +&
              val0(7) + val0(6)*60)     + val0(8)*.001
    t3=float( val1(2) + val1(3)*3600*24 + val1(5)*3600 +&
              val1(7) + val1(6)*60)     + val1(8)*.001

    print *, "Value", r
    write(6,*)"Time Elapsed ", t3-t2
    write(6,*)"Time Cpu     ", t1-t0
    write(6,*)"Utilization  ", (t1-t0)/(16*(t3-t2))
end
```

Exercise: threaded C code

Parallelize this code with OpenMP

1. Put in openmp directives
2. Compile with additional openmp flags

This might be tough if you are not used to programming



```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(){
  clock_t t0,t1;
  struct timespec t2, t3;

  t0=clock();
  clock_gettime(CLOCK_REALTIME, &t2);

  int i,j; float r=0.0f;
  for (i=1;i<100000; i++){
    for (j=1;j<100000; j++){
      if ((i%11==0) && (j%13==0)) {
        r=r+(i*13.0f)/(j*11.0f);
  }}}

  t1=clock();
  clock_gettime(CLOCK_REALTIME, &t3);

  float etime=(float)(t3.tv_sec+t3.tv_nsec*1e-9 - \
                      t2.tv_sec-t2.tv_nsec*1e-9);
  float ctime=(float)(t1-t0)/CLOCKS_PER_SEC;

  printf("Value    %g\n",r);
  printf("Time Cpu %f\n",ctime);
  printf("Time Elp %f\n",etime);

  return 0;
}
```

MPI libraries

	Name	**MPI Library**				Default serial compiler
Cluster Resource		Mvapich	Mvapich2	Openmpi	mpich	
LONI	Queen Bee	.98, 1.1	1.4, 1.6, 1.8.1	1.3.4	X	Intel 11.1
	Other LONI	.98, 1.1	1.4, 1.6	1.3.4	X	Intel 11.1
LSU	Tezpur	.98, 1.1	1.4, 1.6	1.3.4	X	Intel 11.1
	Philip	X	X	1.4.3, 1.6.1	1.2.7, 1.3.2, 1.4.1	Intel 11.1
	SuperMikeII	X	1.6, 1.9	1.6.x, 1.9ax	3.0.x	Intel 13.0.0
	Pandora	X	X	X	X	AIX

MPI Compilers

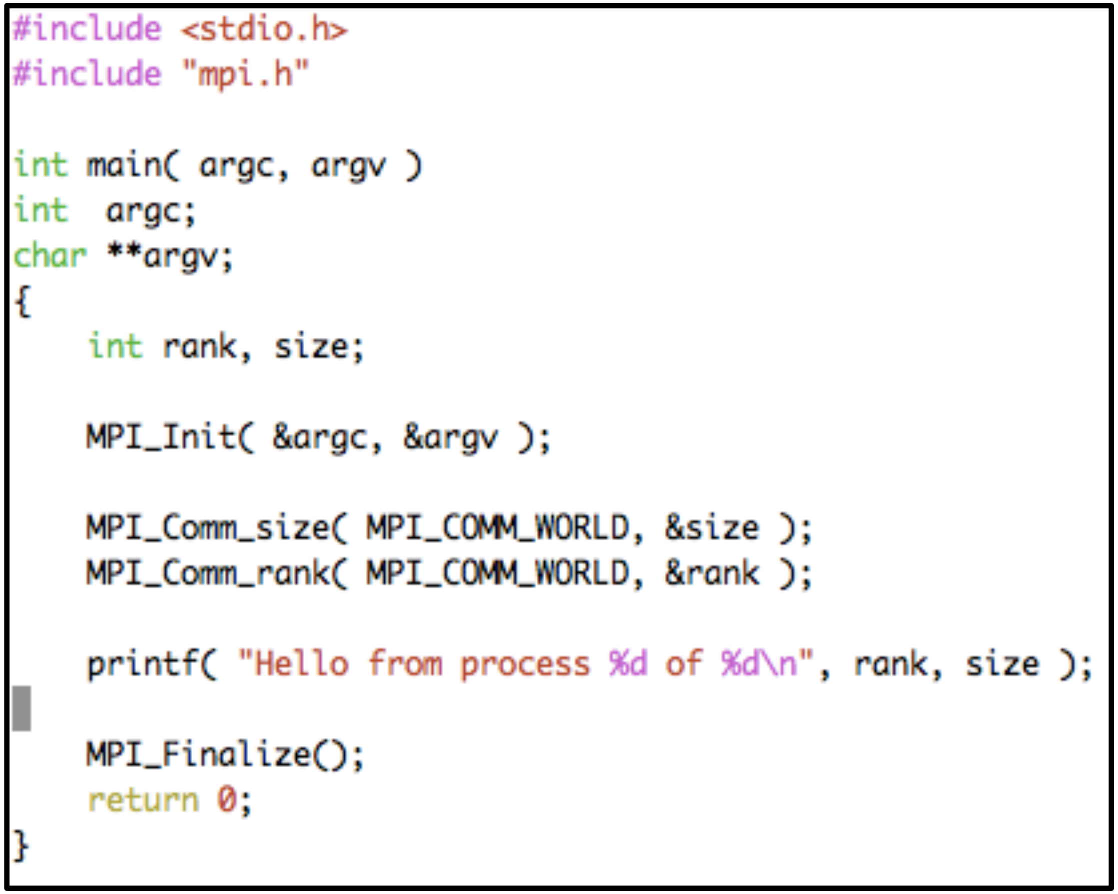
Language	Linux clusters	AIX clusters
Fortran	mpif77, mpif90	mpxlf, mpxlf90
C	mpicc	mpcc
C++	mpiCC	mpCC

```
mpif90  hello.f90

mpicc hello.c

mpicxx hello.cpp
```

Compiling a MPI C program

Compiling Hello world:

mpicc hello_mpi.c

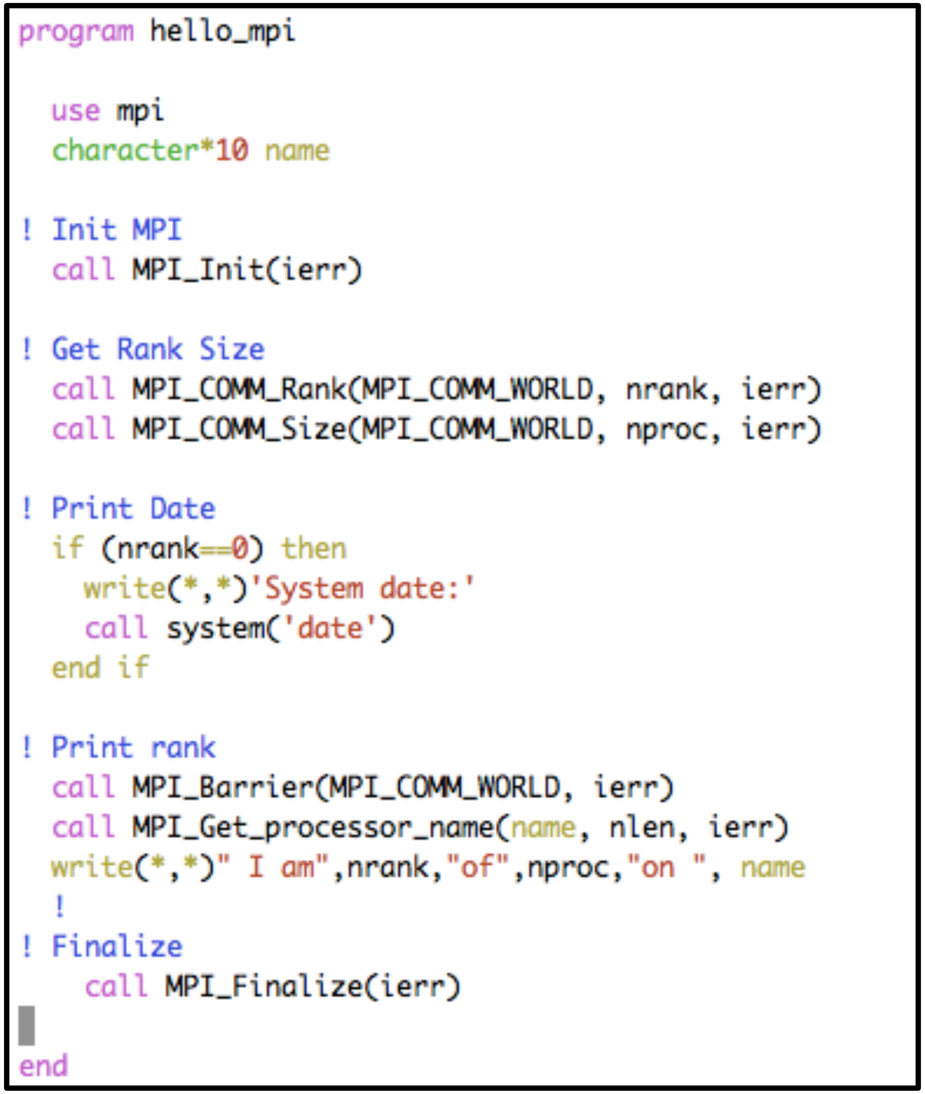


```
#include <stdio.h>
#include "mpi.h"

int main( argc, argv )
int  argc;
char **argv;
{
    int rank, size;

    MPI_Init( &argc, &argv );

    MPI_Comm_size( MPI_COMM_WORLD, &size );
    MPI_Comm_rank( MPI_COMM_WORLD, &rank );

    printf( "Hello from process %d of %d\n", rank, size );

    MPI_Finalize();
    return 0;
}
```

Compiling a MPI Fortran program

Compiling Hello world:

mpif90 hello_more.f90



```
program hello_mpi

  use mpi
  character*10 name

! Init MPI
  call MPI_Init(ierr)

! Get Rank Size
  call MPI_COMM_Rank(MPI_COMM_WORLD, nrank, ierr)
  call MPI_COMM_Size(MPI_COMM_WORLD, nproc, ierr)

! Print Date
  if (nrank==0) then
    write(*,*)'System date:'
    call system('date')
  end if

! Print rank
  call MPI_Barrier(MPI_COMM_WORLD, ierr)
  call MPI_Get_processor_name(name, nlen, ierr)
  write(*,*)" I am",nrank,"of",nproc,"on ", name
  !
! Finalize
    call MPI_Finalize(ierr)

end
```

Compiling a MPI program

Always verify what compiler/library is being used

```
$ mpicc -show
icc -I/usr/local/packages/openmpi/1.6.2/Intel-13.0.0/include
    -L/usr/local/packages/openmpi/1.6.2/Intel-13.0.0/lib
    -lmpi -ldl -lm -Wl,--export-dynamic -lrt -lnsl
    -libverbs -libumad -lpthread -lutil

$ mpif90 -show
ifort -I/usr/local/packages/openmpi/1.6.2/Intel-13.0.0/include
      -L/usr/local/packages/openmpi/1.6.2/Intel-13.0.0/lib
      -lmpi_f90 -lmpi_f77 -lmpi
      -ldl -lm -Wl,--export-dynamic -lrt -lnsl -libverbs -libumad -lpthread -lutil
```

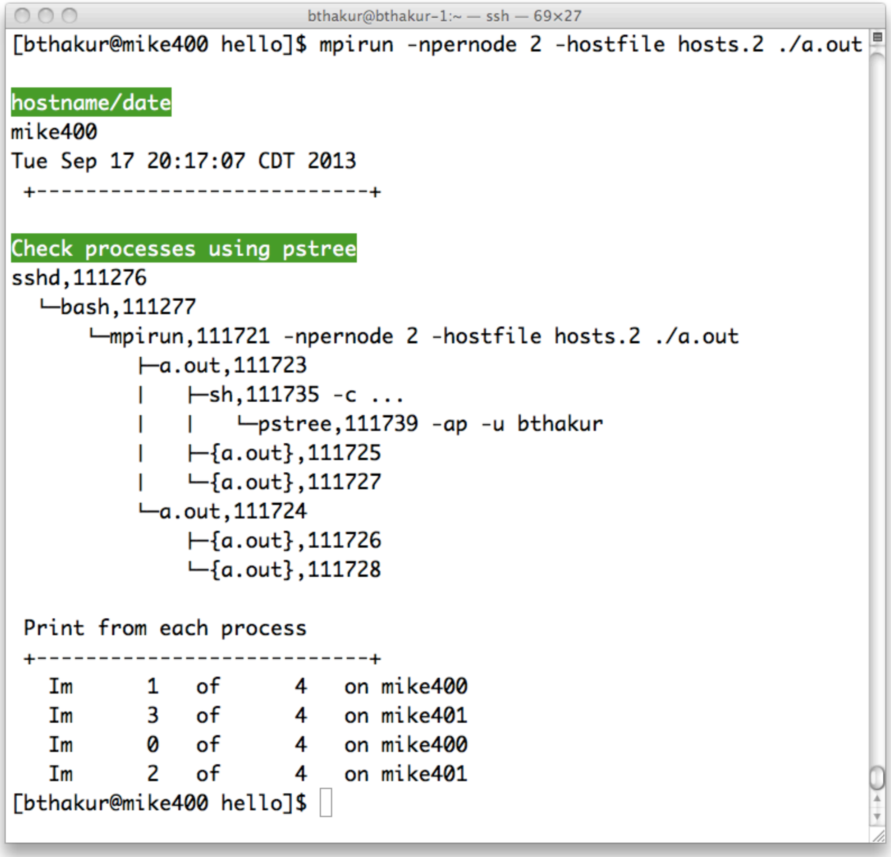
Compiling a MPI program

Always verify what library is being used: Before and after !

```
$ ldd a.out
…
libmpi_f90.so.1 => /usr/local/packages/openmpi/1.6.2/Intel-13.0.0/lib/
libmpi_f90.so.1 (0x00002ba5fb16b000)
libmpi.so.1 => /usr/local/packages/openmpi/1.6.2/Intel-13.0.0/lib/
libmpi.so.1 (0x00002ba5fb5a6000)
libibverbs.so.1 => /usr/lib64/libibverbs.so.1 (0x0000003ec5c00000)
…
libpthread.so.0 => /lib64/libpthread.so.0 (0x0000003e53e00000)
…
libifport.so.5 => /usr/local/compilers/Intel/composer_xe_2013.0.079/
compiler/lib/intel64/libifport.so.5 (0x00002ba5fbbdb000)
```

Analysing a parallel(mpi) program

Running a mpi program:
A process perspective

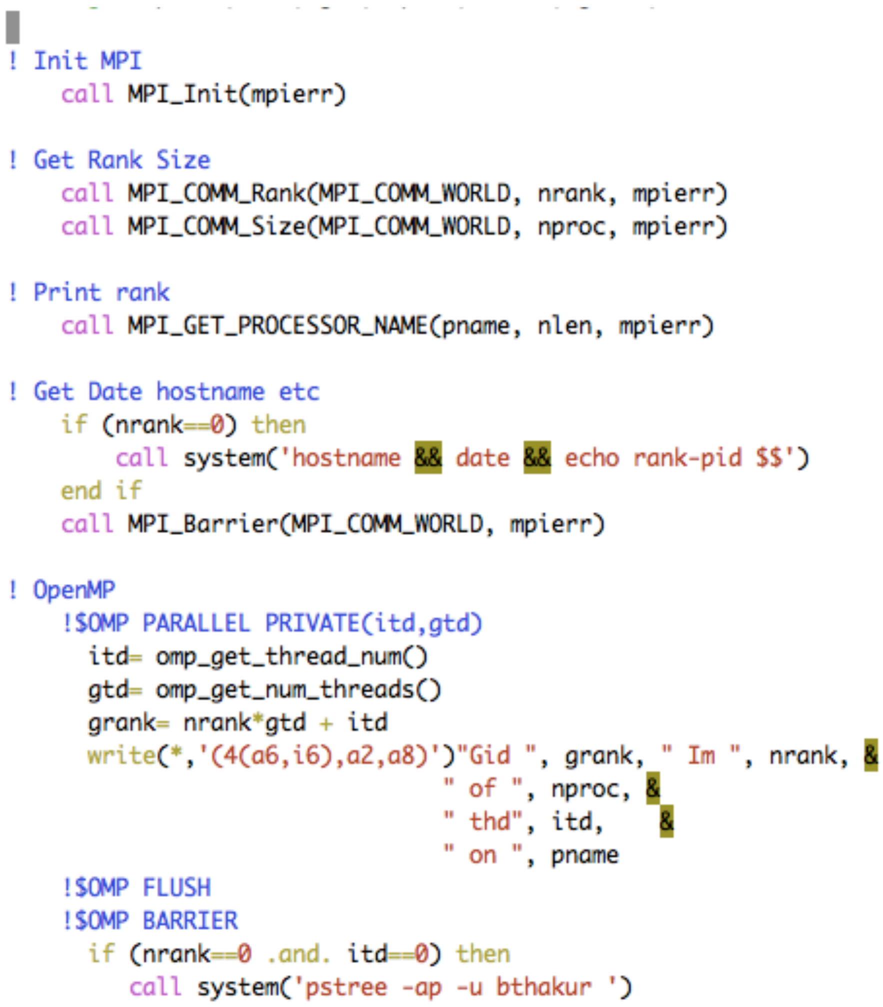


```
[bthakur@mike400 hello]$ mpirun -npernode 2 -hostfile hosts.2 ./a.out

hostname/date
mike400
Tue Sep 17 20:17:07 CDT 2013
 +---------------------------+

Check processes using pstree
sshd,111276
  └─bash,111277
      └─mpirun,111721 -npernode 2 -hostfile hosts.2 ./a.out
          ├─a.out,111723
          |   ├─sh,111735 -c ...
          |   |   └─pstree,111739 -ap -u bthakur
          |   ├─{a.out},111725
          |   └─{a.out},111727
          └─a.out,111724
              ├─{a.out},111726
              └─{a.out},111728

 Print from each process
 +---------------------------+
   Im       1   of      4   on mike400
   Im       3   of      4   on mike401
   Im       0   of      4   on mike400
   Im       2   of      4   on mike401
[bthakur@mike400 hello]$ 
```

Analyzing a Hybrid parallel program

Compiling Hybrid Hello world:

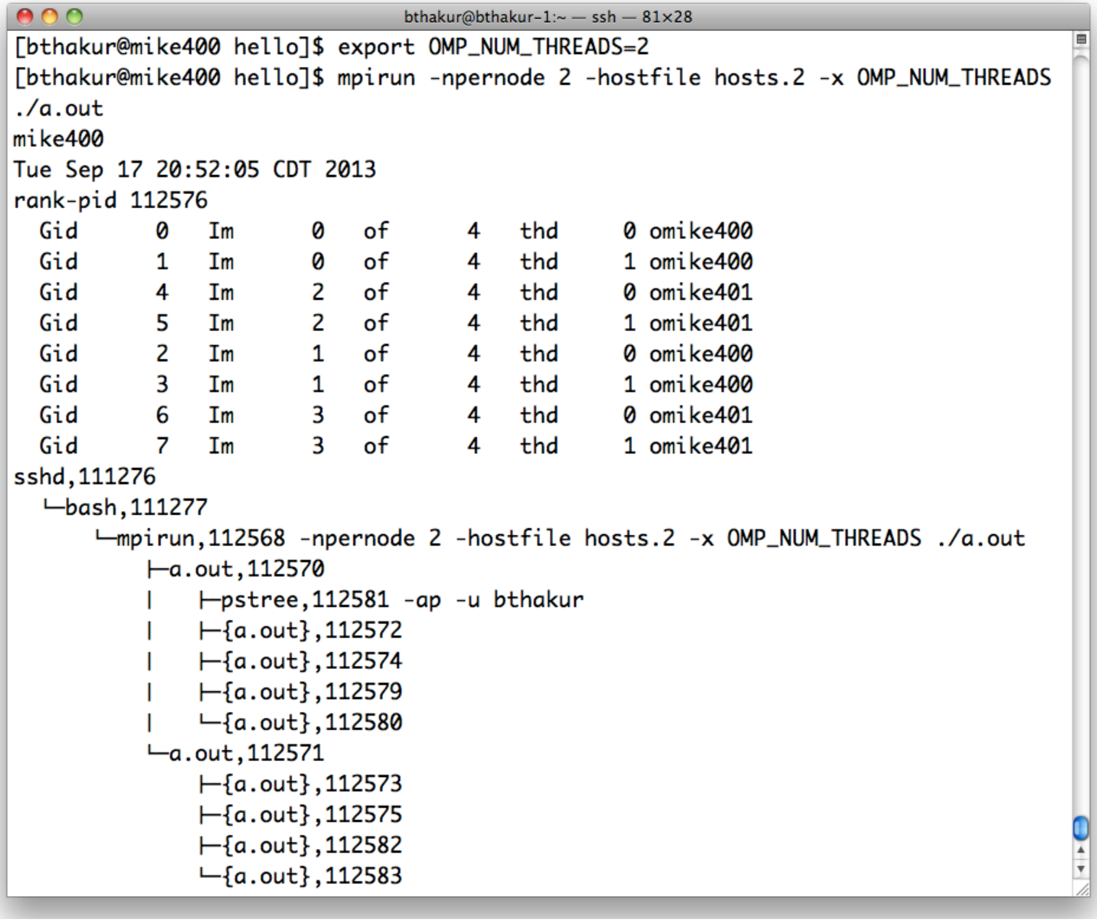
mpif90 –openmp hello_hybrid.f90



```
! Init MPI
   call MPI_Init(mpierr)

! Get Rank Size
   call MPI_COMM_Rank(MPI_COMM_WORLD, nrank, mpierr)
   call MPI_COMM_Size(MPI_COMM_WORLD, nproc, mpierr)

! Print rank
   call MPI_GET_PROCESSOR_NAME(pname, nlen, mpierr)

! Get Date hostname etc
   if (nrank==0) then
      call system('hostname && date && echo rank-pid $$')
   end if
   call MPI_Barrier(MPI_COMM_WORLD, mpierr)

! OpenMP
   !$OMP PARALLEL PRIVATE(itd,gtd)
    itd= omp_get_thread_num()
    gtd= omp_get_num_threads()
    grank= nrank*gtd + itd
    write(*,'(4(a6,i6),a2,a8)')"Gid ", grank, " Im ", nrank, &
                               " of ", nproc, &
                               " thd", itd,    &
                               " on ", pname
   !$OMP FLUSH
   !$OMP BARRIER
    if (nrank==0 .and. itd==0) then
       call system('pstree -ap -u bthakur ')
```

Analyzing a Hybrid parallel program

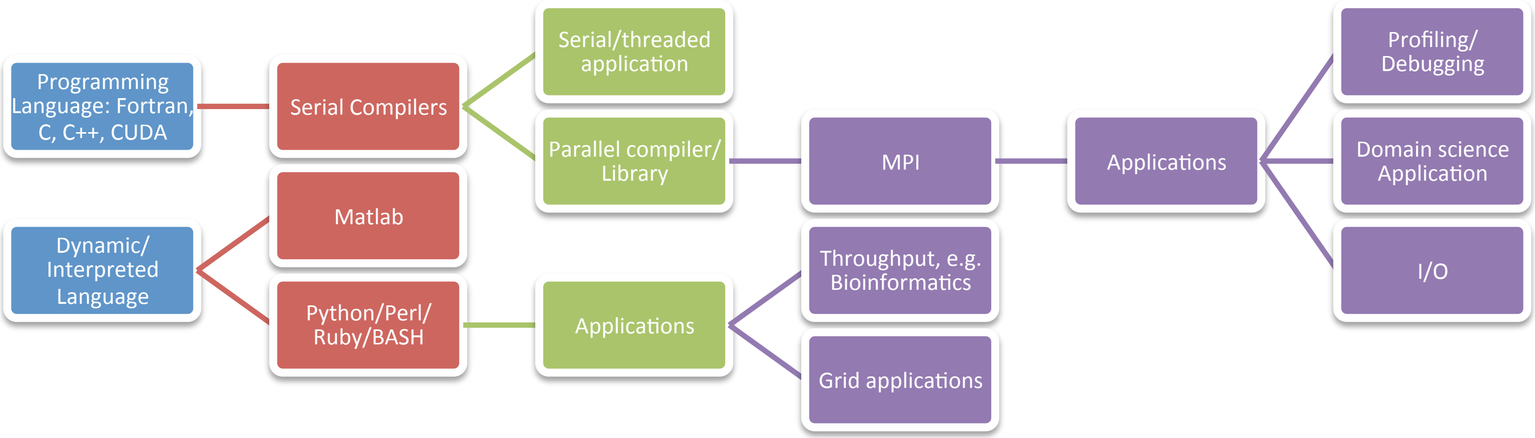
Running a hybrid (mpi +openmp) process



```
[bthakur@mike400 hello]$ export OMP_NUM_THREADS=2
[bthakur@mike400 hello]$ mpirun -npernode 2 -hostfile hosts.2 -x OMP_NUM_THREADS
./a.out
mike400
Tue Sep 17 20:52:05 CDT 2013
rank-pid 112576
  Gid      0   Im      0   of      4   thd     0 omike400
  Gid      1   Im      0   of      4   thd     1 omike400
  Gid      4   Im      2   of      4   thd     0 omike401
  Gid      5   Im      2   of      4   thd     1 omike401
  Gid      2   Im      1   of      4   thd     0 omike400
  Gid      3   Im      1   of      4   thd     1 omike400
  Gid      6   Im      3   of      4   thd     0 omike401
  Gid      7   Im      3   of      4   thd     1 omike401
sshd,111276
  └─bash,111277
      └─mpirun,112568 -npernode 2 -hostfile hosts.2 -x OMP_NUM_THREADS ./a.out
          ├─a.out,112570
          │   ├─pstree,112581 -ap -u bthakur
          │   ├─{a.out},112572
          │   ├─{a.out},112574
          │   ├─{a.out},112579
          │   └─{a.out},112580
          └─a.out,112571
              ├─{a.out},112573
              ├─{a.out},112575
              ├─{a.out},112582
              └─{a.out},112583
```

Application Software

Broadly we can classify them as


Programming Language: Fortran, C, C++, CUDA
Serial Compilers
Serial/threaded application
Parallel compiler/ Library
MPI
Applications
Profiling/ Debugging
Domain science Application
I/O
Dynamic/ Interpreted Language
Matlab
Python/Perl/ Ruby/BASH
Applications
Throughput, e.g. Bioinformatics
Grid applications

Application Software

- ***List of software***

  http://www.hpc.lsu.edu/resources/software/index.php
  /usr/local/packages and /usr/local/compilers

  Run softenv

- ***Installed Software***

Numerical , I/O libraries:	Lapack, FFTW, HDF5, NetCDF, PETSc
Molecular Dynamics:	Amber, Gromacs, NAMD, LAMMPS…
Programming Tools:	Totalview, DDT, TAU
Licensed	Matlab, Fluent

- ***User requested packages***

  Usually installed in user space, unless request by a group of users, in which case it will be installed under /usr/local/packages

Exercises

1. Serial:
   Compare the speed of serial C code with Intel, GCC and PGI compiler.
   Can you tune the compile options to produce best timing?

2. OpenMP:
   Modify the serial C code to be OMP threaded.
   Find compile time option for creating threaded cide with PGI compiler(pgcc)
   Compare performance vs Intel and GCC compilers

3. MPI:

Job management

- Job management basics
  - Find appropriate queue
  - Understand the queuing system and your requirements and proceed to submit jobs
  - Monitor jobs

Job Queues

- Nodes are organized into queues. Nodes can be shared.
  Each job queue differs in
  - Number of available nodes
  - Max run time
  - Max running jobs per user
  - Nodes may have special characteristics: GPU's, Large memory etc

- Jobs need to specify resource requirements
  - Nodes, time, queue

- Its called a queue for a reason, but jobs don't run on a 'First come first served' policy.

Queue Characteristics – LONI clusters

Machine	Queue	Max Runtime	# of nodes	Max running jobs per user	Max nodes per job	Use
Queen Bee	workq	3 days	530	8	128	Unpreemptable
	checkpt		668		256	Preemptable
Others	workq	3 days	128	8	40	Unpreemptable
	checkpt		96		64	Preemptable
	single	14 days	16	64	1	Single processor

Queue Characteristics – LSU Linux clusters

Machine	Queue	Max Runtime	# of nodes	Max running jobs per user	Max nodes per job	Use
SuperMikeII	workq	3 days	128	48	128	Unpreemptable
	checkpt		96		200	Preemptable
	bigmem	2 days	8		2	Big memory
	gpu	1 day	50		32	Job using GPU
Tezpur	workq	3 days	180	8	90	Unpreemptable
	checkpt		344		180	Preemptable
	single	14 days	16	64	1	Single processor
Philip	workq	3 days	28	12	5	Unpreemptable
	checkpt		28			Preemptable
	gpu		2			Job using GPU
	bigmem		5			Big memory
	single	14 days	24		1	Single processor

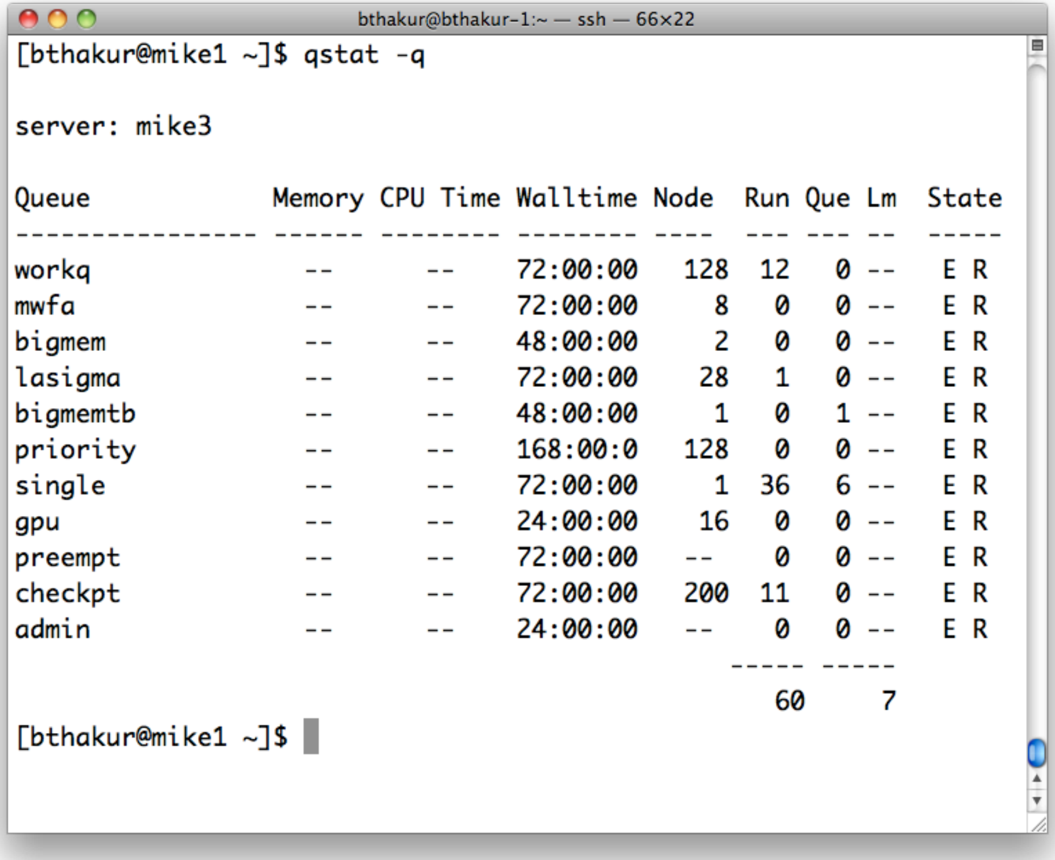
Queue Characteristics – LSU AIX Clusters

Machine	Queue	Max Runtime	# of cores	Max running jobs per user	Max cores per job	Use
Pandora	Interactive	30 minutes	8	6	8	Unpreemptable
	Workq	3 days	224		128	Preemptable
	Single	7 days	64		32	Single processor

Queue Characteristics

"qstat –q" will give you more info on the queues

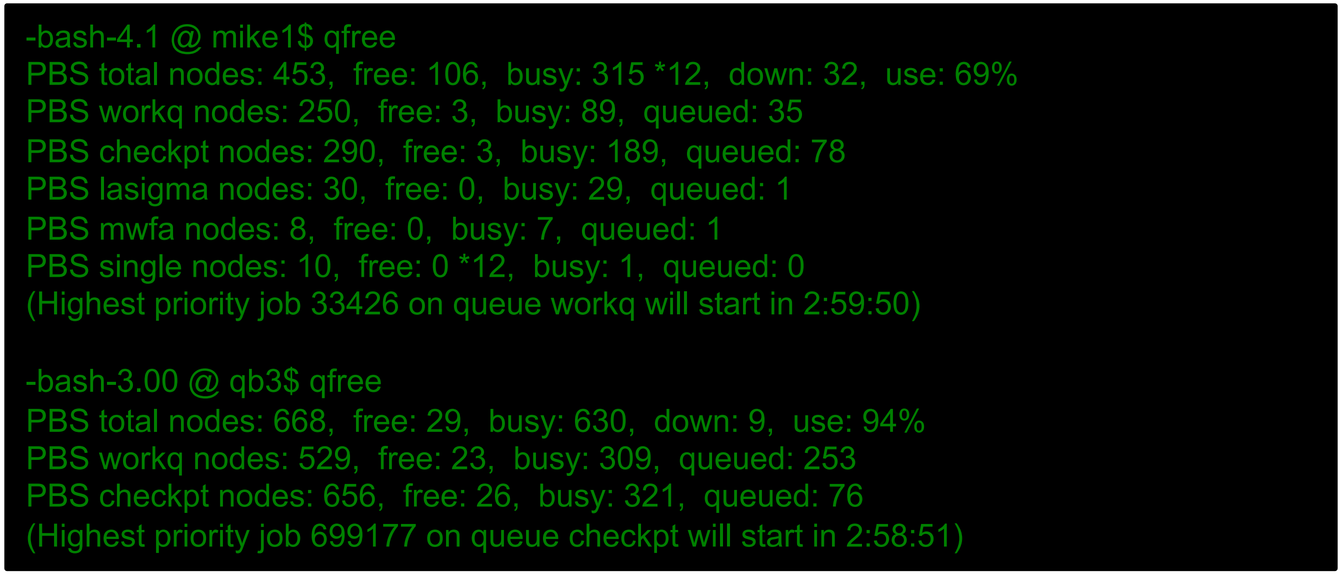
For a more detailed desctiption use mdiag



```
[bthakur@mike1 ~]$ qstat -q

server: mike3

Queue            Memory CPU Time Walltime Node  Run Que Lm  State
---------------- ------ -------- -------- ----  --- --- --  -----
workq              --      --    72:00:00   128  12   0 --   E R
mwfa               --      --    72:00:00     8   0   0 --   E R
bigmem             --      --    48:00:00     2   0   0 --   E R
lasigma            --      --    72:00:00    28   1   0 --   E R
bigmemtb           --      --    48:00:00     1   0   1 --   E R
priority           --      --    168:00:0   128   0   0 --   E R
single             --      --    72:00:00     1  36   6 --   E R
gpu                --      --    24:00:00    16   0   0 --   E R
preempt            --      --    72:00:00   --    0   0 --   E R
checkpt            --      --    72:00:00   200  11   0 --   E R
admin              --      --    24:00:00   --    0   0 --   E R
                                               ----- -----
                                                  60     7

[bthakur@mike1 ~]$ 
```

Queue Querying – Linux Clusters

- Command: qfree
  - Show the number of free, busy and queued nodes
- Command: qfreeloni
  - Equivalent to run qfree on all LONI Linux clusters



```
-bash-4.1 @ mike1$ qfree
PBS total nodes: 453,  free: 106,  busy: 315 *12,  down: 32,  use: 69%
PBS workq nodes: 250,  free: 3,  busy: 89,  queued: 35
PBS checkpt nodes: 290,  free: 3,  busy: 189,  queued: 78
PBS lasigma nodes: 30,  free: 0,  busy: 29,  queued: 1
PBS mwfa nodes: 8,  free: 0,  busy: 7,  queued: 1
PBS single nodes: 10,  free: 0 *12,  busy: 1,  queued: 0
(Highest priority job 33426 on queue workq will start in 2:59:50)

-bash-3.00 @ qb3$ qfree
PBS total nodes: 668,  free: 29,  busy: 630,  down: 9,  use: 94%
PBS workq nodes: 529,  free: 23,  busy: 309,  queued: 253
PBS checkpt nodes: 656,  free: 26,  busy: 321,  queued: 76
(Highest priority job 699177 on queue checkpt will start in 2:58:51)
```

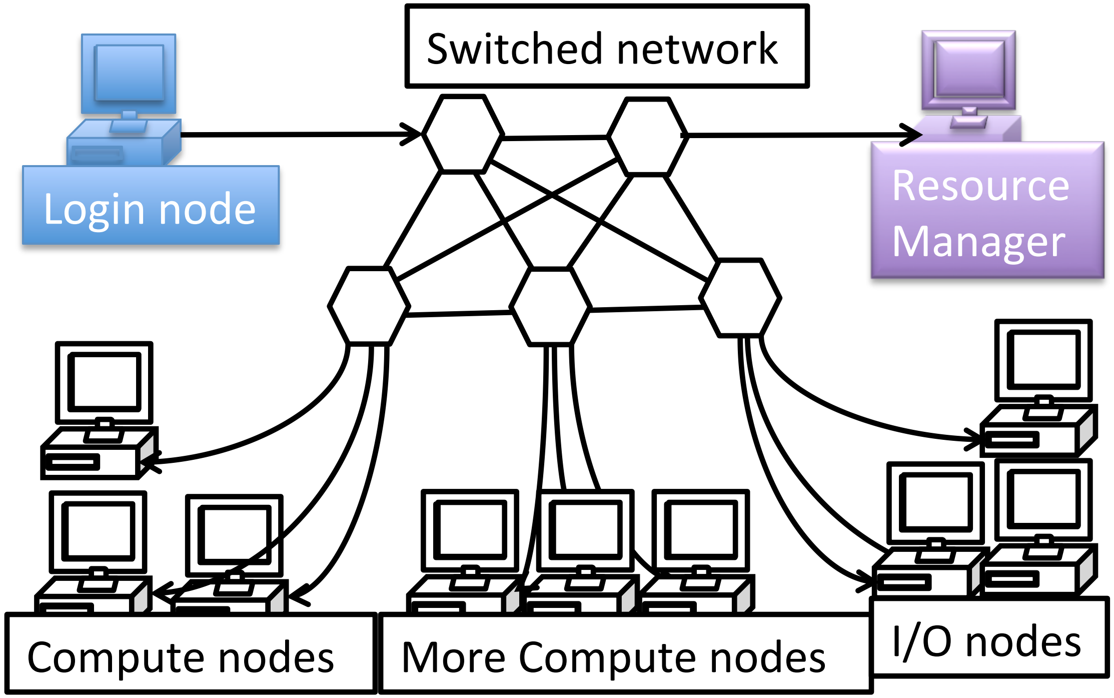
Back to Cluster Architecture

Resource managers give access to compute resource

- Takes in a request on login node
- Finds appropriate resource and assigns you a priority number
- Positions your job in a queue based on the priority assigned.
- Starts running jobs until it cannot run more jobs with what is available.

Note

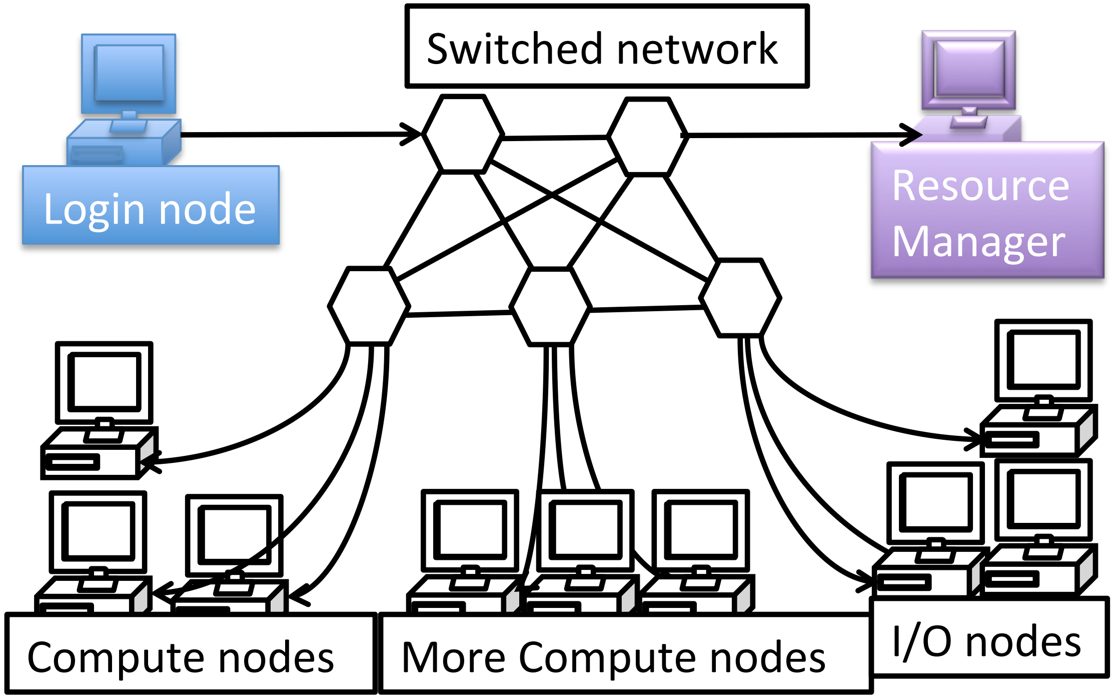
- Newer jobs coming in can have a higher priority as It follows a complex calculation for priority number



Resource manager philosophy

Working Philosophy

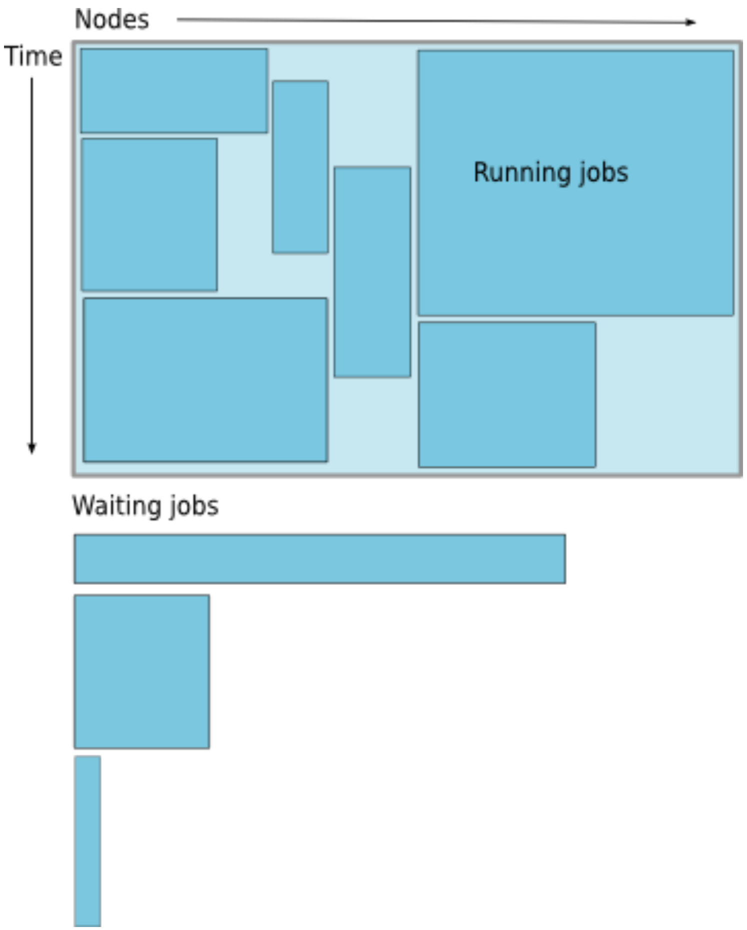
- Prioritize workload into a queue for jobs
- Backfill idle nodes to maximize utilization



Job priorities

Job priorities have contributions from the following

- Resource requirements.
- Time spent in queue
- User Credentials
- Fair-share



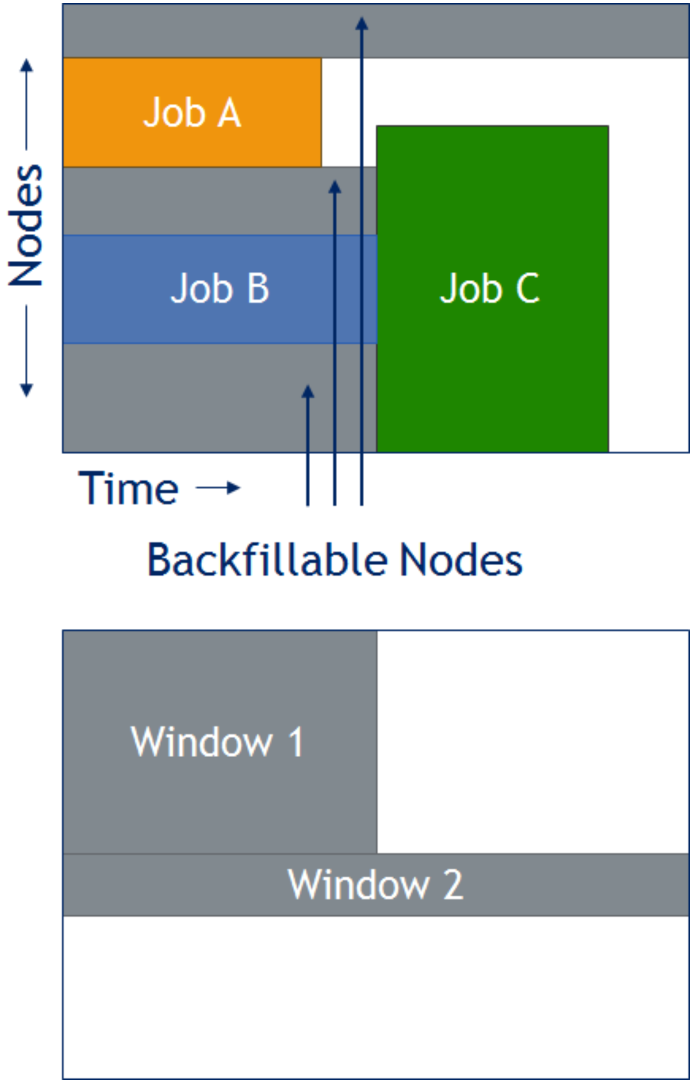
"qstat –a" to see what's running/queued
Don't run it too often as it an intensive query
"qstat –u $USER" to see your jobs

Backfilling

Backfilling aims to utilize idle nodes by running jobs out of order. Enabling backfill allows the scheduler to start other, lower-priority jobs so long as they do not delay the highest priority job.

If the *FIRSTFIT* algorithm is applied, the following steps are taken:

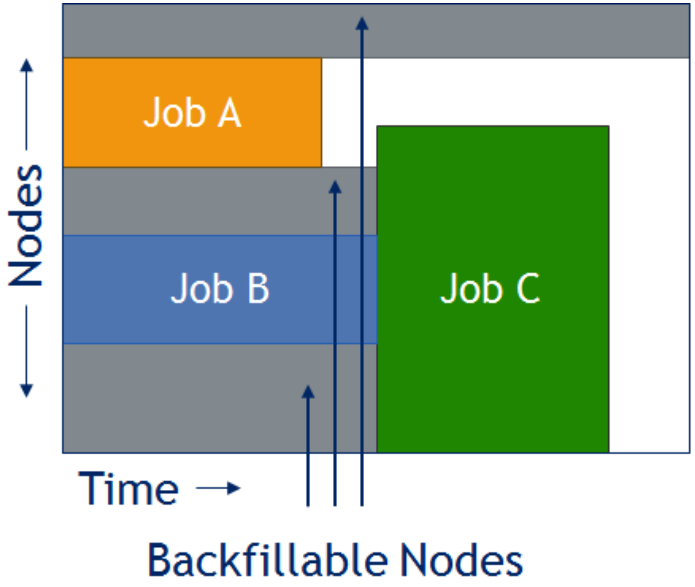
- The list of feasible backfill jobs is filtered, selecting only those that will actually fit in the current backfill window.
- The first job is started.
- While backfill jobs and idle resources remain, repeat step 1

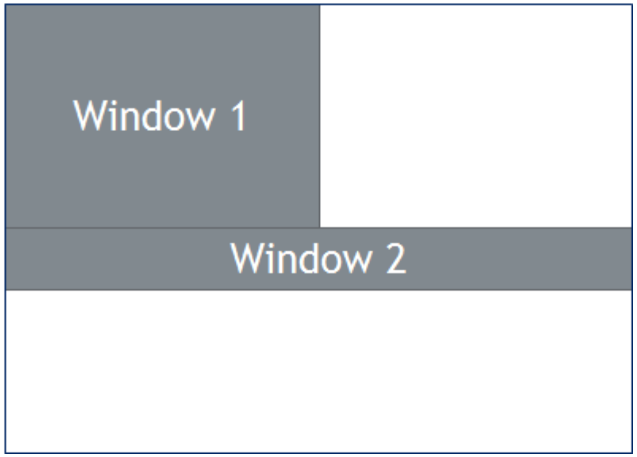


Backfilling

Backfilling aims to utilize idle nodes by running jobs out of order. Enabling backfill allows the scheduler to start other, lower-priority jobs so long as they do not delay the highest priority job.

Although the highest priority job is protected, there is nothing to prevent the third priority job from starting early and possibly delaying the start of the second priority job.





showbf will show you the current backfill windows

Job Types

- Interactive job
  - Set up an interactive environment on compute nodes for users
    - Advantage: can run programs interactively
    - Disadvantage: must be present when the job starts
  - Purpose: testing and debugging
    - Do not run on the head node !!!
    - Try not to run interactive jobs with large core count, which is a waste of resources)
- Batch job
  - Executed without user intervention using a job script
    - Advantage: the system takes care of everything
    - Disadvantage: can only execute one sequence of commands which cannot changed after submission
  - Purpose: production run

Submitting Jobs – Linux Clusters

- Interactive job

```
qsub    -I -V \
        -l walltime=<hh:mm:ss>,nodes=<num nodes>:ppn=<num cores> \
        -A <Allocation> \
        -q <queue name>
```

- Batch job

  qsub job_script

- Add -X to enable X11 forwarding

PBS Job Script – Parallel Jobs

```
#!/bin/bash
#PBS -l nodes=4:ppn=4                  Number of nodes and processors per node
#PBS -l walltime=24:00:00              Maximum wall time
#PBS -N myjob                          Job name
#PBS -o <file name>                    File name for standard output
#PBS -e <file name>                    File name for standard error
#PBS -q checkpt                        Queue name
#PBS -A <allocation_if_needed>         Allocation name
#PBS -m e                              Send mail when job ends
#PBS -M <email address>                Send mail to this address

<shell commands>
mpirun  -machinefile $PBS_NODEFILE -np 16 <path_to_executable> <options>
<shell commands>
```

PBS Job Script – Serial Jobs

```
#!/bin/bash
#PBS -l nodes=1:ppn=1              Number of nodes and processor
#PBS -l walltime=24:00:00          Maximum wall time
#PBS -N myjob                      Job name
#PBS -o <file name>                File name for standard output
#PBS -e <file name>                File name for standard error
#PBS -q single                     The only queue that accepts serial jobs
#PBS -A <loni_allocation>          Allocation name
#PBS -m e                          Send mail when job ends
#PBS -M <email address>            Send mail to this address

<shell commands>
<path_to_executable> <options>
<shell commands>
```

Job Monitoring – Linux Clusters

- Check details on your job using qstat

  ```
  $ qstat –f jobid           : For details on your job
  $ qstat –n –u $USER        : For quick look at nodes assigned to you
  $ qdel jobid               : To delete job
  ```

- Check approximate start time using showstart

  ```
   $ showstart jobid
  ```

- Check details of your job using checkjob

  ```
  $ checkjob jobid
  ```

- Check health of your job using qshow

  ```
  $ qshow –j jobid
  ```

Pay close attention to the load and the memory consumed by your job.

Queue Querying – AIX Clusters

- Command: llclass

```
lyan1@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-proceesor parallel jobs
workq              unlimited      unlimited    51    56  Default queue (5 days), up to 56 processors
priority           unlimited      unlimited    40    40  priority queue resevered for on-demand jobs (5 days), up to 48 processors
preempt             unlimited      unlimited    40    40  preemption queue resevered 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
--------------------------------------------------------------------------------
```

LoadLeveler Job Script - Parallel

```
#!/bin/sh
#@ job_type = parallel                                  Job type
#@ output = /work/default/username/$(jobid).out         Standard output
#@ error = /work/default/username/$(jobid).err          Standard error
#@ notify_user = youremail@domain                       Notification
#@ notification = error                                 Notify on error
#@ class = checkpt                                      Queue
#@ wall_clock_limit = 24:00:00                          Wall clock time
#@ node_usage = shared                                  node usage
#@ node = 2                                             # of nodes
#@ total_tasks = 16                                     # of processors
#@ requirements = (Arch == “POWER7”                     Job requirement
#@ environment = COPY_ALL                               Environment
#@ queue
<shell commands>
poe  <path_to_executable>  <options>
<shell commands>
```

Loadleveler Job Script - Serial

```
#!/bin/sh
#@ job_type = serial                                    Job type
#@ output = /work/default/username/$(jobid).out         Standard output
#@ error = /work/default/username/$(jobid).err          Standard error
#@ notify_user = youremail@domain                       Notification
#@ notification = error                                 Notify on error
#@ class = single                                       Queue
#@ wall_clock_limit = 24:00:00                          Wall clock time
#@ requirements = (Arch == “POWER5”)                    Job requirement
#@ environment = COPY_ALL
         Environment
#@ queue

<shell commands>
poe  <path_to_executable>  <options>
<shell commands>
```

Submitting Jobs – AIX clusters

- Submit jobs using llsubmit
  - llsubmit jobscript : submit job
  - llcancel jobid : delete job

- Check job status using llq and cluster status using llstatus

Job Monitoring – AIX Clusters

- Command: `showllstatus.py`
  - Show job status and nodes running on
- Command: `llq <options> <job_id>`
  - All jobs are displayed if `<job_id>` is omitted
  - Display detailed information: `llq -l <job_id>`
  - Check the estimated start time: `llq -s <job_id>`
  - Show jobs from a specific user: `llq -u <username>`

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

Exercise

Submit a small job to run “sleep 180”and “print PBS variables”

- Create a script to submit a 5 min job and print from within the job script PBS variables $PBS_NODEFILE, $PBS_WORKDIR. Also run “sleep 180” to give you a few minutes to verify status.
- Once the job is running, find out the Mother Superior node and other slave nodes assigned to your job using qstat.
- Log into MS node and verify that your job is running and find your temporary output file
- Modify your script to print hello from each of your assigned nodes

Run it within an interactive job session

- Verify using hostname that you are not on the head-node
- Check available PBS variables and print them

Run a shell script using mpirun to print process id of shell

Exercise

Run hello_hybrid.f90 as a batch job

- On SM-II run on 2 nodes with 2 mpi-processes per node and 8 threads per mpi process.
- On QB run 4 threads per mpi process

Future Trainings

- Weekly trainings during regular semester
  - Wednesdays “10am-12pm + afternoon” sessions, Frey 307
- Programming/Parallel Programming workshops
  - **Usually in summer**

Keep an eye on our webpage: www.hpc.lsu.edu