

HPC User Environment 1

Feng Chen
HPC User Services
LSU HPC LONI
sys-help@loni.org

Louisiana State University
Baton Rouge
June 24, 2015

Outline

- **Things to be covered in the training**
 - Available HPC resources
 - LONI & LSU HPC
 - Account and allocations
 - HPC software environment
 - How to access HPC & LONI clusters
 - The software management tool softenv and modules
 - Job management basics

What is HPC

- **High Performance Computing (HPC) is computation at the cutting edge of modern technology, often done on a supercomputer**
- **A supercomputer is in the class of machines that rank among the fastest in the world**
 - Rule of thumb: a supercomputer could be defined to be at least 100 times as powerful as a PC
- **How do you evaluate the performance of HPC?**

Measure HPC performance-FLOPS

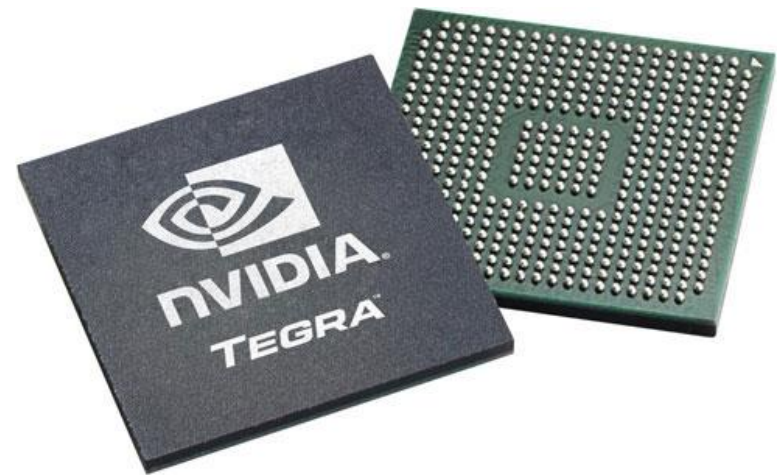
- Performance is measured in ***FL*loating *P*oint *O*perations *P*er *S*econd** (FLOPS or flop/s)
- $$FLOPS = cores \times clock \times \frac{FLOPs}{cycle}$$
 - Most microprocessors today can do 4 FLOPs per clock cycle. Therefore a 2.5-GHz processor has a theoretical performance of 10 billion FLOPs = 10 GFLOPs
 - Dual Core? Quad Core?

Computer performance

Name	FLOPS
yottaFLOPS	10^{24}
zettaFLOPS	10^{21}
exaFLOPS	10^{18}
petaFLOPS	10^{15}
teraFLOPS	10^{12}
gigaFLOPS	10^9
megaFLOPS	10^6
kiloFLOPS	10^3

Supercomputing on a cell phone?

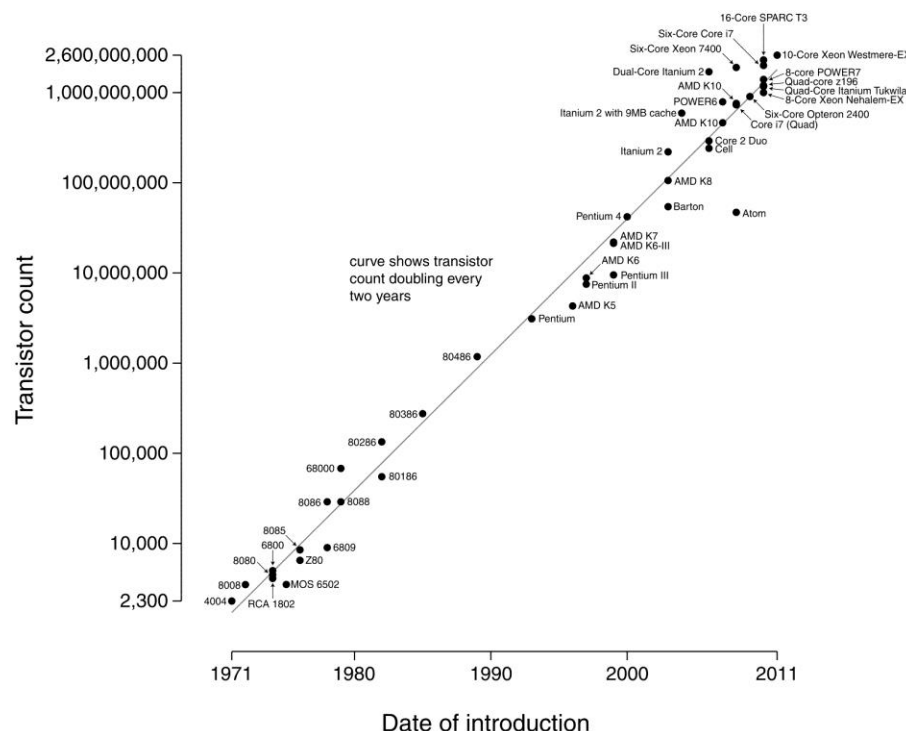
- **Quad-core processors are coming to your phone**
 - Nvidia, TI, Qualcomm...
 - Processing power in the neighborhood of 10 GigaFLOPS
 - Would make the top 500 list 15 years ago
 - What is your phone's FLOPS?
 - iPhone 5 1.3 GHz dual-core Cyclone
 - Compare to ENIAC (500 FLOPS)
 - Compare to top 500 in 1993 #1 (59.7 GFLOPS), #500 (0.42 GFLOPS)



What is driving the Change?

- **What is driving this continuous change?**
 - Moore's Law
 - What was a supercomputer 15 years ago now sits on your desk, or even in your hand.
 - Who measures/ranks the supercomputing performance?

Microprocessor Transistor Counts 1971-2011 & Moore's Law



Source:
http://en.wikipedia.org/wiki/Moore's_law

The Top 500 List

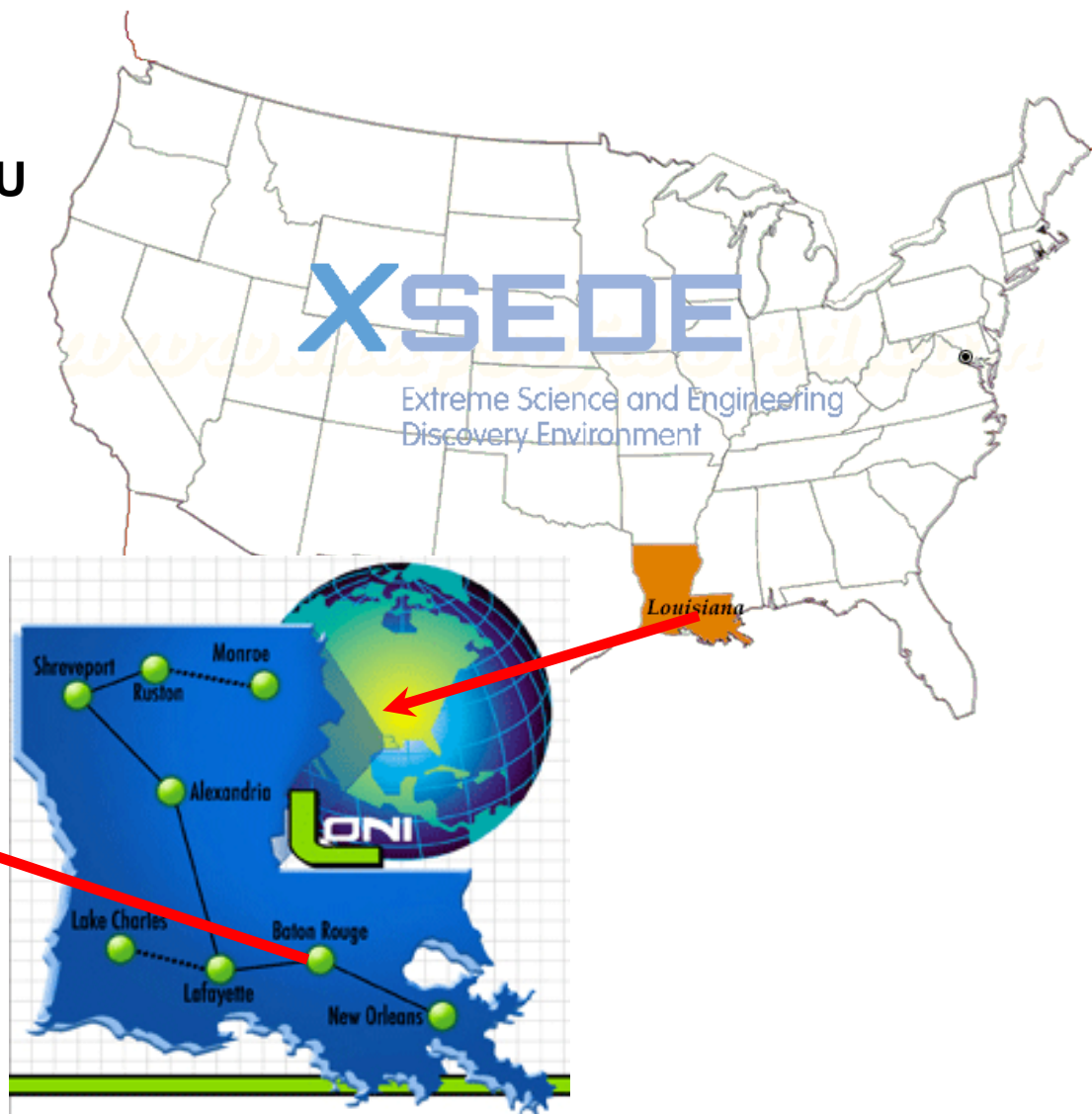


- The TOP500 project provides a list of 500 fastest super computers in the world ranked by their LINPACK performance.
- Semi-annually published (in the public domain)
- As of Nov 2014, China's Tianhe-2 (Milky Way-2) supercomputer is the fastest in the world.
 - Nodes: 16,000
 - Cores: 3,120,000
 - Peak Performance: 33.86 PFlop/s
- As of Nov 2014
 - LSU SuperMIC ranks **88**
 - LONI QB2 ranks **46**



Available Computing Resources

- State wide-LONI
- University wide-HPC@LSU
- Nation wide-XSEDE



University Level: HPC@LSU

- **University Level: LSU HPC resources available to LSU Faculty and their affiliates.**
- **LONI and LSU HPC administered and supported by HPC@LSU**



Available LSU HPC resources

SuperMIC		SuperMike II		Philip		Pandora	
Hostname	smic.hpc.lsu.edu	Hostname	mike.hpc.lsu.edu	Hostname	philip.hpc.lsu.edu	Hostname	pandora.hpc.lsu.edu
Peak Performance/TFlops	1000	Peak Performance/TFlops	146	Peak Performance/TFlops	3.469	Peak Performance/TFlops	6.8
Compute nodes	360	Compute nodes	440	Compute nodes	37	Compute nodes	8
Processor/node	2 Deca-core	Processor/node	2 Octa-core	Processor/node	2 Quad-Core	Processor/node	32 (4 threads each)
Processor Speed	2.8GHz	Processor Speed	2.6GHz	Processor Speed	2.93GHz	Processor Speed	3.3GHz
Processor Type	Intel Xeon 64bit	Processor Type	Intel Xeon 64bit	Processor Type	Intel Xeon 64bit	Processor Type	IBM POWER7
Nodes with Accelerators	360	Nodes with Accelerators	50	Nodes with Accelerators	2	Nodes with Accelerators	0
Accelerator Type	Xeon Phi 7120P	Accelerator Type	2 nVidia M2090	Accelerator Type	3 nVidia M2070	Accelerator Type	
OS	RHEL v6	OS	RHEL v6	OS	RHEL v5	OS	AIX v7.1
Vendor		Vendor	Dell	Vendor	Dell	Vendor	IBM
Memory per node	64 GB	Memory per node	32/64/256 GB	Memory per node	24/48/96 GB	Memory per node	128 GB
Detailed Cluster Description		Detailed Cluster Description		Detailed Cluster Description		Detailed Cluster Description	
User Guide		User Guide		User Guide		User Guide	
Available Software		Available Software		Available Software		Available Software	

Ref: <http://www.hpc.lsu.edu/resources/hpc/index.php#lsuhpc>

State Level - Louisiana Optical Network Initiative (LONI)

- A state-of-the-art fiber optic network that runs throughout Louisiana and connects Louisiana and Mississippi research universities.
- \$40M Optical Network, 10Gb Ethernet over fiber optics.
- \$10M Supercomputers installed at 6 sites.



LONI-Louisiana Optical Network Initiative

➤ **LONI connects supercomputers at Louisiana's universities:**

- Louisiana State University
- Louisiana Tech University
- LSU Health Sciences Center in New Orleans
- LSU Health Sciences Center in Shreveport
- Southern University
- Tulane University
- University of Louisiana at Lafayette
- University of New Orleans

Available LONI resources

QB2	
Hostname	qb2.loni.org
Peak Performance/TFlops	1,500
Compute nodes	504
Processor/node	2 10-Core
Processor Speed	2.8GHz
Processor Type	Intel Ivy Bridge-EP Xeon 64bit
Nodes with Accelerators	480
Accelerator Type	NVIDIA Tesla K20x
OS	RHEL v6
Vendor	Dell
Memory per node	64 GB
Location	Information Systems Building, Baton Rouge
Detailed Cluster Description	
User Guide	
Available Software	

Eric	
Hostname	eric.loni.org
Peak Performance/TFlops	9.544
Compute nodes	128
Processor/node	2 4-Core
Processor Speed	2.33GHz
Processor Type	Intel Xeon 64bit
Nodes with Accelerators	0
Accelerator Type	
OS	RHEL v4
Vendor	Dell
Memory per node	8 GB
Location	Louisiana State University, Baton Rouge
Detailed Cluster Description	
User Guide	
Available Software	

Ref: <http://www.hpc.lsu.edu/resources/hpc/index.php#loni>

Summary of clusters for LSU and LONI

	Name	Performance (TFLOPS)	Location	Vendor	Architecture
LONI	Eric	9.5	LSU	Dell	Linux x86_64
	QB2	1500	ISB	Dell	Linux x86_64
LSU	Philip	3.5	LSU	Dell	Linux x86_64
	SuperMIC	1000	LSU	Dell	Linux x86_64
	SuperMike	212 (CPU+GPU)	LSU	Dell	Linux x86_64
	Pandora	6.8	LSU	IBM	Power7

National Level

- **National Level: Extreme Science and Engineering Discovery Environment (xSEDE)**
 - 5 year, \$121M project supported by NSF
 - Supports 16 supercomputers and high-end visualization and data analysis resources across the country.
 - <https://www.xsede.org/>
 - LSU SuperMIC is one XSEDE cluster.

XSEDE

Extreme Science and Engineering
Discovery Environment

Account Eligibility-*LONI*

- All faculty and research staff at a [LONI Member Institution](#), as well as students pursuing sponsored research activities at these facilities, are eligible for a LONI account.
- Requests for accounts by research associates not affiliated with a LONI Member Institution will be handled on a case by case basis.
- For prospective LONI Users from a non-LONI Member Institution, you are required to have a faculty or research staff in one of LONI Member Institutions as your Collaborator to sponsor you a LONI account.

Account Eligibility-*LSU HPC*

- All faculty and research staff at Louisiana State University, as well as students pursuing sponsored research activities at LSU, are eligible for a LSU HPC account.
- For prospective LSU HPC Users from outside LSU, you are required to have a faculty or research staff at LSU as your Collaborator to sponsor you a LSU HPC account.

LONI & LSU HPC Accounts

- **LSU HPC and LONI systems are two distinct computational resources administered by HPC@LSU.**
- **Having an account on one does not grant the user access to the other.**

How do I get a **LONI** Account?

- Visit https://allocations.loni.org/login_request.php
- Enter your **INSTITUTIONAL** Email Address and captcha code.
- Check your email and click on the link provided (link is active for 24hrs only)
- Fill the form provided
- For LONI Contact/Collaborator field enter the name of your research advisor/supervisor who must be a Full Time Faculty member at a LONI member institution.
- Click Submit button
- Your account will be activated once we have verified your credentials.

How do I get a *LSU HPC* Account?

- Visit https://accounts.hpc.lsu.edu/login_request.php
- Enter your **INSTITUTIONAL** Email Address and captcha code.
- Check your email and click on the link provided (link is active for 24hrs only)
- Fill the form provided
- For HPC Contact/Collaborator field enter the name of your research advisor/supervisor who must be a **Full Time** Faculty member at LSU
- Click Submit button
- Your account will be activated once we have verified your credentials

Allocation

- **An allocation is a block of service unit (SUs) that allow a user to run jobs on a cluster**
 - One SU is one cpu-hour
 - Example
 - 40 SUs will be charged for a job that runs 10 hours on 4 cores
- **LONI & HPC users: All LONI clusters, SuperMikell and SuperMIC jobs need to be charged to valid allocation.**

Allocation Types

➤ **Startup: Allocations upto 50K SUs**

- Can be requested at any time during the year.
- Reviewed and Approved by the LONI Resource Allocation Committee.
- Only two active allocations per PI at any time.

➤ **Large: Allocations between 50K - 4M SUs.**

- Reviewed and Approved by the LONI Resource Allocation Committee every Quarter.
- Decision will be made on January 1, April 1, July 1 and October 1 of each year
- Users can have multiple Large Allocations.
- **LSU HPC:** Each request is limited to 3 million SUs, and a PI may have a total of 5 million SUs active at any given time.
- **LONI:** Each requests is limited to 4 million SU, and a PI may have a total of 6M SU active at any given time

Who can request an Allocation?

- **Only Full Time Faculty member at LONI member institutions can act as Principle Investigators (PI) and request Allocations.**
- **Rule of Thumb: If you can sponsor user accounts, you can request allocations.**
- **Everyone else will need to join an existing allocation of a PI, usually your advisor/supervision or course instructor (if your course requires a LONI account).**

How to request/join an Allocation

- **Login to your LONI Profile at <https://allocations.loni.org>**
- **Click on "Request Allocation" in the right sidebar.**
- **Click "New Allocation" to request a New Allocation.**
 - Fill out the form provided.
 - All requests require submission of a proposal justifying the use of the resources.
 - Click "Submit Request" button.
- **Click "Join Allocation" to join an existing Allocation.**
 - Search for PI using his/her email address, full name or LONI username
 - Click "Join Projects" button associated with the PI's information.
 - You will be presented with a list of allocations associated with the PI. Click "Join" for the allocation you wish to join.
 - Your PI will receive an email requesting him to confirm adding you to the allocation.
 - Please do not contact the helpdesk to do this.

Account Management

- **LONI account**
 - <https://allocations.loni.org>
- **LSU HPC account**
 - <https://accounts.hpc.lsu.edu>
- **The default Login shell is bash**
 - Supported Shells: bash, tcsh, ksh, csh, sh
 - Change Login Shell at the profile page

How do I reset my password?

- **LONI:** Visit https://allocations.loni.org/user_reset.php
- **LSU HPC:** Visit https://accounts.hpc.lsu.edu/user_reset.php
- Enter the email address attached to your account and captcha code
- You will receive an email with link to reset your password, link must be used within 24 hours.
- Once you have entered your password, one of the HPC Admins need to approve the password reset.
- The Password approval can take anything from 10 mins to a few hours depending on the schedule of the Admins and also time of day
- You will receive a confirmation email stating that your password reset has been approved.

Password Security

- **Passwords should be changed as soon as your account is activated for added security.**
- **Password must be at least 12 and at most 32 characters long, must contain three of the four classes of characters:**
 - lowercase letters,
 - uppercase letters,
 - digits, and
 - other special characters (punctuation, spaces, etc.).
- **Do not use a word or phrase from a dictionary,**
- **Do not use a word that can be obviously tied to the user which are less likely to be compromised.**
- **Changing the password on a regular basis also helps to maintain security.**
 - <http://www.thegeekstuff.com/2008/06/the-ultimate-guide-for-creating-strong-passwords/>
 - http://en.wikipedia.org/wiki/Password_policy

HPC@LSU User Services

- **Hardware resources**
 - Currently manages 6 clusters
- **Software stack**
 - Communication software
 - Programming support: compilers and libraries
 - Application software
- **Contact user services**
 - Email Help Ticket: sys-help@loni.org
 - Telephone Help Desk: +1 (225) 578-0900
 - Instant Message: AOL, Yahoo!, Gmail, User name: lsuhpchelp

General Cluster Architecture

➤ **Login (head) nodes get you access to the cluster.**

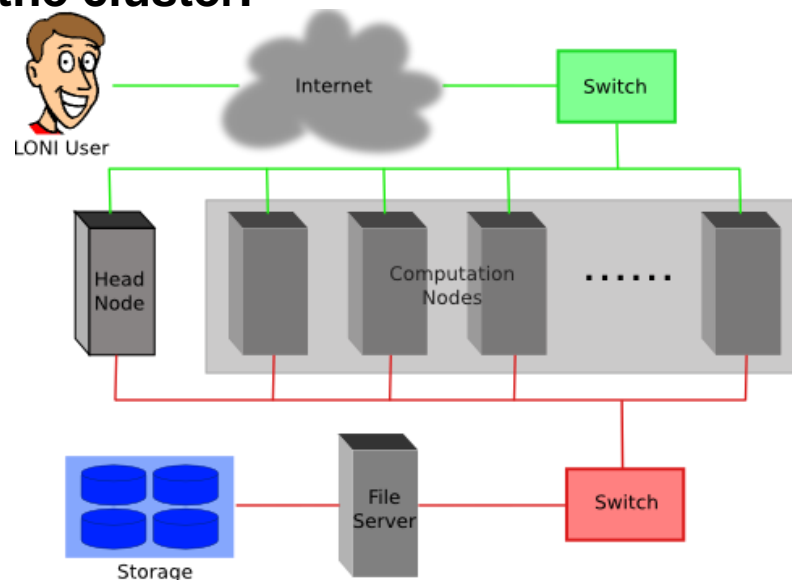
- Individual nodes are not accessible.
- Login via ssh
- Node are not meant to run jobs

➤ **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**

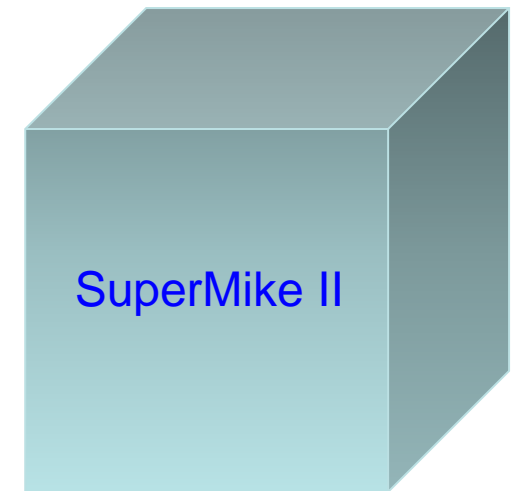
- PBS/loadleveler installed
- Run commands qsub, qstat, qdel



LSU HPC :What should I use?

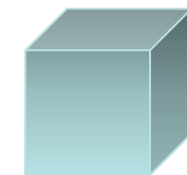
➤ Why would you use SuperMike II?

- 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 Pandora?

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



Pandora

➤ 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 SuperMikeII
 - /project not shared with SuperMikeII.



Philip

LSU HPC :What should I use? *SuperMIC*

➤ 360 Compute Nodes

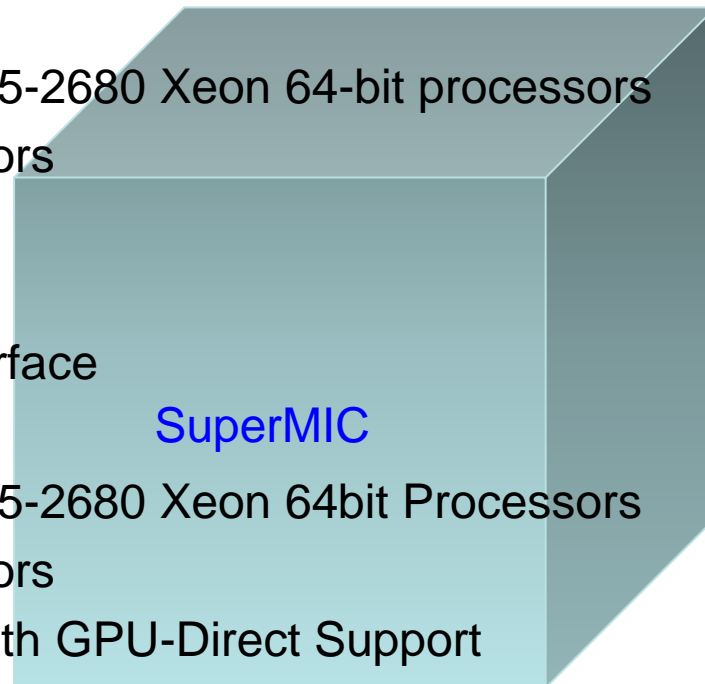
- Two 2.8GHz 10-Core Ivy Bridge-EP E5-2680 Xeon 64-bit processors
- Two Intel Xeon Phi 7120P Coprocessors
- 64GB DDR3 1866MHz Ram
- 500GB HD
- 56 Gigabit/sec Infiniband network interface

➤ 20 Hybrid Compute Nodes

- Two 2.8GHz 10-Core Ivy Bridge-EP E5-2680 Xeon 64bit Processors
- One Intel Xeon Phi 7120P Coprocessors
- One NVIDIA Tesla K20X 6GB GPU with GPU-Direct 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



SuperMIC

LONI:What should I use? **QB2**

➤ **480 Compute Nodes**

- Two 10-core 2.8 GHz E5-2680v2 Xeon processors.
- 64 GB memory
- 2 NVIDIA Tesla K20x GPU's

➤ **16 Compute Nodes**

- Two 10-core 2.8 GHz E5-2680v2 Xeon processors.
- 64 GB memory
- 2 Intel Xeon Phi 7120P's

➤ **4 Visualization Nodes, each with:**

- Two 10-core 2.8 GHz E5-2680v2 Xeon processors.
- Two NVIDIA Tesla K40 GPU's
- 128 GB memory

➤ **4 Big Memory Nodes, each with:**

- Four 12-core 2.6 GHz E7-4860v2 Xeon processors.
- 1.5 TB memory
- Two 1 TB HDD's

Accessing cluster using ssh (Secure Shell)

➤ On Unix and Mac

- use ssh on a terminal to connect

➤ Windows box (ssh client):

- Putty
(<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>)
- MobaXterm (<http://mobaxterm.mobatek.net/>)
- Cygwin

➤ Host name

- LONI: <cluster_name>.loni.org
 - <cluster_name> can be:
 - eric.loni.org
 - qb.loni.org
- LSU HPC: <cluster_name>.hpc.lsu.edu
 - <cluster_name> can be:
 - mike.hpc.lsu.edu
 - smic.hpc.lsu.edu
 - philip.hpc.lsu.edu
 - pandora.hpc.lsu.edu

Accessing cluster on Linux and Mac

```

fchen14@mike.hpc.lsu.edu:~$ ssh fchen14@mike.hpc.lsu.edu
fchen14@mike.hpc.lsu.edu's password:
Last login: Mon Aug 18 11:26:16 2014 from fchen14-4.lsu.edu
#####
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.

1-Feb-2013

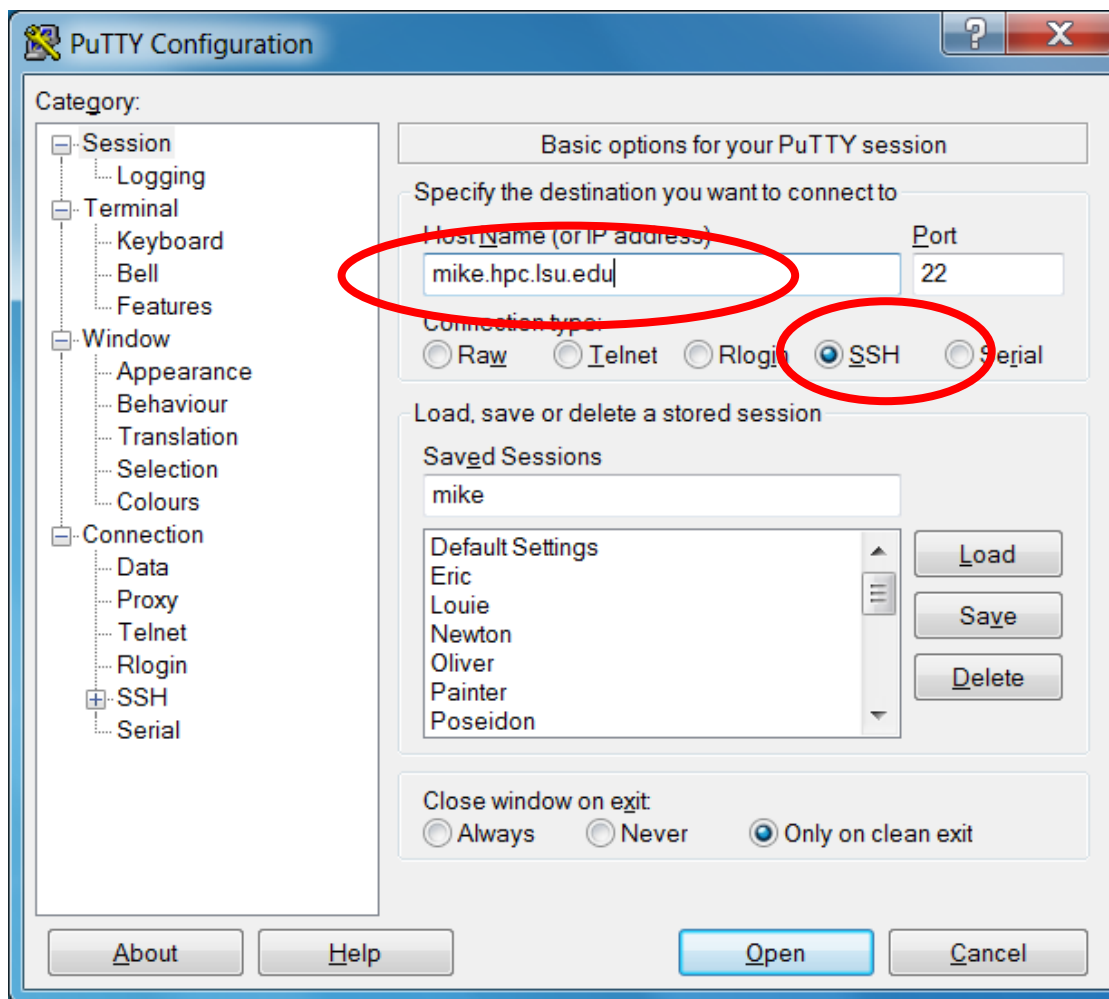
SuperMike-II is open for general use. Please report problems to our email ticke
t
system at sys-help@loni.org so that we can address them.

Quotas for the /home volume are enabled at 5 GB. Please do

```

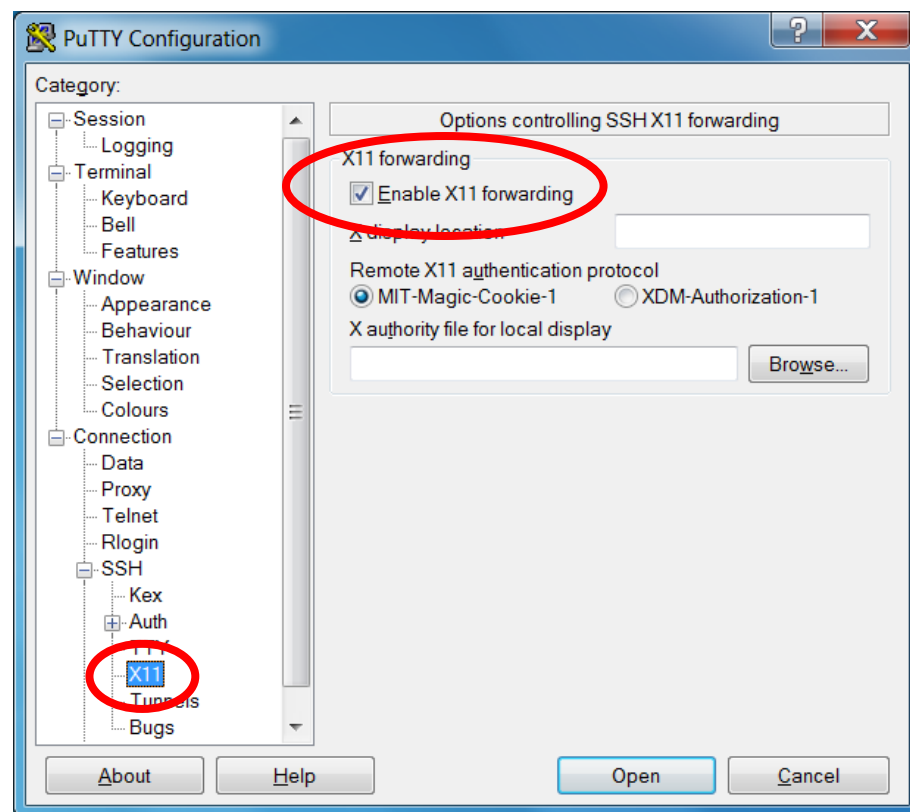
Accessing cluster on Windows - Putty

➤ Example use Putty



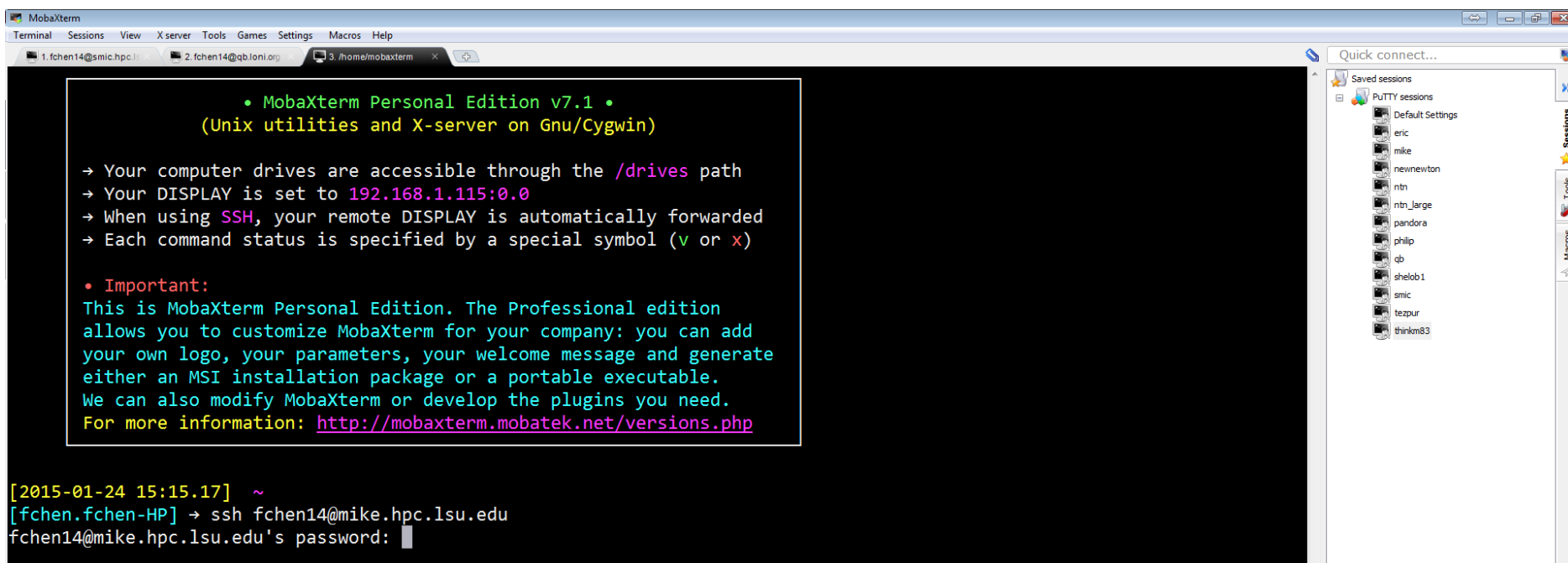
Enable X11 forwarding - Putty

- **On Linux or Mac, simply pass the -X option to the ssh command line**
 - `ssh -X username@mike.hpc.lsu.edu`
- **On windows using putty**
 - Connection->SSH->X11->Enable X11 forwarding
 - Install X server (e.g. Xming)



Accessing cluster on Windows - MobaXterm

- Example using MobaXterm (X11 forwarding is built-in and automatic)



- MobaXterm supports
 - command line scp and rsync
 - sftp file transfer through GUI

File Systems

	Distributed	Throughput	File life time	Best used for
Home	Yes	Low	Unlimited	Code in development, compiled executable
Work/Scratch	Yes	High	30 days	Job input/output
Local scratch	No		Job duration	Temporary files

➤ Tips

- Never let your job write output to your home directory
- Do not write temporary files to /tmp
 - Write to the local scratch or work space via /var/scratch
- The work space is not for long-term storage
 - Files are 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 Philip, SuperMikell and SuperMIC**
- **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 Philip, SuperMikell**
- **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
- **Only PI is eligible for applying storage allocation**

File Transfer (nix/Mac)

➤ From/to a Unix/Linux/Mac machine (including between the clusters)

- scp command
 - Syntax: scp <options> <source> <destination>
- rsync command
 - Syntax: rsync <options> <source> <destination>

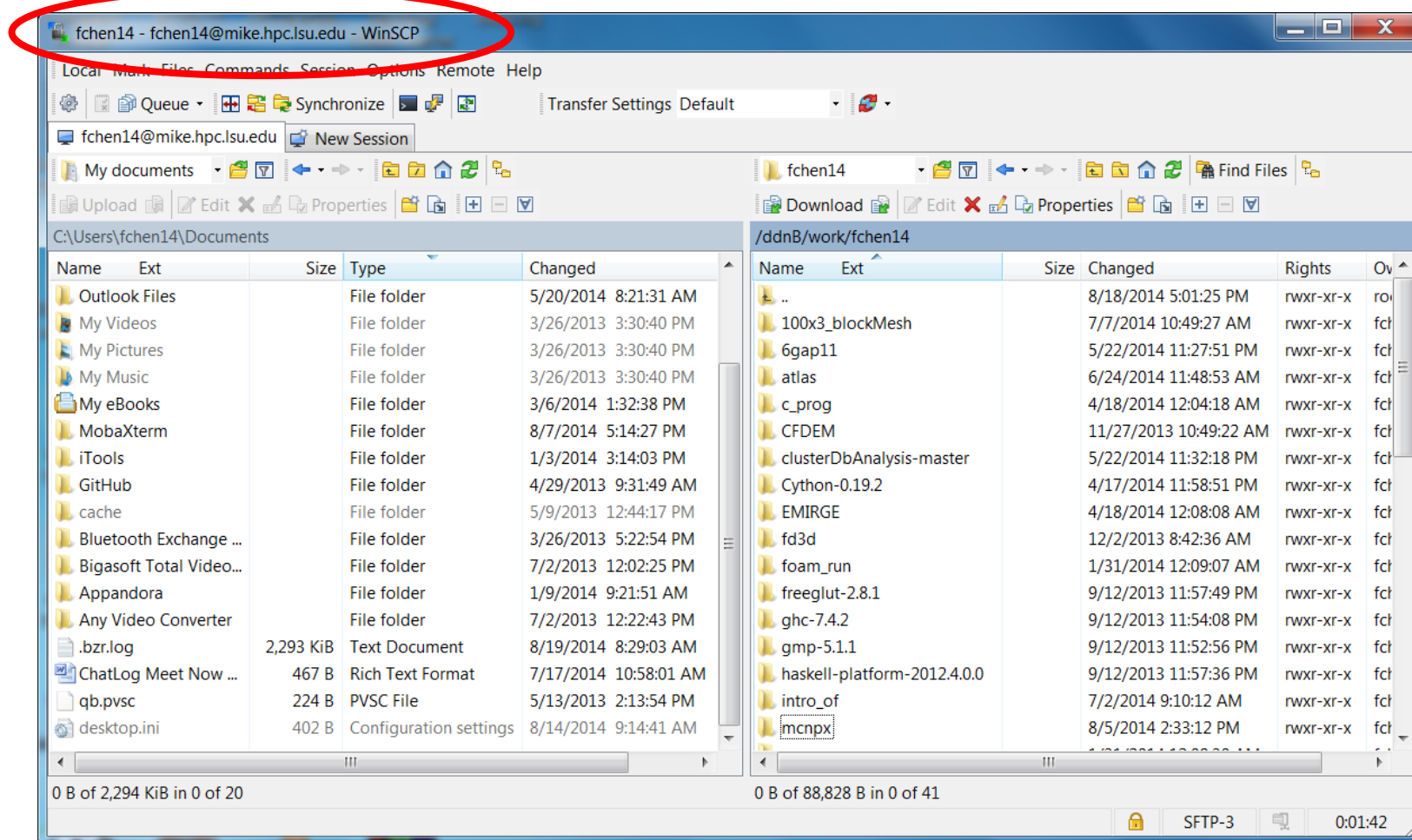
```
[fchen14@mike2 ~]$ scp
usage: scp [-1246BCpqrsv] [-c cipher] [-F ssh_config] [-i identity_file]
        [-l limit] [-o ssh_option] [-P port] [-S program]
        [[user@]host1:]file1 ... [[user@]host2:]file2
```

```
[fchen14@mike2 ~]$ rsync
rsync version 3.0.6 protocol version 30
Usage: rsync [OPTION]... SRC [SRC]... DEST
or rsync [OPTION]... SRC [SRC]... [USER@]HOST:DEST
or rsync [OPTION]... SRC [SRC]... [USER@]HOST::DEST
or rsync [OPTION]... SRC [SRC]... rsync://[USER@]HOST[:PORT]/DEST
...
```

File Transfer (Windows)

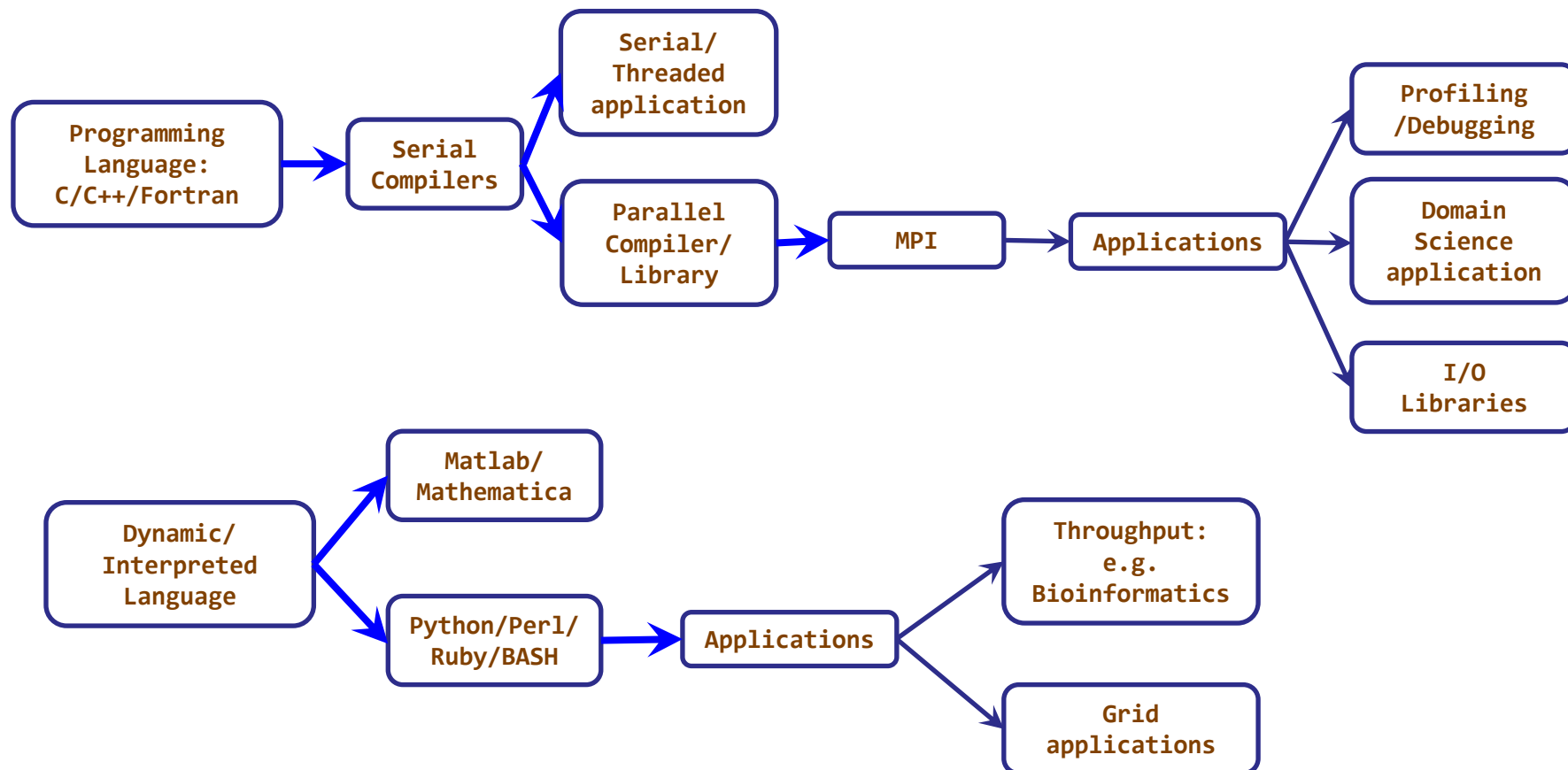
➤ From/to a Windows machine

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



Application Software

➤ We can roughly classify them as:



Application Software

➤ Installed Software

- Mathematical and utility libraries
 - FFTW, HDF5, NetCDF, PETSc...
- Applications
 - Amber, CPMD, NWChem, NAMD, Gromacs, R, LAMMPS...
- Visualization
 - VisIt, VMD, GaussView
- Programming Tools
 - Totalview, DDT, TAU...

➤ List of software

- <http://www.hpc.lsu.edu/resources/software/index.php>

➤ Installed under [/usr/local/packages](#)

➤ 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](#)

Software Environment: Module and 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

- A software that helps users set up environment variables properly to use other software package. Much more convenient than setting variables in `.bashrc`
- Pandora, Philip, SuperMike2 and Eric

➤ Modules

- Another software that helps users set up their environment. Most supercomputing sites (including XSEDE) use modules.
- SuperMIC and QB2

Softenv: Listing All Packages

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

softenv on SuperMike II
example

```
[fchen14@mike2 ~]$ softenv
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:

    @bio-all
  *   @default

These are the keywords explicitly available:

+Intel-12.1.4      @types: Programming/Compiler @name: Intel
                   @version: 12.1.4 @build: Binary
                   installation @internal: @external:
```

softenv key

Softenv: Searching A Specific Package

➤ Use “-k” option with softenv”:

- softenv -k fftw

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
```

➤ Or use grep with softenv

- softenv | grep “fftw”

```
[fchen14@mike2 ~]$ softenv | grep fftw
+fftw-3.3.2-Intel-13.0.0      @types: Library/Math @name: fftw @version:
                             @external: www.fftw.org @about: A fast,
+fftw-3.3.3-Intel-13.0.0      @types: Library/Math @name: fftw @version:
                             @external: www.fftw.org @about: A fast,
+fftw-3.3.3-Intel-13.0.0-openmpi-1.6.2
                             @types: Library/Math @name: fftw @version:
                             @internal: @external: www.fftw.org @about:
[fchen14@mike2 ~]$
```

Softenv: One time change of environment

- **Set up the environment variables to use a certain software package in the current login session only**
 - Add a package: `soft add <key>`
 - Remove a package: `soft delete <key>`

```
[fchen14@mike2 ~]$ which python
/usr/bin/python
[fchen14@mike2 ~]$ soft add +Python-2.7.3-gcc-4.4.6
[fchen14@mike2 ~]$ which python
/usr/local/packages/Python/2.7.3/gcc-4.4.6/bin/python
[fchen14@mike2 ~]$
```


Softenv: Permanent change of environment

- **Set up the environment variables to use a certain software package**
 - First add the key to ~/.soft
 - Then execute resoft at the command line
 - The environment will be the same next time you log in

```
[fchen14@mike2 ~]$ cat ~/.soft
#+Python-2.7.3-gcc-4.4.6
@default
[fchen14@mike2 ~]$ which python
/usr/bin/python
[fchen14@mike2 ~]$ vi ~/.soft
[fchen14@mike2 ~]$ resoft
[fchen14@mike2 ~]$ cat ~/.soft
+Python-2.7.3-gcc-4.4.6
@default
[fchen14@mike2 ~]$ which python
/usr/local/packages/Python/2.7.3/gcc-4.4.6/bin/python
```

Softenv: Querying a software key

- Command “soft-dbq” shows which variables are set by a SOFTENV key

- soft-dbq +Python-2.7.3-gcc-4.4.6

```
[fchen14@mike2 ~]$ soft-dbq +Python-2.7.3-gcc-4.4.6
```

```
This is all the information associated with  
the key or macro +Python-2.7.3-gcc-4.4.6.
```

```
-----  
Name: +Python-2.7.3-gcc-4.4.6  
Description: @types: Programming/Languages @name: Python @version: 2.7.3 @build: gcc-4.4.6 @internal: @external: http://www.python.org @about: Python is a general-purpose, interpreted high-level programming language.  
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_INCLUDE_PATH = ${LD_INCLUDE_PATH}:/usr/local/packages/Python/2.7.3/gcc-4.4.6/include  
LD_LIBRARY_PATH = ${LD_LIBRARY_PATH}:/usr/local/packages/Python/2.7.3/gcc-4.4.6/lib:/usr/local/packages/Python/2.7.3/gcc-4.4.6/lib64  
MANPATH = ${MANPATH}:/usr/local/packages/Python/2.7.3/gcc-4.4.6/man  
PATH = ${PATH}:/usr/local/packages/Python/2.7.3/gcc-4.4.6/bin:/usr/local/packages/Python/2.7.3/gcc-4.4.6/sbin  
PYTHONHOME = /usr/local/packages/Python/2.7.3/gcc-4.4.6
```

Quiz for softenv (1)

- Which mpif90/mpirun will the system use if you just call mpif90/mpirun?
- Which compiler will be used?

```
[fchen14@mike2 ~]$ cat ~/.soft  
+openmpi-1.6.2-gcc-4.7.2  
@default
```

Quiz for softenv (2)

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

```
[fchen14@mike2 ~]$ cat ~/.soft  
+mvapich2-1.9-Intel-13.0.0  
@default  
+openmpi-1.6.2-gcc-4.7.2
```

Quiz for softenv (3)

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

```
[fchen14@mike2 ~]$ cat ~/.soft  
PATH += /usr/local/packages/mpich/3.0.2/Intel-13.0.0/bin  
@default  
+openmpi-1.6.2-gcc-4.7.2
```

Quiz for softenv (4)

- Which version of intel C compiler will be displayed by the commands “**icc --version**”

```
[fchen14@mike2 intel64]$ cat ~/.soft
PATH += /usr/local/compilers/Intel/composer_xe_2013.5.192/bin/intel64
LD_LIBRARY_PATH +=
/usr/local/compilers/Intel/composer_xe_2013.5.192/compiler/lib/intel64
LD_INCLUDE_PATH +=
/usr/local/compilers/Intel/composer_xe_2013.5.192/compiler/include/intel
64:/usr/local/compilers/Intel/composer_xe_2013.5.192/compiler/include
+openmpi-1.6.2-Intel-13.0.0
@default
```

Exercise: Use Softenv

- **Find the key for perl-5.16.3**
 - Check what variables are set through the key
 - Set up your environment to use perl-5.16.3

- **Find the key for Python-2.7.3**
 - Check if the variables are correctly set by “which python”
 - Check if you have access to ipython, scipy, numpy, matplotlib

Using Environment Modules

- **Similar to Softenv, Environment Modules is another framework to manage what software is loaded into a user's environment. Its functionality includes**
 - List all software packages currently available in the Environment Modules system,
 - List all software packages loaded into a user's environment,
 - Load/Unload/Switch software packages into a user's environment
 - Unload a software package from a user's environment.

Modules: List All Available Packages

- **The command to list all available packages is:** `module avail/av`

```
[fchen14@smic1 ~]$ module av
----- /usr/local/packages/Modules/modulefiles/apps -----
abyss/1.5.2/INTEL-140-MVAPICH2-2.0          mpich/3.1.1/INTEL-14.0.2
ansys/15.0                                mummer/3.23/INTEL-14.0.2
beast/1.7.5                                mumps/4.10.0/INTEL-140-MVAPICH2-2.0
blast/2.2.22                               muscle/3.8.31
blast/2.2.28/INTEL-14.0.2                 mvapich2/2.0/INTEL-14.0.2(default)
...
----- /usr/local/packages/Modules/modulefiles/xsede -----
ant/1.9.4          java/1.7.0          tgresid/2.3.4          unicore/6.6.0
globus/5.0.4-r1    pacman/3.29-r3        tgusage/3.0           xsede/1.0(default)
gx-map/0.5.3.3-r1  tginfo/1.1.4          uberftp/2.6
```

- **The format of the listed packages is <package name>/<package version>. For example, gcc/4.4.2 is version 4.4.2 of gcc.**

Modules: List Currently Loaded Packages

- To see what packages are currently loaded into a user's environment, the command is: `module list`

```
[fchen14@smic1 ~]$ module list
```

Currently Loaded Modulefiles:

1) intel/14.0.2	6) tginfo/1.1.4	11) xsede/1.0
2) mvapich2/2.0/INTEL-14.0.2	7) ant/1.9.4	12) ansys/15.0
3) gx-map/0.5.3.3-r1	8) java/1.7.0	13) EasyBuild/1.13.0
4) tgusage/3.0	9) uberftp/2.6	14) INTEL/14.0.2
5) globus/5.0.4-r1	10) tgresid/2.3.4	15) impi/4.1.3.048/intel64

- The above listing shows that this user has 15 packages loaded

Modules: Load/Unload a Package

- **The command for loading a package into a user's environment is:** `module load <package name>`.
- **The command for unloading a package is:** `module unload <package name>`.
- **If a specific version of a package is desired, the command can be expanded to:** `module load <package name>/<package version>`.

```
[fchen14@smic1 ~]$ module av intel
----- /usr/local/packages/Modules/modulefiles/apps -----
intel/13.0.0          intel/13.1.3          intel/14.0.2(default)
[fchen14@smic1 ~]$ module load intel
[fchen14@smic1 ~]$ icc -v
icc version 14.0.2 (gcc version 4.4.7 compatibility)
[fchen14@smic1 ~]$ module unload intel
[fchen14@smic1 ~]$ module load intel/13.1.3
[fchen14@smic1 ~]$ icc -v
icc version 13.1.3 (gcc version 4.4.7 compatibility)
```

Modules: Unload All Loaded Packages

➤ **To unload all loaded modulefiles, use the purge method:**

```
[fchen14@smic1 ~]$ module list
Currently Loaded Modulefiles:
  1) intel/14.0.2              3) r/3.1.0/INTEL-14.0.2      5) ansys/15.0
  2) INTEL/14.0.2             4) impi/4.1.3.048/intel64
[fchen14@smic1 ~]$ module purge
[fchen14@smic1 ~]$ module list
No Modulefiles Currently Loaded.
[fchen14@smic1 ~]$
```

Modules: Dependencies

- **Note that Modules will load any prerequisites (dependencies) for a package when that package is loaded. (SoftEnv does not)**

```
[fchen14@smic1 ~]$ module list
No Modulefiles Currently Loaded.
[fchen14@smic1 ~]$ module av gromacs
```

```
----- /usr/local/packages/Modules/modulefiles/apps -----
gromacs/5.0/INTEL-140-MVAPICH2-2.0(default)
[fchen14@smic1 ~]$ module load gromacs/5.0
gromacs/5.0                                gromacs/5.0/INTEL-140-MVAPICH2-2.0
[fchen14@smic1 ~]$ module load gromacs
[fchen14@smic1 ~]$ module list
Currently Loaded Modulefiles:
  1) intel/14.0.2                        3) mvapich2/2.0/INTEL-14.0.2
  2) INTEL/14.0.2                      4) gromacs/5.0/INTEL-140-MVAPICH2-2.0
```

Modules: Switch Between Packages

- Different versions of entire software packages can be replaced with the module switch/swap command by unloading module1 and loading module2: `module switch <package1> <package2>`.

```
[fchen14@smic1 ~]$ module load impi/4.1.3.048
```

```
[fchen14@smic1 ~]$ module list
```

Currently Loaded Modulefiles:

```
1) intel/14.0.2          2) impi/4.1.3.048/intel64
```

```
[fchen14@smic1 ~]$ module switch impi/4.1.3.048/intel64 mvapich2/2.0/INTEL-14.0.2
```

```
[fchen14@smic1 ~]$ module list
```

Currently Loaded Modulefiles:

```
1) intel/14.0.2          2) mvapich2/2.0/INTEL-14.0.2
```

Modules: Display the module changes

- **The display/show command will detail all changes that will be made to the user's environment:** `module disp <package name>`.

```
[fchen14@smic1 ~]$ module disp python/2.7.7-anaconda
```

```
-----  
/usr/local/packages/Modules/modulefiles/apps/python/2.7.7-anaconda:
```

```
module-whatis    Description: Python is a programming language that lets you work  
more quickly and integrate your systems more effectively. - Homepage:  
http://python.org/
```

```
conflict         python
```

```
prepend-path     CPATH /usr/local/packages/python/2.7.7-anaconda/include
```

```
prepend-path     LD_LIBRARY_PATH /usr/local/packages/python/2.7.7-anaconda/lib
```

```
prepend-path     LIBRARY_PATH /usr/local/packages/python/2.7.7-anaconda/lib
```

```
prepend-path     MANPATH /usr/local/packages/python/2.7.7-anaconda/share/man
```

```
prepend-path     PATH /usr/local/packages/python/2.7.7-anaconda/bin
```

```
prepend-path     PKG_CONFIG_PATH /usr/local/packages/python/2.7.7-  
anaconda/lib/pkgconfig
```

```
prepend-path     PYTHONPATH /usr/local/packages/python/2.7.7-  
anaconda/lib/python2.7/site-packages
```

```
setenv           LHPC_ROOTPYTHON /usr/local/packages/python/2.7.7-anaconda
```

```
setenv           LHPC_VERSIONPYTHON 2.7.7  
-----
```

Modules: Load Automatically on Login

- On HPC and LONI clusters, Modules can be loaded automatically on login by adding the appropriate module load commands to a user's `~/.bashrc` or `~/.modules` (recommended) file
- The following example shows a `.modules` file that automatically loads R, intel mpi and ansys-15.0

```
[fchen14@smic1 ~]$ cat ~/.modules
## This is the default .modules file for smic
# It is used to customize your Modules environment
# variables such as PATH and LD_LIBRARY_PATH.
## The default software stack on smic uses
#   Intel/cluster_studio_xe_2013.1.046/composer_xe_2013_sp1.2.144
#   mvapich2/2.0/INTEL-14.0.2
## To learn more about available software, try:
# module --help
## Your default software
module load r/3.1.0/INTEL-14.0.2
module load impi/4.1.3.048/intel64
module load ansys/15.0
# You can add additional software here
```


Creating Your Own Module File

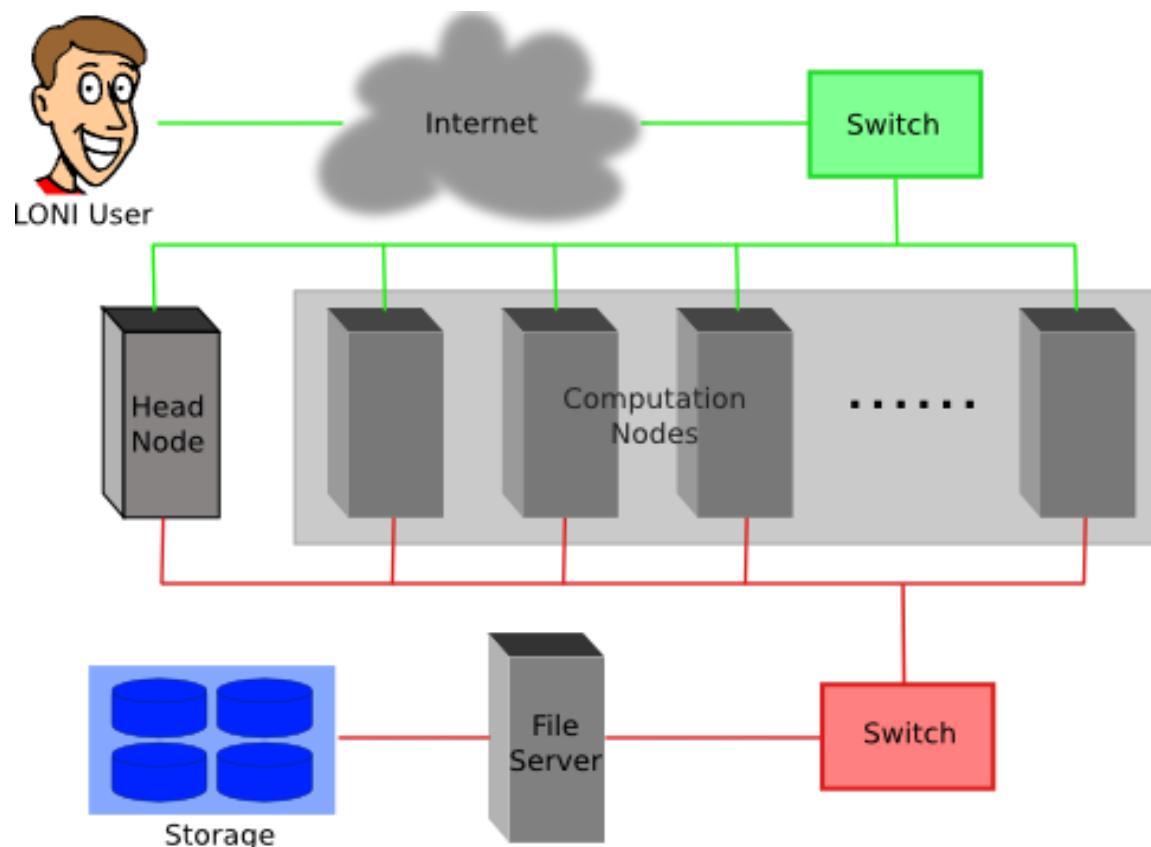
- An example of a simple module file (`~/my_module/gitkey`):

```
#!/Module
proc ModulesHelp { } {
    puts stderr { my compiled version of git.
}
}
module-whatis {version control using git}
set GIT_HOME /home/fchen14/packages/git-master/install
prepend-path PATH $GIT_HOME/bin
```

- Add the path to the key to the **MODULEPATH** environment variable:
`$ export MODULEPATH=~/my_module:$MODULEPATH`
- Then try to use:
`$ module load gitkey`
`$ which git`
`$ module unload gitkey`
`$ Which git`

Cluster Environment

- Multiple compute nodes
- Multiple users
- Each user may have multiple jobs running simultaneously



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	ppn	Max running jobs	Max nodes per job	Use
Eric	workq	3 days	8	16	24	Unpreemptable
	checkpt		8		48	Preemptable
	single		1	32	1	ppn < =8
QB2	workq	3 days	20	44	128	Unpreemptable
	checkpt		20		256	Preemptable
	single	7 days	1,2,4,8		1	Single node jobs

Queue Characteristics – LSU Linux clusters

Machine	Queue	Max Runtime	ppn	Max running jobs	Max nodes per job	Use
SuperMike II	workq	3 days	16	34	128	Unpreemptable
	checkpt		16		128	Preemptable
	bigmem	2 days	16		1	Big memory
	gpu	3 days	16		16	Job using GPU
	single	3 days	1,2,4,8		1	Single node jobs
Philip	workq	3 days	8	5	4	Unpreemptable
	checkpt		8		4	Preemptable
	bigmem		8		2	Big memory
	single	14 days	4	50	1	Single processor
SuperMIC	workq	3 days	20	34	128	Unpreemptable
	checkpt		20		360	Preemptable

Queue Characteristics – LSU AIX Clusters

Machine	Queue	Max Runtime	ppn	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

```
[fchen14@mike2 ~]$ qstat -q
```

```
server: mike3
```

Queue	Memory	CPU Time	Walltime	Node	Run	Que	Lm	State
workq	--	--	72:00:00	128	31	6	--	E R
mwfa	--	--	72:00:00	8	3	0	--	E R
bigmem	--	--	48:00:00	1	0	0	--	E R
lasigma	--	--	72:00:00	28	28	7	--	E R
bigmemtb	--	--	48:00:00	1	0	0	--	E R
priority	--	--	168:00:0	128	0	0	--	E R
single	--	--	72:00:00	1	62	0	--	E R
gpu	--	--	24:00:00	16	1	0	--	E R
preempt	--	--	72:00:00	--	0	0	--	E R
checkpt	--	--	72:00:00	128	31	137	--	E R
admin	--	--	24:00:00	--	0	0	--	E R
scalemp	--	--	24:00:00	1	0	0	--	E R
					156	150		

- For a more detailed description use mdiag

Queue Querying – Linux Clusters

- Displays information about active, eligible, blocked, and/or recently completed jobs: `showq` command

```
[fchen14@mike2 ~]$ showq
```

```
active jobs-----
```

JOBID	USERNAME	STATE	PROCS	REMAINING	STARTTIME
294690	michal	Running	64	00:22:41	Sat Jan 24 23:05:26
294693	michal	Running	64	00:22:41	Sat Jan 24 23:05:26

```
...
```

```
183 active jobs          6399 of 7512 processors in use by local jobs (85.18%)
                        401 of 468 nodes active          (85.68%)
```

```
eligible jobs-----
```

JOBID	USERNAME	STATE	PROCS	WCLIMIT	QUEUE TIME
295775	stumusii	Idle	32	3:00:00:00	Tue Jan 27 21:15:35
295776	stumusii	Idle	32	3:00:00:00	Tue Jan 27 21:17:43

```
...
```

```
9 eligible jobs
```

```
blocked jobs-----
```

JOBID	USERNAME	STATE	PROCS	WCLIMIT	QUEUE TIME
-------	----------	-------	-------	---------	------------

```
0 blocked jobs
```

```
Total jobs: 192
```

- **Command:** `qfree`, `qfreeioni`

- Show the number of free, busy and queued

Queue Querying - AIX clusters

➤ Command: llclass

```
-bash-3.2$ llclass
```

Name	MaxJobCPU d+hh:mm:ss	MaxProcCPU d+hh:mm:ss	Free Slots	Max Slots	Description
interactive	unlimited	unlimited	8	8	Queue for interactive jobs; maximum runtime of 30 minutes.
workq	unlimited	unlimited	32	224	Standard queue for job submissions; maximum runtime of 3 days.
cheme	unlimited	unlimited	32	96	Queue for Chemical Engineering; maximum runtime of 3 days.
single	unlimited	unlimited	32	64	Queue for single-node job submissions; maximum runtime of 3 days.

"Free Slots" values of the classes "workq", "cheme", "single" are constrained by the MAX_STARTERS limit(s).

Two 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 be changed after submission
- Purpose: production run

Submitting Jobs on Linux Clusters

➤ **Interactive job example:**

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

➤ **Add -X to enable X11 forwarding**

➤ **For batch jobs, PBS script is needed. A PBS script is a standard Unix/Linux shell script that contains a few extra comments at the beginning that specify directives to PBS. These comments all begin with **#PBS**.**

➤ **Batch Job example:**

```
qsub job_script
```

❖ **For details, refer to: <http://www.hpc.lsu.edu/docs/pbs.php>**

PBS Job Script – Serial Job

```
#!/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 <lioni_allocation> # Allocation name
#PBS -m ab                 # Send mail when job ends
#PBS -M <email address>    # Send mail to this address

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

PBS Job Script – Parallel Job

```
#!/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>
```

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!**

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 = checkpoint          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>`

```
-bash-3.2$ llq
```

Id	Owner	Submitted	ST	PRI	Class	Running On
-----	-----	-----	---	---	-----	-----
pandora1.19106.0	mainak	9/1 23:41	R	50	workq	pandora008
pandora1.19108.0	ghoshbd	9/2 14:58	R	50	workq	pandora005
pandora1.19109.0	ghoshbd	9/2 15:08	R	50	workq	pandora007
pandora1.19110.0	ghoshbd	9/2 15:33	R	50	workq	pandora002
pandora1.19111.0	ghoshbd	9/2 15:44	R	50	workq	pandora004
pandora1.19112.0	ghoshbd	9/2 15:58	I	50	workq	
pandora1.19113.0	ghoshbd	9/2 16:10	I	50	workq	
pandora1.19114.0	mainak	9/4 08:16	I	50	workq	

```
8 job step(s) in queue, 3 waiting, 0 pending, 5 running, 0 held, 0 preempted
```

Exercise

- **Submit a batch job to single queue, using nodes=1:ppn=1, run the python script to calculate the value of pi**
 - You can use the sample file in example directory, modify it to your environment (on SuperMikell):
`/home/fchen14/userenv/pbs_script/single.pbs`
- **Run an interactive job session for 5 min, using nodes=1:ppn=16**
 - Verify using hostname that you are not on the headnode
 - Check available PBS variables and print them
 - Print \$PBS_NODEFILE and count the total lines

Next Week Training

- **HPC User Environment 2, Feb. 4**
 - More on job management
 - Compiling serial/parallel program

- **Weekly trainings during regular semester**
 - Wednesdays “9:30am-11:30am” session, Frey 307 CSC

- **Programming/Parallel Programming workshops**
 - Usually in summer

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