HPC User Environment 1

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

Louisiana State University
Baton Rouge
September 5, 2018

Outline

- **Things to be covered in the training**
  - Basic HPC concepts
    - Computational speed evaluation
    - Top 500 list
  - Available HPC resources
    - LONI & LSU HPC
    - Account and allocations
  - HPC software environment
    - General cluster architecture
    - How to access HPC & LONI clusters
    - How to check file systems, allocation balance
    - How to transfer files between cluster and local PC
    - The software management tool Module
  - Job submission and 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

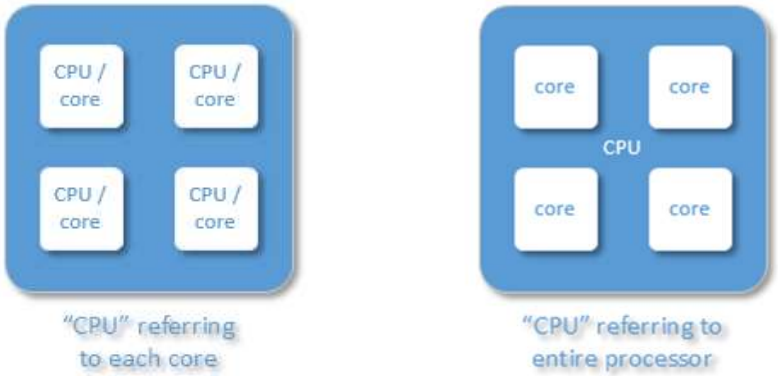
600 mph

60 mph

- **How do we evaluate the performance of HPC?**

Core, CPU, Processor?

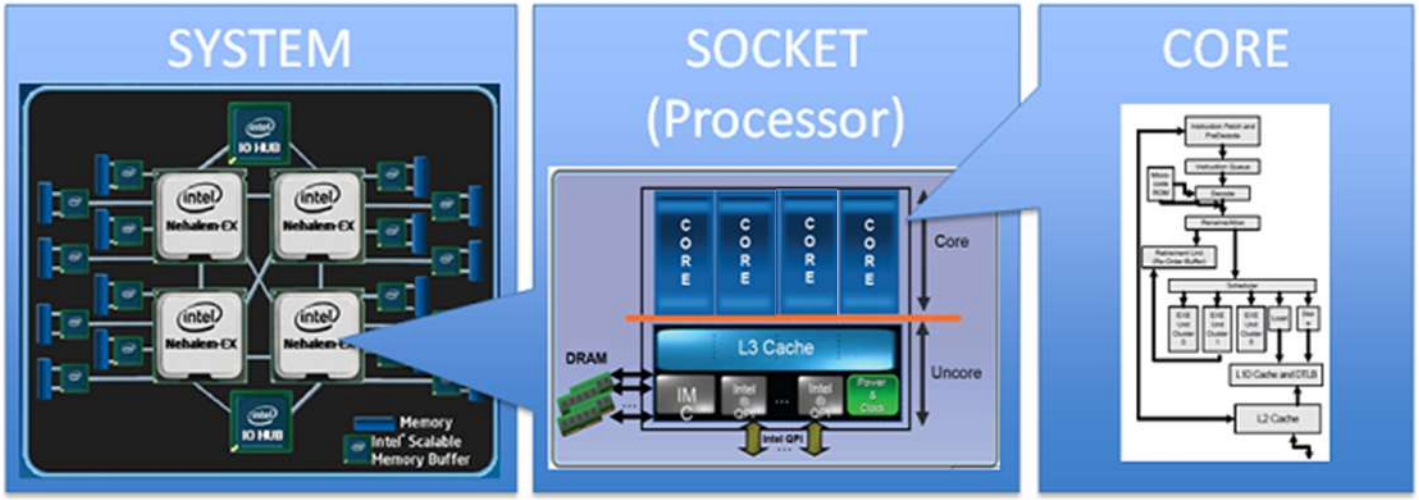
- **A core is usually the basic computation unit.**
- **A CPU may have one or more cores to perform tasks at a given time.**



- **In this training, CPU = processor, which has multiple cores. The term "CPU" will not be used.**

Case study – core and processor

- **How many cores does this computer have?**



- **4 cores/processor * 4 processors = 16 total cores**

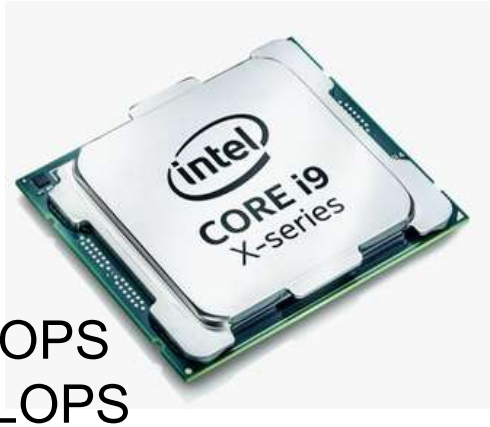
Measure HPC performance-FLOPS

- **Performance is measured in *Floating Point Operations Per Second* (FLOPS or flop/s)**
- $FLOPS = cores \times clock \times \frac{FLOPs}{cycle}$
  - Most processors today can do 4 FLOPs per clock cycle. Therefore a single-core 2.5-GHz processor has a theoretical performance of 10 billion FLOPs = 10 GFLOPs
  - Dual core, quad core? (i3, i5, i7)

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}

- Intel i9-7980XE was released in September 2017
  - 18 cores
  - "the first teraflop desktop PC processor ever"
  - $1999



- 18 * 4.4 GHz (when Turbo Boost) *16 = 1267 GFLOPS
  = 1. 267 TFLOPS

Supercomputing on a Cell Phone?

- **Hex-core processors are coming to your phone**
  - Nvidia, TI, QualComm…
  - Processing power in the neighborhood of 50 GigaFLOPS
  - Would make the top 500 supercomputer list 20 years ago
  - What is your phone's FLOPS?

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}

- iPhone X's A11 Bionic chip: 2×Monsoon + 4×Mistral (2.39 GHz)
- 57.4 GFLOPS
- Compare to ENIAC (500 FLOPS)
- Compare to top 500 in 1993 #1 (59.7 GFLOPS), #500 (0.42 GFLOPS)

The Top 500 List

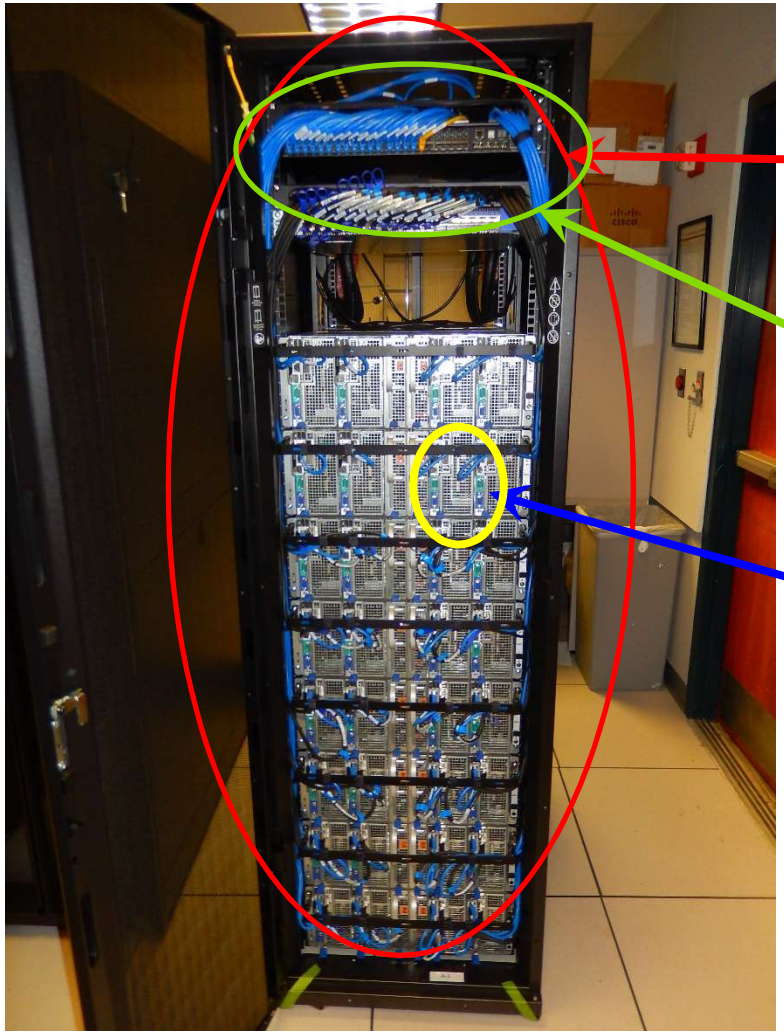
TOP500 SUPERCOMPUTER SITES

- **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 June 2018, an IBM-built supercomputer called Summit is the fastest in the world.**
  - Cores: 2,282,544
  - Peak Performance: 187,659TFlop/s
- **As of June 2018**
  - LONI QB2 ranks **232** 1474TFlop/s

Intel i9-7980XE: 1.267 TFlop/s

Supercomputer Cluster Racks

Inside A Cluster Rack

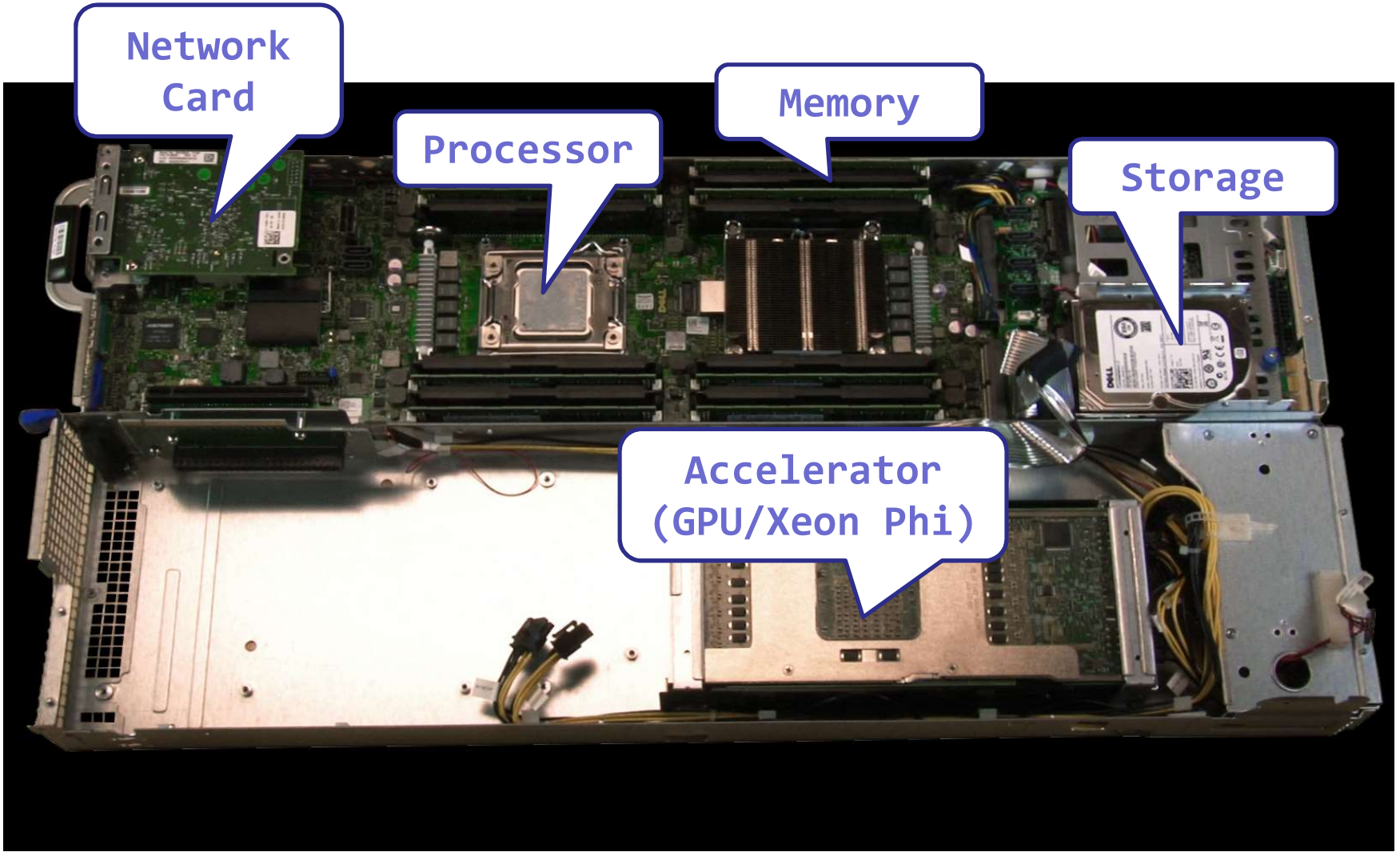


Rack

Interconnect: Infiniband Switch

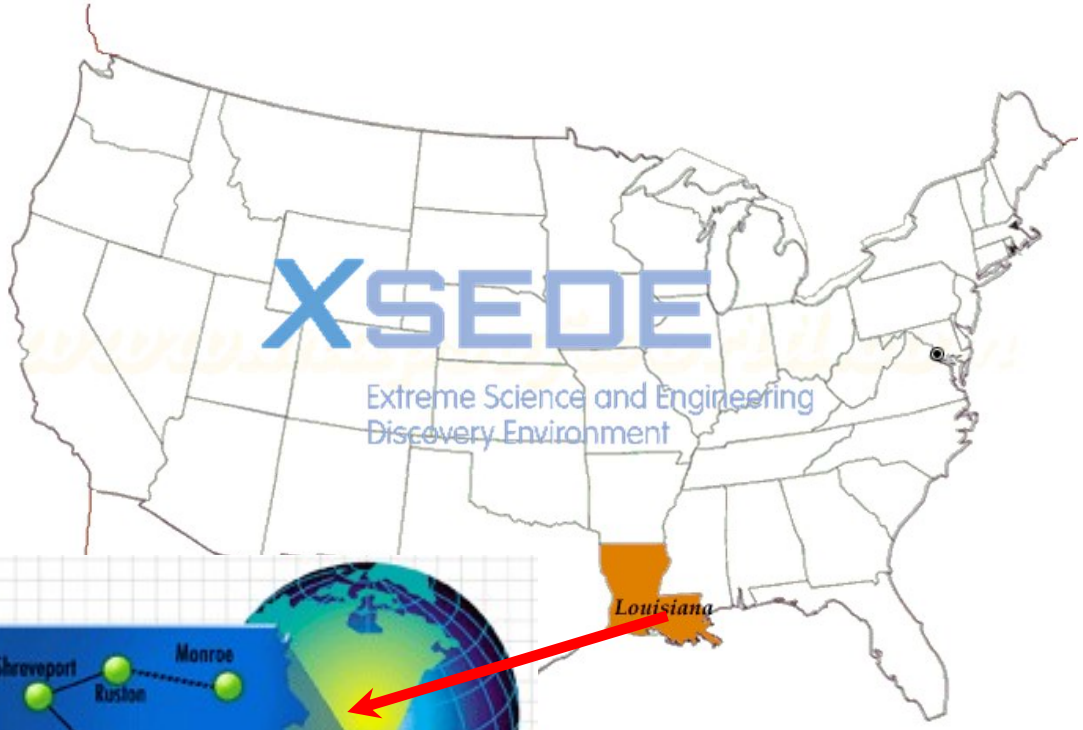
Compute Node

Inside A Compute Node



Available Computing Resources

- **University wide-HPC@LSU**
- **State wide-LONI**
- **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	
Hostname	smic.hpc.lsu.edu
Peak Performance/TFlops	1000
Compute nodes	360
Processor/node	2 Deca-core
Processor Speed	2.8GHz
Processor Type	Intel Xeon 64bit
Nodes with Accelerators	360
Accelerator Type	Xeon Phi 7120P
OS	RHEL v6
Vendor	
Memory per node	64 GB
Detailed Cluster Description	
User Guide	
Available Software	

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	

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	

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

State Level - Louisiana Optical Network Infrastructure (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.**

LONI-Louisiana Optical Network Infrastructure

- **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
  - Grambling State University
  - Southeastern Louisiana University

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	

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	QB2	1474	ISB	Dell	Linux x86_64
LSU	Philip	3.5	LSU	Dell	Linux x86_64
	SuperMIC	925	LSU	Dell	Linux x86_64
	SuperMike	212 (CPU+GPU)	LSU	Dell	Linux x86_64

ISB: Information Services Building (Downtown Baton Rouge)

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 of the XSEDE clusters.

XSEDE
Extreme Science and Engineering Discovery Environment

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.**
- User accounts on LSU HPC require a valid LSU email address, and LSU affiliated sponsor.
- **ACCOUNT SPONSOR: The person who is responsible for your activities on the cluster.**
- Only faculty members or certain research staff from LSU Baton Rouge campus can sponsor accounts.
- LSU students, postdoctoral researchers, or research associates may choose their advisor as their sponsor.
- For prospective LSU HPC Users from outside LSU, you are required to have a **faculty member** at LSU as your Collaborator to sponsor you a LSU HPC account.

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 member** in one of LONI Member Institutions as your Collaborator to **sponsor** you a LONI account.

Account Eligibility Test

- **I can be granted a LSU HPC or LONI account if:**
  a) I am using HPC resource for my research, sponsored by my PI
  b) I am attending HPC training sessions, sponsored by the HPC staff
  c) I am taking a class that requires using HPC resource, sponsored by the course instructor
  d) a and b
  e) a and c ☺
  f) All of the above

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.**
- **Contact user services**
  - Email Help Ticket: sys-help@loni.org
  - Telephone Help Desk: +1 (225) 578-0900

How Do I Get a *LONI* Account?

- **Visit https://allocations.loni.org/login_request.php**
- **Enter your INSTITUTIONAL Email Address.**
- **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 or course instructor who must be a Full Time Faculty member at a LONI member institution. Never enter HPC staff's name.**
- **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.**
- **Check your email and click on the link provided (link is active for 24hrs only)**
- **Fill the form provided**
- **For LSU HPC CONTACT/COLLABORATOR field enter the name of your research advisor/supervisor or course instructor who must be a Full Time Faculty member at LSU. Never enter HPC staff's name.**
- **Click Submit button**
- **Your account will be activated once we have verified your credentials**

Account Management
- LSU HPC and LONI User Portals

- **Both portals can be found at the top of http://www.hpc.lsu.edu/**
- **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
- **May keep the account after graduation under certain circumstances**

Allocation

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

- **It is FREE to have allocation and use LSU HPC/LONI resources, but there is a value of approximately $0.1/core-hour from other sources (e.g. taxpayer)**

Who Can Request Allocation?

- **Only Full Time LSU/LONI Faculty member at LONI member institutions can act as Principle Investigators (PI) and request LSU HPC/LONI 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 or course instructor (if your course requires a LSU HPC/LONI account).**
- **As a non-PI, your goal is to help your PI understand the allocation policy and prepare the allocation request.**

Allocation Types

- **Startup: Allocations upto 50K SUs**
  - Can be requested at any time during the year.
  - **Begins on the first day of the quarter** in which we received the request
  - Only two active allocations per PI at any time.
- **Large (research): Allocations between 50K - 4M SUs.**
  - Decision will be made on January 1, April 1, July 1 and October 1 of each year
  - A request must be submitted one month before the decision day.
  - 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 SUs, and a PI may have a total of 6M SUs active at any given time

LSU HPC Research Allocation Proposals

- **Depending on the amount of SUs being requested, research allocations are categorized into three types**

Research allocation types and requirements

Type	Size (SU)	Previous usage and outcome	External funding or LSU demand	Number of pages
A	50,000 to 300,000	Optional	Optional	4
B	300,000 to 1,000,000	Required	Optional	5
C	>1,000,000	Required	Required	6

- Sample proposals for each type are available now.
- http://www.hpc.lsu.edu/users/hpcpolicy.php#research
- ***LONI allocation procedure is not affected.***

Case study – startup allocation

User: “My allocation hpc_xyz was cancelled on 12/31/16, even though the allocation itself was activated less than a year ago (see email below). Could you please explain me why?”

Forwarded email:

From: Accounts Admin [mailto:sys-help@loni.org]

Sent: Thursday, February 11, 2016 11:11 AM

Subject: [CyS-HPC] Project hpc_xyz has been activated

User Services: “The startup allocation begins at the first day of the quarter when you applied. So when you applied for this allocation in Feb 2016, it starts 1/1/16 and ends 12/31/16.”

How to Request/join an Allocation

- **LONI: Login to your profile at https://allocations.loni.org**
- **LSU HPC: Login at https://accounts.hpc.lsu.edu/allocations.php**
- **Click on "Request Allocation" in the right sidebar**
- **Click "New Allocation" to request a New Allocation. (PI only)**
  - 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 (Non-PI).**
  - 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.**

Case study – startup allocation

User: “Hi, my PI recently applied for an allocation on SuperMike II which was approved (see forwarded email below). However, when I view my allocation balances at https://accounts.hpc.lsu.edu/balances.php, I do not see that this allocation is available for my use. What steps should I take to access this allocation?”

User Services: “You should either ask your PI add you to the allocation, or request to be added through the user portal”

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
- You will receive an email with link to reset your password, link must be used within 24 hours.
- Once you have entered your password, your password reset request needs to be manually reviewed. **IMPORTANT:**
  - **DO NOT** assume your new password is available to use right after the reset request submission
  - **DO NOT** submit the reset request multiple times if you didn't see your new password worked at once.
- 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.

Case study - password reset

User: “I have been trying to access my accounts on Mike and Queen Bee via an SSH client, but the connection won't go through. I ***reset*** my passwords this weekend and the terminals keep giving me a "Password Authentication Failed" error message…..”

User Services: “When you send a password reset request, it has to be manually processed for security reason before your new password becomes available.”

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.**
- **Do not tell your password to others including your advisor.**

During the break…

- **Login to one of the user portals (LSU HPC or LONI) with your HPC username and password. Enter "Certificate of completion for HPC User Environment One training session" and update your profile**
- **Download MobaXterm (if you are Windows user)**
- **Review commands in Linux and the vim editor**

Cheat sheet of Commands in Linux

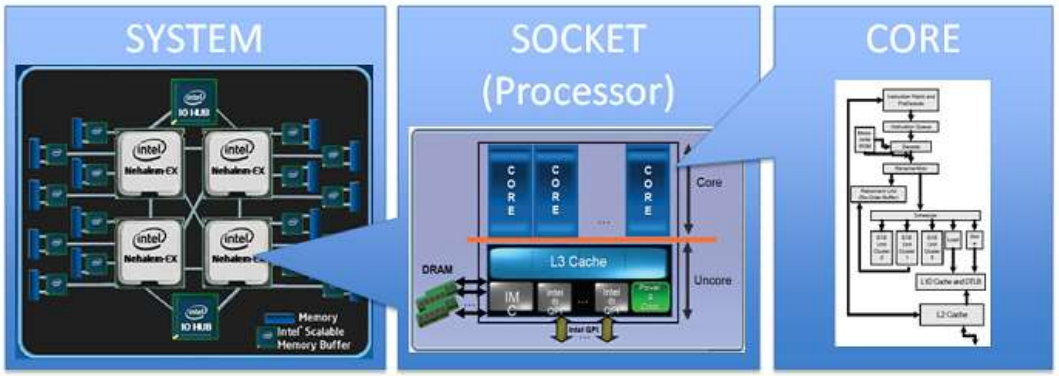
- History
- mkdir (name of file) -> makes a folder
- ls -> list
  - -a list all files including hidden
  - -l shows files with a long listing format
- cd -> change directory
- pwd -> shows location
- cp -> copy
- rm -> Remove files (careful)
- Up arrow (↑) -> moves back in history
- Tab -> fills in unique file name
- Tab Tab -> press tab twice, shows all available file names

Cheat sheet of vi

- vi (name of file)
- Commands in VI
  - i enter insert mode (-- INSERT -- shows in the bottom left corner)
  - esc exits insert mode, back to the command mode
    - dd -> deletes line
    - u -> Undo
    - Shift Z shift Z or :wq -> saves and exits VI
    - :q! -> exit without saving
    - : (some number) -> moves through file to row #
    - /(indicator) -> search
      - Use N to find Next
    - [(page up)] (page down)
- NO CAPS (for example :q! is not :Q!)

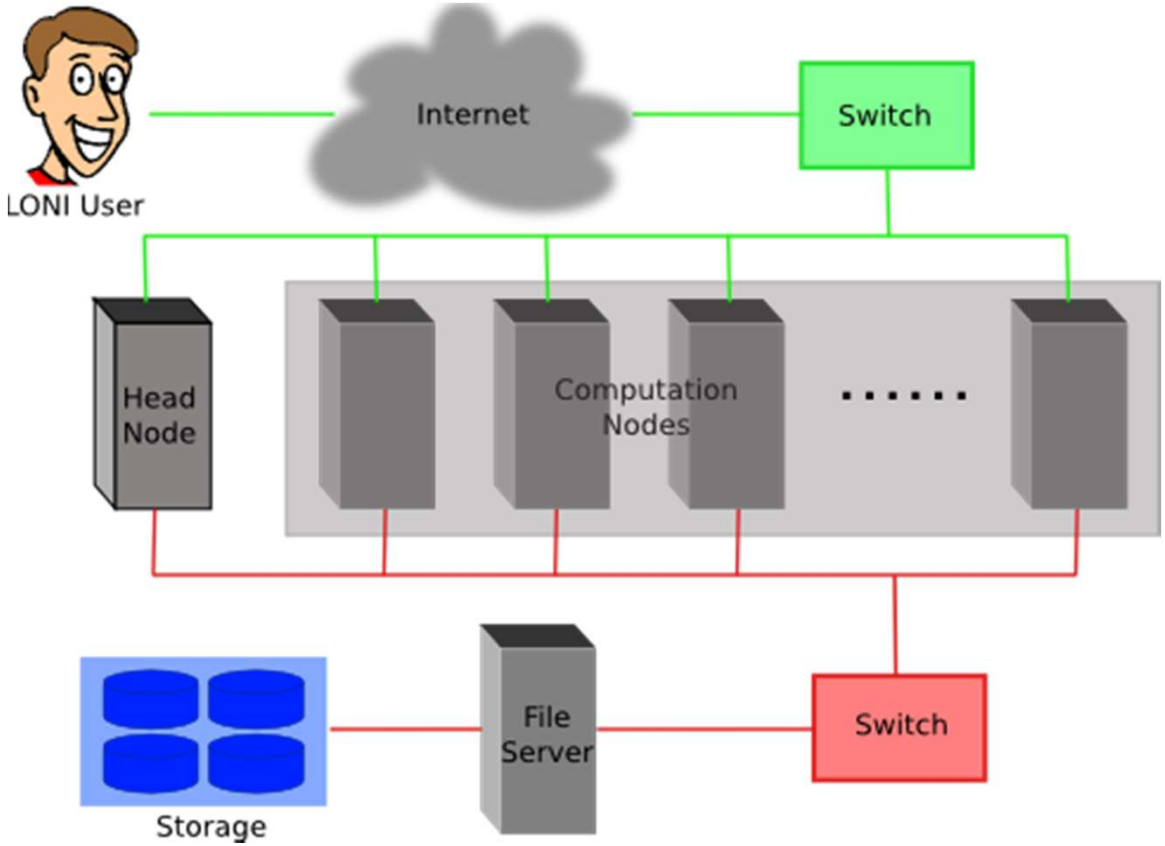
Cluster Nomenclature

Term	Definition
Cluster	A set of connected computer nodes that work together, each node set to perform the same kind of task (job).
Node	A single, named host machine in the cluster.
Core	The basic computation unit in the processor (CPU). For example, a quad-core processor has 4 cores.
Job	A user's request to use a certain amount of resources for a certain amount of time on cluster for his/her work.



General Cluster Architecture

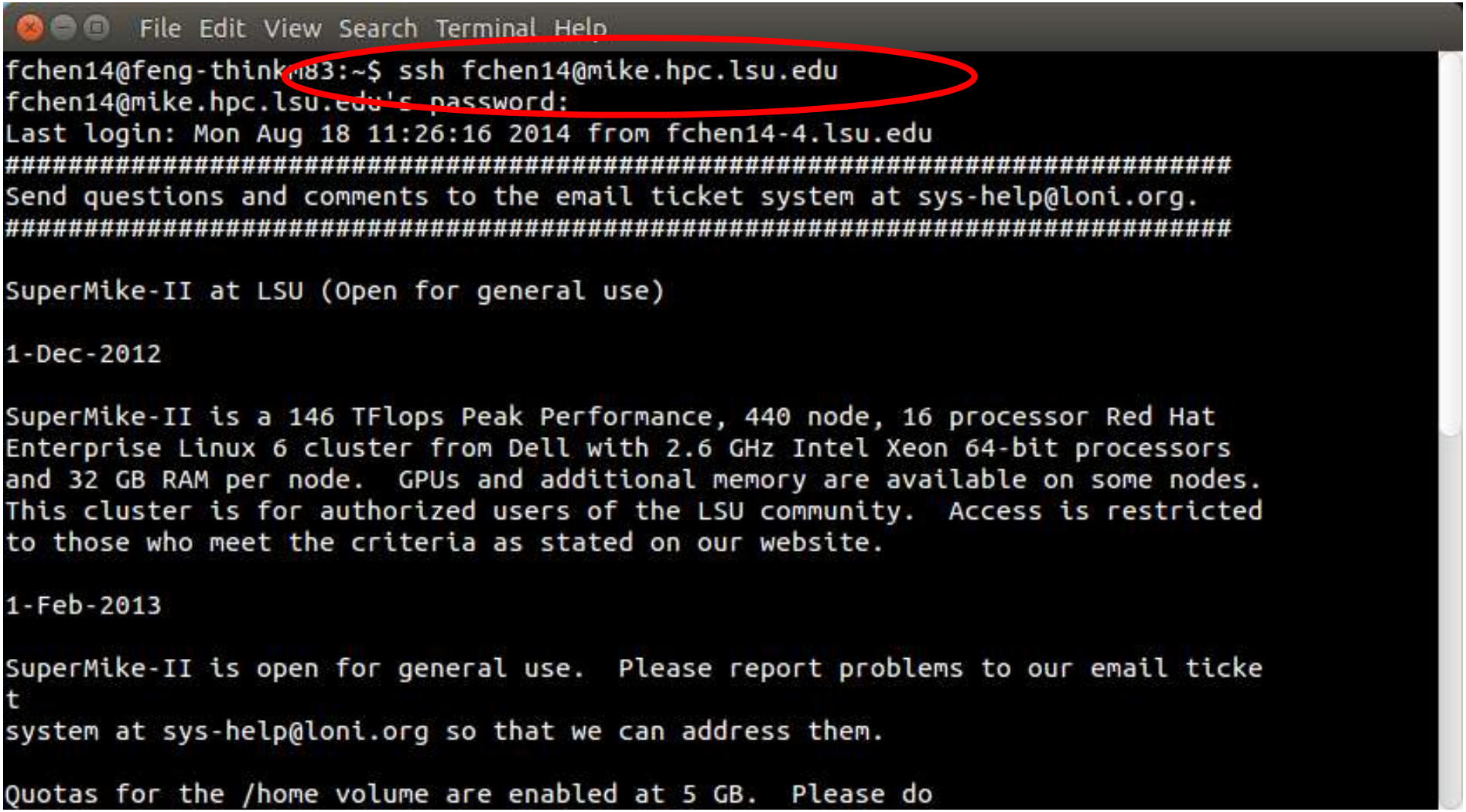
- **Multiple compute nodes**
- **Multiple users**
- **Each user may have multiple jobs running simultaneously**



Accessing Cluster via SSH (Secure Shell)

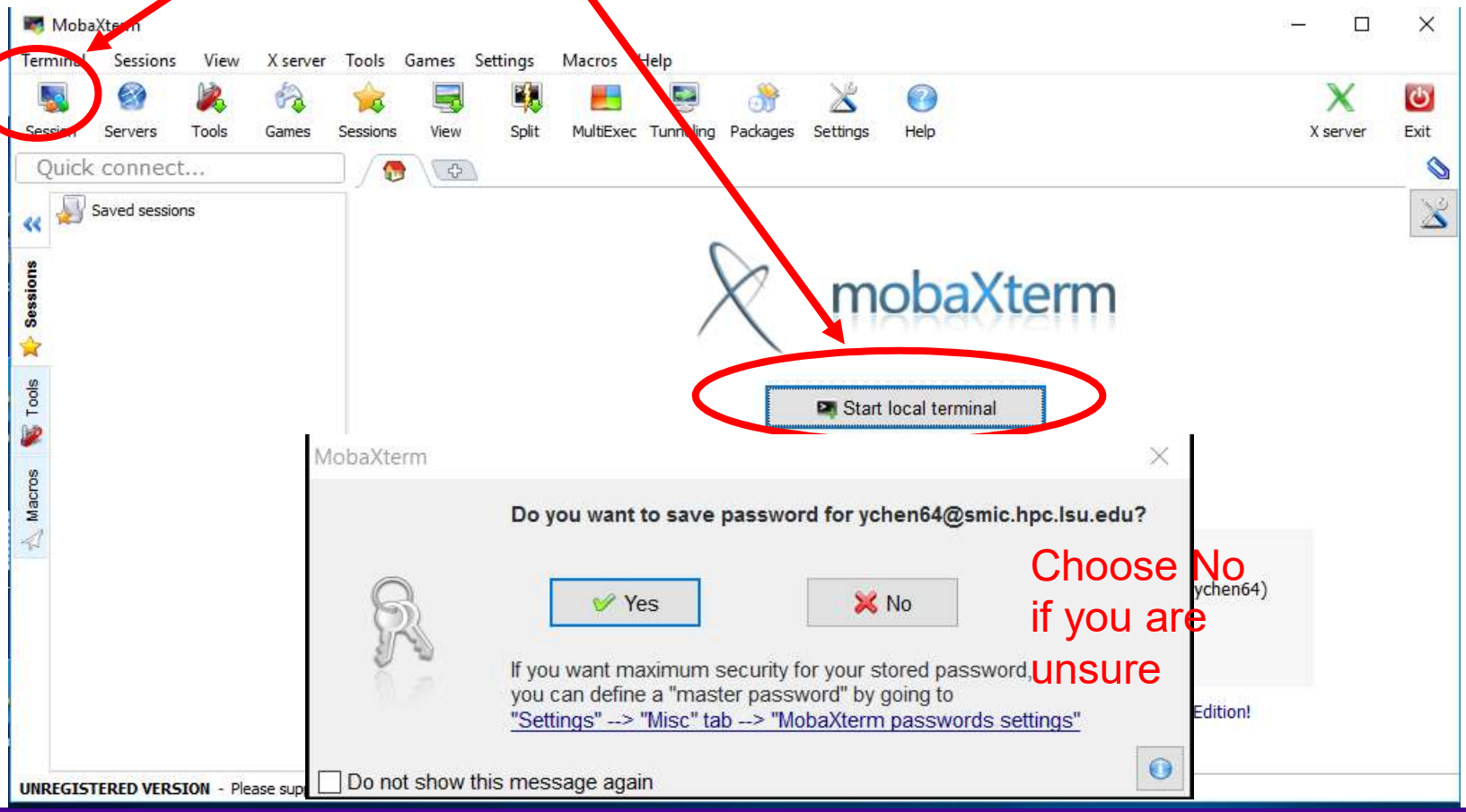
- **On Linux and Mac**
  - use ssh on a terminal to connect
- **Windows box (ssh client):**
  - MobaXterm (recommended)
  - SSH Secure Shell Client
  - Putty (Better use Bitvise SSH Client from http://www.putty.org/)
- **Username and password**
- **Host name**
  - LONI: <cluster_name>.loni.org
    - <cluster_name> can be:
      - **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**

Accessing Cluster on Linux and Mac

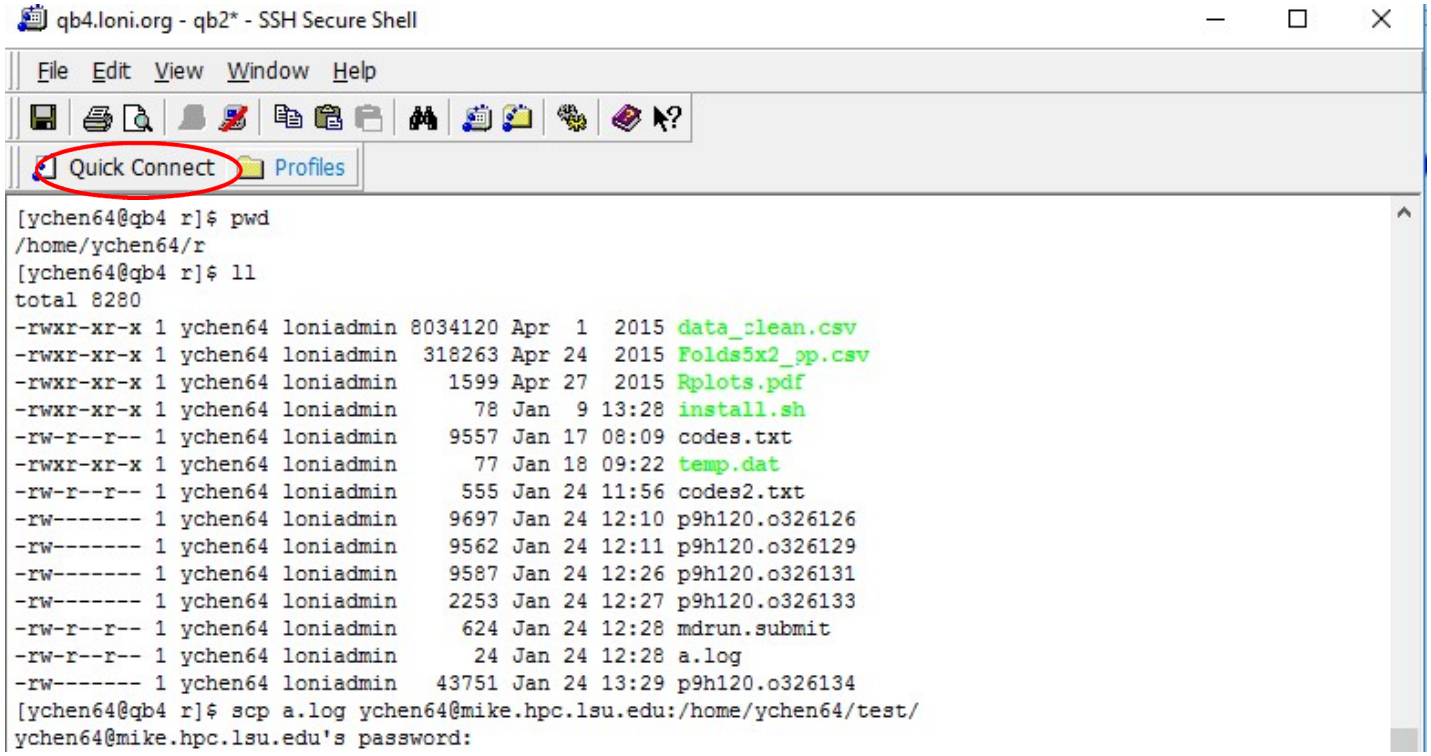


Accessing Cluster on Windows - MobaXterm

- **First time user, choose either one:**
  - use ssh on a terminal
  - start a new remote session -> SSH

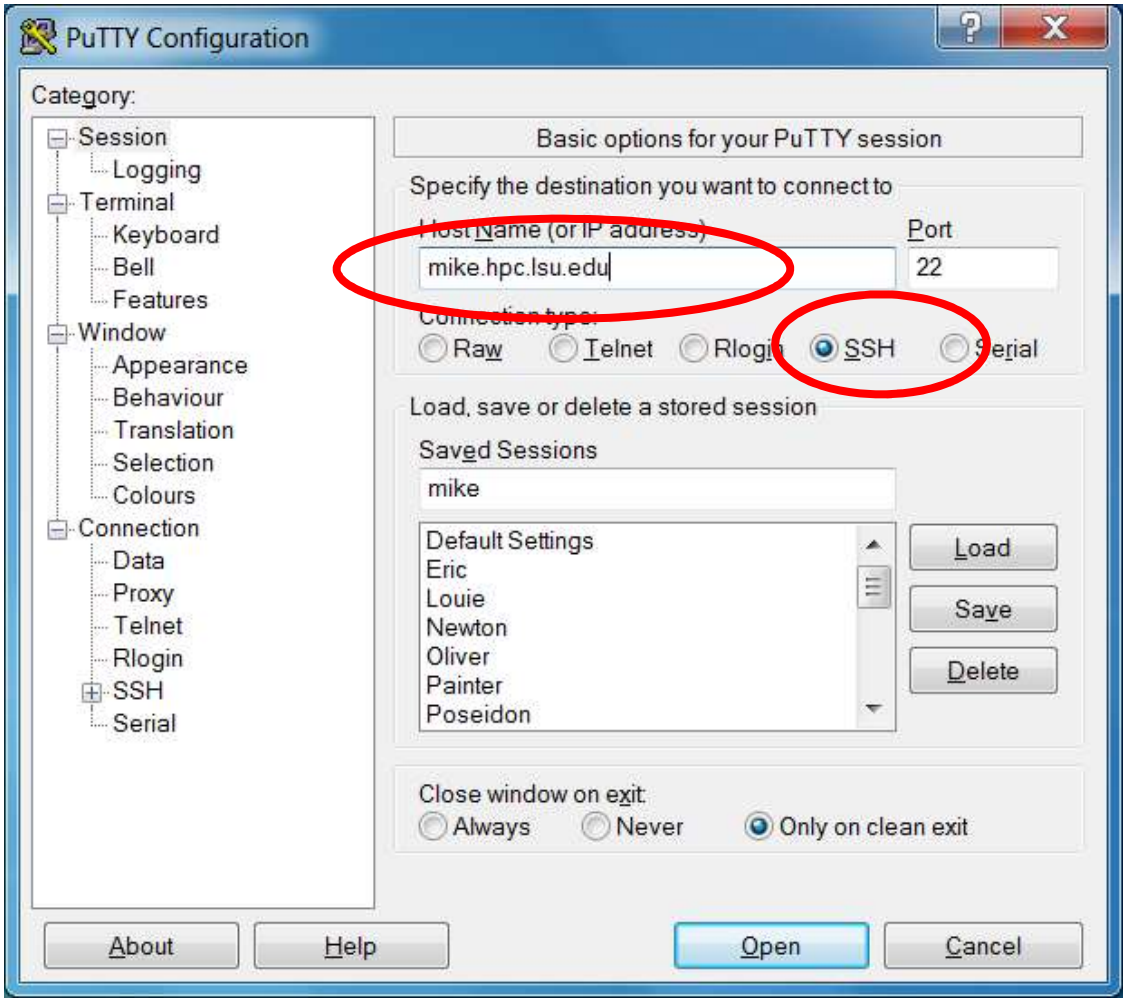


Accessing cluster on Windows - SSH Secure Shell Client



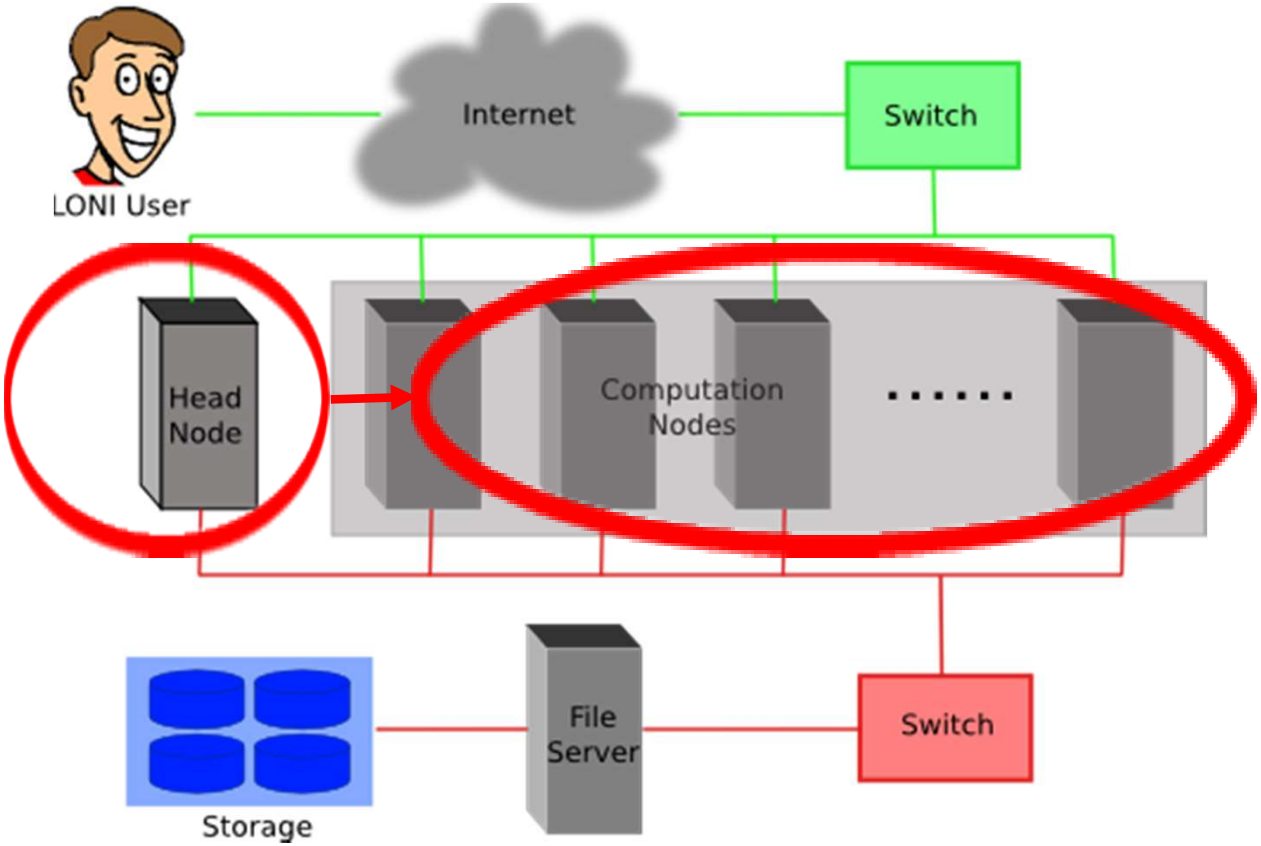
- **SSH Secure Shell Client**
  - command line scp and rsync
  - sftp file transfer through GUI

Accessing cluster on Windows - Putty



Cluster Environment

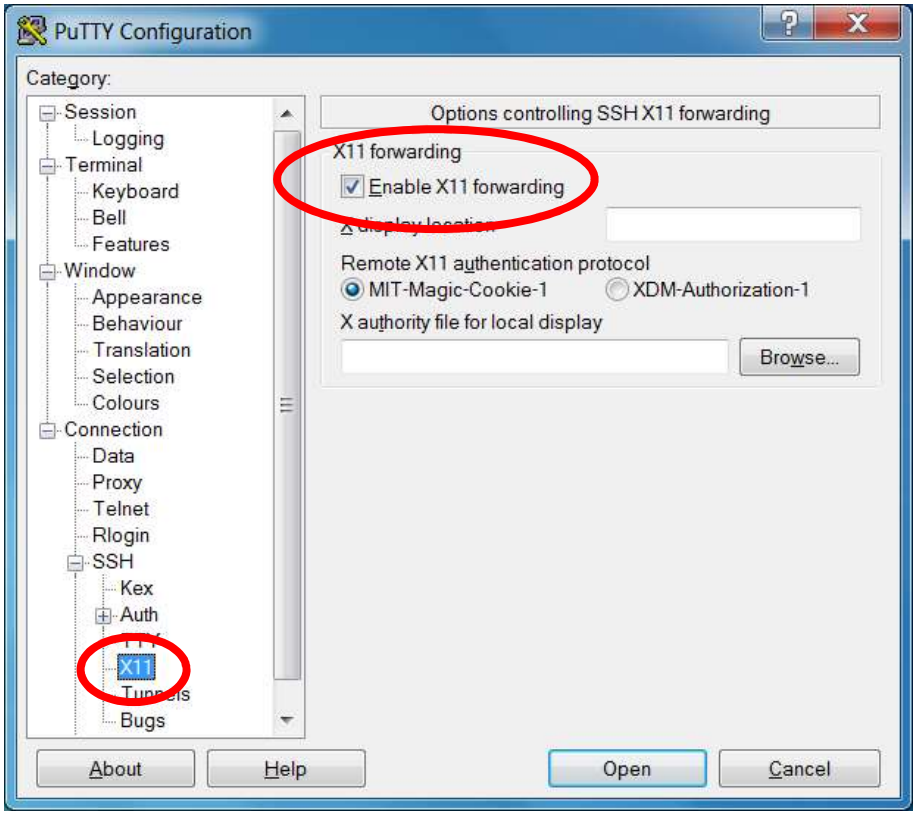
- **Multiple compute nodes**
- **Multiple users**
- **Each user may have multiple jobs running simultaneously**



DO NOT RUN YOUR JOB ON THE HEAD NODE!!!

Enable X11 Forwarding

- **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)
- **On Windows using MobaXterm**
  - X server already set up
  - Automatically start X server at start up (Settings -> X11)

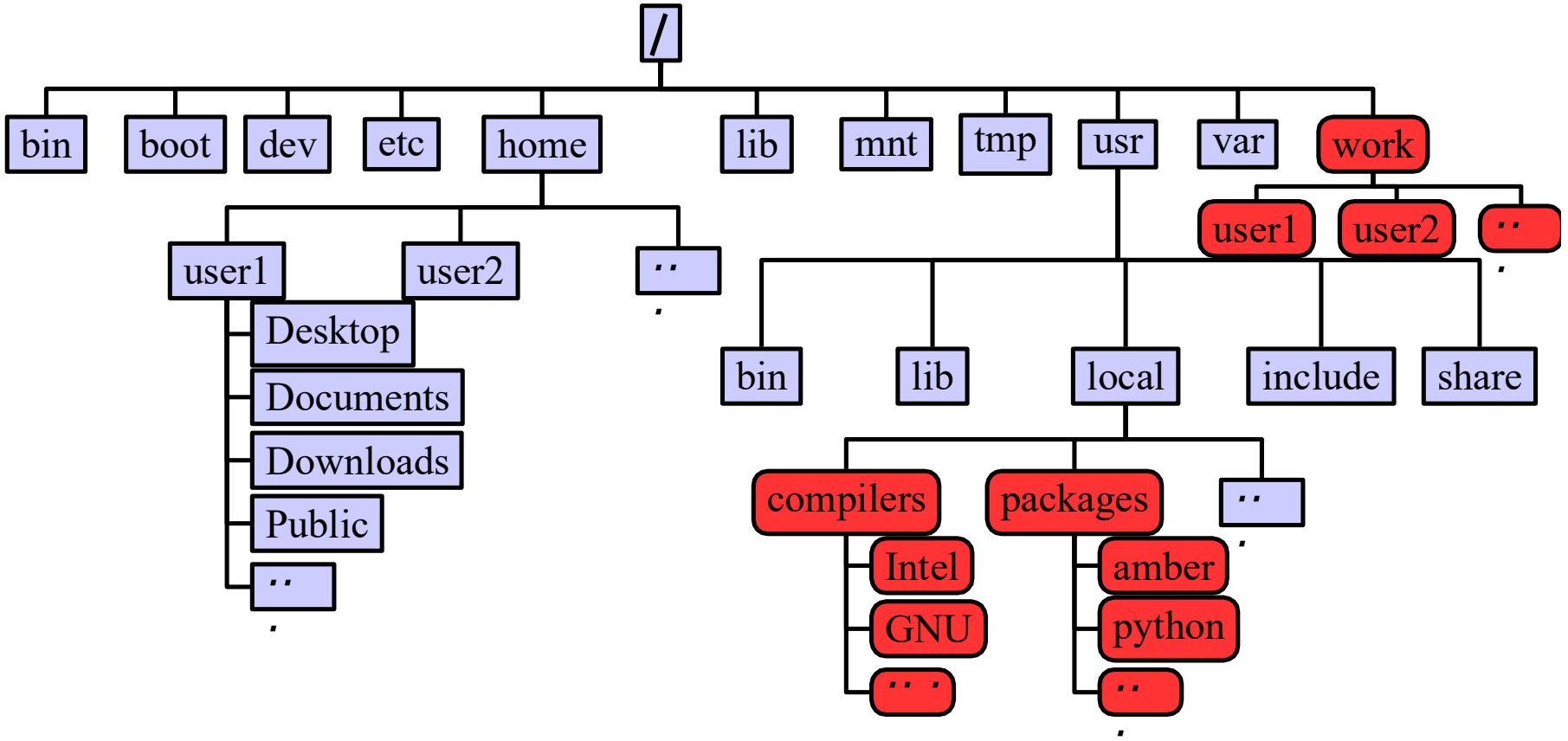


Cluster Environment

- **Useful commands on the head node**
  - pwd, print current directory
  - who, check who is on the node
  - balance, check allocation balance

File Systems

- **All files are arranged in directories.**
- **These directories are organized into the file system**



File Systems

Directory (folder)	Distributed	Throughput	File life time	Best used for
Home	Yes	Low	Unlimited	Code in development, compiled executable
Work	Yes	High	60 days	Job input/output
Project	Yes	Medium/High	1 year	storage space for a specific project, NOT for archival purposes

- **Tips**
  - The work directory is not for long-term storage
    - Files are subject to be purged after 60-90 days
  - The work directory will be created 1 hour after the first cluster login
  - The project directory provides storage space for a specific project
    - **PI** can apply storage allocation to use and renew periodically
    - NOT for archival purposes
    - For more info about the storage allocation, see appendix or contact us

Disk Quota

Cluster	Home		Work		Local scratch
	Access point	Quota	Access Point	Quota	Access point
LONI	/home/$USER	5 GB	/work/$USER	N/A	/var/scratch
HPC					

- **Never let you job write output to your home directory**
- **Check current disk quota and usage**
  - showquota

File Transfer (Linux/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>

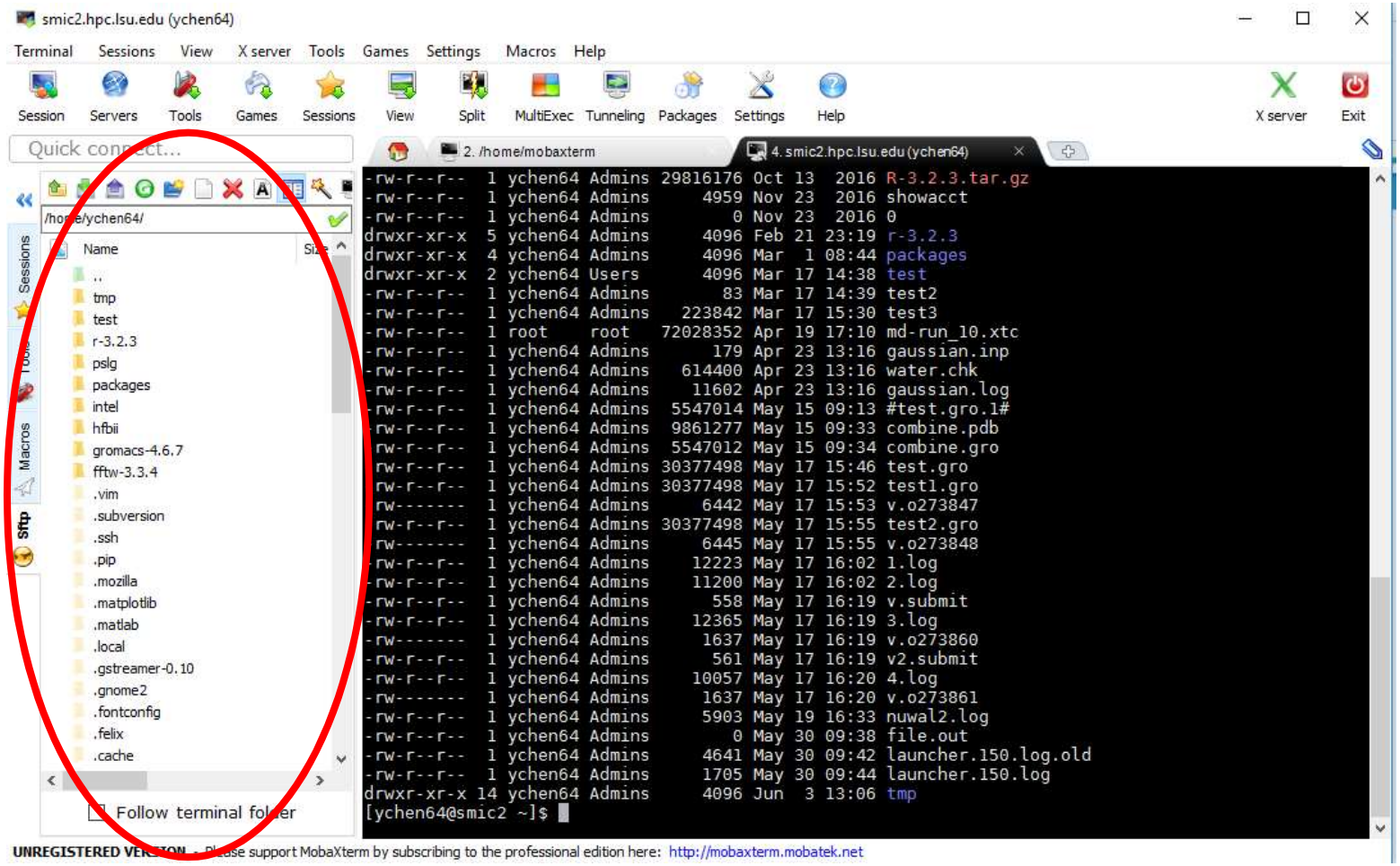
```
[ychen64@mike2 ~]$ scp
usage: scp [-1246BCpqrv] [-c cipher] [-F ssh_config] [-i identity_file]
           [-l limit] [-o ssh_option] [-P port] [-S program]
           [[user@]host1:]file1 ... [[user@]host2:]file2

[ychen64@mike2 ~]$ rsync
rsync  version 3.0.6  protocol version 30
Usage: rsync [OPTION]... SRC [SRC]... DEST
```

- **From a download link on a website (usually opened with a web browser)**
  - Right click on the link and then copy the link location
  - wget command

```
[ychen64@mike2 ~]$ wget <paste_your_copied_link_here>
```

File Transfer (Windows)

- **From/to a Windows machine**
  - Use a client that supports the scp protocol (e.g. SSH Secure Shell Client)



Review the use of vim editor

- vi (name of file)
- Commands in VI
  - i is insert mode (-- INSERT -- shows in the bottom left corner)
  - esc exits insert mode, back to the command mode
    - dd -> deletes line
    - u -> Undo
    - Shift Z shift Z or :wq -> saves and exits VI
    - :q! -> exit without saving
    - : (some number) -> moves through file to row #
    - /(indicator) -> search
      - Use N to find Next
    - [(page up)] (page down)
- NO CAPS (e.g. :q! ≠ :Q!)

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 home directory, unless request by a group of users, in which case it will be installed under `/project` or `/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`
- **Module**
  - An application that helps users set up their environment. Most supercomputing sites (including XSEDE) use modules. Much more convenient than setting variables in .bashrc
  - SuperMike2, SuperMIC, Philip and QB2
- **SoftEnv**
  - HPC user services will no longer support issues related to Softenv
  - SuperMike2 users: http://www.hpc.lsu.edu/docs/guides.php?system=SuperMike2#env

Using Environment Modules

- **Environment Modules is a 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/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 module files, 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: 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
```

Exercise: Use Modules

- **Find the key for gromacs**
  - Set up your environment to use gromacs you choose (one time change)
  - Check if the variables are correctly set by “which mdrun”

- **Find the key for Python-2.7**
  - Set up your environment to permanently use Python-2.7
  - Check if the variables are correctly set by “which python”

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 YOUR JOB 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 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>

Note: ppn=16 for SuperMike2; ppn=20 for QueenBee2 and SuperMic
```

- **Add `-X` to enable X11 forwarding**

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!**

Exercise

- **Run an interactive job session for 30 min, using `nodes=1:ppn=16`**
  - Verify you are not on the head node by checking the # after hostname
  - Download the tarball from HPC website to the home directory
  - `$ tar -xvzf a.tar.gz` :untar it
  - `$ cd serialpi` : cd to the directory “serialpi”
  - `$ cp a.out /work/$USER` :Copy it to the work directory
  - `$ cd /work/$USER` : cd to your work directory
  - `$ ./a.out #####` :Execute a.out along with a number argument
- **At job runtime**
  - Open a new terminal and login to the head node
  - `$ qstat -u $USER` : Find out your current job id and status
  - `$ qstat -n jobid` : Quick look at the compute node assigned to you
  - `$ qshow -j jobid` : Check health of your job using qshow
  - `$ ssh mike/qb/smic###` :SSH to (your own) compute node
  - Now on the compute node, check
  - `$ free -m` Check free memory
  - `$ top` Monitoring job

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

- "**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            QUEUETIME

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            QUEUETIME
0 blocked jobs
Total jobs:  192
```

- **Command: `qfree`, `qfreeloni`**
  - Show the number of free, busy and queued

Take-home message

- **All words in red font are very important, CAPITALIZED ARE EVEN MORE IMPORTANT!**
- **What is HPC and how to evaluate the computational speed (FLOPS)**
- **Available HPC resources**
  - LSU HPC, LONI, XSEDE
  - SuperMike2, SuperMic, Queenbee, Philip, Eric
- **Account eligibility, allocation, password reset procedure.**
  - Need account sponsor (most likely a faculty) to support HPC account
  - It is free to run your simulation on cluster, but you need allocation
- **Infrastructure**
  - What is (1) node, (2) core (3) cluster (4) job
  - Where are you upon login to the cluster, where your job should be on
- **Practice on the cluster**
  - How to login via SSH
  - How to check your quota and balance
  - How to transfer files, add software by Softenv or Modules

Next Week Training

- **HPC User Environment 2, September 12**
  - More on job management and queue
  - Job submission and monitoring

- **Weekly trainings during regular semester**
  - Wednesdays “9:00am-11:00am” session, Frey 307 CSC

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

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

HPC@LSU User Services

- **Hardware resources**
  - Currently manages 4 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

Appendix
Storage Allocation on /project

- **Don't be confused with the computational allocation introduced in this training**
- **PI can apply for extra disk space on the /project volume for you and his/her entire research group 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**
- **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

Appendix
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
```

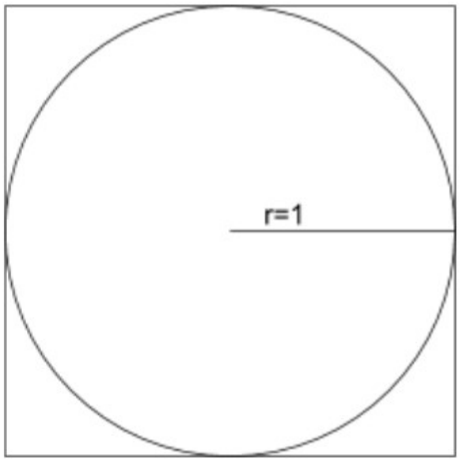
Exercise: Use Softenv

- **Find the key for gromacs-4.5.5**
  - Set up your environment to use gromacs-4.5.5 (one time change)
  - Check if the variables are correctly set by “which mdrun”

- **Find the key for Python-2.7.3**
  - Set up your environment to permanently use Python-2.7.3
  - Check if the variables are correctly set by “which python”

Appendix
Computing an approximate value for PI

- The a.out in this training calculates the value for PI based on the math which is actually quite simple: Imagine a square dartboard with circle inscribed within it such that the diameter of the circle is the length of a side of the square.



- We can observe that the ratio of the area of the circle to the area of the square is equal to some constant, π/4 (since the square's area is 2*2 = 4 and area_circle = π*r^2 = π). If we randomly place many points (darts) inside the square, we can count how many are also inside the circle (satisfy x^2+y^2 <= 1) vs the total number of points and compute an estimate for the value of π. (Problem description is from Jared Baker, UW; Ben Matthews, NCAR)