Welcome New Users!

Getting Started with HPC

Erin Shaw & Cesar Sul

Advanced Cyberinfrastructure

Research & Education Facilitation (ACI-REF)

USC Center for High-Performance Computing (HPC)

Spring 2019

1. What is HPC?

- HPC is USC's Center for High-Performance Computing.
  - Supports USC's research mission to advance scientific discovery.
  - Provides the infrastructure and support necessary for **research computing.**
- HPC is a world-class super-computing center!
  - As part of "standing up" an upgraded system, HPC runs and publishes standard performance benchmarks to Top500.org.
- HPC resources are available at no charge to USC faculty, staff, graduate students, and iVIP collaborators.
  - .

1. HPC Management

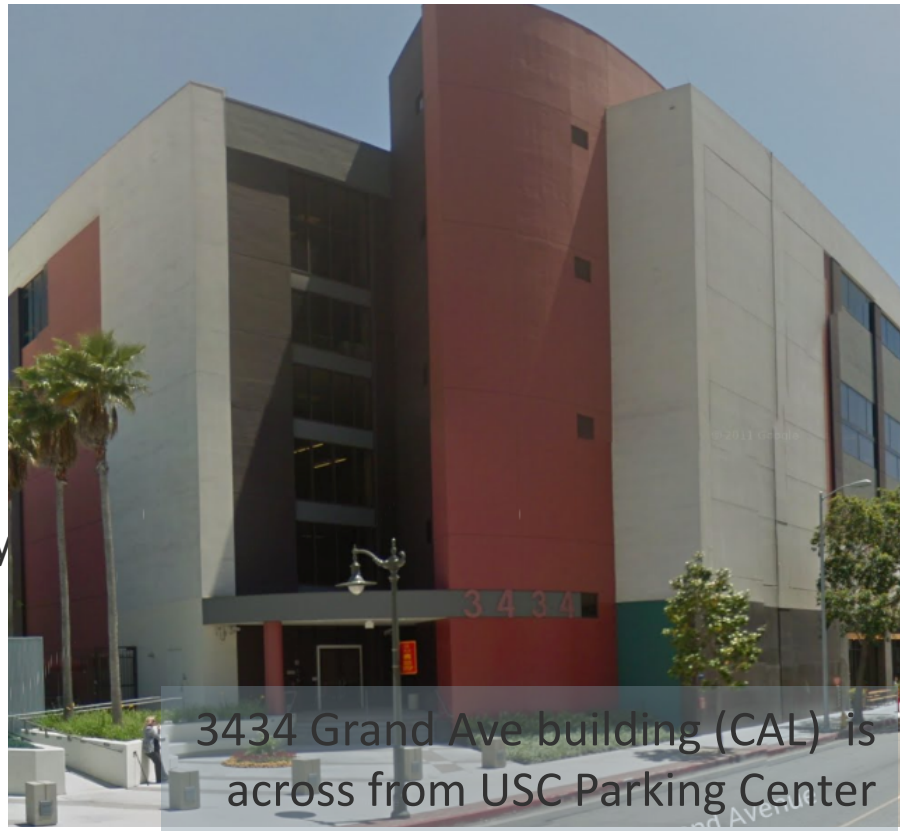
Douglas Shook
USC CIO

Randolph Hall
*USC VP of Research & HPC Faculty Executive Director**

HPC Director

*The HPC Faculty Advisory Committee advises the CIO about the faculty's research needs related to the university's HPC resources.

HPC is housed within the ITS data center and is monitored around-the-clock by ITS staff.

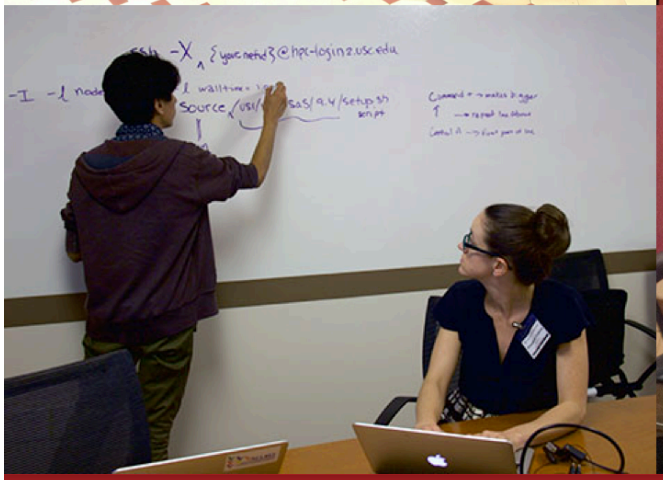


3434 Grand Ave building (CAL) is across from USC Parking Center

1. HPC Staff

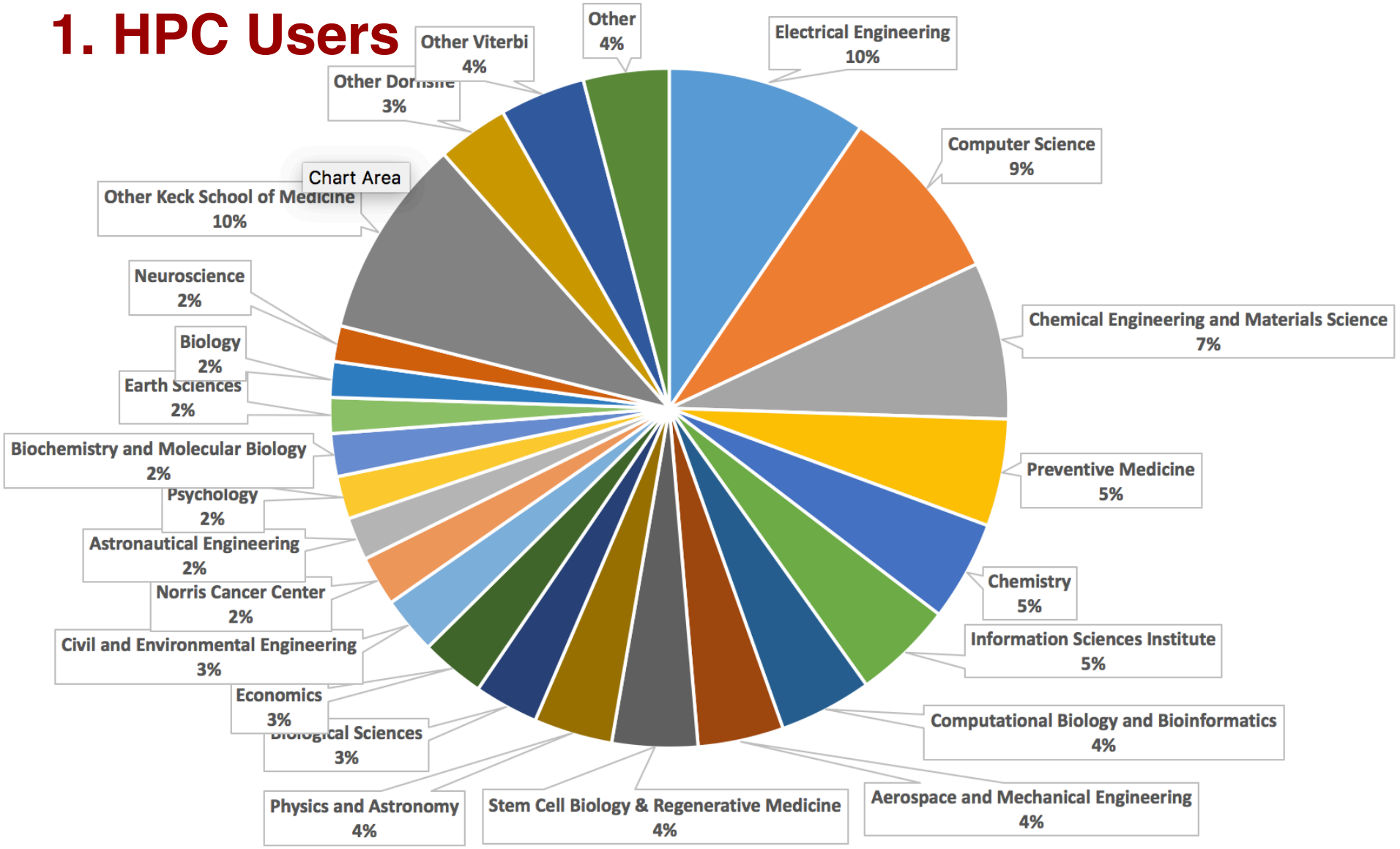

USC
USC University of Southern California


Artificial neurons
Artificial neurons




HPC

1. HPC Users



Our researchers are from all over campus. Historically they were from Dornsife, Viterbi, and Keck. Increasingly from Business, Psychology, Pharmacy and International Relations

2. HPC Accounts

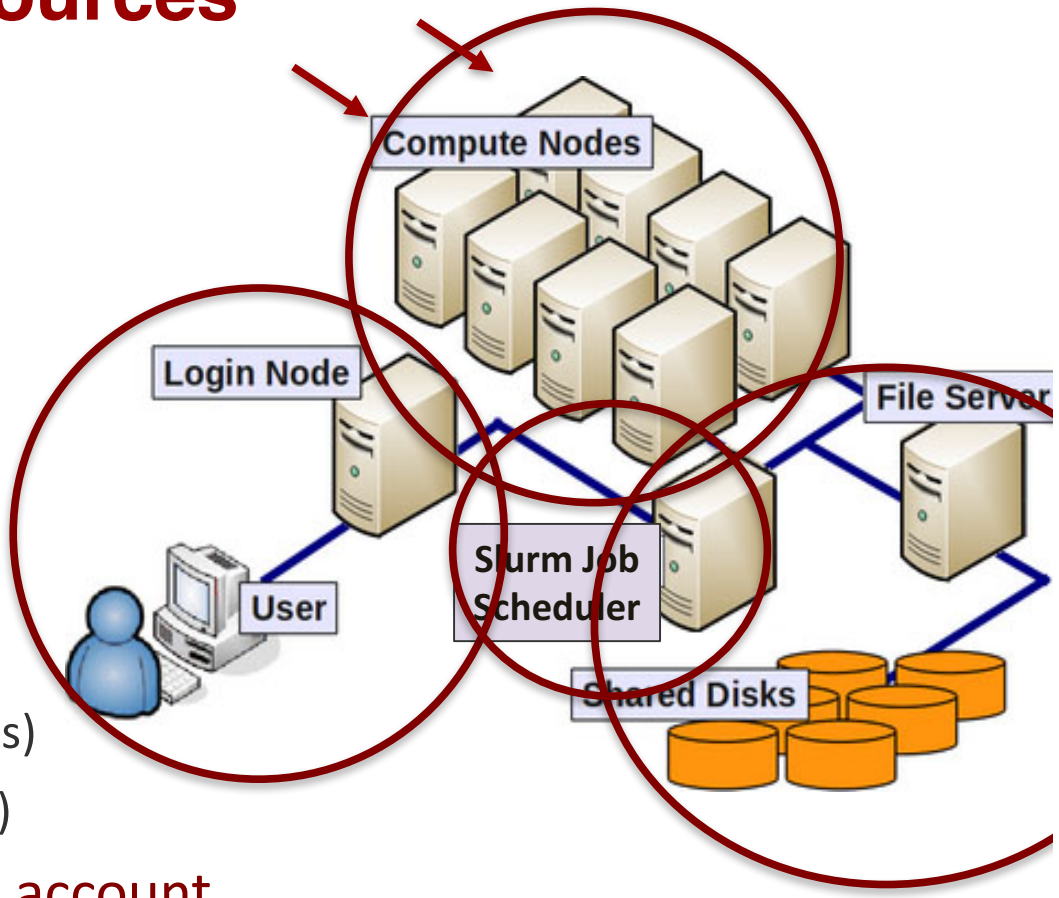
- Accounts are project-based
  - Project “PI”s “own” accounts
  - A PI can add members or be the sole member
  - Members can belong to multiple projects
- Projects are allocated quotas for files, disk space and ~~core hours~~
  - PI can request increases on **project website**
    - *https://hpc-web.usc.edu/project*
  - Project members **share** quotas
  - To monitor quotas:
    - *$ myquota #disk space & number of files*
    - ~~*$ mybalance -h #cpu/core hours (now infinite)*~~

2. HPC Special Accounts

- Class accounts
  - Instructors can create class accounts for their students, for teaching and class assignments
- HPC secure data accounts (HSDA)
  - An HSDA is required if you wish to store or process legally protected or high-risk information on HPC
  - There is a special application process for HSDA
- See “Accounts” under hpcc.usc.edu for more information

3. HPC Computing Resources

- Head nodes (Login nodes)
  - e.g., hpc-login3.usc.edu
- Compute nodes
  - ~2700 nodes running CentOS 7
- Networked storage
  - Disk arrays and file servers
- Two low-latency networks
  - Infiniband 56-gbps (1,390 nodes)
  - Myricom 10-gbps (1,300 nodes)
- SLURM resource scheduler and account manager



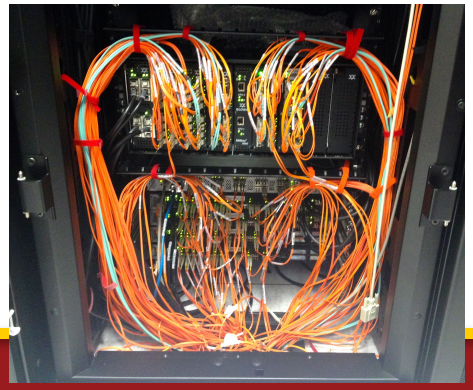
3. HPC Computing Cluster

A simple, home-built cluster

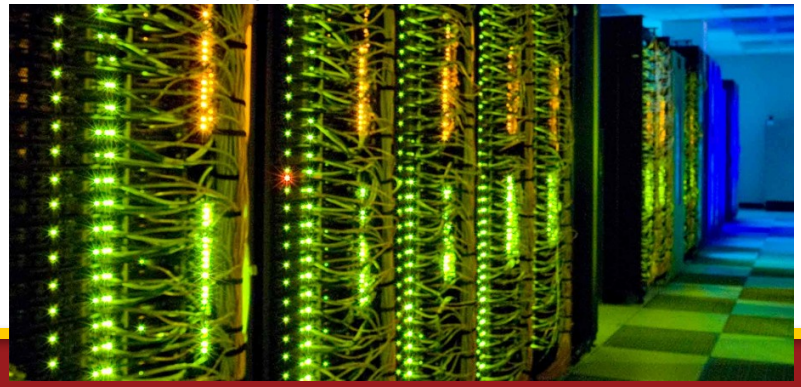
One rack in HPC cluster

Rows of racks in HPC cluster!

Network cables



Multiple racks in HPC cluster



3. HPC Computing Cluster

- There are over 1300 nodes on newer cluster
  - 264 Hewlett-Packard SL250, dual Xeon 8-core 2.6GHz, dual NVIDIA K20 GPUs containing 2,496 cores, each with 64GB memory
  - 448 Hewlett-Packard SL230, dual Xeon 8-core 2.6GHz CPUs, with 64GB memory
  - 288 Lenovo nx360m5 dual Xeon 8-core 2.6GHz CPUs with 64GB memory
  - 19 Lenovo nx360m5 2.6GHz dual NVIDIA K40 GPUs containing 2,880 cores, each with 64GB memory
- Recently installed
  - 11/18, Added 120 dual 12-core, 96G (83 condo) and c34 dual 12-core V100 GPU, 192G memory
  - 3/18, Added 132 new x86, DELL R430 compute nodes (decommissioned 120)
  - 10/17, Added 64 GPU nodes and 54 non-GPU nodes (all condo'd)
- Next upgrade on 4/2019

3. HPC Use Policies

- Downtime policy
  - Every fall and often also in spring, HPC installs new nodes, upgrades, & patches.
  - All HPC resources are unavailable during these times.
    - *No, we cannot make exceptions*
    - *Please plan ahead!*
- Fair share use policies
  - Limited use of shared head nodes.
    - *Move computation to compute nodes*
  - Limited use of compute nodes:
    - *Max walltime per job = 24 hours on general nodes*
      (Max walltime is 300 hours on a few specific nodes)
    - *Max nodes per job = 99 nodes*
    - *Max jobs queued = 10*

3. HPC Computing Cluster

Network Firewall

Head* Nodes

hpc-login2

hpc-login3

$ ssh NetID@hpc-login3.usc.edu

Data Transfer Node*

hpc-transfer

$ ssh NetID@hpc-transfer.usc.edu

Compute Nodes (CentOS)

>2,700 nodes, >32K cores

Data Storage

2.4 PB (total)

328 TB (/staging)

Fast Networks

Myrinet (10Gbit/sec)

Infiniband (56Gbit/sec)

***Only head nodes can access the Internet. *Head nodes and DTN are shared by all users**

Connecting to the Cluster

4. To Work Remotely on HPC

- A *secure network* is required
  - Use USC Secure Wireless or USC Ethernet to connect from USC
  - Use a Virtual Private Network (VPN) client to connect from outside USC
- A *secure shell* (ssh) is required
  - On Macs, use Terminal, a native ssh client
    - *Install XQuartz (www.xquartz.com) for viewing graphics displays*
  - On Windows, install PuTTY, X-Win32 (software.usc.edu), etc.
- DUO is now required!
  - See USC/ITS page on how to set up.

*A shell is Linux's command line interface

4. To Work Remotely on HPC

- To connect from a terminal window (Mac)
  - **ssh <your_USC_NetId>@hpc-login3.usc.edu**
  - Authenticate via DUO
- To connect from a 3rd party client application
  - Set the SSH hostname to hpc-login3.usc.edu
  - Connect or launch, etc.
  - If asked, accept/trust the HPC machine
  - Authenticate via DUO
    - *X-Win32 has been tested with DUO*

*A shell is Linux's command line interface

```
Erins-MacBook-Pro-3:~ erinshaw$ ssh erinshaw@hpc-login3.usc.edu

Enter passphrase for key '/Users/erinshaw/.ssh/id_rsa':
Last login: Wed Jan 17 11:56:12 2018 from usc-secure-wireless-088-116.usc.edu
*******************************************************************************
      Thursday, November 9, 2017
      All users of this computer system acknowledge that activities on it
      may be subject to monitoring;  the privacy of activities on this
      system cannot be ensured.  All computer account users are required
      to read and abide by the ITS Computing and Usage Policies.  Please
      refer to the web page at:  https://policy.usc.edu/technology/
*******************************************************************************

[erinshaw@hpc-login3 ~]$ pwd
/auto/rcf-40/erinshaw

[erinshaw@hpc-login3 ~]$ myquota
-------------------------------------------
Disk Quota for /home/rcf-40/erinshaw ID 22418

             Used     Soft    Hard
     Files   10672    100000  101000
     Bytes   692.21M  1.00G   1.00G
-------------------------------------------
Disk Quota for /home/rcf-proj2/ess ID 735

             Used     Soft     Hard
     Files   725663   1000000  1001000
     Bytes   119.23G  1.00T    1.02T
-------------------------------------------

[erinshaw@hpc-login3 ~]$ cd /home/rcf-proj2/ess

[erinshaw@hpc-login3 ess]$ pwd
/auto/rcf-proj2/ess
```

4. From desk/laptop: To transfer files to/from HPC

- Directly transfer files to/from HPC on a local computer.
  - Use Linux commands scp and rsync from a terminal window.
- User a graphical *secure file transfer protocol* (sftp) client.
  - Install one of the many sftp client applications available:
    - *Fetch, Filezilla (uncheck Yahoo box!), etc.*
  - Use host hpc-transfer.usc.edu
    - *hpc-transfer is a dedicated DTN (data transfer node)*

4. From HPC: To move files onto HPC

- You can move files TO HPC from the public Internet
  - Only works from a head node

    (compute nodes cannot connect to Internet)
- Login to hpc-transfer.usc.edu
  - Use wget, git clone and curl to transfer files
  - Examples:

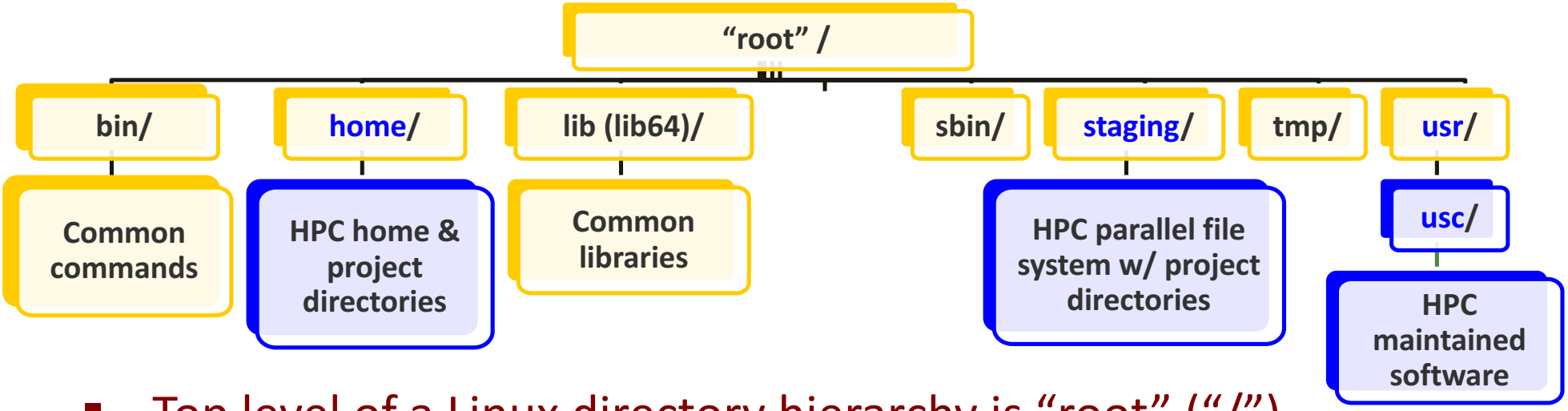
    $ wget http://software.usc.edu/app.tar.gz

    $ git clone https://github.io/myrepo/myrepo.git

HPC File System

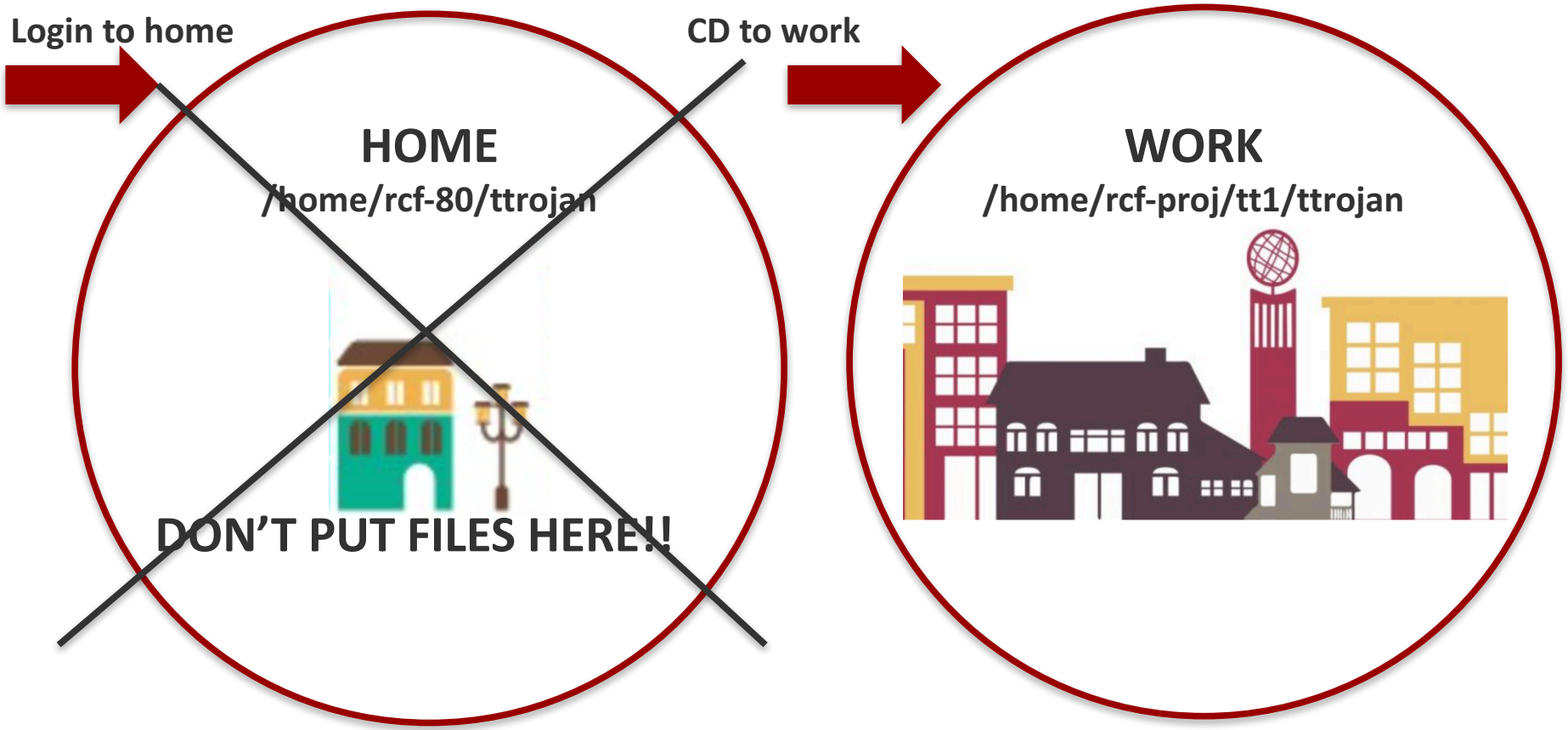
5. HPC File System

- HPC nodes run CentOS 7, an open source version of Red Hat Enterprise Linux (RHEL).



- Top level of a Linux directory hierarchy is "root" ("/")
  - Your data files are located under /home/ and /staging/.
  - HPC maintained software is located under /usr/usc/.

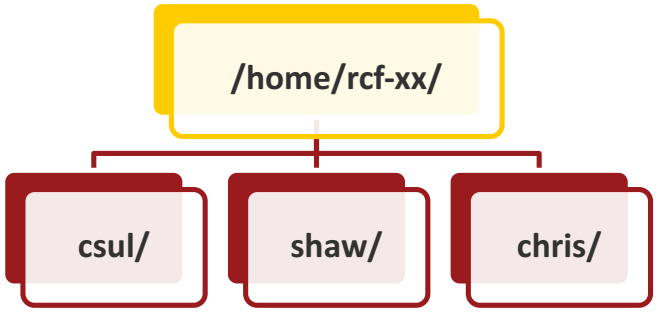
5. HPC File System



5. Home Directory (private)

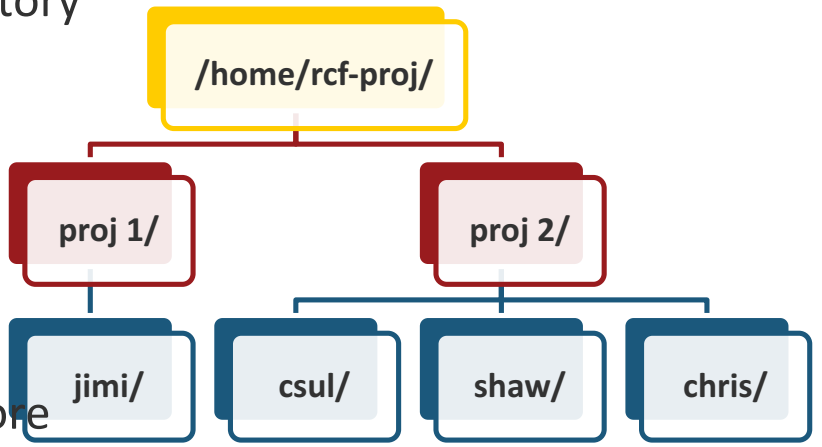
- Users login to their home directory
  - Personal, private directory
  - Only user can modify files here
  - Backed up daily
- User quotas
  - 1 GB of disk quota & 100,000 files
    - *Warning: applications will install hidden files here! (use **ls –a** to view)*
- Used for
  - Logging in, setting up environment

/home/rcf-40/<user_name>



5. Project Directory (shared)

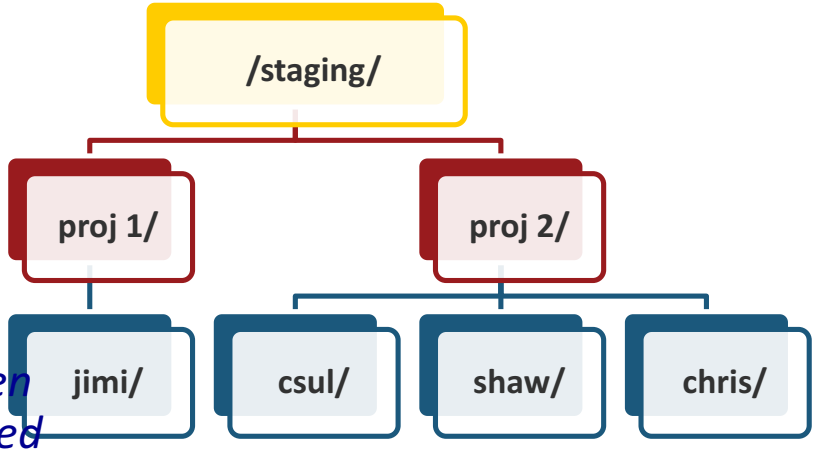
- Every project has its own directory **/home/rcf-proj/<project_name>**
  - PI is owner, each member has a subdirectory
  - Backed up daily
- User quotas
  - 5TB per project, shared among members
- Used for
  - Installing software, running jobs, data store
- Permissions
  - By default, member directories are group-readable
    - *Members can make private, never set permission so others can write*
    - *Only PI can create shared subdirectories and install software at top level of project directory*



5. Staging Directory (328TB, short term)

- Every project has a staging directory
  - Same structure as project directory
  - On parallel file system
- User quotas
  - 10 TB per person, for 2 weeks
  - Data is not backed up!
    - *Store original data somewhere else*
    - *Data cleared during downtimes, and when data idle for 2 weeks or capacity is reached*
- Used for
  - Staging data for jobs, i.e., copying data and results to/from program
  - Optimized for concurrent file access, w faster r/w access rates than /home file system

/staging/<project_name>



5. Local Storage (on node, only while running jobs)

- Every job has access to local storage on compute node

  $TMPDIR

  - *Equal to /tmp/{your_job_id}*
  - *Fastest r/w rates*
  - *60GB-1.8TB capacity, depending on node*

- Multi-node jobs can use shared storage

  $SCRATCHDIR

  - *Equal to /staging/scratch/{your_job_id}*
  - *Same r/w rates as /staging*

- **All job files cleared automatically at end of job!**
  - Copy results back to more permanent storage area

Working on the Cluster

6. Using the compute cluster

- HPC's cluster consists of about 2700 compute nodes
  - Must be managed to improve utilization.
- HPC uses *Slurm*, a resource manager that:
  - Performs job queuing, scheduling, and execution
  - Performs node allocation and clean up
  - Monitors memory usage and communication

- What is a job???
  - What you "run" on HPC
    - *Slurm resource specifications*
    - *Programs and data*
    - *Linux shell commands that will be executed remotely to run a program*
  - Put in text file and submit
    - *$sbatch myjob.slurm*

```
#!/bin/bash
#SBATCH --ntasks=1
#SBATCH –mem-per-cpu=1G
#SBATCH --time=01:00:00

#move to your working directory
cd /home/rcf-proj/{myproj}/{myworkdir}

#set path, environment variables
source /usr/usc/sas/default/setup.sh

#run program
sas my.sas
```

6. Using the compute cluster

- Slurm (Simple Linux utility for resource management)
  - Slurm job control commands:

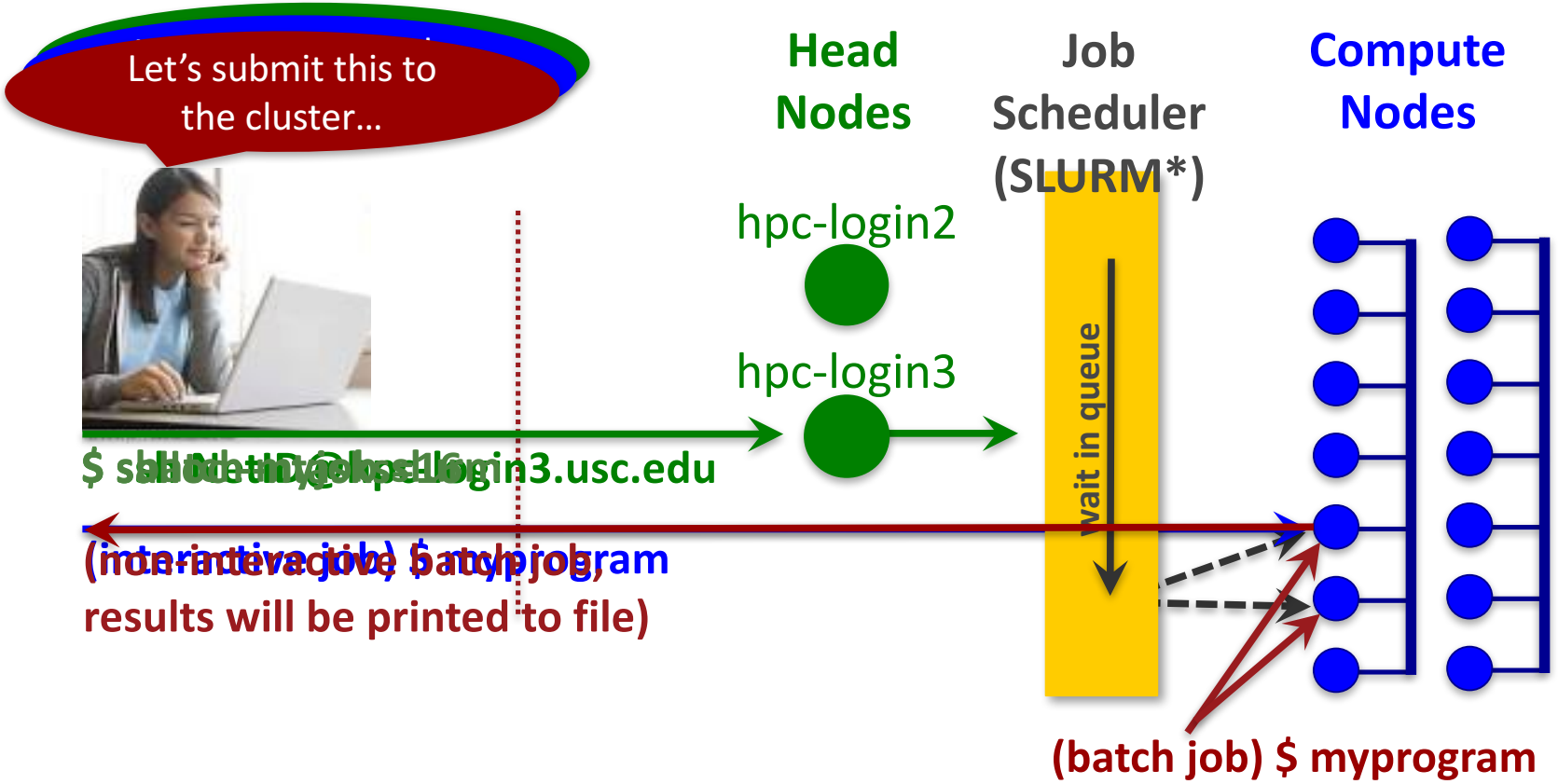
sbatch	submit a batch job
salloc	run an interactive job
scancel	delete a job

  - Slurm job monitoring commands:

squeue -u <uname>	display your queued jobs
scancel -j <jobid> -- start	display estimated start time
scontrol show job <jobid>	display detailed job statistics

  - More SLURM documentation at https://slurm.schedmd.com/

6. Running jobs on the cluster



*HPC uses the SLURM job scheduler and resource manager. Jobs are scheduled based on order submitted, number & types of nodes requested and time required.

6. Test your job interactively

- Slurm has a special job submission mode that allows you to access computing resources interactively
  - Example: 1 processor/cpu/core for one hour

```
$ salloc --ntasks=1 --time=1:00:00
```

  - You can test your program until the requested time expires

```
$ source /usr/usc/hello_usc/3.0/setup.sh
$ hello_usc
```

  - Extremely useful for compiling/debugging/testing your code and preparing job scripts

6. Submit a batch job

- Use a job script to submit a batch job to the cluster

1. Add SLURM computing resource requests
2. Add the commands you need to run your program
3. Submit your job to the queue by typing:

   $ sbatch myjob.slurm

Example job script: myjob.slurm

```
#!/bin/bash
#SBATCH –export=none
#SBATCH --ntasks=1
#SBATCH –mem-per-cpu=1G
#SBATCH --time=01:00:00

#move to your working directory
cd /home/rcf-proj/{myproj}/{myworkdir}

#set path, environment variables
source /usr/usc/sas/default/setup.sh

#run program
sas my.sas
```

6. Running on HPC: Easy as 1-2-3!

1. Plan ahead

- Organize project directories
- Install software, transfer data
- Think about your job requirements
  - amount of memory
  - number of i/o files
  - size of data

2. Test interactively

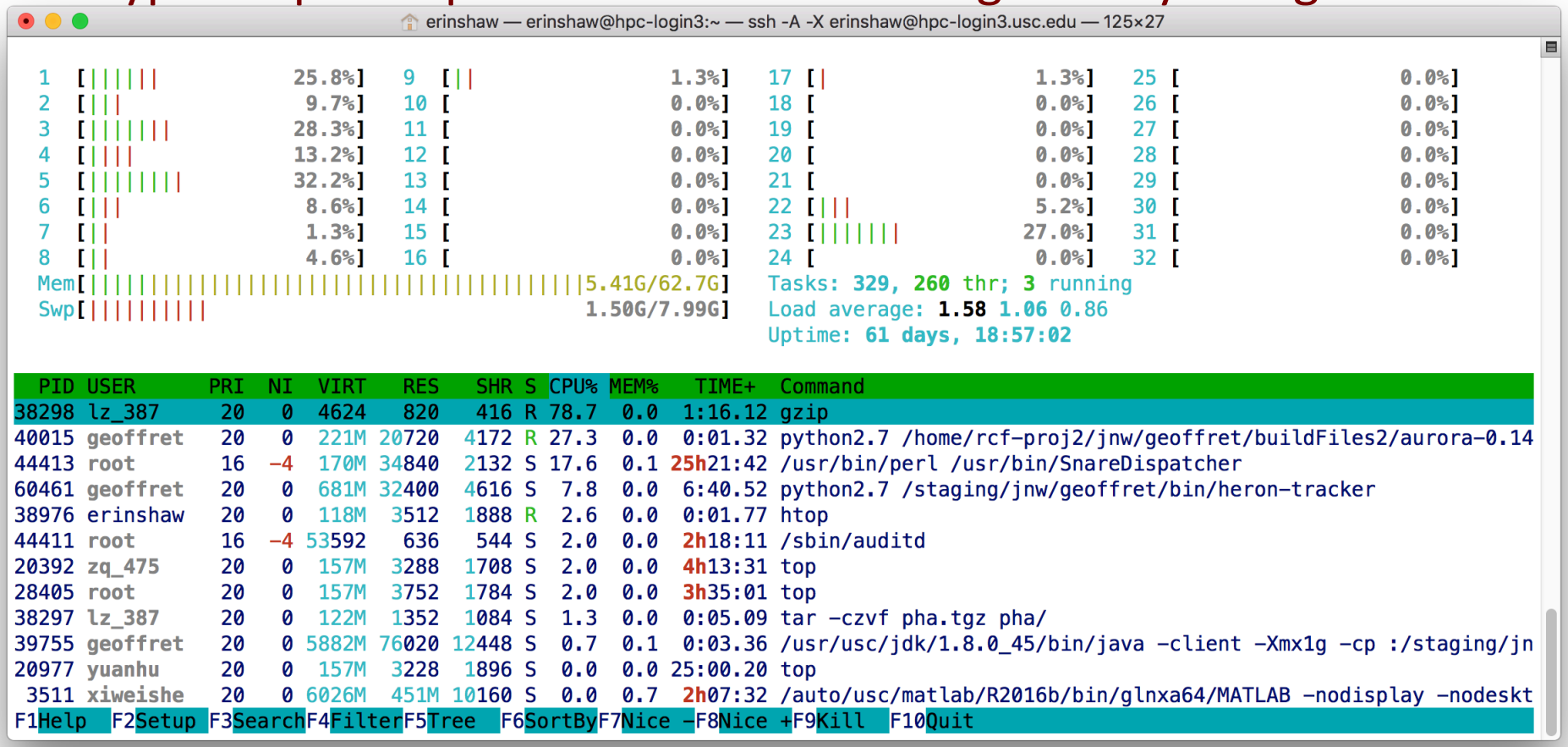
- Use salloc/srun
  - test run commands
  - check paths, results
- Experiment with resource allocation
  - number cores (cpus)
  - amount of memory
- Start small!
  - *then* scale

3. Run remotely

- Create Slurm script
  e.g., myjob.slurm
- Submit job to queue
  $sbatch myjob.slurm
- Monitor your job
  $squeue --user <uname>
- Check results
  $less slurm-<jobid>.out

6. Monitoring Processes

- Type htop or top to view what's running when you login.



7. Installed Software

- HPC maintains software in /usr/usc/
  - Includes compilers, statistical, mathematical, and simulation programs; numerical libraries, licensed applications and more…

```
$ ls /usr/usc
acml/          fftw/          imp/           mpich2/      qespresso/
amber/         gaussian/      intel/         mpich-mx/    qiime/
aspera/        gflags/        iperf/         mvapich2/    R/
bbcp/          git/           java@          NAMD/        root/
bin/           globus/        jdk/           ncview/      sas/
(many more)
```

- You can also install software in your project directory
  - HPC can help with this

7. Installed Software

- HPC may have multiple versions of each application

```
$ ls -F /usr/usc/python
2.7.6/     2.7.8/     3.3.3/     3.4.3/     3.4.5/     3.5.2/     3.6.0/     default@

$ ls -l /usr/usc/python
drwxr-xr-x  7 root    root 4096 Sep 12  2017 2.7.8
. . . . .
drwxrwxr-x  6 root   root 4096 Oct 26  2016 3.5.2
drwxrwxr-x  6 root   root 4096 Aug 21 2018 3.6.0
lrwxrwxrwx  1 root   root      5 Apr  9   2018 default -> 3.6.0

$ ls -F /usr/usc/python/default/
bin/  include/  lib/  man/  setup.csh  setup.sh  share/
```

=> **default@** is a symbolic link to the most recently tested version

7. Installed Software

- The HPC script **setup.sh** is added to all installed software.

```
$ ls -F /usr/usc/python/3.6.0/
bin/  include/  lib/  man/  setup.csh  setup.sh  share/
```

  - When run, **setup.sh** will *setup (configure)* your shell environment so you can access the proper version of software

- To run setup.sh, use the **source** command:

```
$ source /usr/usc/python/3.6.0/setup.sh
```

Policies
Getting Help

8. HPC Policies

- Required reading
  - https://hpcc.usc.edu/support/accounts/hpcc-policies/
- Resource limits
  - On head nodes, jobs (24 hours), allocations (core hours, disk space),
- Scheduled downtimes
  - Twice a year, for upgrades and maintenance
- Protected data allowed within HPC Secure Data Accounts (only)
  - HPC is now HIPAA-compliant
- Play well with others, practice safe computing
  - i.e., share public nodes, not private data!

8. USC Policies

- Recommended reading
  - It is your responsibility to abide by these
- Information Security
  - http://policy.usc.edu/info-security/
- Network Infrastructure Use
  - http://policy.usc.edu/network-infrastructure/
- Privacy of Personal Information
  - http://policy.usc.edu/info-privacy/
- Digital Millennium Copyright Act Compliance
  - http://cio.usc.edu/copyright/policy/

9. Getting Help

- Request assistance
  - Email hpc@usc.edu (email again!)
  - Office Hours (drop-in)
    - *Every Tuesday@2:30pm (UPC LVL 3M)*
    - *1st Thursday month @Noon (HSC NML 203)*
    - *3rd Thursday month @Noon (HSC SSB 106)*
- Learn more!
  - Visit https://hpcc.usc.edu
  - Request a consultation (anytime)
  - Attend a Workshop (when scheduled)

9. Getting Help

"My program doesn't run on HPC."

- **Describing Your HPC Problem**
  - When you send HPC staff an email or submit a request through our ticketing system, it's important to provide the detailed information that will help us understand your problem and find the right solutions.
- If you are **trying to run a job**, please include the following information, if appropriate:
  - Job ID
  - Slurm script (or interactive commands)
  - Error messages
  - Slurm output or error files
  - Screenshots
  - Portions of your code, for example, any libraries you install or import

9. Getting help

- Finally, please **tell us if there are inconsistencies**, such as:
  - Did some jobs run on some nodes but not on others?
  - Is the behavior reproducible or erratic?
  - Has the behavior you're experiencing changed recently? When?
- If you are trying to **use or install software**, please include:
  - Name and version of the software
  - Commands that give you an error, and error messages you receive
  - A link to the website or github repo you installed the software from
  - A link to the installation directions

10. Linux Commands & Concepts

Environment

bash shell, .bashrc

environment variables ($PATH)

Keyboard navigation

<up>/<down> : show prev/next cmd

<ctl-a>/<ctl-e> : mv to beg/end of line

<alt-f>/<alt-b> : mv forwd/back a word

<tab>, <tab-tab>: autocomplete

Special characters

"*": wild card

"/", "~" : root dir, home dir

".", ".." : current dir, parent dir

">", "<" : redirect output/input

"|" : pipe output to input of next cmd

Navigation

$ pwd

$ cd

$ ls (-alh)

$ cp/mv

$ touch/ rm (-i)

$ mkdir/rmdir

$ history

Reading/Editing files

$ cat, $ less

$ nano ($vi, $emacs)

Permissions

$ chmod

$ chgrp

Information

$ mybalance -h

$ myquota

$ top

$ du –h

$ man

$ echo [$PATH]

$ wc

$ sort

Tools

$ alias

$ wget

$ for (loop)

10. Linux Commands & Concepts (applied)

$ mybalance -h
$ myquota
$ top
$ pwd
$ ls, ls -l, ls -F --color
$ man ls
$ echo hello
$ ↑↑↑↓↓↓
$ echo $PATH
$ cd .., pwd, ls

$ cd /home/rcf-proj/<myproj>/<mydir>
$ mkdir test, ls, rmdir test, ls
$ mkdir workshop, ls
$ cd workshop, pwd
$ touch a.a, ls
$ cp a.a b.b, ls
$ mv b.b c.c, ls
$ rm c.c, ls
$ alias rm='rm -i'
$ rm a.a, ls

10. Linux Commands & Concepts (applied)

$ alias cdp='cd /path/to/proj'
$ cd ~, pwd
$ cdp, pwd
$ cd ~, ls –alh .bash*
$ cat .bashrc
$ cp .bashrc .bashrc_ori
$ nano ~/.bashrc

alias rm='rm-i'
alias cdp='cd /path/to/proj'
alias ll='ls -hlt'

$ cdp
$ ll workshop
$ chmod g+w workshop, ls -l
$ chmod g-rw workshop, ls –l
$ cd workshop
$ wget http://hpcc.usc.edu, ls
$ wc -l index.html
$ cat index.html | grep @
$ cat index.html | grep jpg

10. Linux Commands & Concepts (applied)

```
$ history
$ history >> history.out
$ less history.out
$ less history.out | grep wget
$ ls -t /usr/usc/ > ls.out
$ less ls.out
$ less ls.out | sort
$ less ls.out | sort -f | head –n 5
$ du –h * | sort –n
$ du –h --summarize
```

```
$ ls /usr/usc/mat
    <tab> to autocomplete (fails)
    <tab><tab> to show candidates
$ ls /usr/usc/matl
    <tab> to autocomplete (succeeds)
    <tab><tab> to show candidates
$ ls /usr/usc/matlab/2
        :

$ ls *
$ for i in *; do echo $i; done
```

10. Linux References

- HPC workshop
  - Introduction to Linux, SLURM, and the HPC Cluster
- Lynda video (access via USC)
  - https://www.lynda.com/Linux-tutorials/Learn-Linux-Command-Line-Basics/435539-2.html
- Software Carpentry tutorial
  - http://swcarpentry.github.io/shell-novice/
- O'Reilly Books directory
  - http://www.linuxdevcenter.com/cmd/
- Many many web sites… use search

Appendix I – DDDT*

- Don’t share your password
  - Goes without saying, and yet….
  - Your HPC password is your USC Shibboleth password, used for everything!
- Don’t set permissions so others can write to your directory
  - Makes it easy for others to overwrite and delete your files
  - Create a group-shared directory for your group, instead

Appendix II – "HPC is not your super laptop"

- Don't run compute intensive jobs on head nodes
  - Use a compute node. Everyone is watching ($top).
  - Headnodes are shared resources, compute nodes are there for you..
- Don't read/write/copy zillions of tiny files
  - "Stressful" for the filesystem
  - Use a database (lmdb, mysql) or a large file to combine data.
- Scale appropriately
  - Do you really need 99 nodes?
  - Think about what resources your program needs
- Google gets you surprisingly far
  - It's what we do!