Introduction to HPC3

Nadya Williams

npw@uci.edu

Agenda

1. What are the differences between HPC3 and HPC
2. Basics of Linux and Shell necessary for running jobs on HPC
3. How to search for modules/programs on HPC? What are the differences between modules and programs? How to download modules/programs? How to set environment?
4. How to run jobs on HPC? What queues are available? What are the common problems?

Some background for HPC and HPC3

- HPC and GreenPlanet catalyzed shared computing at UCI
- HPC survey indicated
  - importance to faculty for research
  - overall utility
  - *room for improvement*
- Scalability of HPC has reached limitations in terms of queues access and the OS (operating system) life expectancy.

  HPC is active till the end of 2020
- Recent MRI (NFS Major Research Infrastructure) award coupled with UCI Campus investment provided an opportunity to adjust shared computing and improve upon the existing cluster via a new HPC3 cluster

HPC3 - Goals

1. Enables users to have access to a larger compute/analysis system than they could reasonably afford "on their own"
2. Enables access to specialized nodes (large memory, 64bit GPU)
3. Fosters a growing community across UCI to utilize scalable computing (HPC and HTC)* for their scientific research program and teaching
4. Provides a well-managed software environment that forms the basis of a *reproducible* and more secure research environment

** HPC – High-Performance Computing*
HTC – High-Throughput Computing

HPC3 accounting: jobs draw from an accounting bank

Accounted jobs vs. free jobs

- Accounted – once a job is started, it cannot be killed or pre-empted
- Free – a free (non-accounted) can be killed at anytime

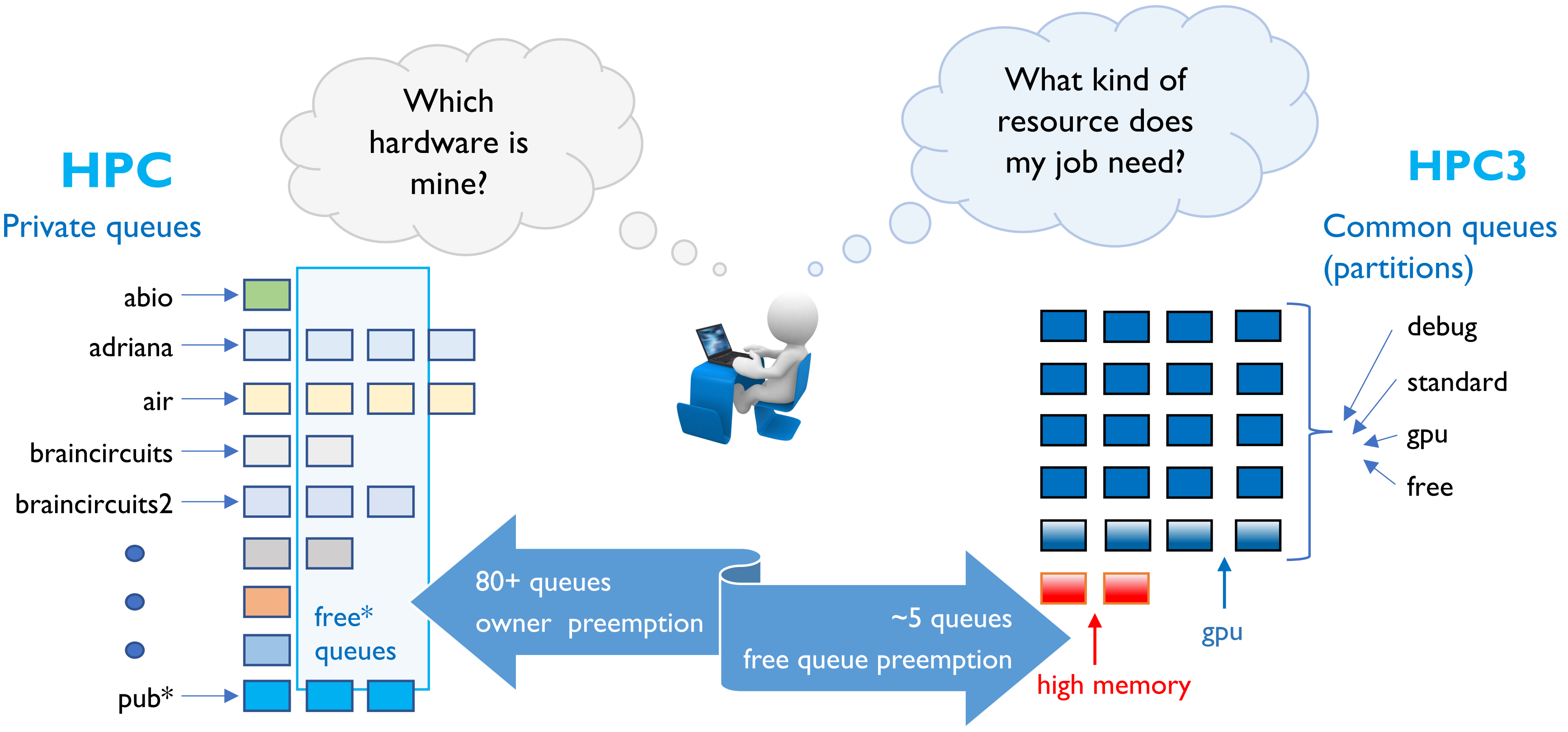
Three ways of filling your account

- Granted cycles: UCI core funds purchase hardware to provide enough resource to support granted cycles
- Converted: condo-style, researchers purchase hardware, RCIC manages. nodes capability is converted to core-hours. Formula:

  Physical hardware can deliver N-core-hours/year. **0.95N** are deposited into an owners account each year the owner has a node (or nodes) in the cluster.
- Purchased: Hours are pre-purchased (~$.0125/core-hour) in chunks (e.g., $100 increments buys ~ 8000 core-hours)

https://rcic.uci.edu/hpc3/index.html
https://rcic.uci.edu/hpc3/hpc3-reference.html

Queueing in HPC vs. HPC3



Major Differences Between HPC and HPC3

HPC	HPC3
Node owners can kill free jobs on their nodes	Only free jobs can be killed on any node
Most users have access to a small number of queues	Users have access to nearly all queues
Only free jobs can span owner and non-owner queues	All jobs can span nodes as needed
Users can purchase hardware	• Users can be granted core-hours • Users can purchase core-hours • Users can purchase hardware, but HPC3 steering group defines supported hardware configurations
Operating system: CentOS 6	Operating system: CentOS 7
SGE job scheduler	SLURM job scheduler
BLCR checkpointing for some jobs	NO checkpointing

No oversubscription + Fair queueing

No oversubscription

If you own X% of the total cluster, your starting account balance is ~ X%
of the total number of hours that can be delivered in a year by the entire cluster.

Fair Queuing

Non-FIFO. Jobs arriving earlier in the queue are not guaranteed to schedule first.
Want to prevent *large number of jobs from user A* blocking *a small number of jobs from user B*

Bias towards *interactive* turnaround for small debugging jobs

Optimize people time for the *debug* process.
Small core count + *short time duration* jobs should schedule as quickly as possible

Fair running

If you are running an accounted job, once your job is started, it will not be pre-empted/killed

Free cycles

Users who *pick up spare cycles* == *run free jobs* can have their jobs killed
so that accounted jobs can run as soon as possible

Moving from HPC to HPC3

HPC end of life ~end of 2020

- All existing accounts will be transferred to HPC3
- Currently, during the **HPC3 Production Ramp Up** we move groups/labs
  See https://rcic.uci.edu/news/content/

Your $HOME on HPC is NOT moving to HPC3, this means

- You transfer IMPORTANT files from HPC $HOME to dfsX or CRSP area

Your files on any of dfs3/dfs4/dfs5 or CRSP are available on HPC3

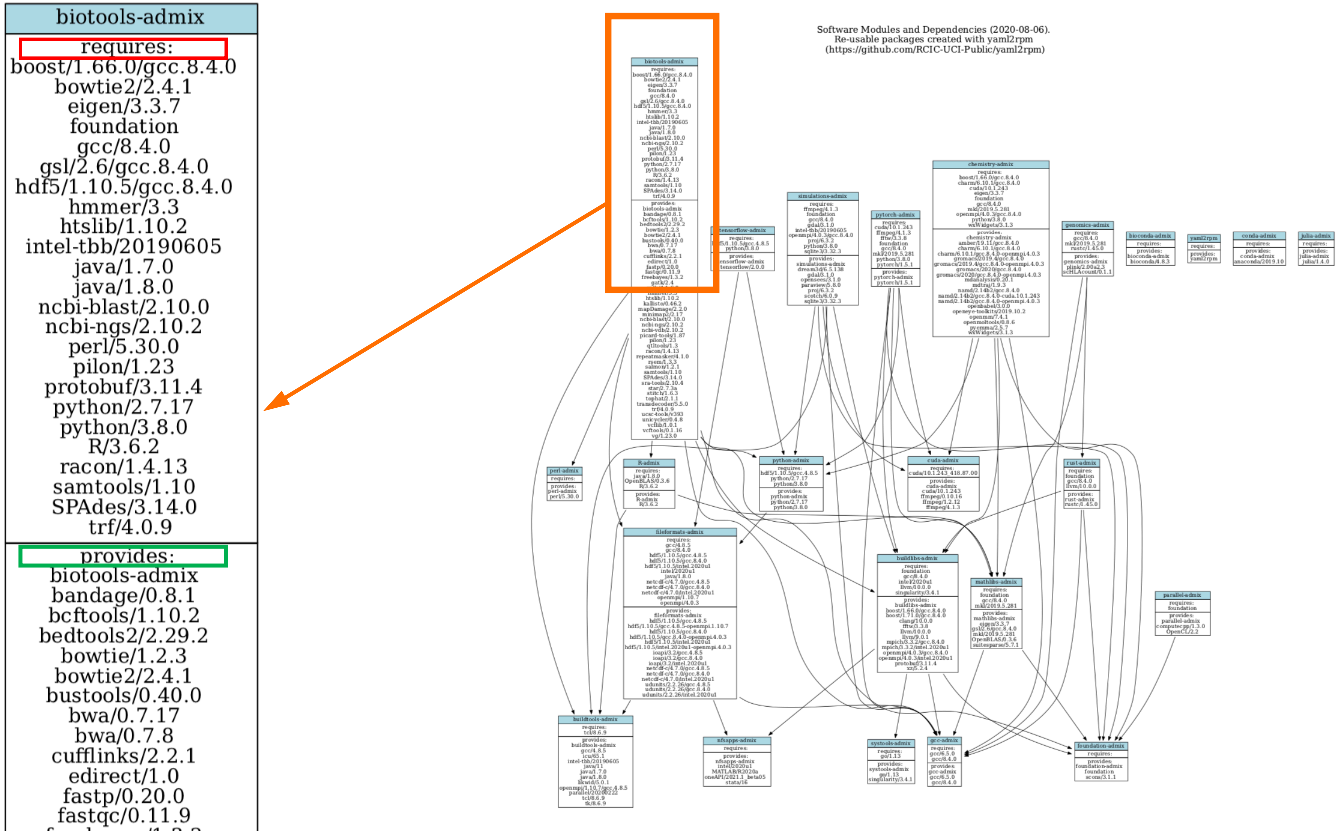
Most of the software will be available on HPC3

- *See Software Map* on https://rcic.uci.edu/hpc3/software-tutorial.html
- If you have your own compiled software, will need to recompile on HPC3

Software Map

See A software Map on https://rcic.uci.edu/hpc3/software-tutorial.html

use modules



How to request software

- Some software is not possible to move from HPC to HPC3
  - Too old for the new OS
  - Versions will be different because of dependencies, new OS.
- Some packages you will need to install yourself
  - R (we have ~350 packages)
  - Perl (we have ~200 packages)
  - Python (we have ~60 packages)
  - Conda (we provide bioconda for PacBio tools and anaconda for python3)
- See https://rcic.uci.edu/hpc3/getting-help.html#askforsoftware guide
  - Request what you really need, we honor group requests
  - Experiment on installing your own with R/Python/Perl/Conda
  - Required elements of software request

Basics of being a good citizen on a cluster

1. Cluster is a shared resource, it is NOT your personal machine
2. What you do affects all the other users, so think before you hit that *Enter* key
   - Do not run interactive jobs on login nodes
   - Do not transfer data on login nodes
3. Secured from mischief and disasters.
   - We restrict users' ability (permissions) to install and run unwanted software applications
   - It is your responsibility to act secure
   - Be careful when bringing applications from unknown sources. DO NOT ask for sudo access
4. For your jobs: use resources you need, don't ask for more
   Study this Slurm guide https://rcic.uci.edu/hpc3/slurm.html
5. Be mindful how you submit tickets
   https://rcic.uci.edu/hpc3/getting-help.html#_how_to_ask_for_help

What makes a bad ticket

1. I am submitting my job with my job script and I think there is something missing in my script and I am unable to find it. Can you look at it?

    1. What is submit script? 2. How job is submitted? 3. What error did you get?

2. When I am running the jobs from the model there is an issue of 'libnetcdff' and I am unable to fix this as well. The path where I am running the job is '/dfs3/pub/userX/PROGY/test2'. More details are there in the screenshot below.

    1. What is submit script ? 2. Screenshot has no info on the cause of error

3. I am unable to access HPC. My connection gets closed on login. Please refer to the image below.

    1. Screen shot has only partial info

4. I need to run a program ThisGreatProgram. Can you install it please. It's commonly used in bioinformatics field so maybe it's better to install it as a public module. The instructions are sudo apt-get …

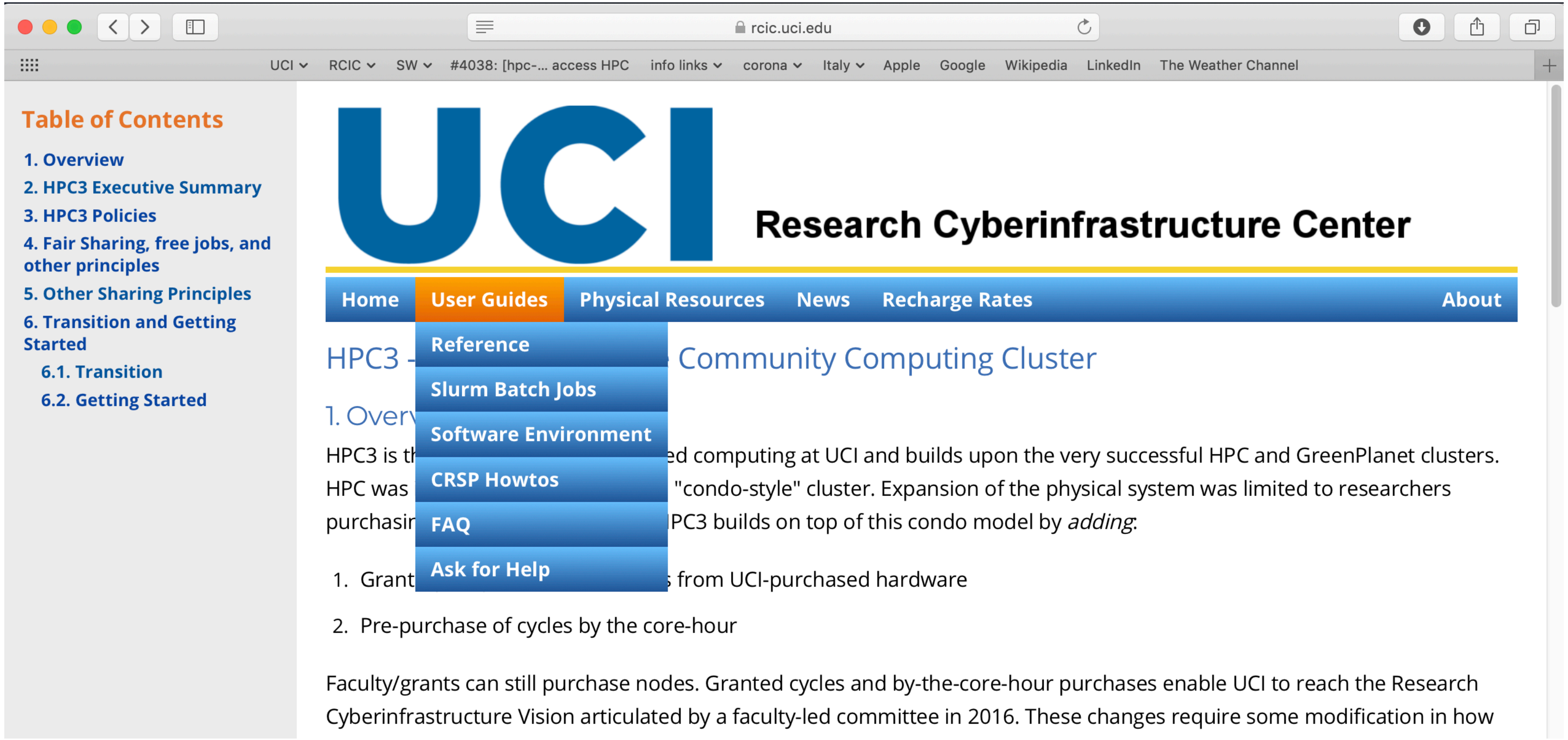
    1. Missing URL & version 2. How big is group who will be using it?

5. I'm having trouble to write any file in the directory I usually work in /wt/panteater/. Do you know why?

    1. On what node? 2. What was the command? 3. What was the error?

Where to get more help and information

http://rcic.uci.edu



rcic.uci.edu
UCI RCIC SW #4038: [hpc-... access HPC info links corona Italy Apple Google Wikipedia LinkedIn The Weather Channel
UCI Research Cyberinfrastructure Center
Home
User Guides
Physical Resources
News
Recharge Rates
About
Reference
Slurm Batch Jobs
Software Environment
CRSP Howtos
FAQ
Ask for Help

Table of Contents

HPC3 - Community Computing Cluster

1. Over

HPC3 is th ed computing at UCI and builds upon the very successful HPC and GreenPlanet clusters. HPC was "condo-style" cluster. Expansion of the physical system was limited to researchers purchasin PC3 builds on top of this condo model by *adding*:

1. Grant from UCI-purchased hardware
2. Pre-purchase of cycles by the core-hour

Faculty/grants can still purchase nodes. Granted cycles and by-the-core-hour purchases enable UCI to reach the Research Cyberinfrastructure Vision articulated by a faculty-led committee in 2016. These changes require some modification in how

Agenda

1. What are the differences between HPC3 and HPC
2. Basics of Linux and Shell necessary for running jobs on HPC
3. How to search for modules/programs on HPC? What are the differences between modules and programs? How to download modules/programs? How to set environment?
4. How to run jobs on HPC? What queues are available? What are the common problems?

SSH login

- Logging in is via ssh with your UCInetID

  ```
  ssh hpc3.rcic.uci.edu -l panteater
  ```
  or
  ```
  ssh panteater@hpc3.rcic.uci.edu
  ```

- Use passphrase and ssh public key authentication

  Do not use empty ssh passphrase!!!

  https://www.ssh.com/ssh/public-key-authentication

- If you plan to run interactive graphics programs

  ```
  ssh  -X -Y hpc3.rcic.uci.edu -l panteater
  ```

  But X11 used by Linux for graphics is high bandwidth and can be sensitive to network latency, some people prefer **x2go** https://wiki.x2go.org/doku.php

HPC3 address:
hpc3.rcic.uci.edu

Your shell is **bash** - GNU **B**ourne-**A**gain **SH**ell

- an implementation of shell and its built-in utilities
- a command language interpreter that intercepts and translates what you type, to tell the computer what to do.
- bash is a full programming language used in scripts and command line

you will be mostly interacting with **bash**, so learn the basics!

- create and list files and directories
- check your disk usage
- create a chain of commands
- create shortcuts of the commands (aliases)
- and MUCH more

Naming schema for files and directories

- names are case-sensitive
- folders == directories, on Linux separated by /, for example /usr/lib/
- single dot . means *this current directory*
  double dot .. means *parent directory, one above current*
  tilde ~ means home directory, a.k.a *$HOME*
- full path means start from the root / as in */dfs3/pub/panteater/work2/*
- relative path means start from the current directory as in *mypath/test23*

IMPORTANT !!!

- Use **alpha-numeric** characters and . - _ (*dot* , *dash*, *underscore*) for file/directory names
- DO NOT use / ? % $ & * < > ( ) { } or space between words

Bash startup files

When invoked, bash executes commands from a set of startup files

Interactive shell - reads and writes to a user terminal

Non-interactive shell - not associated with user terminal, for example executing a script

login shell – a user login to the terminal either remotely via ssh or locally, or when bash is launched with the --login option. Executes .bash_profile

non-login shell - is invoked from the login shell, such as when typing bash in the shell prompt. Executes .bashrc

1. Use *.bash_profile* to run commands that should run only once
   - customizing $PATH *RARELY NEED THIS !*
2. Use *.bashrc* for the commands that should run every time you launch a new shell
   - aliases and environment variables
   - history
   - custom prompts

Creating bash aliases and environment variables

Alias syntax:

alias aliasName=”command to run”

For a current session, can execute from the command line.

For a permanent effect put in *$HOME/.bashrc*

alias rm=‘rm -i’	confirm before removing the files
alias m=less	filter for paging through text
alias vi=”vim”	set text editor to vim
alias c='clear’	clear terminal screen if possible
alias h='history’	list history
alias la='ls -la’	list files/directories, including the hidden
alias llt='ls -lat’	list files/directories, sort newest on top
alias myip='curl ipinfo.io/ip’	print host IP

Environment variables syntax:

```
export  EDITOR=vim
export  TMPDIR=/tmp
export  MYVAR=/dfs3/pub/me/myprog
export  PATH=$PATH:/my/acct/dir1
```

export PATH=/my/acct/dir1/ditr2 NO!!!

Bash history

history - bash build-in, displays all available history of commands with the line numbers.
About ~1000-2000 depending on the system configuration.

```
$ history
…
 931  ls
 932  module avail igv
 933  grep igv out-parsemod
 934  ls /data/apps/igv/2.5.0/
 935  pwd
 936  qstat -s p
```

Examples of using history

!934	– run nth command
!grep	– run last command that starts with grep
history | more	– page history output
history 5	– see last 5 commands
!!	– execute last command
history | grep rpm	– filter a specific command

- By default *.bash_history* is used for saving a list of commands
- If several sessions are opened only the history of last closed one is saved
- To save any current session history use history -a

Custom prompt

Shell prompt is set by default in one of the system files

```
Mon Sep 14 14:13:38 [1.09 0.97 0.91]  panteater@login-i15:~
```

To customize:

```
cp .bashrc .bashrc.save          If you make a mistake, can recover
vim  .bashrc                     make edits
```

Setting for PS1	Resulting prompt
`PS1="\[\033[01;36m\]\\h  \!%  \[\e[0m\]  ”`	`login-i16  183%`
`PS1="\[\033[01;36m\][\u@\h  \W]  \!%  \[\e[0m\]  "`	`[panteater@login-i16  ~]  157%`

Common commands

mkdir dname make a dir

rmdir dname remove a dir

mv from to move or rename

cp from to copy file(s)

rm fnames delete file(s)

wget URL download a file

less fname view files read-only

cat fname print file on STDOUT

head fname print first lines of file

tail fname – print last lines of file

wc   word and line count

pwd print current directory

ls [options] list file

cd dirname change directory

find mydir/ -name ‘fname*.sub’ find files

tree options show the directory tree

file names what is this?

w show about active users and their processes

du -h disk usage

df -h disk free

top show info about processes

time cmd arg1 arg2 how long it takes

cmd –h   or   cmd --help   or   cmd –help   or man cmd

Files in your $HOME and backup

```
login-i16 34%   ls -l out
-rw-rw-r-- 1 npw npw 4004 Sep 17 15:13 out
login-i16 35%  rm  -rf out
login-i16 36%  ls -l out
ls: cannot access out: No such file or directory
login-i16 37%  ls .zfs/snapshot/
zfs-auto-snap_daily-2020-09-16-1017
zfs-auto-snap_daily-2020-09-17-1045
zfs-auto-snap_daily-2020-09-18-1048
login-i16 38%  ls .zfs/snapshot/zfs-auto-snap_daily-2020-09-17-1045/out
ls: cannot access .zfs/snapshot/zfs-auto-snap_daily-2020-09-17-1045/out: No such file or directory
login-i16 39%  ls .zfs/snapshot/zfs-auto-snap_daily-2020-09-18-1048/out
.zfs/snapshot/zfs-auto-snap_daily-2020-09-18-1048/out
login-i16 40%  cp !$ .
cp .zfs/snapshot/zfs-auto-snap_daily-2020-09-18-1048/out .
login-i16 41%  ls -l out
-rw-rw-r-- 1 npw npw 4004 Sep 18 10:53 out
```

Your $HOME:

- Quota 50GB, keep it clean and organized
- ZFS filesystem, we take snapshots (backup capability)
  daily, keep last 8   weekly, keep last 6
- Location $HOME/.zfs/snapshots/ READ ONLY!

Agenda

1. What are the differences between HPC3 and HPC
2. Basics of Linux and Shell necessary for running jobs on HPC
3. How to search for modules/programs on HPC? What are the differences between modules and programs? How to download modules/programs? How to set environment?
4. How to run jobs on HPC? What queues are available? What are the common problems?

Using software on HPC clusters

Where are programs and modules and how to access them?

You install in your user area. Use ~/bin/	added to your $PATH by shell in .bashrc_profile
System utilities and commands /usr/bin	added to your $PATH by shell in .bashrc
Modules for Perl, Python, R /opt/apps	user need to use module command interface
Scientific applications /opt/apps /data/opt/apps	user need to use module command interface

How to search for modules/programs on HPC?

To access a **program**, you need to have its directory in your **$PATH**

```
hpc3-14-00 2046%  echo $PATH
/data/utils/system-files/system-wide-env-setup:/opt/rocks/yaml2rpm:
/opt/apps/python/3.8.0/bin:/data/homezvol0/n/perl5/bin:/usr/local/bin:
/usr/bin:/usr/local/sbin:/usr/sbin:/data/homezvol0/n/bin
hpc3-14-00 2047%  which pip
/opt/apps/python/3.8.0/bin/pip
```

What is my $PATH?

lines are broken for readability

What pip am I using ?

1

language modules are collections of related variables, functions and subroutines that perform a set of specific programming tasks. Simply put – files consisting **Perl/Python/R** code. *Search using its language methods.*

To access a language module, you need to use its language: **Perl / Python / R**

To access a language **Perl / Python / R** you need … **environment modules**

2

What are environment modules

- **Environment module** is a user interface to the Modules package which provides for the dynamic modification of the user's environment via modulefiles.
- Each **modulefile** contains all the info needed to configure the shell to use a specific application.
- Command **module load** interprets the modulefiles and
  - Sets aliases
  - Sets environment variables
  - Loads depended modules
- Command **module avail** lists all installed software and their versions

General info for Linux https://modules.readthedocs.io/en/latest/

Read User guide for HPC3 https://rcic.uci.edu/hpc3/software-tutorial.html

Environment modules update your environment

Case 1: usage of multiple versions of software

```
login-i16 47%  which R
/usr/bin/which: no R in (/usr/local/bin:/usr/bin:/usr/sbin:/data/homezvol0/npw/bin)
login-i16 48%  module avail R
----------- /opt/rcic/Modules/modulefiles/LANGUAGES -------
R/3.6.2 R/4.0.2
login-i16 49%  module load R/4.0.2
login-i16 50%  which R
/opt/apps/R/4.0.2/bin/R
login-i16 51%  module list
Currently Loaded Modulefiles:
1) OpenBLAS/0.3.6   2) java/1.8.0   3) icu/65.1   4) R/4.0.2
login-i16 52%  module unload R/4.0.2
login-i16 53%  module list
No Modulefiles Currently Loaded.
login-i16 54%  module load R/3.6.2
login-i16 55%  which R
/opt/apps/R/3.6.2/bin/R
```

Case 2: load/unload different software modules

```
login-i16 38%  module load gcc/8.4.0
login-i16 39%  module list
Currently Loaded Modulefiles:
  1) gcc/8.4.0
login-i16 40%  module load hdf5/1.10.5/gcc.8.4.0
login-i16 41%  module list
Currently Loaded Modulefiles:
  1) gcc/8.4.0    2) java/1.8.0   3) hdf5/1.10.5/gcc.8.4.0
login-i16 42%  module unload hdf5/1.10.5/gcc.8.4.0
login-i16 43%  module list
Currently Loaded Modulefiles:
  1) gcc/8.4.0
```

Always unload module in reverse order: FILO!

Environment module commands summary

search	$ module avail	shows all installed software environment modules
	$ module avail R	show R modules
	$ module keyword salmon	check all modules for a keyword
info	$ module display R	shows environment modification + description
	$ module help R	show module specific help (description)
use	$ module load R	loads R at whatever latest version not ideal
	$ module load R/4.0.2	loads R at specified version preferred method
	$ module list	lists currently loaded modules
	$ module unload R/4.0.2	unloads specified module (in reverse order if many)
	$ module purge	removes all loaded modules

```
salmon/1.1.0        : Name____________ salmon
salmon/1.1.0        :                  salmon_1.1.0
```

What Perl modules are installed?

Method 1: instmodsh

```
hpc3-14-00 2001%  module load perl/5.30.0
hpc3-14-00 2002%  instmodsh
Available commands are:
   l              - List all installed modules
   m <module>     - Select a module
   q              - Quit the program
cmd? l
Installed modules are:
   Algorithm::Diff
   Alien::Build
    ...
```

Method 3: perl test sript

```
          use strict;
          use warnings;

          use Unicode::Map;
          use Bio::Perl;
          say STDERR "No errors";
hpc3-14-00 2007%  module load perl/5.30.0
hpc3-14-00 2008%  perl test.pl
```

Method 2: cpan

```
hpc3-14-00 2006%  cpan
Terminal does not support AddHistory.
To fix enter>  install Term::ReadLine::Perl
cpan shell -- CPAN exploration ...(v2.22)
Enter 'h' for help.
cpan[1]> r
Fetching with LWP:
...
DONE
Writing /data/homezvol0/npw/.local/share/.cpan/Metadata

Package namespace   installed latest  in CPAN file
Alien::Build           2.15      2.32    Alien-Build-2.32.tar.gz
Alien::Libxml2         0.14      0.16    Alien-Libxml2-0.16.tar.gz
```

Note, it creates cache

What python modules are installed?

Method 1: pip

```
hpc3-14-00 2022%  module load python/3.8.0
hpc3-14-00 2023%  pip list
Package                  Version
------------------------ --------------
absl-py                    0.9.0
appdirs                    1.4.4
astor                      0.8.1
backports.weakref          1.0.post1
…
```

Method 2: quick test for one module

```
hpc3-14-00 2027%  python -c "import mmtf"

hpc3-14-00 2028%  python -c "import bla"
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ModuleNotFoundError: No module named 'bla’
```

What R modules are installed?

```
hpc3-14-00 2022%  module load R/4.0.2

hpc3-14-00 2022%  R

> installed.packages()
…
xlsxjars          "4.0.2"
XML               "4.0.2"
xml2              "4.0.2"
xopen             "4.0.2"
…

> find.package("XML")

[1] "/opt/apps/R/4.0.2/lib64/R/library/XML"

> find.package("XML2")

Error in find.package("XML2") : there is no package called 'XML2'
```

How to install language modules

Many language modules can be installed in user space

There is no single repository for download for all, use main ones to start with

Perl	https://www.cpan.org
Python	https://pypi.org
R	https://cran.r-project.org

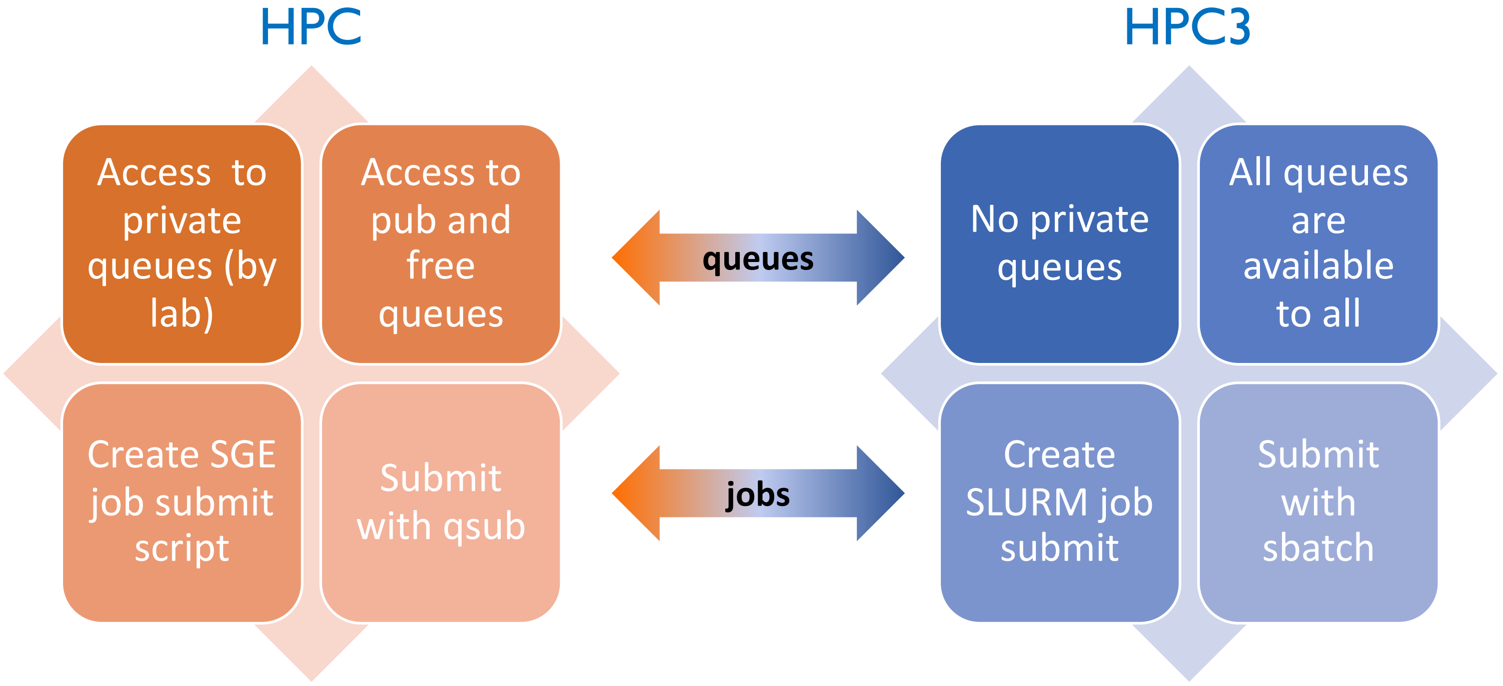
How to install your desired language module, see section *3.1 Install it yourself*
in Getting Help user guide https://rcic.uci.edu/hpc3/getting-help.html

R packages	Python packages with pip
Bulding Conda local environments	Perl CPAN modules

Agenda

1. What are the differences between HPC3 and HPC
2. Basics of Linux and Shell necessary for running jobs on HPC
3. How to search for modules/programs on HPC? What are the differences between modules and programs? How to download modules/programs? How to set environment?
4. How to run jobs on HPC? What queues are available? What are the common problems?

How to run jobs on HPC/HPC3


HPC
Access to private queues (by lab)
Access to pub and free queues
Create SGE job submit script
Submit with qsub
queues
jobs
HPC3
No private queues
All queues are available to all
Create SLURM job submit
Submit with sbatch

Mostly a 1:1 correspondence among commands, directives and concepts

Major difference: accounting

Common problems

1. Submit gazillions of jobs with nearly the same info
   Fix: convert to an array job   Crucial for many users!
2. Run heavy computational jobs on login nodes
   Fix: claim an interactive node or submit a batch job
3. Do not set job environment properly / use too much resources
   Fix:
   - use modules
   - ask resources you need and not more
4. Running out of space and not checking the disk quotas
   Fix: watch your usage and do periodic cleaning
5. Not testing your jobs submissions.
   Fix:
   - test on a small input first
   - check all names and variables are correct
   - after submitting a job check the status
6. Not providing correct information when submitting a ticket for help
   Fix: read the User Guides and follow the directions

See User Guides
http://rcic.uci.edu

An array job in SGE

```
#!/bin/bash
#$ -N MyArray
#$ -q pub
#$ -pe openmp 1
#$ -ckpt restart
#$ -tc 100                         ## run concurrent tasks
#$ -t 1-2000                       ## number of tasks to send
#$ -e $JOB_NAME.e$TASK_ID    ## sge error per task
#$ -o $JOB_NAME.o$TASK_ID   ## sge output per task

module load myprog/1.2.3

INPUT=/pub/panteater/Test/Week1/Input.txt
OUTPUT=/pub/panteater/Test/Week1/Outputs
Args=$(awk "NR==$SGE_TASK_ID" $INPUT)              # get arguments from file,  a line per task
myprog $Args  –o $OUTPUT/out.$JOB_ID–$SGE_TASK_ID   # output of each task in a separate file
```

SGE array examples
https://docs.hpc.shef.ac.uk/en/latest/parallel/JobArray.html

Input.txt content:

```
/pub/bio/u/dir1 file12 34
/pub/bio/u/dir2 file22 34
/pub/bio/u/dir3 file33 30
```

An array job in SLURM

```
#!/bin/bash
#SBATCH --job-name=MyArray
#SBATCH -p free
#SBATCH -t 1-2000%100          ## number of tasks to send and concurrency
#SBATCH -e %x.e%A_%a           ## %x – job name, %A – job id, %a – task id
#SBATCH -o %x.o%A_%a

module load myprog/1.2.3

INPUT=/pub/panteater/Test/Week1/Input.txt
OUTPUT=/pub/panteater/Test/Week1/Outputs
Args=$(awk "NR==$SLURM_ARRAY_TASK_ID" $INPUT)                          # get arguments from file
myprog $Args  –o $OUTPUT/out.$SLURM_JOB_ID–$SLURM_ARRAY_TASK_ID  # separate task output
```

Migrating from SGE to Slurm

	SGE	Slurm
Interactive login	qrsh –q edu qlogin	srun --pty bash srun --pty bash -p standard --time=4:0:0
Job Submission	qsub myjob.sub	sbatch myjob.sub (1) srun (2) salloc
Accounting	qacct	sacct sacctmgr
Job deletion Job status	qdel jobID qstat u panteater qhold jobID | qrls jobID	scancel jobID squeue –u panteater scontrol hold jobID | scontrol release jobID
Queue list Cluster status	qconf –sql qhost -q	squeue (1) sinfo (2) scontrol show nodes

Environment variables

	SGE	Slurm
Job ID	$JOB_ID	$SLURM_JOBID
Job name	$JOB_NAME	$SLURM_JOB_NAME
Submit directory	$SGE_O_WORKDIR	$SLURM_SUBMIT_DIR
Submit host	$SGE_O_HOST	$SLURM_SUBMIT_HOST
Node list	$PE_HOSTFILE	$SLURM_JOB_NODELIST
Job array index	$SGE_TASK_ID	$SLURM_ARRAY_TASK_ID

Job specification

	SGE	Slurm
Script directive	#$	#SBATCH
Queue	-q queueName	-p partitionName
Node count	N/A	-N [min[-max]]
CPU count	-pe name count	-n count
Wall clock limit	-l h_rt=hh:mm:ss	-t [min] or -t [days-hh:mm:ss]
Standard output file Standard error file Combine stdout/error	-o [filename] -e [filename] -j yes	-o filename -e filename -o filename (without -e)
Copy Environment	-V	--export=[ALL | NONE | variables]

Job specification cont’d I

	SGE	Slurm
Event Notification	-m beas (begin, end, abort, stop)	--mail-type=[events] use sparingly!
Email Address	-M address	--mail-user=address
Job Name	-N name	--job-name=name
Job Restart	-r [yes | no]	--requeue OR --no-requeue
Work Directory	-wd directory	--workdir=dirname
Resource Sharing	-l exclusive	--exclusive OR --shared
Memory Size	-l mem_free=[memory][K|M|G]	--mem=[mem][M|G|T] --mem-per-cpu=[mem][M|G|T]
Account to charge	-A account	--account=account

Job specification cont’d 2

	SGE	Slurm
Tasks per node	(Fixed allocation rule in PE)	--tasks-per-node=[count]
CPUs per task	N/A	--cpus-per-task=[count]
Job dependency	-hold_jid [jobid | jobname]	--depend=[state:jobid]
Job Project	-P [name]	-wckey=[name]
Job host preference	-q [queue]@[node] OR -q [queue]@@[hostgroup]	--nodelist=[nodes] AND/OR --exclude=[nodes]
Quality of Service	N/A	--qos=[name]
Job Arrays	-t [array_spec] -tc [arrayConcurrency]	-array=[array_spec%concurrency]