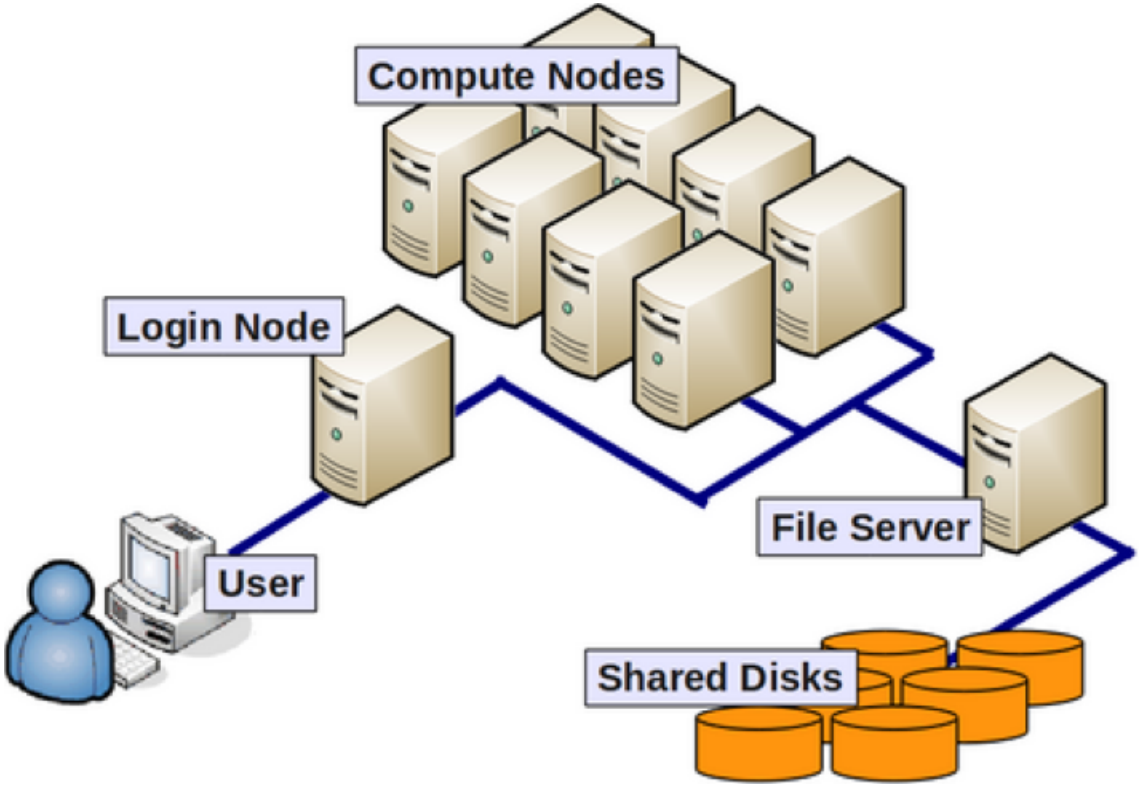
High-Performance Compute Cluster

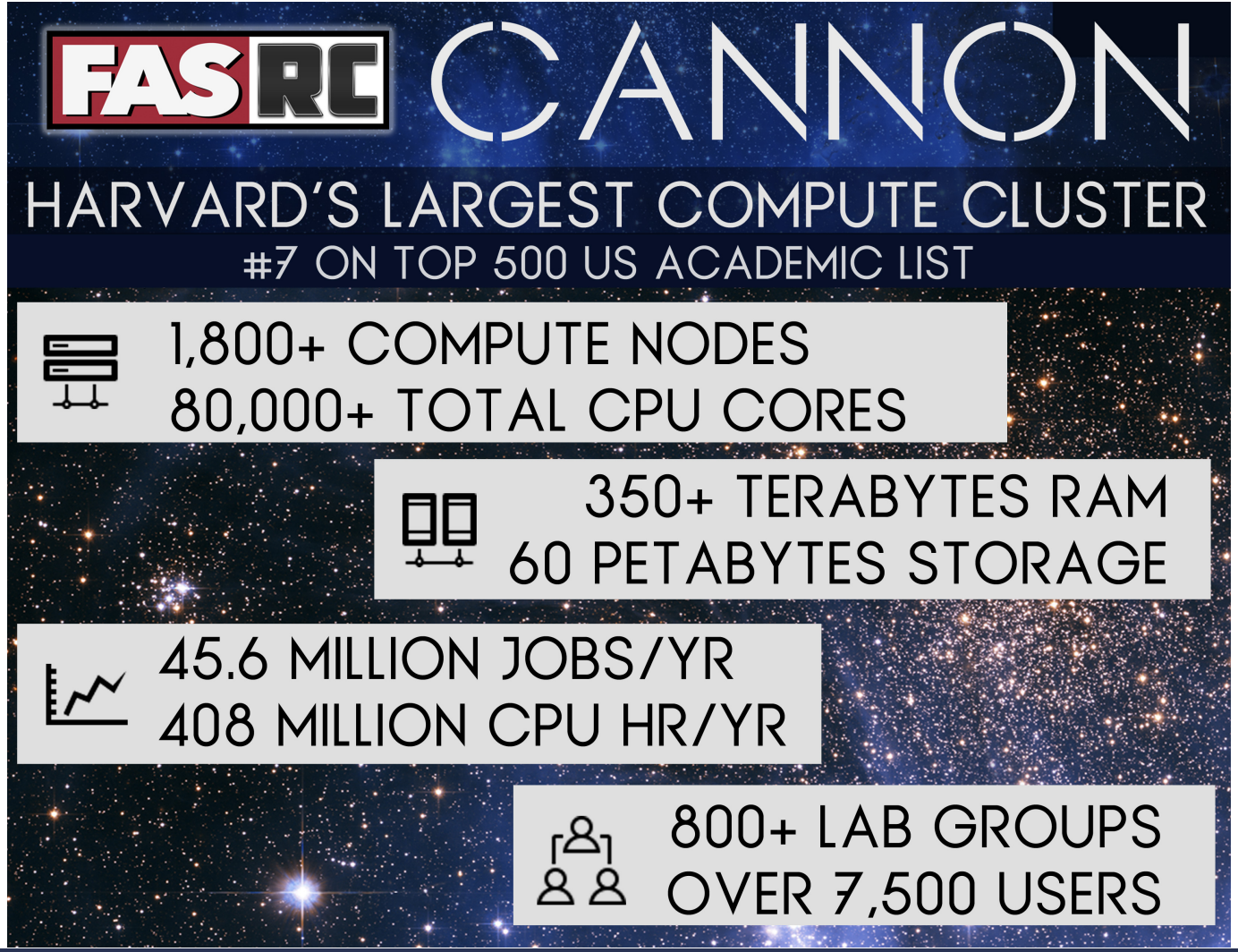
Thanks to

Raminder Singh, Daniel Caunt, Maggie McFee and Muneeba Syed

at FAS-RC for help with these slides

Cluster Architecture




FAS RC
CANNON
HARVARD'S LARGEST COMPUTE CLUSTER
#7 ON TOP 500 US ACADEMIC LIST
1,800+ COMPUTE NODES
80,000+ TOTAL CPU CORES
350+ TERABYTES RAM
60 PETABYTES STORAGE
45.6 MILLION JOBS/YR
408 MILLION CPU HR/YR
800+ LAB GROUPS
OVER 7,500 USERS

The FAS Research Computing cluster's current iteration is named 'Cannon' in honor of astronomy pioneer Annie Jump Cannon.

- The cluster is a multi-user shared environment that runs tens of thousands of jobs each day from thousands of users
- It is comprised of over 1800 nodes and is connected to more than 60 Petabytes of storage.
- Each lab group on the cluster has an equal allotment of usage called 'fairshare' that is then shared amongst that lab's users. As a lab runs jobs, its fairshare decreases, but Fairshare is replenished for each lab over time. This ensure that all labs have equal access to the cluster.
- The SLURM job scheduler ensures that jobs are fairly and equally distributed and scheduled based on available resources and fairshare score.
- Backfill queues (aka 'requeue') allow anyone to take advantage of idle computer resources, no matter their faisrshare score.

The FAS Research Computing cluster's current iteration is named 'Cannon' in honor of astronomy pioneer Annie Jump Cannon.

- The cluster is a multi-user shared environment that runs tens of thousands of jobs each day from thousands of users
- It is comprised of over 1800 nodes and is connected to more than 60 Petabytes of storage.
- Each lab group on the cluster has an equal allotment of usage called 'fairshare' that is then shared amongst that lab's users. As a lab runs jobs, its fairshare decreases, but Fairshare is replenished for each lab over time. This ensure that all labs have equal access to the cluster.
- The SLURM job scheduler ensures that jobs are fairly and equally distributed and scheduled based on available resources and fairshare score.
- Backfill queues (aka 'requeue') allow anyone to take advantage of idle computer resources, no matter their faisrshare score.

The FAS Research Computing cluster's current iteration is named 'Cannon' in honor of astronomy pioneer Annie Jump Cannon.

- The cluster is a multi-user shared environment that runs tens of thousands of jobs each day from thousands of users
- It is comprised of over 1800 nodes and is connected to more than 60 Petabytes of storage.
- Each lab group on the cluster has an equal allotment of usage called 'fairshare' that is then shared amongst that lab's users. As a lab runs jobs, its fairshare decreases, but Fairshare is replenished for each lab over time. This ensure that all labs have equal access to the cluster.
- The SLURM job scheduler ensures that jobs are fairly and equally distributed and scheduled based on available resources and fairshare score.
- Backfill queues (aka 'requeue') allow anyone to take advantage of idle computer resources, no matter their faisrshare score.

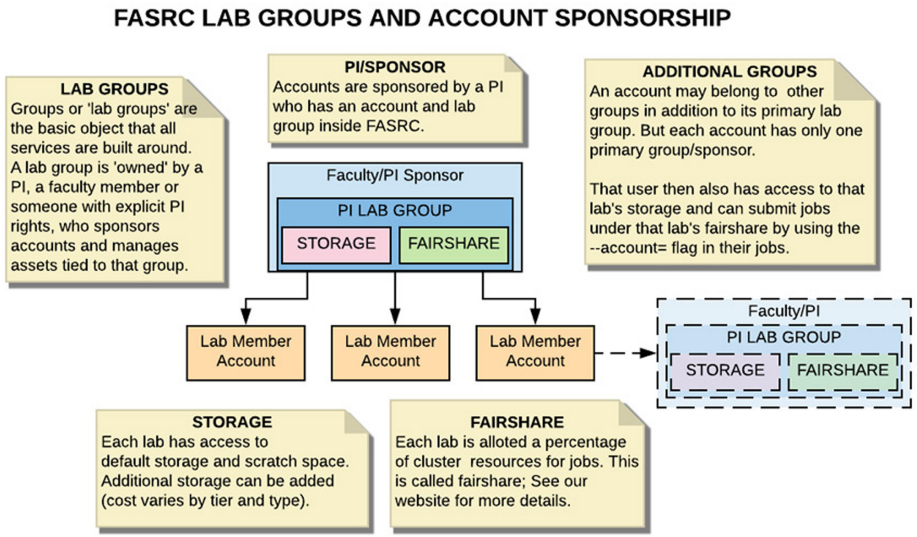
The FAS Research Computing cluster's current iteration is named 'Cannon' in honor of astronomy pioneer Annie Jump Cannon.

- The cluster is a multi-user shared environment that runs tens of thousands of jobs each day from thousands of users
- It is comprised of over 1800 nodes and is connected to more than 60 Petabytes of storage.
- Each lab group on the cluster has an equal allotment of usage called 'fairshare' that is then shared amongst that lab's users. As a lab runs jobs, its fairshare decreases, but Fairshare is replenished for each lab over time. This ensure that all labs have equal access to the cluster.
- The SLURM job scheduler ensures that jobs are fairly and equally distributed and scheduled based on available resources and fairshare score.
- Backfill queues (aka 'requeue') allow anyone to take advantage of idle computer resources, no matter their faisrshare score.

The FAS Research Computing cluster's current iteration is named 'Cannon' in honor of astronomy pioneer Annie Jump Cannon.

- The cluster is a multi-user shared environment that runs tens of thousands of jobs each day from thousands of users
- It is comprised of over 1800 nodes and is connected to more than 60 Petabytes of storage.
- Each lab group on the cluster has an equal allotment of usage called 'fairshare' that is then shared amongst that lab's users. As a lab runs jobs, its fairshare decreases, but Fairshare is replenished for each lab over time. This ensure that all labs have equal access to the cluster.
- The SLURM job scheduler ensures that jobs are fairly and equally distributed and scheduled based on available resources and fairshare score.
- Backfill queues (aka 'requeue') allow anyone to take advantage of idle computer resources, no matter their faisrshare score.

To access the FASRC cluster, you will need a FASRC account sponsored by a PI on the cluster. PI's have lab groups, lab groups have resources, lab members have access to those resources. Users of the cluster can belong to multiple groups, but only one group is your primary group and sponsor.



Using the cluster!

https://www.youtube.com/watch?v=Ay8oR5n-yyQ

Login & Access

https://docs.rc.fas.harvard.edu/kb/quickstart-guide/

Cluster Quick Start Guide

Table of Contents > [show]

This guide will provide you with the basic information needed to get up and running on the FASRC cluster for simple command line access. If you'd like more detailed information, each section has a link to fuller documentation

PREREQUISITES

1. Get a FASRC account using the account request tool.

Before you can access the cluster you need to request a Research Computing account.

See How Do I Get a Research Computing Account for instructions if you do not yet have an account.

See the account confirmation email for instructions on setting your password and getting started.

Login & Access

Once you have an account you can use the Terminal to connect to the cluster


– Mac: Terminal


– Linux: Xterm or Terminal


– Windows: SSH client - Putty or Bash Emulator - Git Bash

```
$ ssh username@login.rc.fas.harvard.edu
```

- ssh stands for Secure SHell
- ssh is a protocol for data transfer that is secure, i.e the data is encrypted as it travels between your computer and the cluster (remote computer)
- Commonly used commands that use the ssh protocol for data transfer are, scp and sftp

Login & Access

Once you have an account you can use the Terminal to connect to the cluster


– Mac: Terminal


– Linux: Xterm or Terminal


– Windows: SSH client - Putty or Bash Emulator - Git Bash

```
$ ssh username@login.rc.fas.harvard.edu
```

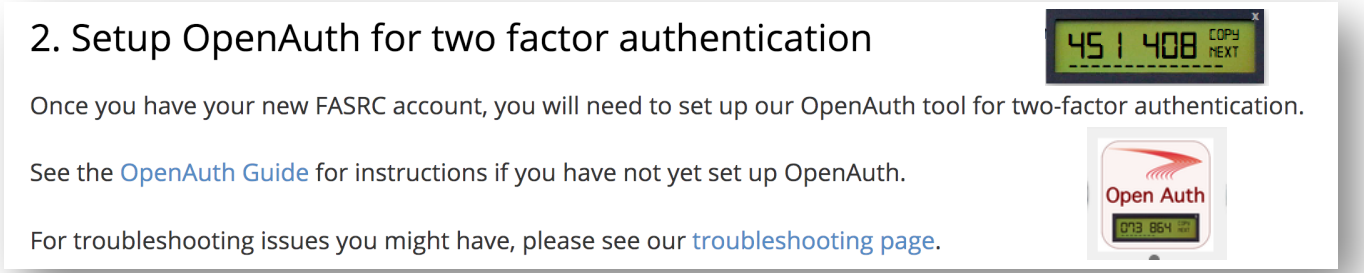
```
Login issues? See https://rc.fas.harvard.edu/resources/support/

Password:
Verification code:
```

Login & Access

https://www.rc.fas.harvard.edu/resources/quickstart-guide/

Once you have run the ssh command:

- Enter your password (*cursor won't move!*)
- Add a verification code (2-Factor Authentication)



OpenAuth is 2-factor authentication separate from HarvardKey and updates the token every 30 seconds

Login & Access

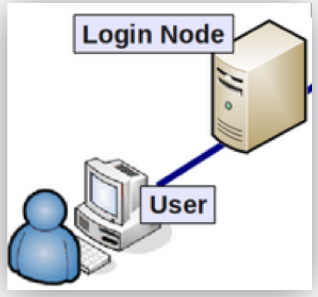
rsk394 — rkhetani@holylogin03:~ — ssh rkhetani@login.rc.fas.harvard.edu — 92×40

```
!!!!!!!!!!!!!!!!!!!!!!!!    Cannon    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Welcome to Cannon, a HPC resource for the research community,
hosted by Research Computing at HU's Faculty of Arts and Sciences.

+--------------- Helpful Documentation: -----------------+
| https://rc.fas.harvard.edu/resources/quickstart-guide/ |
| https://rc.fas.harvard.edu/running-jobs/               |
| https://rc.fas.harvard.edu/convenient-slurm-commands/  |
+--------------------------------------------------------+

+------------------- NEWS & UPDATES: ----------------------------------------------------+
+ OFFICE HOURS: Wednesdays noon-3pm, 38 Oxford, ROOM 100 (1st Flooor conf room)          +
+ Check our consulting calendar at: https://www.rc.fas.harvard.edu/consulting-calendar/  +
+ Check our training schedule at: https://www.rc.fas.harvard.edu/upcoming-training/      +
+-------------------------------------------------------------------------------------+

NEXT MAINTENANCE: NOVEMBER 4TH 7-11AM

https://www.rc.fas.harvard.edu/maintenance

CANNON: Cannon is live!  See the Running Jobs page for information about
the updated partitions.

https://www.rc.fas.harvard.edu/resources/running-jobs/#Slurm_partitions

For more about the new cluster see:

https://www.rc.fas.harvard.edu/fasrc-cluster-refresh-2019/

GENERAL: The general partition has been decommissioned.  Please use
the shared partition.  For high memory jobs use bigmem.

WINTER MAINTENANCE DECEMBER 3RD 7AM-5PM: We are doing an all day major
maintenance on December 3rd which will involve all running jobs being
cancelled. More details forthcoming soon. Please plan accordingly.

[rkhetani@holylogin03 ~]$
```

You have logged into the login node!

```
[joesmith@holylogin03 ~]$
```

Name of the login node assigned to you



Access to resources on a compute node

- Login node:
  - not designed for analysis
  - not anything compute- or memory-intensive
  - best practice is to request a compute node as soon as you log in

Access to resources on a compute node

- Login node:
  - not designed for analysis
  - not anything compute- or memory-intensive
  - best practice is to request a compute node as soon as you log in
- Interactive session on a compute node:
  - work on a compute node “interactively”
  - request resources from SLURM using the **`salloc`** command
  - ongoing analysis will terminate when the remote connection is terminated

```
[rkhetani@holylogin03 ~]$ salloc -p test -t 0-2:30 --mem 1G
```

```
[rkhetani@holy7c26602 ~]$
```

Name of the compute node assigned to you

Access to resources on a compute node

- Login node:
  - not designed for analysis
  - not anything compute- or memory-intensive
  - best practice is to request a compute node as soon as you log in
- Interactive session on a compute node:
  - work on a compute node “interactively”
  - request resources from SLURM using the `salloc` command
  - ongoing analysis will terminate when the remote connection is terminated
- Batch job (compute node):
  - Send an analysis script to a compute node
  - request resources from SLURM using #SBATCH directives in a shell script
  - use the sbatch command to submit the shell script
  - disconnecting the connection to the cluster has no impact on analysis

The Job Scheduler, SLURM

Access to resources on a compute node

Simple Linux Utility for Resource Management - SLURM job scheduler:

- Fairly allocates access to resources to users on compute nodes
- Manages a queue of pending jobs; ensures that no single user or group monopolizes the cluster
- Ensures users do not exceed their resource requests
- Provides a framework for starting, executing, and monitoring batch jobs

The resources managed by SLURM

Commonly used cluster resources

1. Time (`-t`)

- Specified as days-hours:minutes (4-12:00) or hours:minutes:seconds (12:00:00)

More information: https://docs.rc.fas.harvard.edu/kb/running-jobs/#Slurm_partitions

Commonly used cluster resources

1. Time (`-t`)

- Specified as days-hours:minutes (4-12:00) or hours:minutes:seconds (12:00:00)

2. Number of cores (`-c`)

More information: https://docs.rc.fas.harvard.edu/kb/running-jobs/#Slurm_partitions

Commonly used cluster resources

1. **Time (`-t`)**
    - Specified as days-hours:minutes (4-12:00) or hours:minutes:seconds (12:00:00)
2. **Number of cores (`-c`)**
3. **Memory (`--mem or --mem-per-cpu`)**
    - `--mem` sets total memory across all cores
    - `--mem-per-cpu` sets the value for each requested core.

More information: https://docs.rc.fas.harvard.edu/kb/running-jobs/#Slurm_partitions

Commonly used cluster resources

1. Time (`-t`)

- Specified as days-hours:minutes (4-12:00) or hours:minutes:seconds (12:00:00)

2. Number of cores (`-c`)

3. Memory (`--mem or --mem-per-cpu`)

- `--mem` sets total memory across all cores
- `--mem-per-cpu` sets the value for each requested core.

4. Cluster partitions (`-p`)

- The many compute nodes on any cluster are grouped together based on the resources available on them, and at the system administrator's discretion
- These groups are called partitions
- Depending on the resources you require, you pick the one best suited for your work

More information: https://docs.rc.fas.harvard.edu/kb/running-jobs/#Slurm_partitions

Partitions and associated resources

Partitions:	shared	gpu	test	serial_requeue	gpu_test	gpu_requeue	bigmem	unrestricted	pi_lab
Time Limit	7 days	7 days	8 hrs	7 days	8 hrs	7 days	7 days	no limit	**no limit**
# Nodes	426	18	16	varies	11	Varies	30	8	**varies**
# Cores / Node	48	64	48	varies	32	varies	64	48	**varies**
Memory / Node (GB)	184	499	184	varies	373	varies	499	184	**varies**

Resources

More information: https://docs.rc.fas.harvard.edu/kb/running-jobs/#Slurm_partitions

Partitions and associated resources

Partitions:	test
Resources: Time Limit	8 hrs
# Nodes	16
# Cores / Node	48
Memory / Node (GB)	184

This partition is dedicated for interactive work and for testing code before submitting in batch and scaling.

More information: https://docs.rc.fas.harvard.edu/kb/running-jobs/#Slurm_partitions

Partitions and associated resources

Partitions:	serial_requeue
Resources: Time Limit	7 days
Resources: # Nodes	varies
Resources: # Cores / Node	varies
Resources: Memory / Node (GB)	varies

This partition is appropriate for single core jobs or jobs that require up to 8 cores for less than 1 day.

If you do not specify a partition you will be sent to this partition by default.

More information: https://docs.rc.fas.harvard.edu/kb/running-jobs/#Slurm_partitions

Partitions and associated resources

Partitions:	shared	gpu	test	serial_requeue	gpu_test	gpu_requeue	bigmem	unrestricted	pi_lab
Resources: Time Limit	7 days	7 days	8 hrs	7 days	8 hrs	7 days	7 days	no limit	**no limit**
# Nodes	426	18	16	varies	11	Varies	30	8	**varies**
# Cores / Node	48	64	48	varies	32	varies	64	48	**varies**
Memory / Node (GB)	184	499	184	varies	373	varies	499	184	**varies**

Looking at a Partition to learn more:

```
sinfo -p shared
scontrol show partition shared
```

More information: https://docs.rc.fas.harvard.edu/kb/running-jobs/#Slurm_partitions

Job submissions for SLURM

Job submissions with Slurm

Slurm-specific commands are used to run jobs on the compute nodes

- The arguments used by `salloc, sbatch` are identical, e.g. `-p, -t, --mem`

Interactive jobs:

```
salloc -p test -t 0-30:00 OTHER_JOB_OPTIONS /bin/bash
```

```
# -p : Specify Partition
# -t : Specify Runtime in D-HH:MM
```

Batch jobs (*"slurm directives"*):

```
#SBATCH -p serial_requeue
#SBATCH -t 0-30:00
```

More information: https://docs.rc.fas.harvard.edu/kb/running-jobs/

Slurm Job Script (for sbatch)

Slurm directives

```
#!/bin/bash

#SBATCH -J bowtie-align          # Job name
#SBATCH -p serial-requeue        # Partition(s) (separate with commas if using multiple)
#SBATCH -c 4                     # Number of cores
#SBATCH -t 0-00:30               # Time (D-HH:MM)
#SBATCH --mem=500M               # Memory
#SBATCH -o %j.o                  # Name of standard output file
#SBATCH -e %j.e                  # Name of standard error file

## LOAD SOFTWARE ENV ##
module load intel/2017c impi/2017.4.239 Bowtie2/2.3.4.1

## EXECUTE CODE ##
bowtie -c 4 hg19 file1_1.fq
```

Slurm Job Script (for `sbatch`)

Slurm directives

```
#!/bin/bash

#SBATCH -J bowtie-align          # Job name
#SBATCH -p serial-requeue        # Partition(s) (separate with commas if using multiple)
#SBATCH -c 4                     # Number of cores
#SBATCH -t 0-00:30               # Time (D-HH:MM)
#SBATCH --mem=500M               # Memory
#SBATCH -o %j.o                  # Name of standard output file
#SBATCH -e %j.e                  # Name of standard error file

## LOAD SOFTWARE ENV ##
module load intel/2017c impi/2017.4.239 Bowtie2/2.3.4.1

## EXECUTE CODE ##
bowtie -c 4 hg19 file1_1.fq
```

Save script as `myJobScript.run` and run it as follows:

```
$ sbatch myJobScript.run
```

Test first

ALWAYS test the job submission script first:

- To ensure the job will complete without any errors
- To ensure you understand the resource needs and have requested them appropriately

Interacting with Slurm

Slurm-specific commands are used to run jobs on the compute nodes

- Several other commands exist to monitor jobs, control/modify them, obtain information about partitions and user-specific information, etc.

	SLURM	SLURM Example
Submit a batch serial job	sbatch	`sbatch runscript.sh`
Run a script or application interactively	salloc	`salloc -p test -t 10 --mem 1000 [script or app]`
Start interactive session	salloc	`salloc -p test -t 10 --mem 1000`
Kill a job	scancel	`scancel 999999`
View status of your jobs	squeue	`squeue -u akitzmiller`
Check current job by id number	sacct	`sacct -j 999999`
Schedule recurring batch job	scrontab	see scrontab document for example

More information: https://docs.rc.fas.harvard.edu/kb/running-jobs/

Using & installing software

LMOD: Software Modules

- Most “software” on the FAS-RC cluster is installed as an environment module.
- LMOD system adds directory paths of software into the $PATH variable, to make sure the program runs without any issues.
- Allows for clean, easy loading, including most dependencies, and switching versions.
- All available sofware is listed on this webpage -
  - https://portal.rc.fas.harvard.edu/p3/build-reports/

LMOD: Software Modules

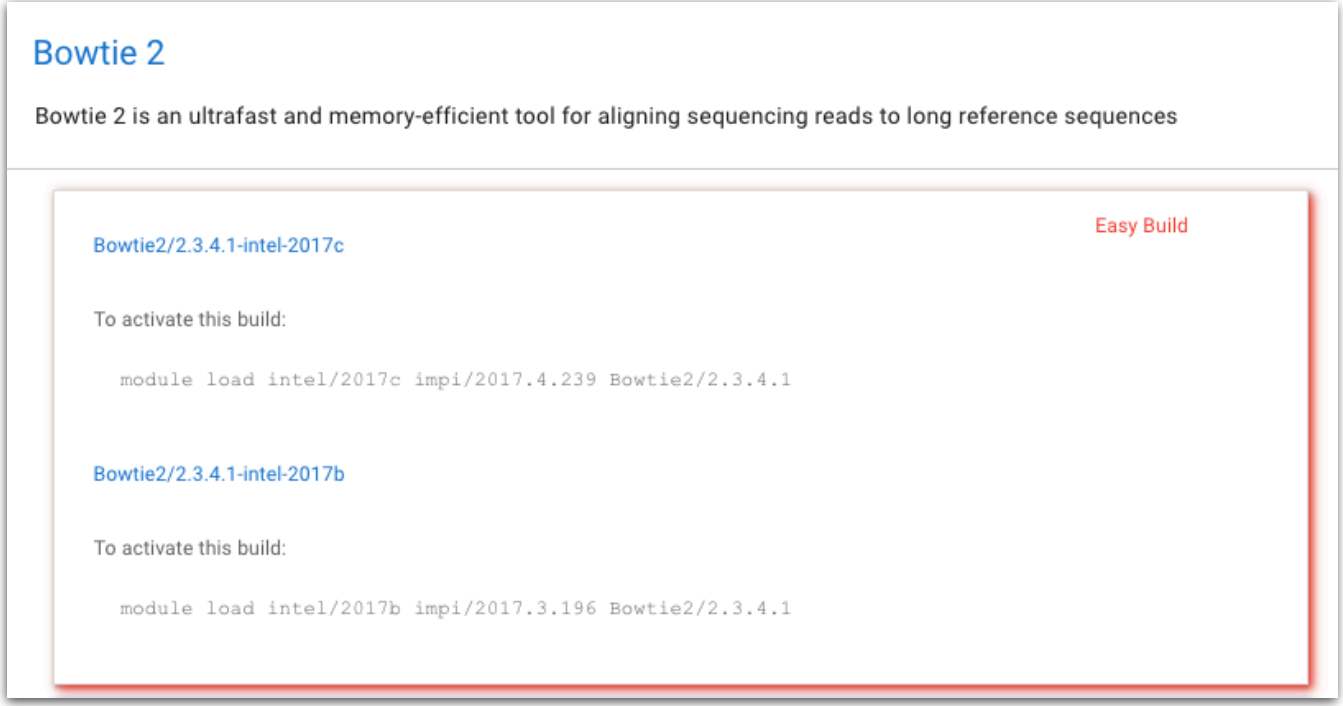
Check module status (e.g. the alignment tool bowtie2)

```
$ module list
$ echo $PATH
$ bowtie2
```

LMOD: Software Modules

Check module status (e.g. the alignment tool bowtie2)

```
$ module list
$ echo $PATH
$ bowtie2
```



https://portal.rc.fas.harvard.edu/p3/build-reports/

LMOD: Software Modules

Check module status (e.g. the alignment tool bowtie2)

```
$ module list
$ echo $PATH
$ bowtie2
```

Load the module

```
$ module load intel/2017c impi/2017.4.239 Bowtie2/2.3.4.1
```

LMOD: Software Modules

Check module status (e.g. the alignment tool bowtie2)

```
$ module list
$ echo $PATH
$ bowtie2
```

Load the module

```
$ module load intel/2017c impi/2017.4.239 Bowtie2/2.3.4.1
```

Recheck module status

```
$ module list
$ echo $PATH
$ bowtie2
$ which bowtie2
```

LMOD: Software Modules

Check module status (e.g. the alignment tool bowtie2)

```
$ module list
$ echo $PATH
$ bowtie2
```

Load the module

```
$ module load intel/2017c impi/2017.4.239 Bowtie2/2.3.4.1
```

Recheck module status

```
$ module list
$ echo $PATH
$ bowtie2
$ which bowtie2
```

Unloading modules

```
$ module unload Bowtie2/2.3.4.1
```

Dump all modules

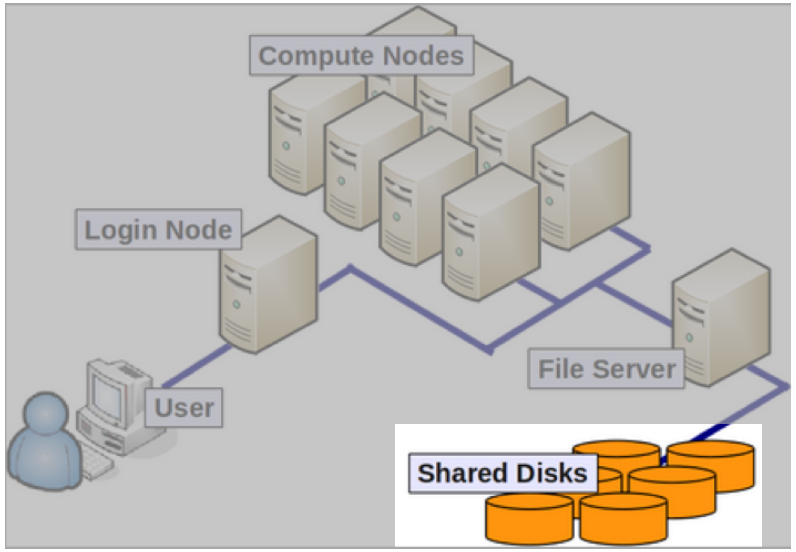
```
$ module purge
```

Filesystems and Storage

Data Storage: https://www.youtube.com/watch?v=rGH96xVQNyM

Moving data around: https://www.youtube.com/watch?v=n1I4-4TGVP8

Filesystems and storage



- Storage on HPC systems is organized differently than on your personal machine
- Physical disks are bundled together into a virtual volume; this volume may represent a single filesystem, or may be divided up, or partitioned, into multiple filesystems
- Filesystems are accessed over the internal network

Storage Grid

	Home Directories	Lab Storage	Local Scratch	Global Scratch	Persistent Research Data
Mount Point	/n/home#/ $USER	/n/pi_lab	/scratch	/n/$SCRATCH	/n/$REPOS
Size Limit	100GB	4TB+	70GB/node	2.4PB total	3PB
Availability	All cluster nodes + Desktop/laptop	All cluster nodes + Desktop/laptop	Local compute node only.	All cluster nodes	All cluster nodes
Retention Policy	Indefinite	Indefinite	Job duration	90 days	3-9 mo
Backup	Hourly snapshot + Daily Offsite	Daily Offsite	No backup	No backup	External Repos No backup
Performance	Moderate. Not suitable for high I/O	Moderate. Not suitable for high I/O	Suited for small file I/O intensive jobs	Appropriate for large file I/O intensive jobs	Appropriate for large I/O intensive jobs
Cost	Free	4TB Free + Expansion at a cost	Free	Free	Free

Storage Grid

	Home Directories
Mount Point	/n/home#/ $USER
Size Limit	100GB
Availability	All cluster nodes + Desktop/laptop
Retention Policy	Indefinite
Backup	Hourly snapshot + Daily Offsite
Performance	Moderate. Not suitable for high I/O
Cost	Free

- Your primary, private, space on the cluster.
- Custom software installations, job scripts, and other important data should live here.

https://docs.rc.fas.harvard.edu/kb/additional-groups/

Storage Grid

	Lab Storage
Mount Point	/n/pi_lab
Size Limit	4TB+
Availability	All cluster nodes + Desktop/laptop
Retention Policy	Indefinite
Backup	Regular Offsite
Performance	Moderate. Not suitable for high I/O
Cost	4TB Free + Expansion at a cost

- The primary shared space for a group/ lab.
- Datasets, lab work, and job results should live here.

Storage Grid

	Lab Storage
Mount Point	/n/pi_lab
Size Limit	4TB+
Availability	All cluster nodes + Desktop/laptop
Retention Policy	Indefinite
Backup	Regular Offsite
Performance	Moderate. Not suitable for high I/O
Cost	4TB Free + Expansion at a cost

- The primary shared space for a group/ lab.
- Datasets, lab work, and job results should live here.

Data Storage: https://www.youtube.com/watch?v=rGH96xVQNyM

Moving data around: https://www.youtube.com/watch?v=n1I4-4TGVP8

Cluster Help + Resources

- Documentation
  - https://docs.rc.fas.harvard.edu/
- Portal
  - http://portal.rc.fas.harvard.edu/rcrt/submit_ticket
- Email
  - rchelp@rc.fas.harvard.edu
- Office Hours
  - Wednesdays 12pm-3pm
  - https://www.rc.fas.harvard.edu/training/office-hours/
- Training
  - https://www.rc.fas.harvard.edu/upcoming-training/