HARVARD UNIVERSITY

FAS RC

FAS Research Computing
Division of Science
https://rc.fas.harvard.edu

New Users Training

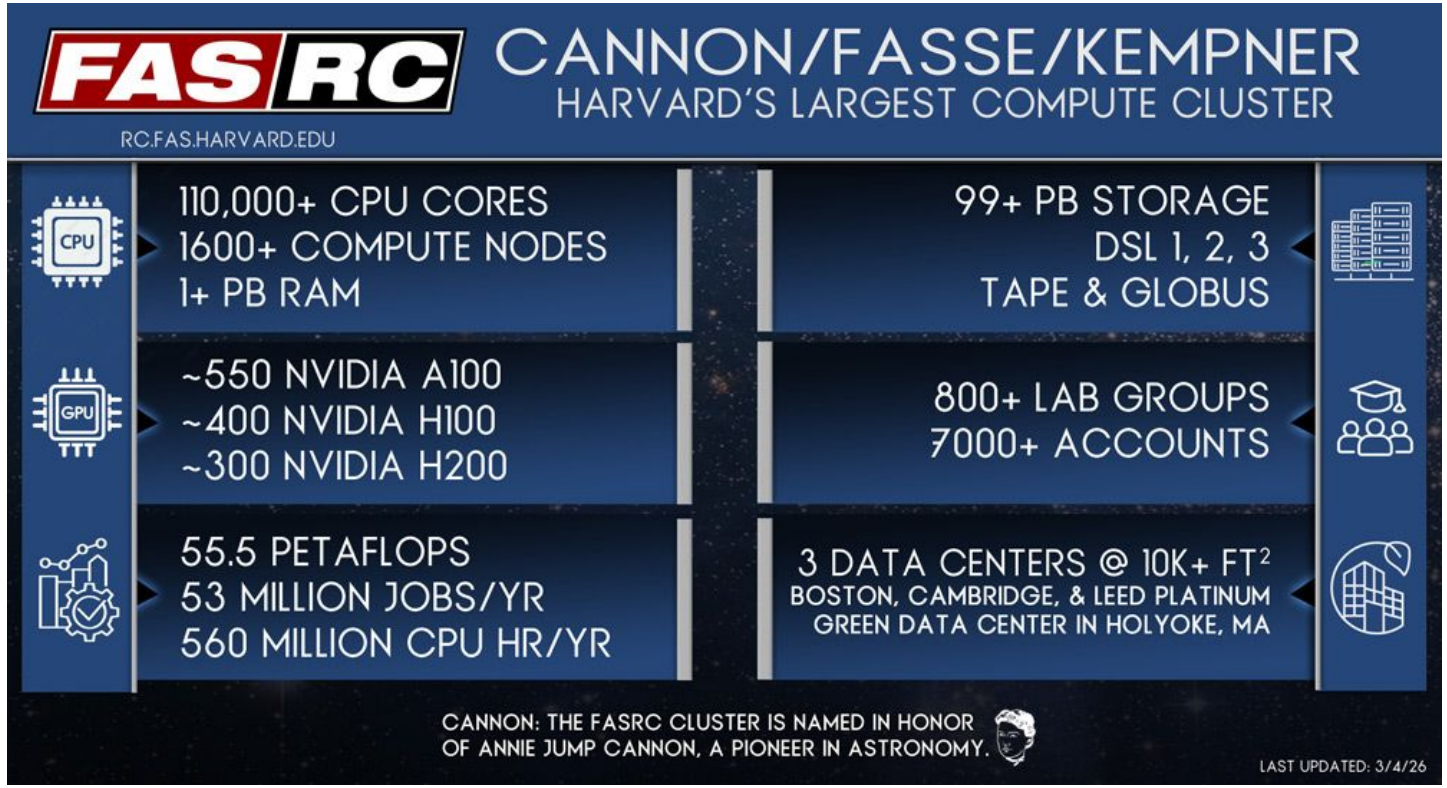
Introduction to FASRC clusters

IQSS

The Institute for Quantitative Social Science

What is FASRC?

- Compute Clusters
- Access and Authentication
- Storage Service
- Infrastructure
- Security
- Research Data Management
- Support and Training



Learning objectives 1 – FASRC account

- Learn how to request an FASRC account
- Activate your new account
- How to modify your account or add groups

Learning objectives 2 – Intro to HPC

- What is high-performance computing (HPC)?
- Laptop vs. Cannon
- Why HPC?
- FASRC clusters
- Cluster architecture
- Job scheduler
- Choose compute resources for jobs
  - Memory, cores
  - Partitions, file systems
- Storage
- Data Management
- Cluster customs and responsibilities

Learning objectives 3 – Documentation and help

- FASRC docs - https://docs.rc.fas.harvard.edu
- GitHub User Codes - https://github.com/fasrc/User_Codes
- Office Hours - https://rc.fas.harvard.edu/training/office-hours
- Tickets
  - Send email to rchelp@rc.fas.harvard.edu

FASRC Account

Request FASRC Account

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

1. Request Account

Use Account Request Tool:
portal.rc.fas.harvard.edu/request/account/new

- **Use HarvardKey option**

2. Set FASRC Password

Reset/Set your password via the portal:
portal.rc.fas.harvard.edu/p3/pwreset/

3. Set Two-Factor Auth (2FA)

Mandatory security setup instructions:
docs.rc.fas.harvard.edu/kb/openauth/

4. Set FASRC VPN

Required for storage, OOD, level 3 data, licenses.
docs.rc.fas.harvard.edu/kb/vpn-setup/

5. Review intro training

How to modify your account

Add a lab

- Portal for lab storage access: portal.rc.fas.harvard.edu
- Work for 1+ PI? Send a ticket for slurm account access.

Change labs

Request to move your primary lab affiliation:

docs.rc.fas.harvard.edu/kb/change-lab-group/

Never request a second account!

Requirement: Membership in the FASRC mailing-list is required.

Inactive Accounts

Account Status

Accounts are disabled after **9 months** of inactivity.
Can be reactivated with **PI/admin approval**.

Data Retention Policy

- Data is **not deleted immediately** upon account disablement.
- Home directories are deleted after **7 years** of account inactivity.
- **Only PIs** can delete lab data.

Full Policy Details:
https://docs.rc.fas.harvard.edu/kb/fas-rc-research-data-retention-and-deletion-policy/

Intro to HPC

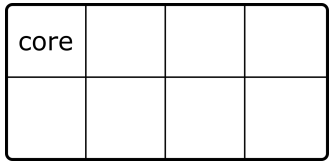
What is HPC?

- HPC: High performance computing
- HPC: biggest and fastest computing machines right now
- Supercomputers: rule of thumb - at least 100 times as powerful as a PC (personal computer)
- Jargon: other terms
  - Supercomputing
  - Cyberinfrastructure (CI)
  - Cluster computing

Laptop vs. Cannon

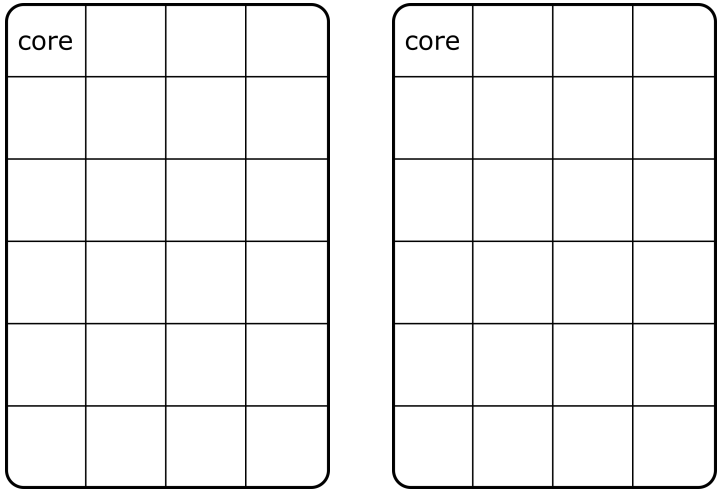
MacBook Pro

- 1 CPU (processor)
- 4-12 cores per CPU
- Memory: 16-96 GB



Cannon typical nodes

- 2 CPUs
- 48-112 cores per node
- Memory: 184-2000 GB
- **1600+ nodes!!!**



Why HPC?

- **Size:** problems that can't fit on a desktop/laptop, for example 500+ GB of RAM or 100s of cores
- **Speed:** problems that take months on a laptop may take a few hours on a supercomputer
- **Amount:** need 1000s of runs

45 miles/hour

600 miles/hour

FASRC clusters

Massachusetts Green HPC Center (MGHPCC)

Cannon cluster

From https://www.servethehome.com/the-harvard-cannon-powered-by-lenovo-neptune/

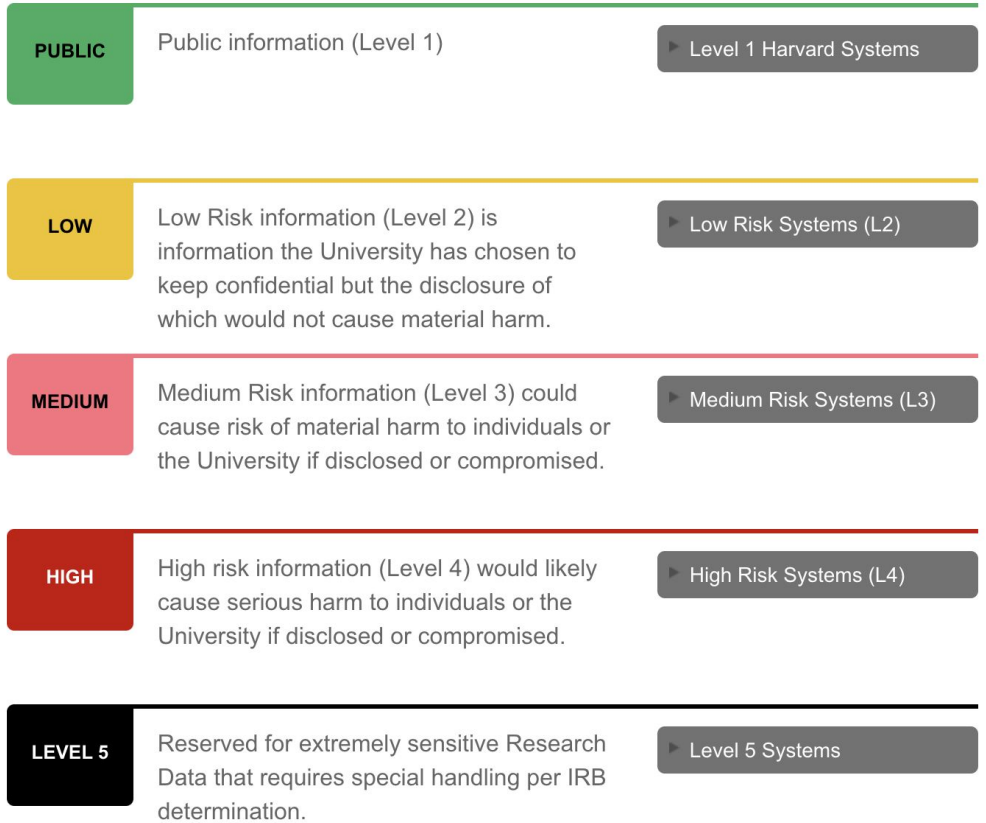
FASRC clusters: Cannon and FASSE

Cannon

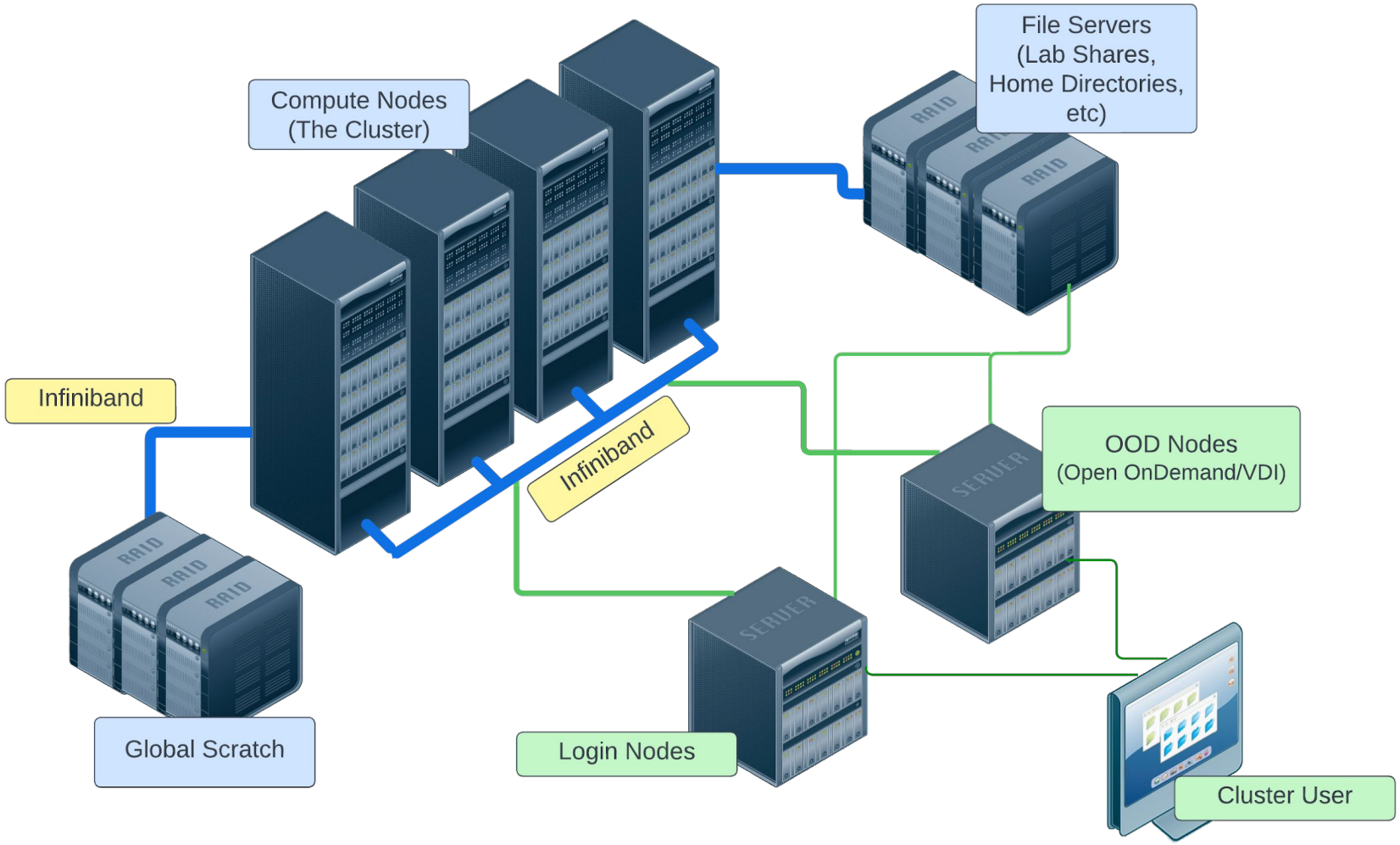
- General purpose
- Only level 1 and 2 data

FASSE

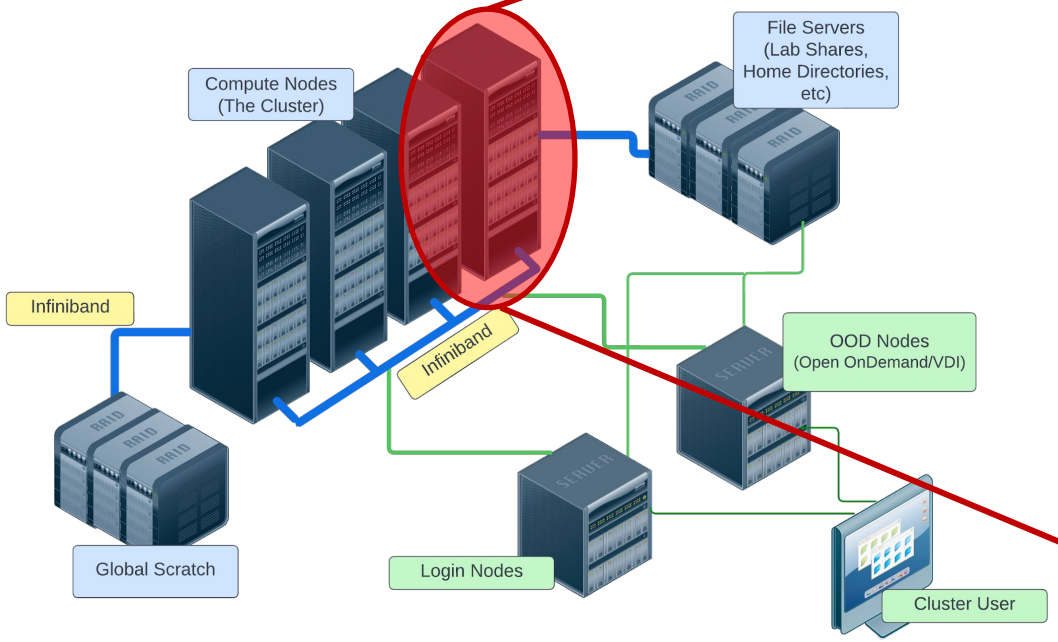
- FAS Secure Environment
  https://docs.rc.fas.harvard.edu/kb/fasse/
- Secure multi-tenant environment
- Analysis of sensitive datasets with DUAs and IRBs
- Level 3 data, no level 4 data
- PI/lab responsibility to know their data
- Information Security and Data Privacy
- DUA:
  https://docs.rc.fas.harvard.edu/kb/data-use-agreements/

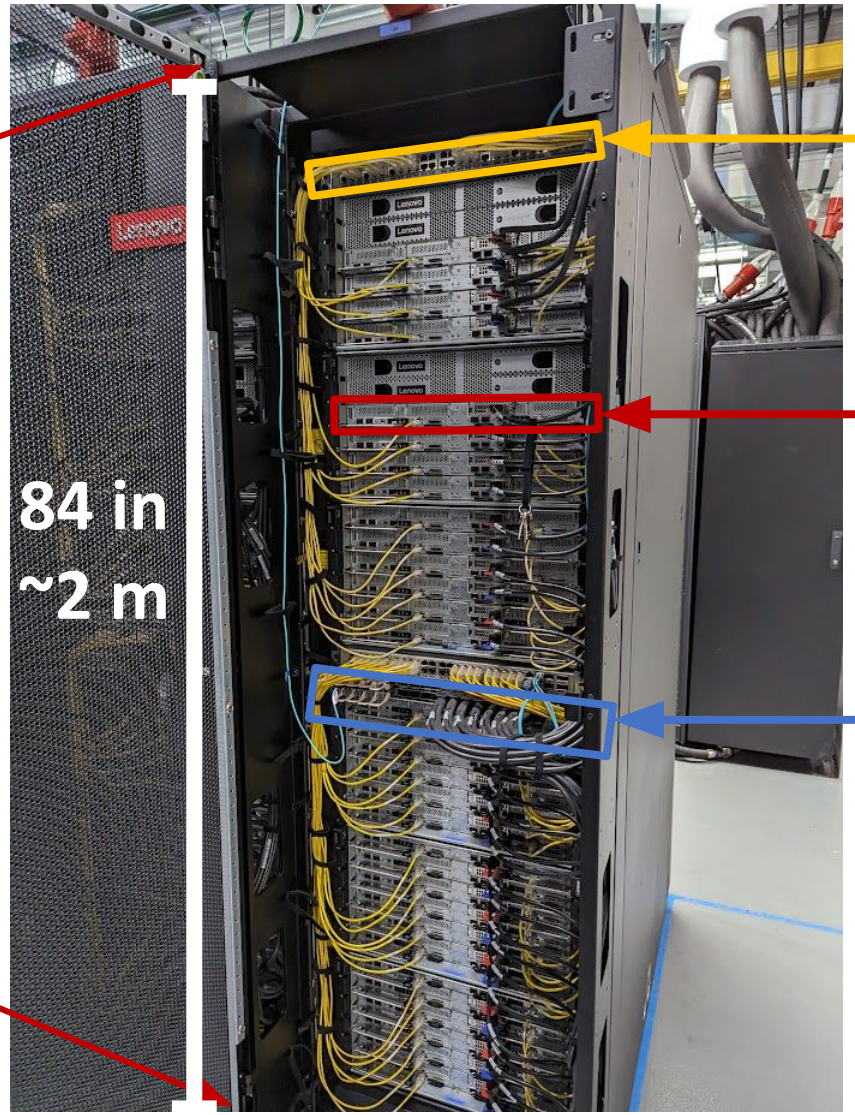


Cluster architecture



Rack



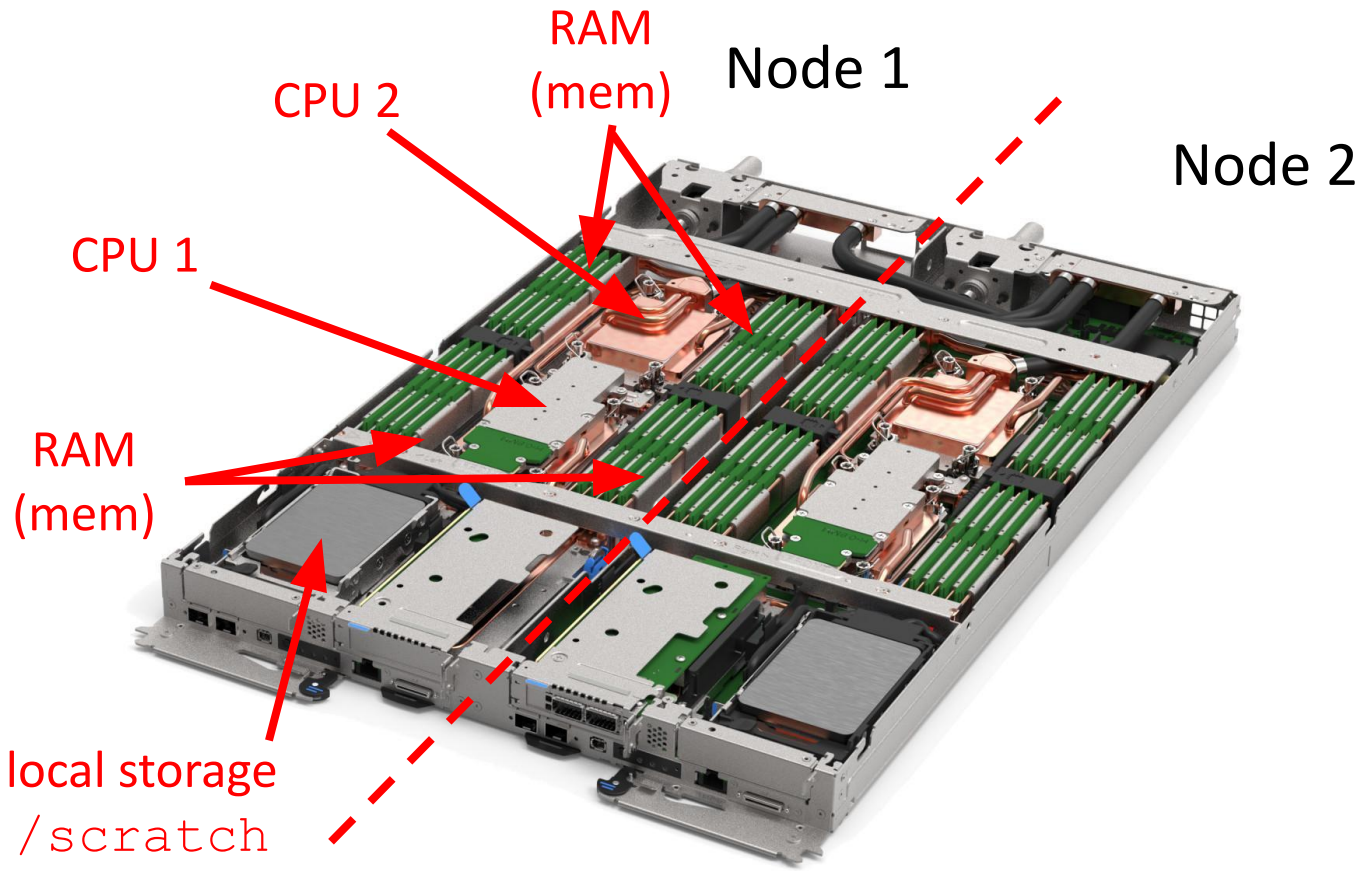


Ethernet switch

Compute nodes

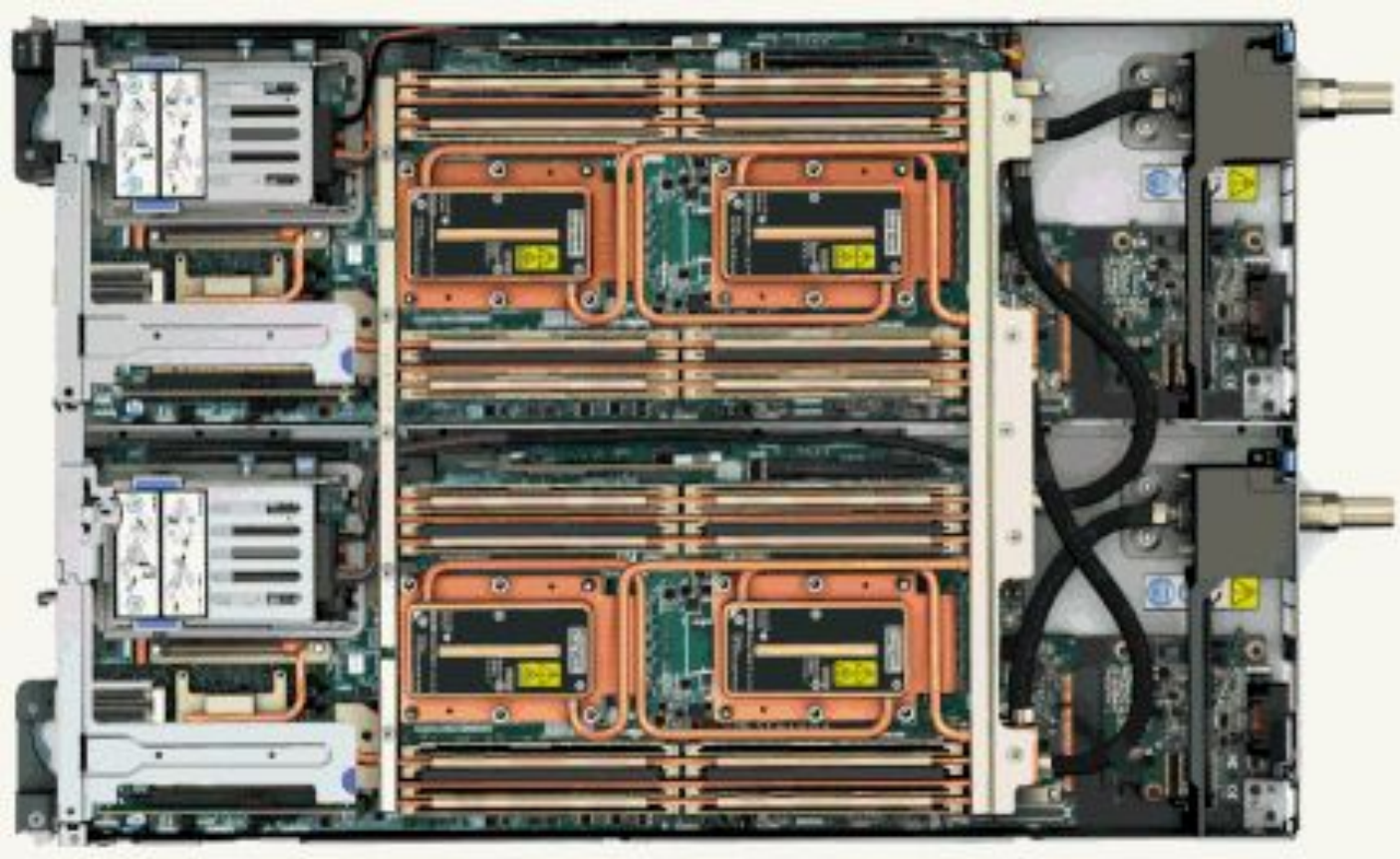
Infiniband switch

CPU Node components



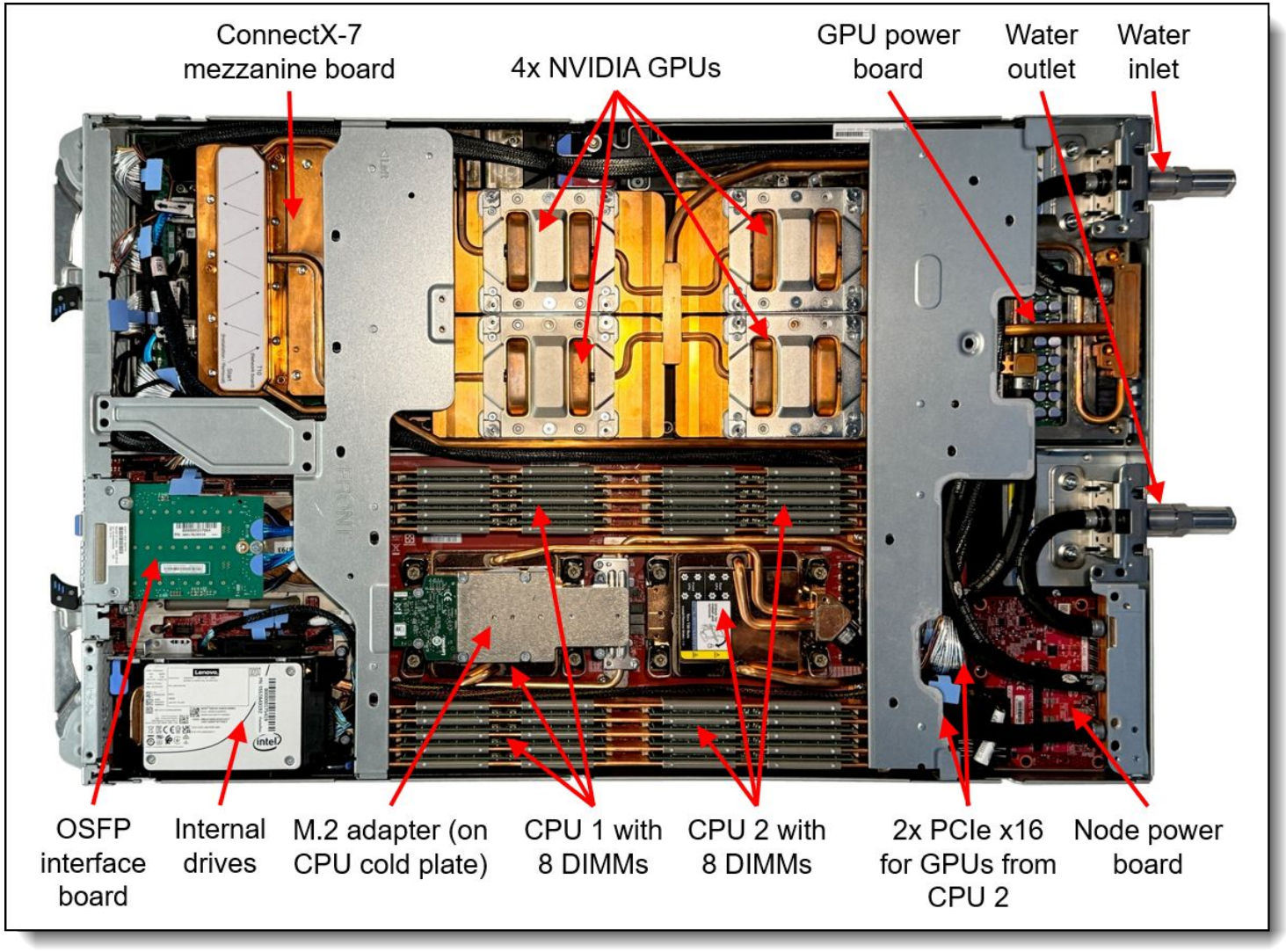
From https://lenovopress.lenovo.com/lp1603-thinksystem-sd650-v3-server

CPU nodes and water cooling



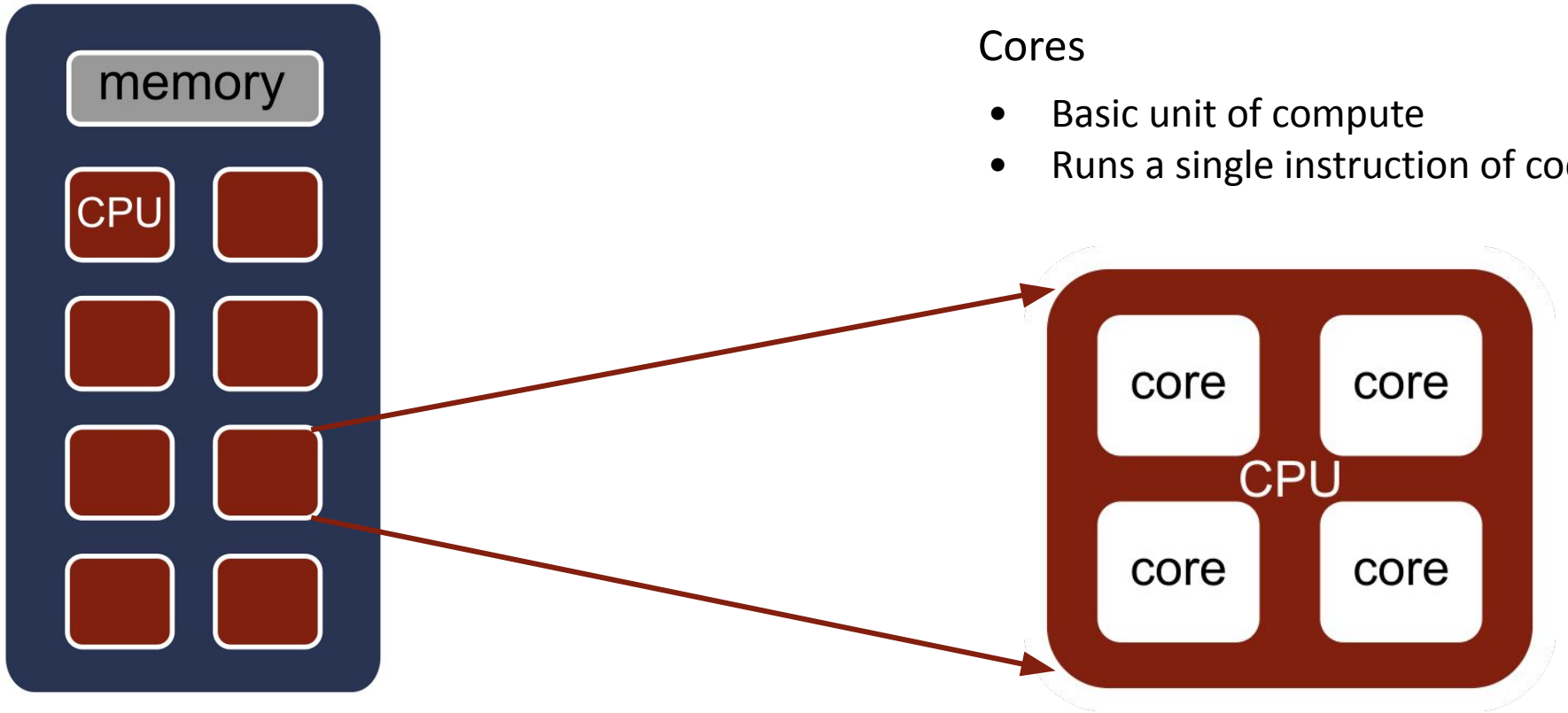
From https://www.rc.fas.harvard.edu/blog/cannon-makes-top-500-list/

GPU node


ConnectX-7 mezzanine board
4x NVIDIA GPUs
GPU power board
Water outlet
Water inlet
OSFP interface board
Internal drives
M.2 adapter (on CPU cold plate)
CPU 1 with 8 DIMMs
CPU 2 with 8 DIMMs
2x PCIe x16 for GPUs from CPU 2
Node power board

Node, processors, core

Node: a computer in the cluster



CPU

- Central processing unit, processor
- Can have many cores

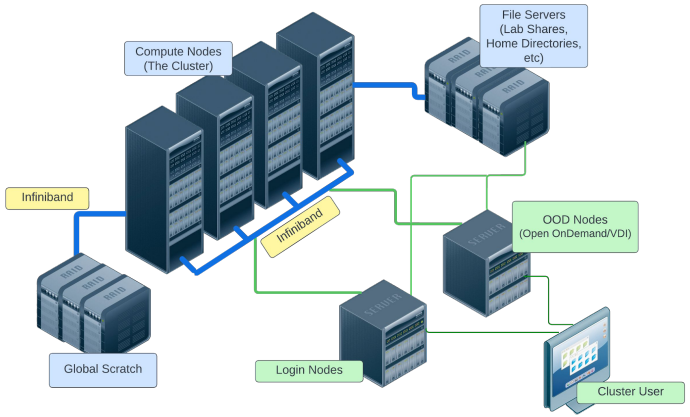
Cores

- Basic unit of compute
- Runs a single instruction of code

Nomenclature summary

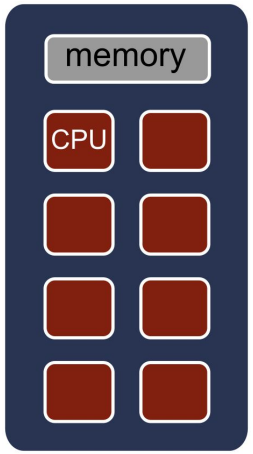
Cluster

Top level unit of a supercomputer



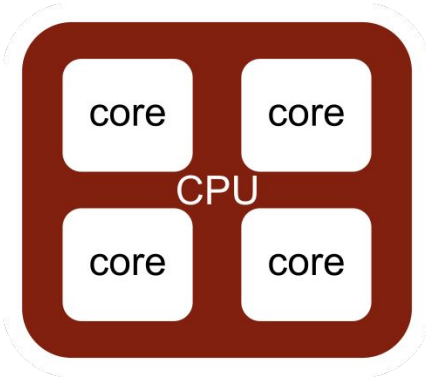
Node

One host in the cluster
(i.e., one computer)



Core

Basic unit of computer



New term: Job

A user's request to use a certain amount of resources for a specific amount of time

Glossary: https://docs.rc.fas.harvard.edu/kb/glossary/

Adapted from HPC@LSU training (http://www.hpc.lsu.edu/training/weekly-materials/2021-Summer/HPC_UserEnv_2021_Summer_session_2.pdf)

Job scheduler

- The Cluster is a multi-tenant environment, so how can everyone use it fairly?
- Job scheduler!
- Slurm: Simple Linux Utility for Resource Management
  - Manages job queue for a cluster of resources
  - Prioritizes jobs
  - Provides status of running, queue, completed and failed jobs
  - Determines the order jobs are executed
  - On which node(s) jobs are executed

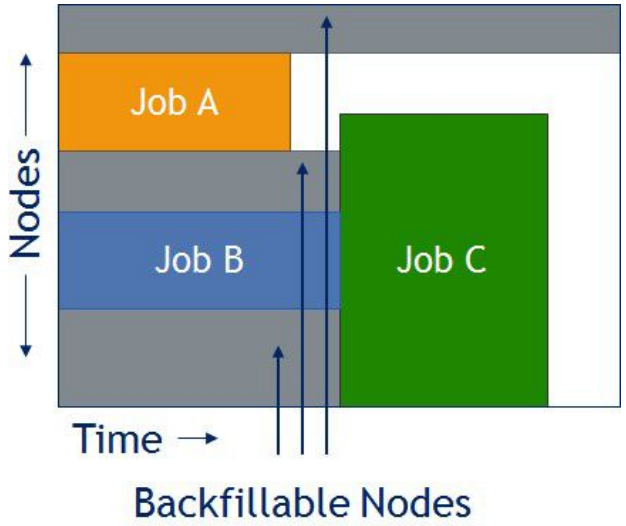
Adapted from HPC@LSU training (http://www.hpc.lsu.edu/training/weekly-materials/2021-Summer/HPC_UserEnv_2021_Summer_session_2.pdf)

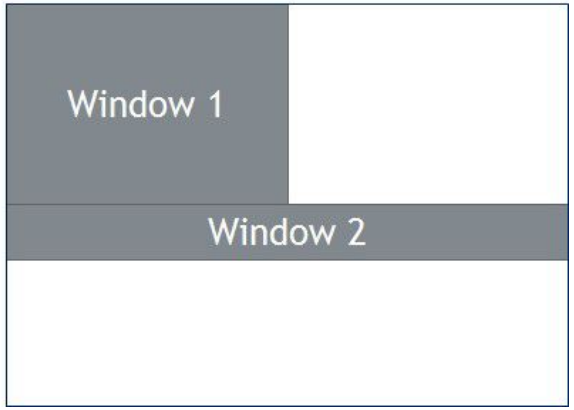
Job management philosophy

- Prioritize workload
- Backfill idle node to maximize cluster use

Job Priority

- **Not** first come, first served
- Job with higher priority scheduled ahead of jobs with lower priority
- Priority depends on
  - Group Fairshare
  - Amount of time pending

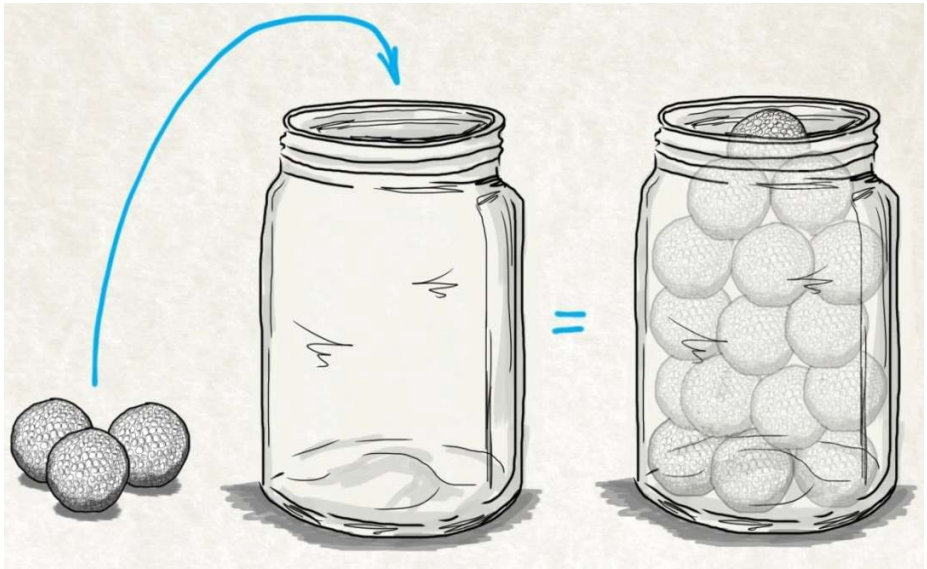




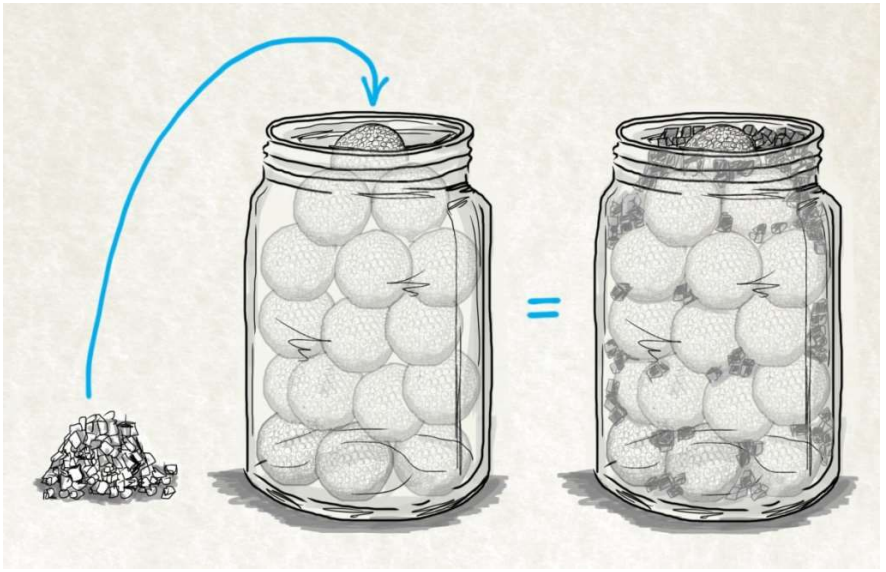
Adapted from HPC@LSU training
(http://www.hpc.lsu.edu/training/weekly-materials/2021-Summer/HPC_UserEnv_2021_Summer_session_2.pdf)

How to maximize cluster usage?

1. Fill in high-priority jobs



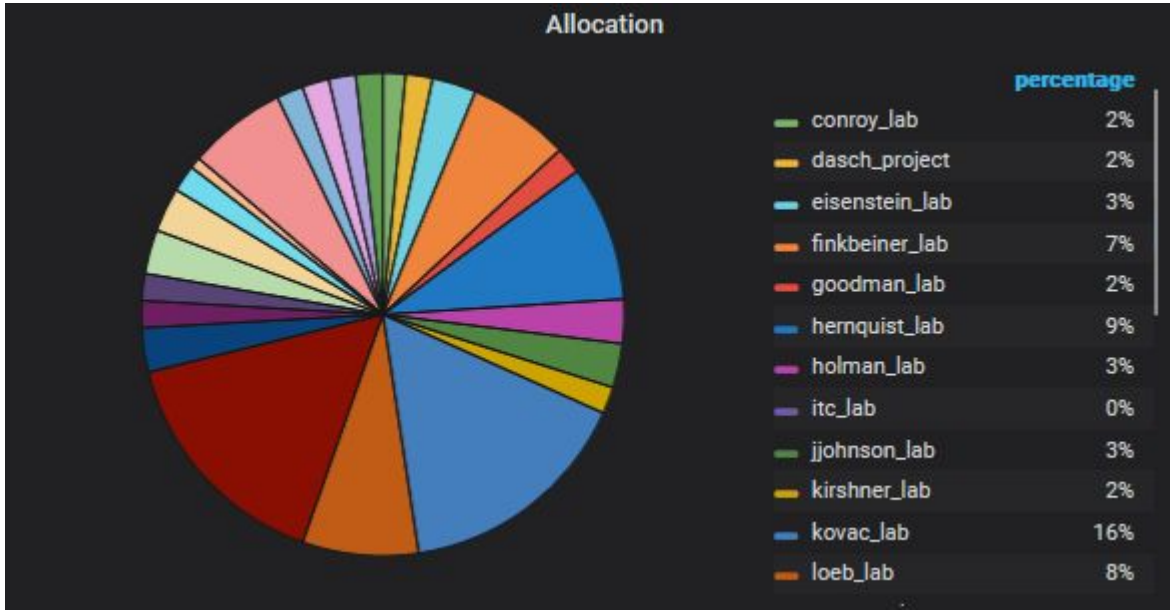
2. Backfill with low-priority jobs

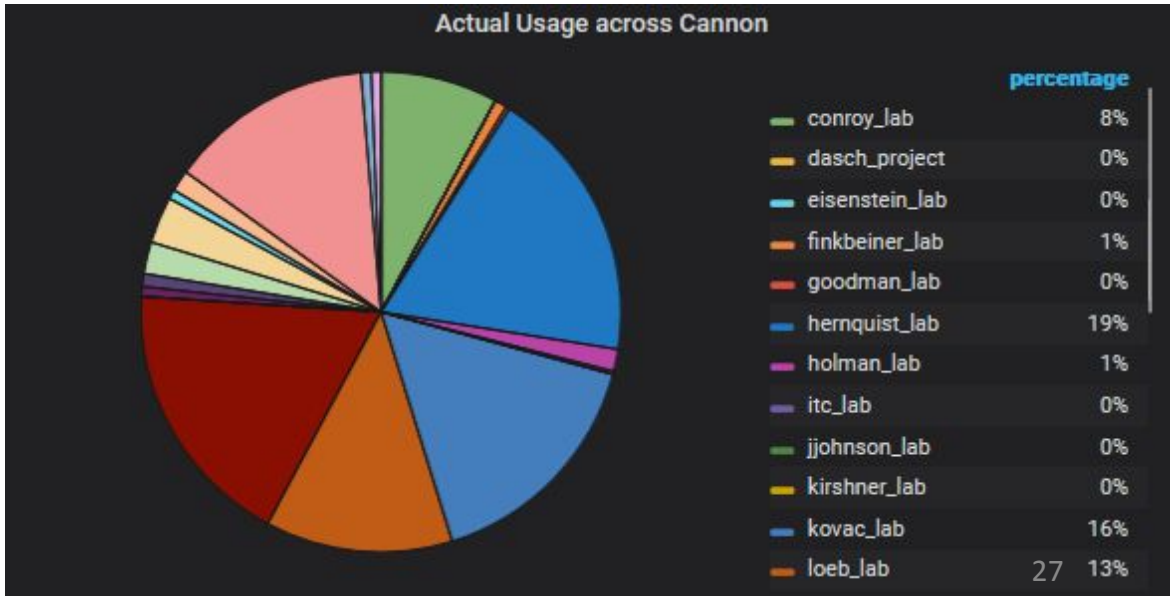


Adapted from HPC@LSU training
(http://www.hpc.lsu.edu/training/weekly-materials/2021-Summer/HPC_UserEnv_2021_Summer_session_2.pdf)

Fairshare

- A method for ensuring the equitable use of a cluster
- The fraction of the cluster a user/group gets
- The score assigned by Slurm to a user/group based on usage
- Priority that users/groups get based on usage





General fairshare principles

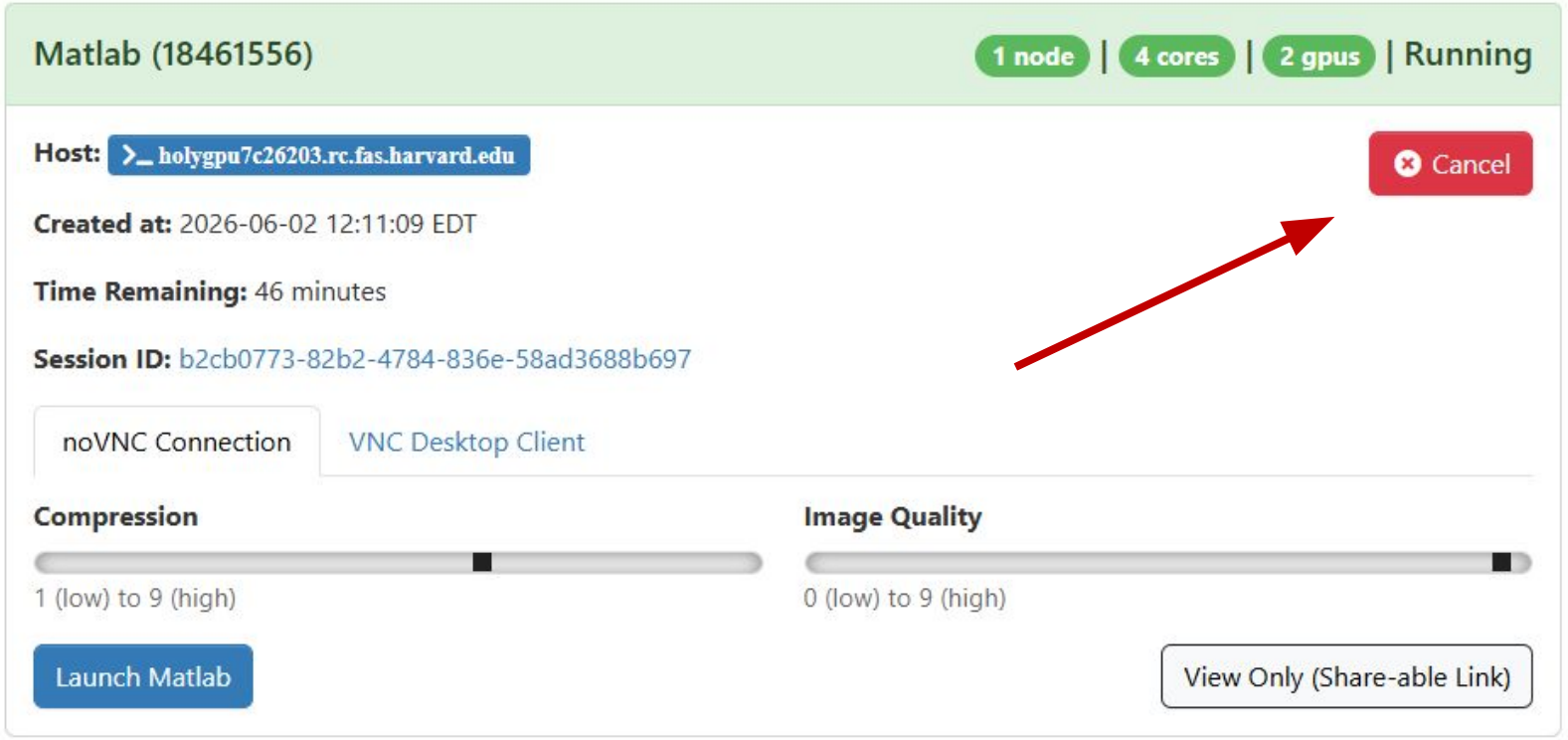
- Fairshare is affected by how much cpu, memory, GPU, and time you request in order to run your calculations.
- GPUs eat up fairshare 200x-500x as fast as CPUs
- Your usage affects every person in your group
- Using either the `test` or the `gpu_test` partitions does not affect fairshare, but you are limited in terms of time and the number of jobs you can submit
- Leaving a session running in OOD consumes resources. You should cancel/delete it if you're not still using it.

Things you can do to decrease your impact on fairshare

- Use fewer GPUs. They are 200x-500x more "expensive" than CPUs in terms of fairshare
- Request less memory
- Request fewer CPUs
- Run your jobs for a shorter time period
- Leaving a session running in OOD consumes resources. You should cancel it if you're not still using it
- Wait. Coordinate jobs with lab members. Space jobs accordingly

Cancelling an OOD job

Press the Cancel button



Choosing computational resources

- How do we choose memory, cores, partitions, and file systems?
- First time ever running on a cluster?
  - Run a test case choosing similar resources as the machine (laptop or desktop) you are currently using
  - Check how efficient your job was and adjust it accordingly
- Increasing a job/analysis/simulation?
  - Run for a small test case (~1h)
  - Increase size by 1.5, 2.0, 2.5x and check how job scales
  - Then you can have a rough estimation of how much a first trial production job of ~10x would require

Cannon “test” partitions

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

Partitions	test	gpu_test
Time Limit	12 h	12 h
# Nodes	18	12
# Cores / Node / GPU	112	112 + 8 A100 MIG
Memory / Node (GB)	990	487

Cannon “cpu” partitions

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

Partitions	sapphire	shared	intermediate	unrestricted
Time Limit	3 days	3 days	3-14 days	14+ days - none
# Nodes	186	310	12	8
# Cores / Node	112	48	112	48
Memory / Node (GB)	990	184	990	184

Cannon “gpu” partitions

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

Partitions	gpu	gpu_h200
Time Limit	3 days	3 days
# Nodes	36	22
# Cores / Node / GPU	64 + 4 A100	112 + 4 H200
Memory / Node (GB)	990	990

Cannon “big memory” partitions

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

Partitions	bigmem	bigmem_intermediate
Time Limit	3 days	3 days
# Nodes	4	3
# Cores / Node	112	64
Memory / Node (GB)	1988	2000

Cannon HSPH partitions

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

Partitions	hsph	hsph_gpu
Time Limit	3 days	3 days
# Nodes	56	2
# Cores / Node / GPU	112	96 + 4 H100
Memory / Node (GB)	990	1500

Cannon “requeue” partitions

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

Partitions	serial_requeue	gpu_requeue
Time Limit	3 days	3 days
# Nodes	varies	varies
# Cores / Node	varies	varies
Memory / Node (GB)	varies	varies

Cannon “other” partitions

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

Partitions	remoteviz	PI_lab
Time Limit	3 days	varies
# Nodes	1	varies
# Cores / Node / GPU	32 + V100 for rendering	varies
Memory / Node (GB)	373	varies

FASSE partitions

Documentation: https://docs.rc.fas.harvard.edu/kb/fasse/

Partitions	test	fasse	serial_requeue	fasse_gpu	fasse_gpu_h200	fasse_bigmem	fasse_ultramem	remoteviz	**pi_lab**
Time Limit	12 h	7 days	7 days	7 days	**3 days**	7 days	7 days	7 days	**varies**
# Nodes	5	42	varies	2	2	17	1	1	**varies**
# Cores / Node	48	48	varies	64 + 4 A100	112 + 4 H200	64	64	32	**varies**
Memory / Node (GB)	184	184	varies	487	990	499	2000	373	**varies**

cpu gpu memory

Which partitions can I use?

Documentation: https://docs.rc.fas.harvard.edu/kb/convenient-slurm-commands/

```
[jharvard@boslogin02 ~]$ spart
```

Partition	State	Cores	GPUs	Average Mem/Node(GB)	Nodes	Time Limit
bigmem	UP	448	0	2015	4	3-00:00:00
bigmem_intermediate	UP	192	0	2015	3	14-00:00:00
gpu	UP	2304	144	1007	36	3-00:00:00
gpu_h200	UP	2464	88	1007	22	3-00:00:00
gpu_requeue	UP	24240	1356	1174	316	3-00:00:00
gpu_test	UP	768	96	503	12	12:00:00
intermediate	UP	1344	0	1007	12	14-00:00:00
remoteviz	UP	32	0	377	1	3-00:00:00
sapphire	UP	20832	0	1007	186	3-00:00:00
serial_requeue	UP	110416	1356	607	1590	3-00:00:00
shared	UP	17760	0	188	370	3-00:00:00
test	UP	2016	0	1007	18	12:00:00
unrestricted	UP	384	0	188	8	365-00:00:00

Storage

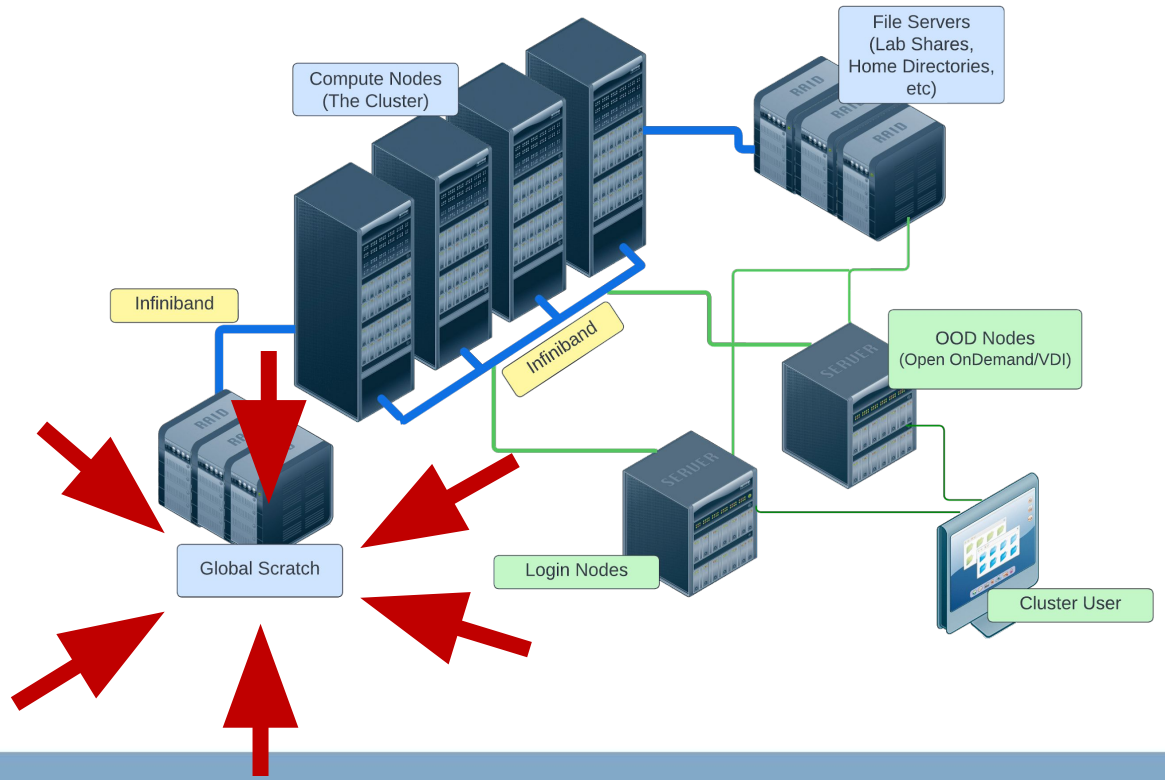
Tier storage documentation: https://www.rc.fas.harvard.edu/services/data-storage/

	Home Directories	Lab Directory (Startup)	Local Scratch	Global Scratch	Tier Storage
Mount Point	/n/home#/$USER /n/home_fasse/$USER	/n/holylabs/pi_lab	/scratch	/n/netscratch/pi_lab	/n/pi_lab
Variable	$HOME			$SCRATCH	
Size Limit	100GB, cannot expand	4TB, cannot expand	70+ GB/node	50 TB, cannot expand	Based on Tier
Availability	All cluster nodes + Desktop/laptop	All cluster nodes	Local compute node only	All cluster nodes	All cluster nodes/ mountable
Retention Policy	Indefinite	Indefinite	Job duration	90 days	Indefinite
Backup	Hourly snapshot + Daily Offsite	No backup	No backup	No backup	Depending on Tier
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	Depending on Tier
Cost	Free	Free	Free	Free	Paid

Storage schematics

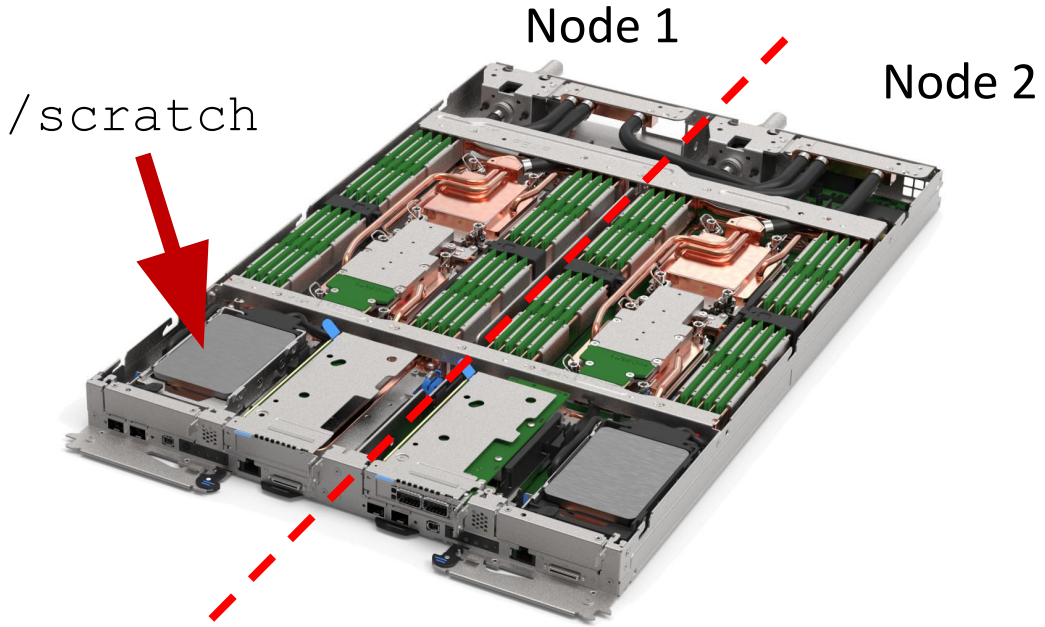
Global Scratch

- Networked scratch
- Global variable: `$SCRATCH`
- Path: `/n/netscratch/pi_lab`



Local Scratch

- Storage on the node
- Path: `/scratch`

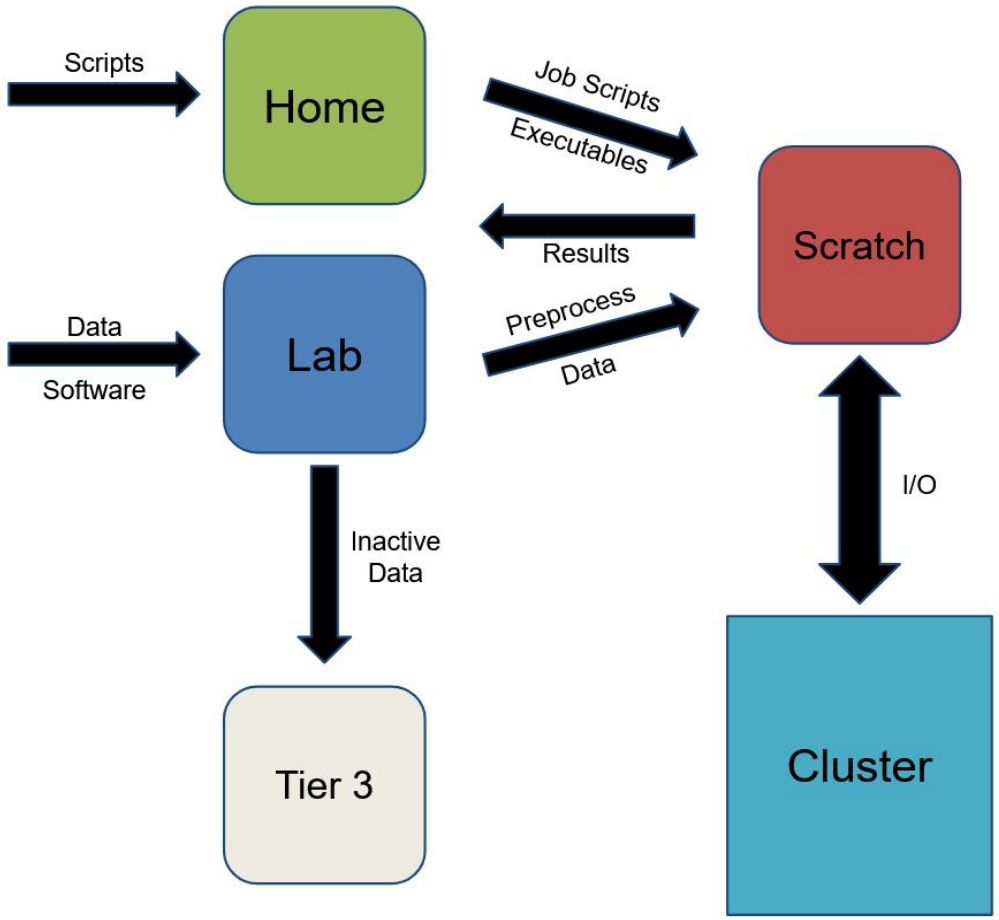


From https://lenovopress.lenovo.com/lp1603-thinksystem-sd650-v3-server

Data management

Documentation:
https://docs.rc.fas.harvard.edu/kb/data-storage-workflow-rdm/

- Home
  - Backed up with daily snapshots (up to 2 weeks)
  - "Valuable" and small code
- Global scratch
  - Temporary storage
  - Copy job scripts and executables for jobs
  - Input data, output results
  - Do not have multiple jobs hitting the same file!!
- Lab storage
  - Permanent storage
  - If you have code here and not backed up, use version control (git)!!
- Specific training about Research Data Management at FASRC - check out Training Calendar!



Cluster customs and responsibilities (1)

Documentation: https://docs.rc.fas.harvard.edu/kb/responsibilities/

- Don’t run anything on the login nodes
- Be as accurate as possible for memory requests
- Keep job counts reasonable: 10,100 job limit per user (scheduled or running)
- Request at least 10 minutes
- Don’t overwhelm scheduler: wait 0.5 to 1 sec for `sbatch` and `sacct` commands
- No production work on test partitions

Cluster customs and responsibilities (2)

Documentation: https://docs.rc.fas.harvard.edu/kb/responsibilities/

- Use appropriate partition
- Use `serial_requeue` and `gpu_requeue` when possible
- Heavy I/O should be done on `/scratch` and `$SCRATCH`
- Keep at most 1000 files per directory (i.e., folder)
- Do not `touch` files on /n/netscratch
- Poorly behaved jobs will be terminated
- Don’t mine digital currency or misuse Harvard resources

Acknowledge using the FASRC Clusters

Documentation: https://docs.rc.fas.harvard.edu/kb/attribution/

If you publish work performed on FASRC clusters, please acknowledge it and let us know:

> *"The computations in this paper were run on the FASRC cluster supported by the FAS Division of Science Research Computing Group at Harvard University."*

Training session evaluation

Please, fill out our training session evaluation. Your feedback is essential for us to improve our trainings!!

https://tinyurl.com/FASRC-training

FASRC documentation

- FASRC docs: https://docs.rc.fas.harvard.edu/
  - If searching on Google add FASRC to your search
- GitHub User_Codes: https://github.com/fasrc/User_Codes/
- Getting help
  - Office hours: https://www.rc.fas.harvard.edu/training/office-hours/
  - Ticket: send email to rchelp@rc.fas.harvard.edu
    - include as much detail as possible
    - please send screenshots, full pathnames, complete word for word error messages, refer to previous relevant tickets, etc.

Upcoming training sessions

Training calendar: https://www.rc.fas.harvard.edu/upcoming-training/

Getting started on the FASRC clusters with Open OnDemand
June 11, 1:00-2:00 pm, Virtual

- Audience
  - New users not familiar with command-line interface
  - Wants to use a GUI
- Requirements
  - Single-node jobs
  - Working FASRC account with cluster access
- Content
  - Access Open OnDemand
  - Launch Jupyter, Rstudio Server, Remote Desktop
  - Install Rstudio Server packages
  - Install python packages for Jupyter
  - Launch software from Remote Desktop

Getting started on the FASRC clusters with command line interface (CLI)
June 12, 1:00-2:00 pm, Virtual

- Audience
  - Users familiar with command-line interface
  - New to Cannon and FASSE, but familiar with HPC systems
- Requirement: working FASRC account with cluster access
- Content
  - Submit interactive job with salloc
  - Submit batch job sbatch
  - Monitor jobs
  - Cluster software overview (modules, spack)

Upcoming training sessions

Training calendar: https://www.rc.fas.harvard.edu/upcoming-training/

AI-ready Data on FASRC Clusters
June 26th, 1:00-2:30 pm, Virtual

- Audience
  - FASRC cluster users
  - Anyone working with AI
- Requirements
  - Working FASRC account with cluster access
- Content
  - What is AI-ready data is?
  - How to obtain AI-ready data?
  - Distinction from traditional data preparation
  - Necessary data-management practices to manage AI-ready data

Training session evaluation

Please, fill out our training session evaluation. Your feedback is essential for us to improve our trainings!!

https://tinyurl.com/FASRC-training

Thank you :)

FAS Research Computing