Introduction to High Performance Computing

A crash course on using the HPC system at the University of Arizona

Outline

1. HPC Overview
    a. What is HPC?
    b. Common Misconceptions
2. System Access
    a. Creating an account
    b. Browser Access
    c. Command Line Access
3. System Layout
    a. Node Types and Acceptable Use
4. Storage and Transfers
    a. File system overview
    b. Transfer Methods
5. Software
    a. System Modules
    b. User Installations
6. Accessing Compute
    a. GUI Jobs
    b. Interactive Jobs
    c. Batch Jobs

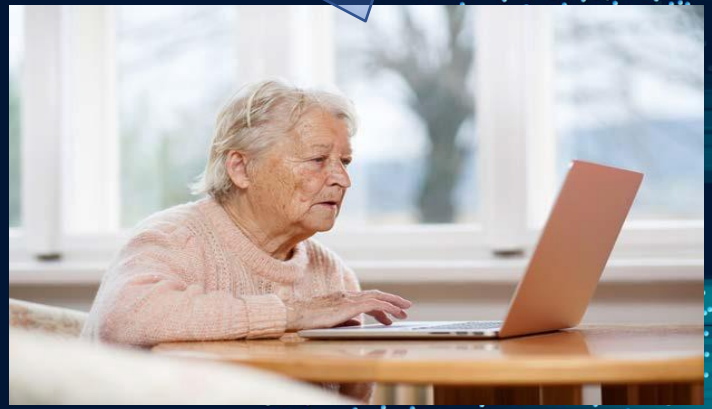
and many more!

HPC Overview

Why HPC?

Research is easy!

It’s still running…



Why HPC?

Problems

- Computation takes too long

  → *Get a more powerful computer*

- Computation is too big

  → *Link multiple computers*

- Too many computations

  → *Use a separate one for each job*

What is HPC?

Modern instrument for High-Performance Computing is a **cluster**, consisting of lots of connected individual computers (nodes).

Supercomputer is a commonly used nickname.

What Is HPC?

Laptop

Supercomputer

What Is HPC?

Laptop

Supercomputer

What Is HPC?

Laptop : Personal

Supercomputer : Shared

What Is HPC?

Laptop : Local

Supercomputer : Remote

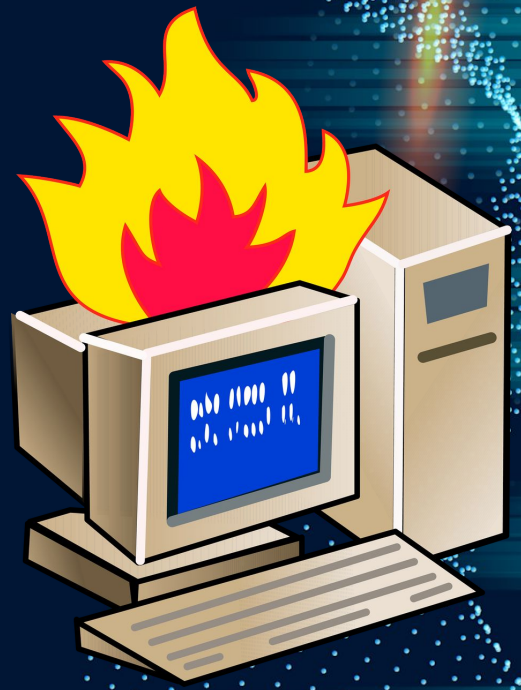
What Is HPC?

Laptop : Hands On

Supercomputer : Hands Free

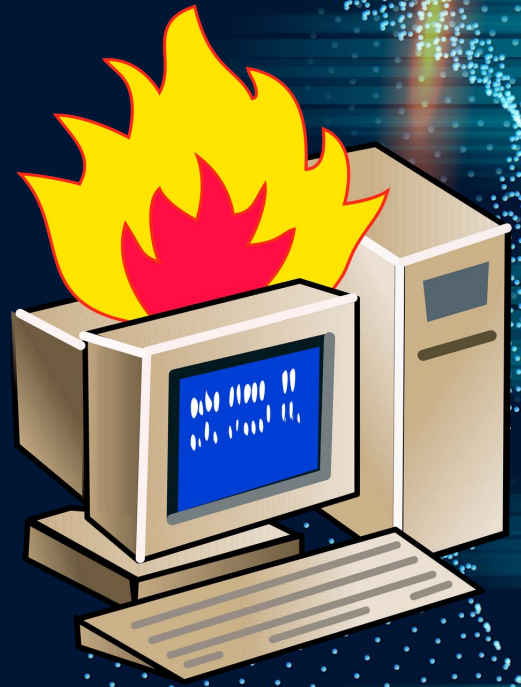
Common Misconceptions

1. My code will automatically run faster
    - *HPC's power comes from **parallelization** - make sure your software has this enabled!*
2. All nodes on a supercomputer are the same
    - *Don't run computations on the **login node**!*
3. N times more CPUs for a job = N times faster
    - *Does your program know those **CPUs** exist?*
    - ***Scaling** can be complicated!*



Common Misconceptions

4. I can perform any action on the HPC as I could on my own system
    a. *HPC is a **shared system**, and you are not **root**!*
5. I am not allowed to install anything on the HPC
    a. *System directories are off limits, but you are in control of your home folder!*



Puma

Implemented in the middle of 2020, Puma is the biggest cat yet. Similar to Ocelote, it has standard CPU nodes (with 94 cores and 512 GB of memory per node), GPU nodes (with Nvidia V100) and two high-memory nodes (3 TB). Local scratch storage increased to ~1.4 TB. Puma runs on CentOS 7.

As is the case for our other supercomputers, we use the RFP process to get the best value for our financial resources, that meet our technical requirements. This time Penguin Computing one with AMD processors. This is tremendously valuable as each node comes with:

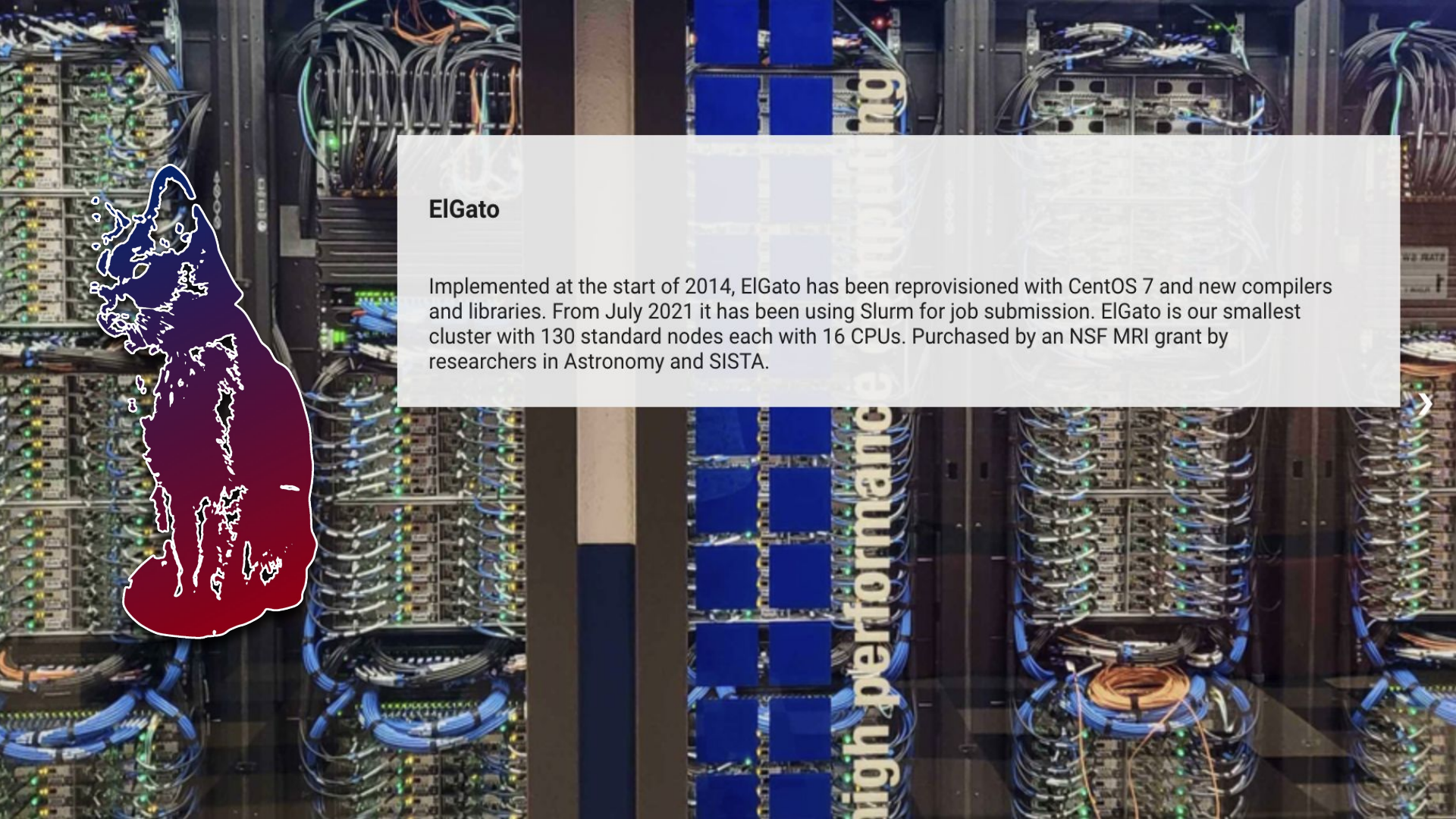
- Two AMD Zen2 48 core processors
- 512GB RAM
- 25Gb path to storage
- 25Gb path to other nodes for MPI
- 2TB internal NVME disk (largely available as /tmp)
- Qumulo all flash storage array for shared filesystems
- Two large memory nodes with 3TB memory and the same processors and memory as the other nodes
- Six nodes with four Nvidia V100S GPU's each

Ocelote

Ocelote arrived in 2016. Lenovo's Nextscale M5 technology was the winner of the RFP mainly on price, performance and meeting our specific requirements. Ocelote has one large memory node with 2TB of memory and 46 nodes with Nvidia P100 GPUs for GPU-accelerated workflows. This cluster is actually the next generation of the IBM cluster we call ElGato. Lenovo purchased IBM's Intel server line in 2015.

In 2021, Ocelote's operating system was upgraded from CentOS6 to CentOS7 and was configured to use SLURM, like Puma. It will continue until it is either too expensive to maintain or it is replaced by something else.

- Intel Haswell V3 28 core processors
- 192GB RAM per node
- FDR infiniband for fast MPI interconnect
- Qumulo all flash storage array (all HPC storage is integrated into one array)
- One large memory node with 2TB RAM, Intel Ivy Bridge V2 48 cores
- 46 nodes with Nvidia P100 GPU's



ElGato

Implemented at the start of 2014, ElGato has been reprovisioned with CentOS 7 and new compilers and libraries. From July 2021 it has been using Slurm for job submission. ElGato is our smallest cluster with 130 standard nodes each with 16 CPUs. Purchased by an NSF MRI grant by researchers in Astronomy and SISTA.

Check In!

When we say the HPC is more like a fleet of Dodge Chargers than an individual Ferrari, what does this mean?

What is the name of the most advanced HPC cluster?

System Access

Outline

1. HPC Overview
    a. What is HPC?
    b. Common Misconceptions
2. **System Access**
    a. Creating an account
    b. Browser Access
    c. Command Line Access
3. System Layout
    a. Node Types and Acceptable Use
4. Storage and Transfers
    a. File system overview
    b. Transfer Methods
5. Software
    a. System Modules
    b. User Installations
6. Accessing Compute
    a. GUI Jobs
    b. Interactive Jobs
    c. Batch Jobs

UA HPC Accounts

PI/Sponsor Accounts

- mostly faculty
- can create and own groups
- receive storage and CPU time allocations

Research Groups

- access both storage and full CPU

Class Groups

- access only storage and limited CPU

Normal/Sponsored Accounts

- students, post-docs, etc
- must request sponsorship from a PI
- can be added to groups
- given access to storage and compute resources according to group membership

What are Groups?

Groups are ways to organize users on the HPC

Research Group membership provides:

- HPC account sponsorship (if not already obtained)
- Access to group owner's full compute allocation
- Access to group owner's storage allocations

Class Groups…

- are intended for a single semester
- provide limited access to compute and storage

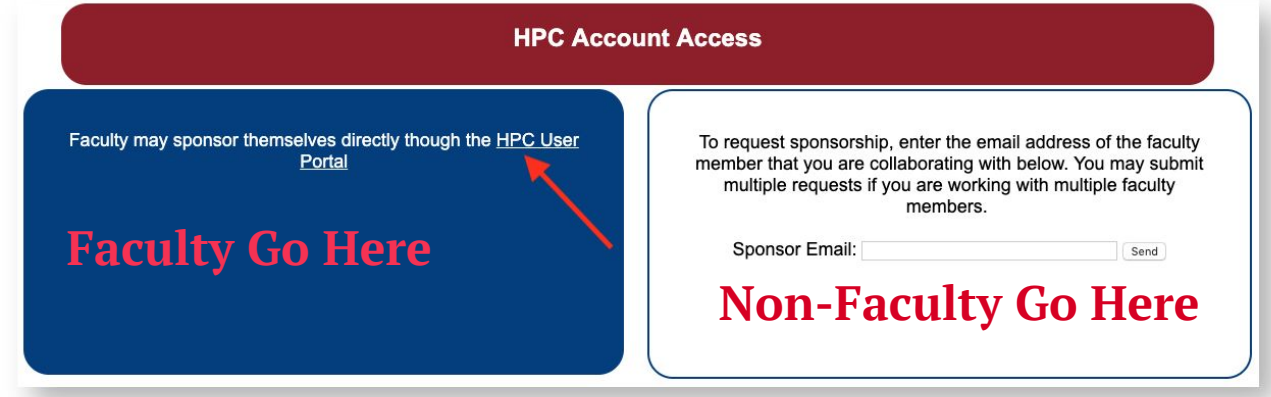
How to Create Your Account: *UA Affiliates*

Step 1: Account Creation

- Visit https://portal.hpc.arizona.edu/. This will automatically create an HPC account for you.

Step 2: Sponsorship

- Go to https://portal.hpc.arizona.edu/portal/sendlink.php



How to Create Your Account: ***Non****-UA Affiliates*

If you or a member of your lab are not officially affiliated with the UA, you can request status as a

Designated Campus Colleague (DCC)

https://hr.arizona.edu/hr-resources/training-guides/designated-campus-colleagues-guides

Once you have created you account and received sponsorship, you can access the HPC!

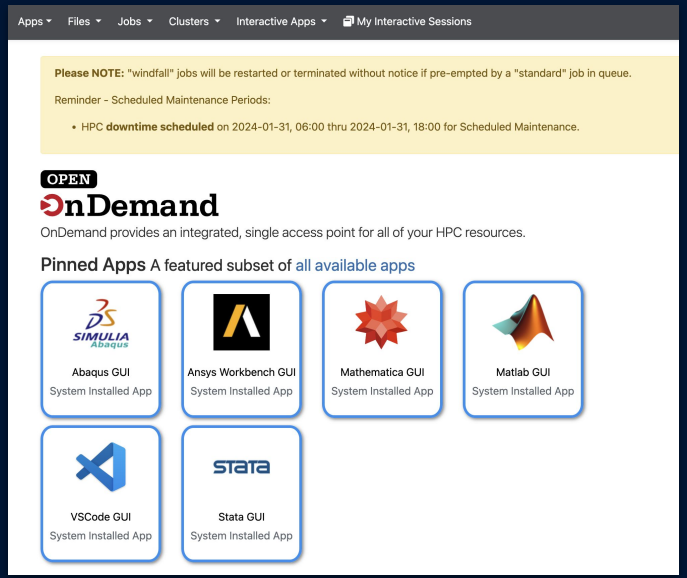
Check In

Give a ✅ reaction if you have already created an account and been sponsored

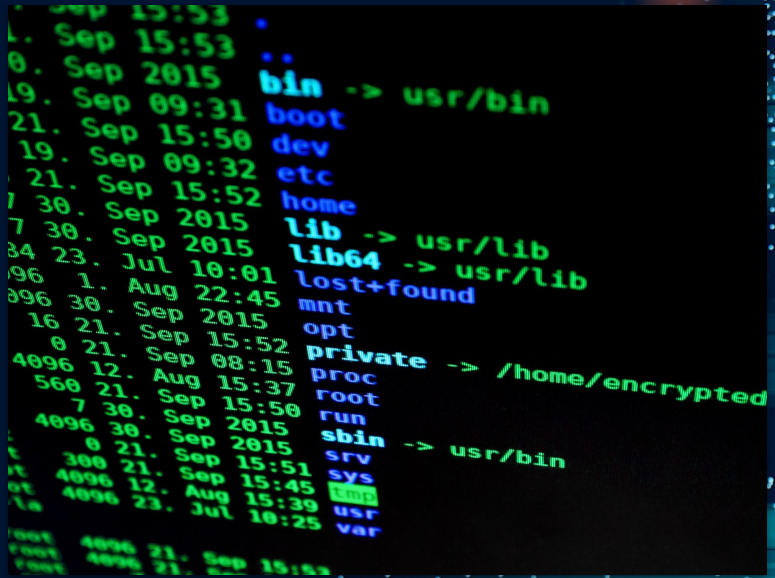
Give a ❌ reaction if you don’t have an HPC account yet

Connecting to the HPC

Method 1: Browser



Method 2: Command Line



Open On Demand

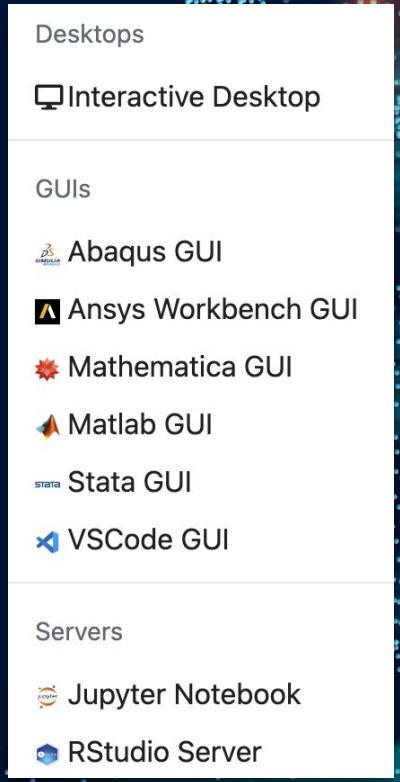
Website: ood.hpc.arizona.edu

Browser-based graphic user portal

- file browser
- interactive desktop
- GUI-based applications
- easy access to terminal

Downside

- can be a little slow
- less control than terminal for certain actions

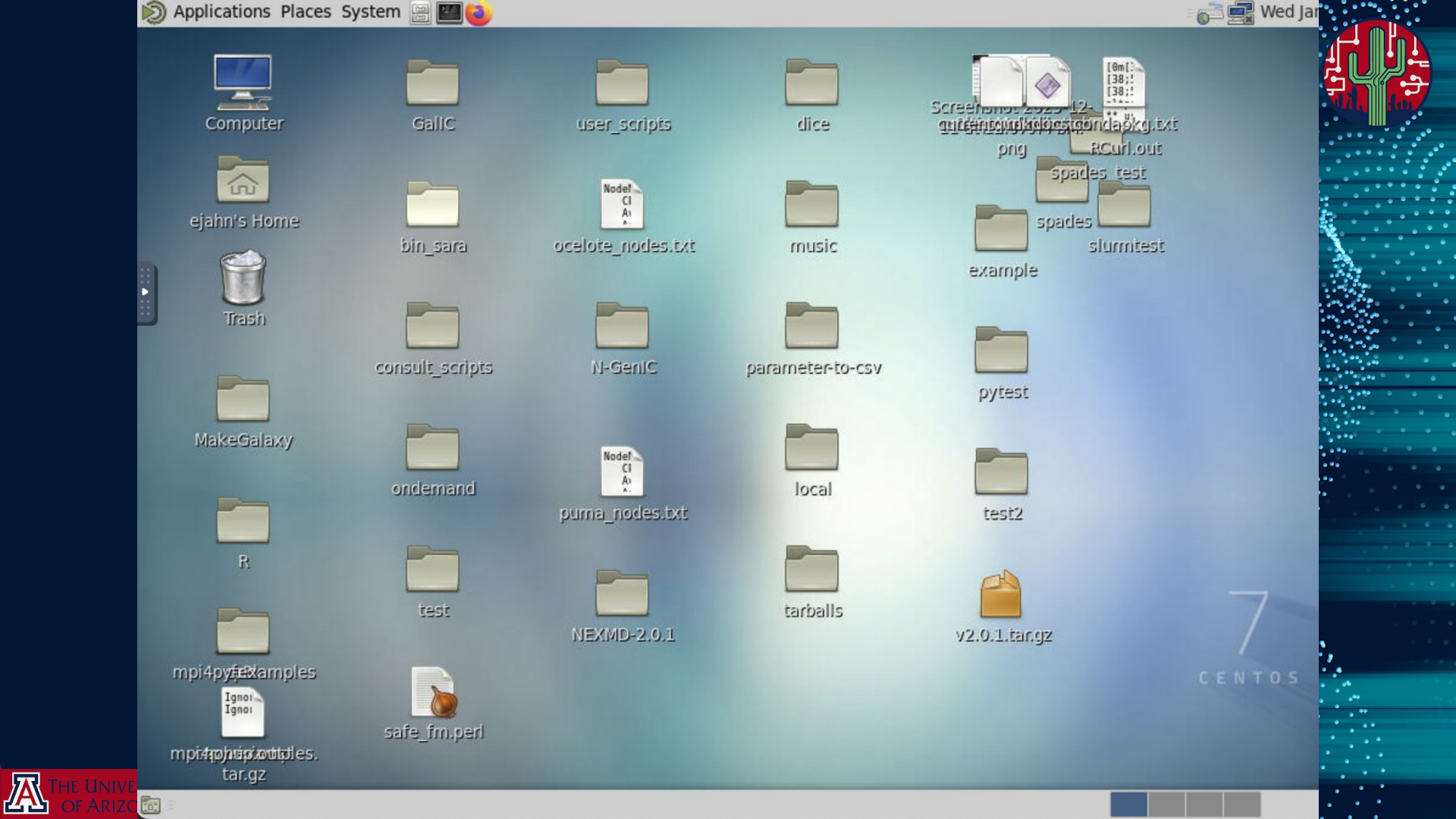


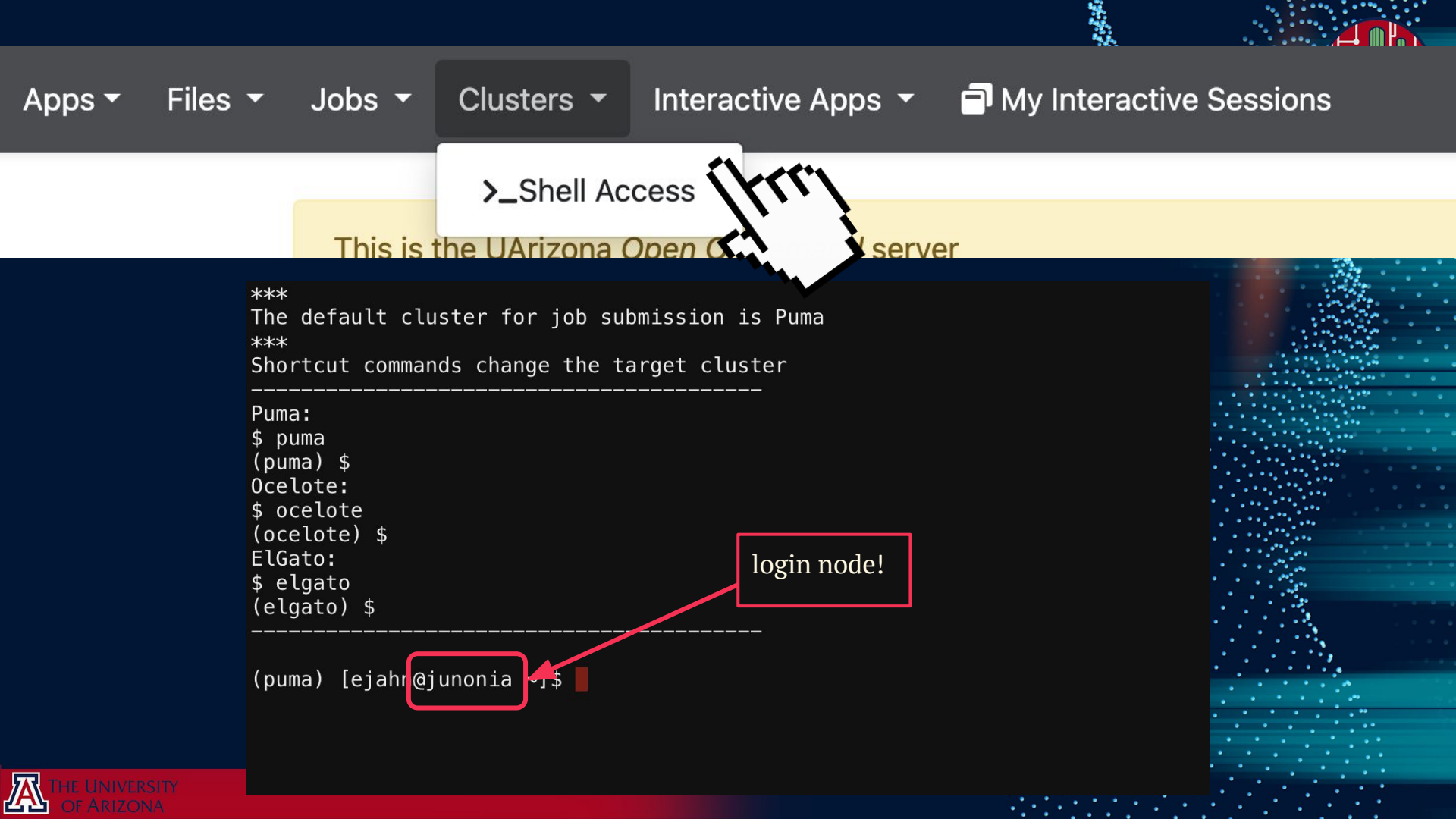
OPEN OnDemand

OnDemand provides an integrated, single access point for all of your HPC resources.

Pinned Apps A featured subset of all available apps

- Abaqus GUI — System Installed App
- Ansys Workbench GUI — System Installed App
- Mathematica GUI — System Installed App
- Matlab GUI — System Installed App
- Stata GUI — System Installed App
- VSCode GUI — System Installed App


Applications Places System
Wed Jan
Computer
GalIC
user_scripts
dice
Screenshot 2023-12-
png
RCurl.out
ejahn's Home
bin_sara
ocelote_nodes.txt
music
spades_test
spades
slurmtest
example
Trash
consult_scripts
N-GenIC
parameter-to-csv
pytest
MakeGalaxy
ondemand
puma_nodes.txt
local
test2
R
test
NEXMD-2.0.1
tarballs
v2.0.1.tar.gz
mpi4pyfex
examples
safe_fm.perl
tar.gz
CENTOS
7


Apps
Files
Jobs
Clusters
Interactive Apps
My Interactive Sessions
>_Shell Access
This is the UArizona Open C server
***
The default cluster for job submission is Puma
***
Shortcut commands change the target cluster

Puma:
$ puma
(puma) $
Ocelote:
$ ocelote
(ocelote) $
ElGato:
$ elgato
(elgato) $

(puma) [ejahr@junonia ~]$
login node!
The University of Arizona

Command Line Access – *Why?*

Faster *(don't waste bandwidth on pesky visuals!)*

More control with use of commands

Write and edit code directly in the terminal using a text editor like vim or nano

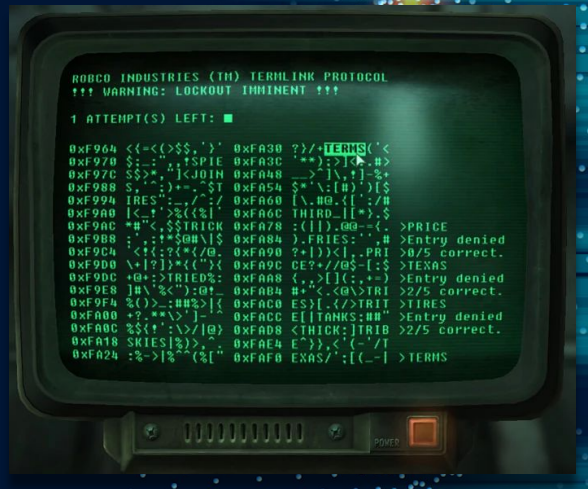
Manually access compute nodes and load software

Submit *batch* jobs

- ***no need to babysit your session!***

<u>Downside</u>: slightly steeper learning curve

<u>Upside</u>: more powerful user in the long run



Command Line Access – *What?*

Text-only interface

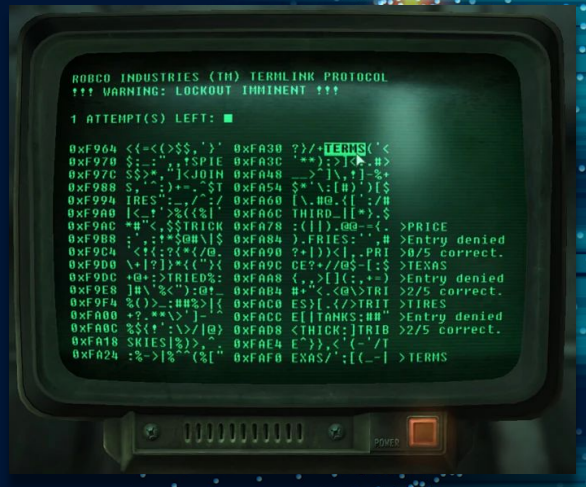
Available on Mac, Windows, and Linux

Uses a *scripting language*

- windows: powershell
- mac/linux: bash (bourne-again shell); zsh
- windows also provides bash shell

Connect to remote server (like HPC) using secure shell protocol: SSH

Can access a terminal through OOD, but is also a program that runs on your personal computer



Command Line Access – *Bash & Linux*

The HPC runs a version of Linux called Centos7

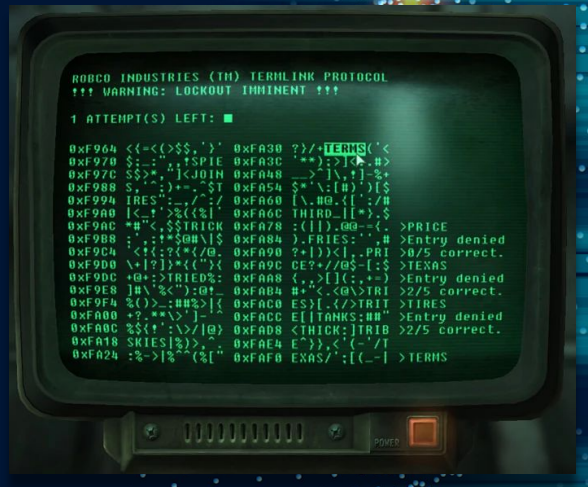
- this is the OS you are using on the HPC regardless of how you accessed it

Remember that you are remotely accessing a *different computer*

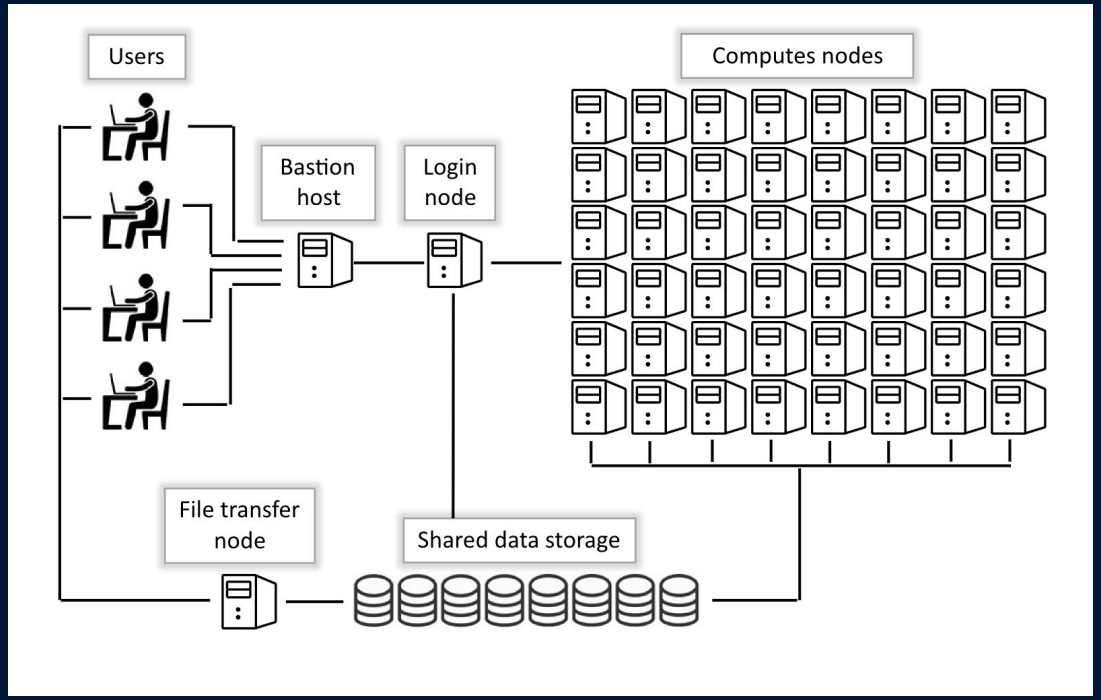
- file system and installed software will be different

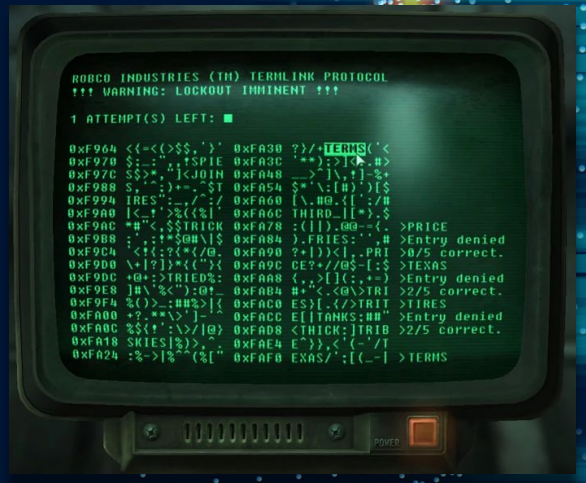
HPC uses bash commands to

- manage files
- navigate between directories
- so much more



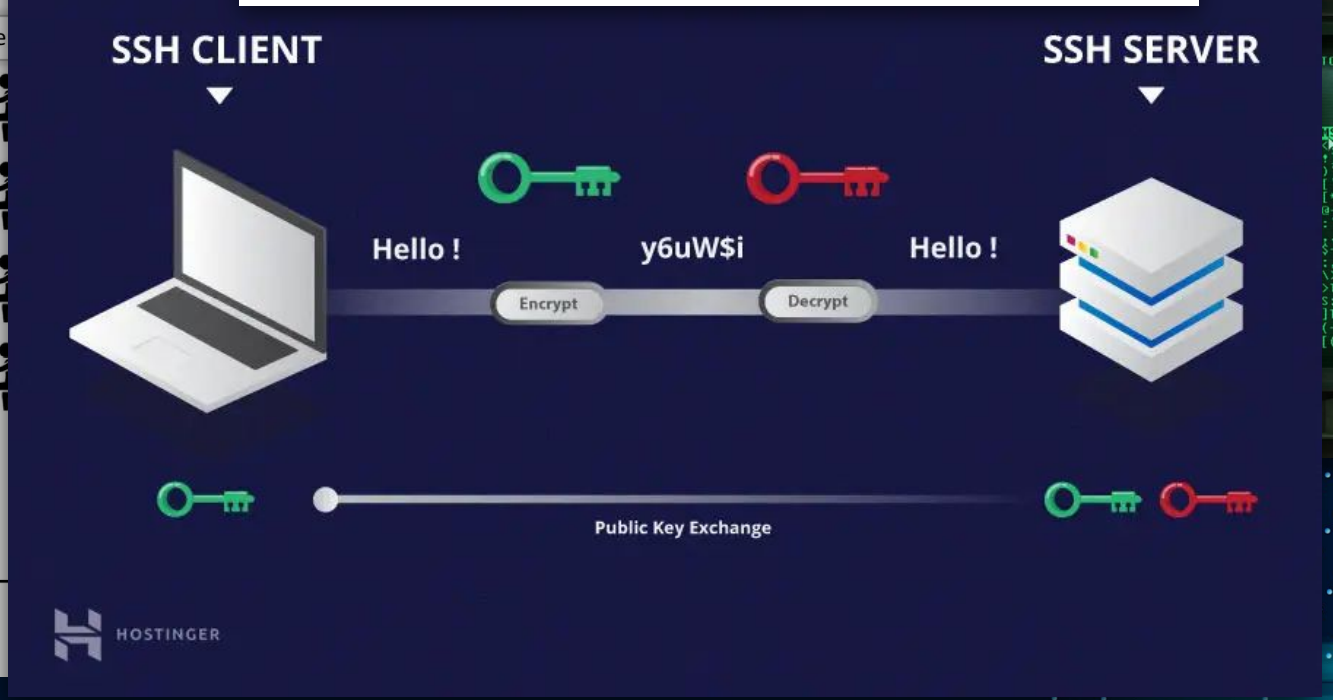
Command Line Access – *how?*

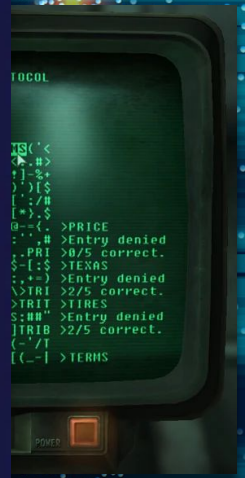




Comma

The **Secure Shell Protocol** (**SSH**) is a cryptographic network protocol for operating network services securely over an unsecured network.[1] Its most notable applications are remote login and command-line execution.





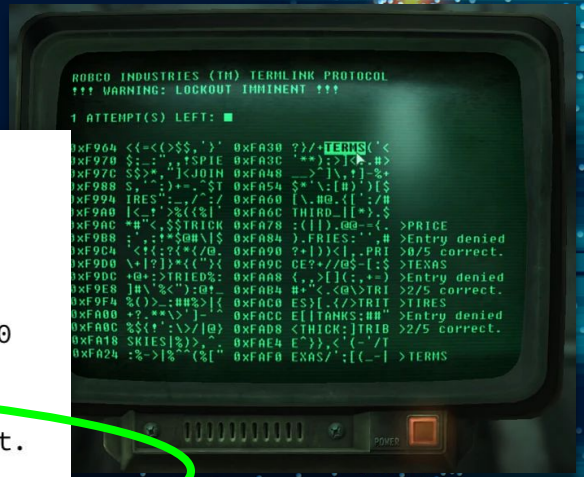
Command Line Access – *how?*

```
~ ssh ejahn@hpc.arizona.edu
Last login: Wed Jan 17 10:14:00 2024 from ip72-201-152-35.ph.ph.cox.net

***
  Reminder - Scheduled Maintenance Periods:
    * HPC downtime scheduled on 2024-01-31, 06:00 thru 2024-01-31, 18:00
      for Scheduled Maintenance.
***
This is a bastion host used to access the rest of the RT/HPC environment.

Type "shell" to access the job submission hosts for all environments
-------------------------------------------------------------------

[ejahn@gatekeeper ~]$
```



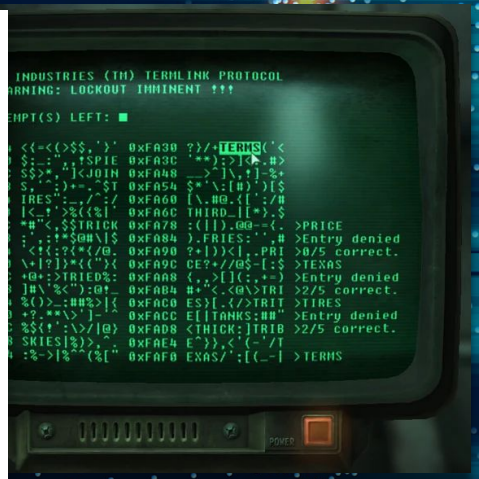
Command Line Access – *how?*

```
[ejahn@gatekeeper ~]$ shell
Last login: Wed Jan 17 10:40:39 2024 from gatekeeper.hpc.arizona.edu
***
The default cluster for job submission is Puma
***
Shortcut commands change the target cluster
-----------------------------------------
Puma:
$ puma
(puma) $
Ocelote:
$ ocelote
(ocelote) $
ElGato:
$ elgato
(elgato) $
-----------------------------------------


(puma) ejahn@wentletrap:~
 ->>>
```

how to change cluster

prompt
- (cluster)
- user
- @
- node



login node names
- junonia
- wentletrap

Command Line Access – *how?*

once you see the prompt with a login node name… *you're in!*

now, what to do?

- use **bash commands** to manage files
- **install** code or software needed for your research
  - *note: compile on a compute node!*
- use an **interactive session** for testing
- submit a **batch job**

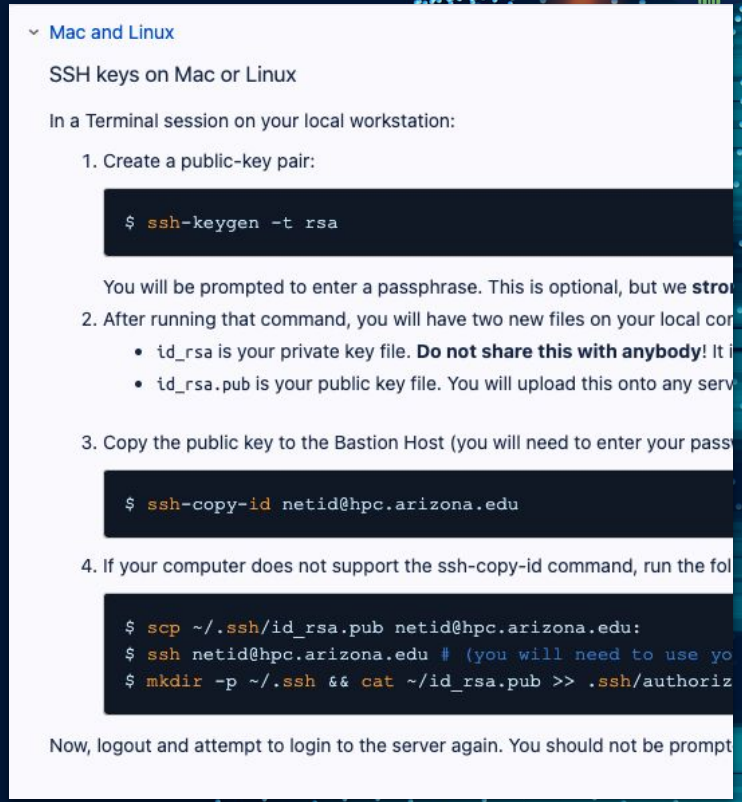
Some Useful Commands

`puma/ocelote/elgato`	: switch cluster
`nodes-busy`	: see the usage of current cluster
`uquota`	: get a summary of your storage usage
`va`	: view allocation – summary of CPU time used on current cluster
`past-jobs -d <N>`	: view jobs from previous N days
`job-history <job ID>`	: detailed report about given job

…and many more

Easy Access: SSH Keys

- create a **private/public key pair** to place on your local machine and the HPC to reduce the number of times you need to enter your password
- **3 failed password attempts means system lockout 1 hour**
  - ***avoid this with SSH keys!***
- instructions in our docs or google :)



Mac and Linux

SSH keys on Mac or Linux

In a Terminal session on your local workstation:

1. Create a public-key pair:

```
$ ssh-keygen -t rsa
```

You will be prompted to enter a passphrase. This is optional, but we stro

2. After running that command, you will have two new files on your local cor
   - `id_rsa` is your private key file. **Do not share this with anybody**! It i
   - `id_rsa.pub` is your public key file. You will upload this onto any serv
3. Copy the public key to the Bastion Host (you will need to enter your pass

```
$ ssh-copy-id netid@hpc.arizona.edu
```

4. If your computer does not support the ssh-copy-id command, run the fol

```
$ scp ~/.ssh/id_rsa.pub netid@hpc.arizona.edu:
$ ssh netid@hpc.arizona.edu # (you will need to use yo
$ mkdir -p ~/.ssh && cat ~/id_rsa.pub >> .ssh/authoriz
```

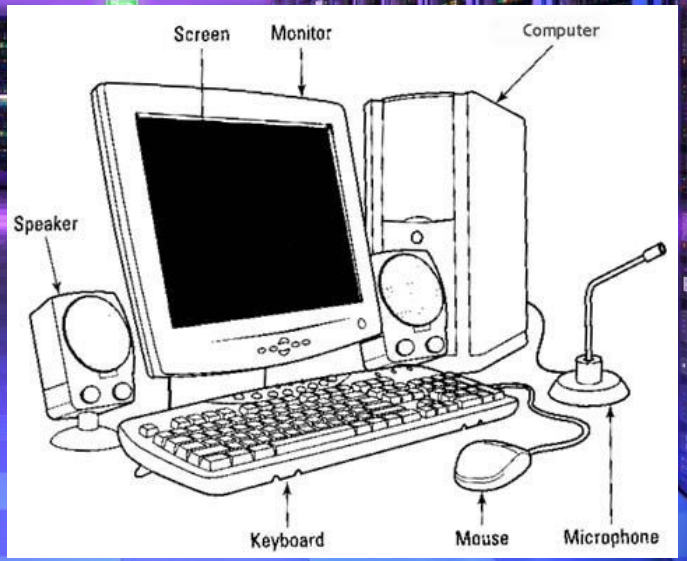
Now, logout and attempt to login to the server again. You should not be prompt

Check In!

Name one pro and one con of using Open OnDemand

Name one pro and one con of using the terminal & SSH

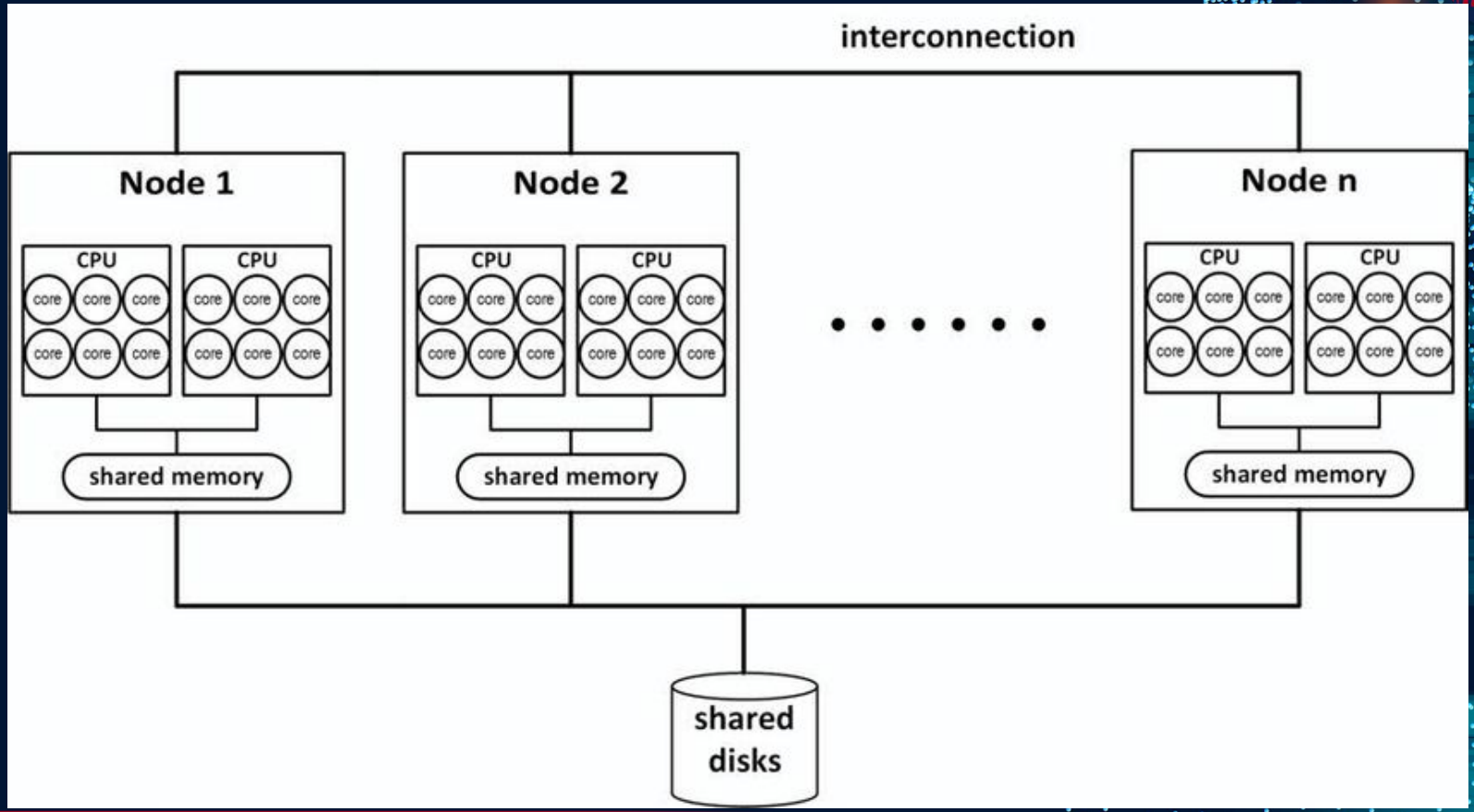
system layout



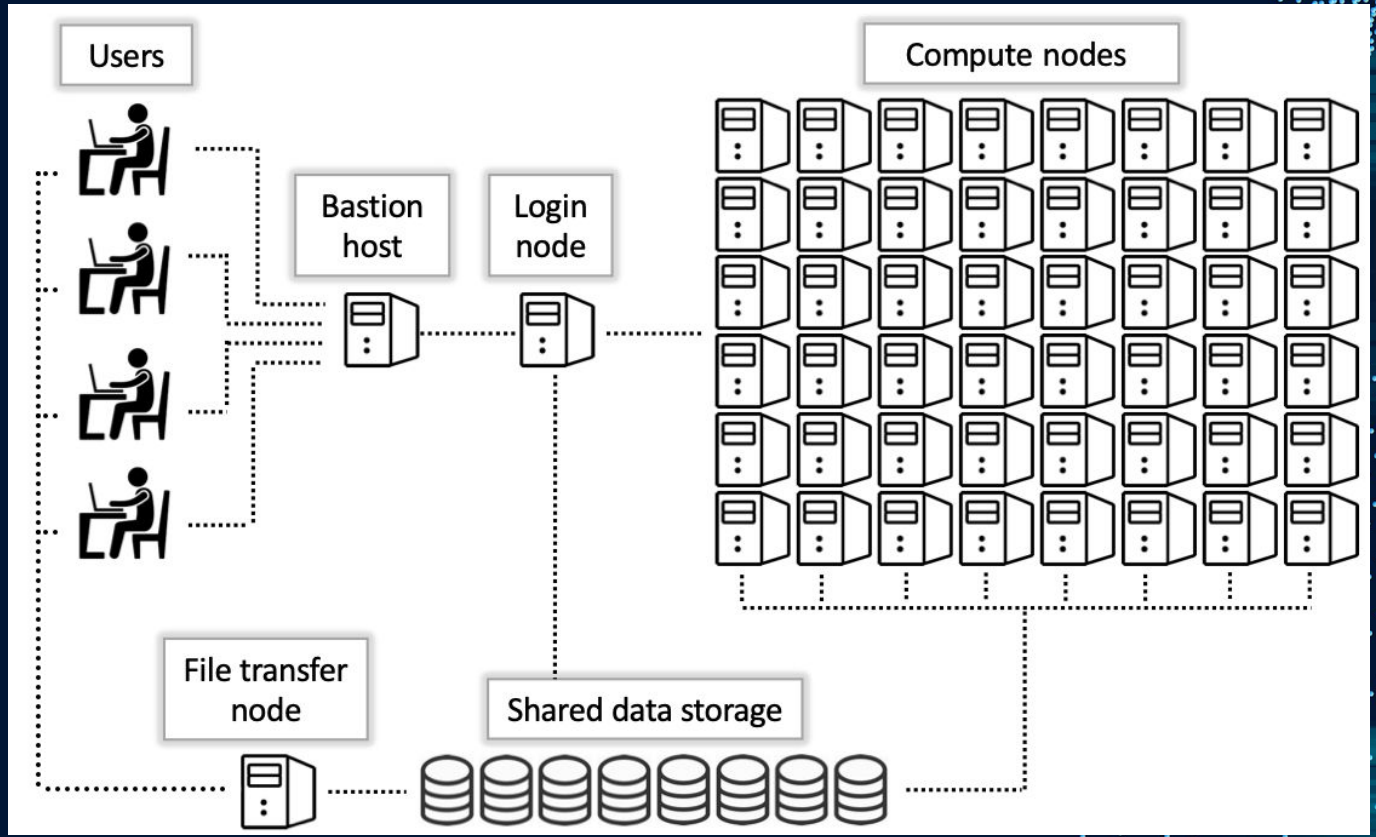
Outline

1. HPC Overview
    a. What is HPC?
    b. Common Misconceptions
2. System Access
    a. Creating an account
    b. Browser Access
    c. Command Line Access
3. **System Layout**
    a. Node Types and Acceptable Use
4. Storage and Transfers
    a. File system overview
    b. Transfer Methods
5. Software
    a. System Modules
    b. User Installations
6. Accessing Compute
    a. GUI Jobs
    b. Interactive Jobs
    c. Batch Jobs

Architecture of a Supercomputer



UA HPC System Architecture



Different Nodes - Different Purposes

Bastion Host - security & validation

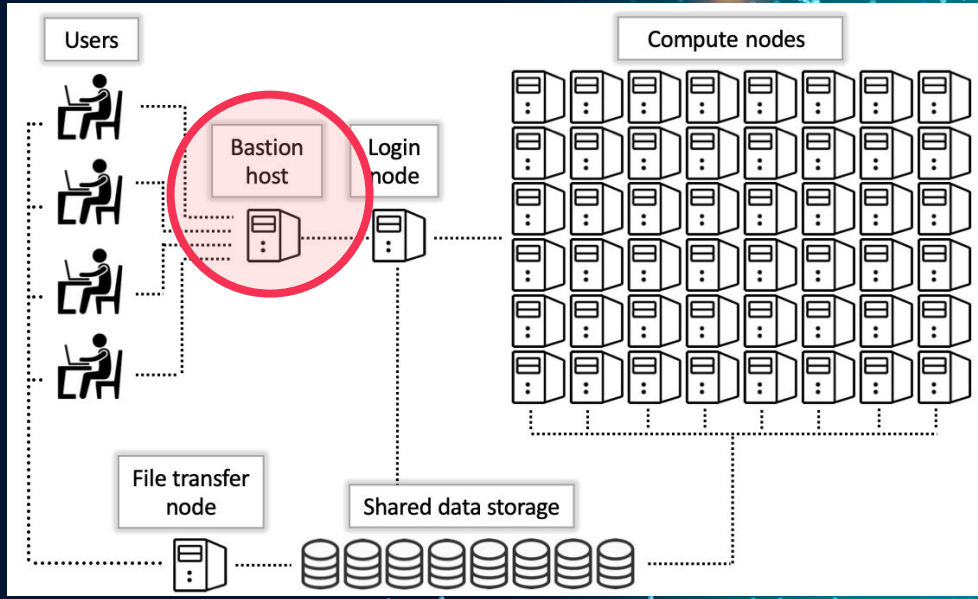
- hostname: gatekeeper
- ***NO File Storage***
- ***NO Compute***
- ***the only thing to do here is type "shell" to access the login nodes***

```
ssh ejahn@hpc.arizona.edu
Last login: Wed Jan 17 10:14:09 2024 from ip72-201-152-35.ph.ph.cox.net

***
  Reminder - Scheduled Maintenance Periods:
    * HPC downtime scheduled on 2024-01-31, 06:00 thru 2024-01-31, 18:00
      for Scheduled Maintenance.
***
This is a bastion host used to access the rest of the RT/HPC environment.

Type "shell" to access the job submission hosts for all environments
-------------------------------------------------------------------

[ejahn@gatekeeper ~]$
```



Different Nodes - Different Purposes

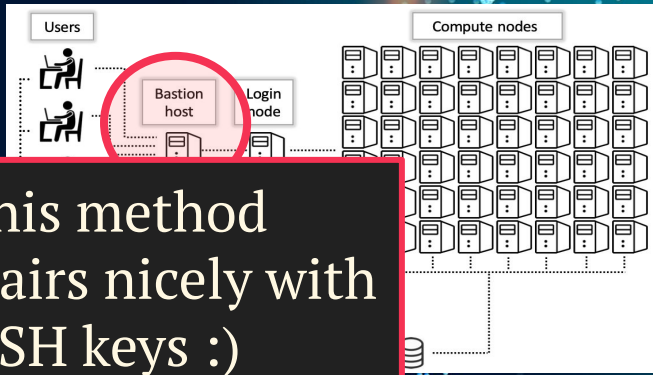
Bastion Host - security & validation

SSH to bastion host can be bypassed by using a proxy jump to the login node:

```
ssh -J <jump server> <remote server>

ssh -J user@hpc.arizona.edu \
                user@shell.hpc.arizona.edu
```

or use ssh config file to save hostnames and proxy jumps!

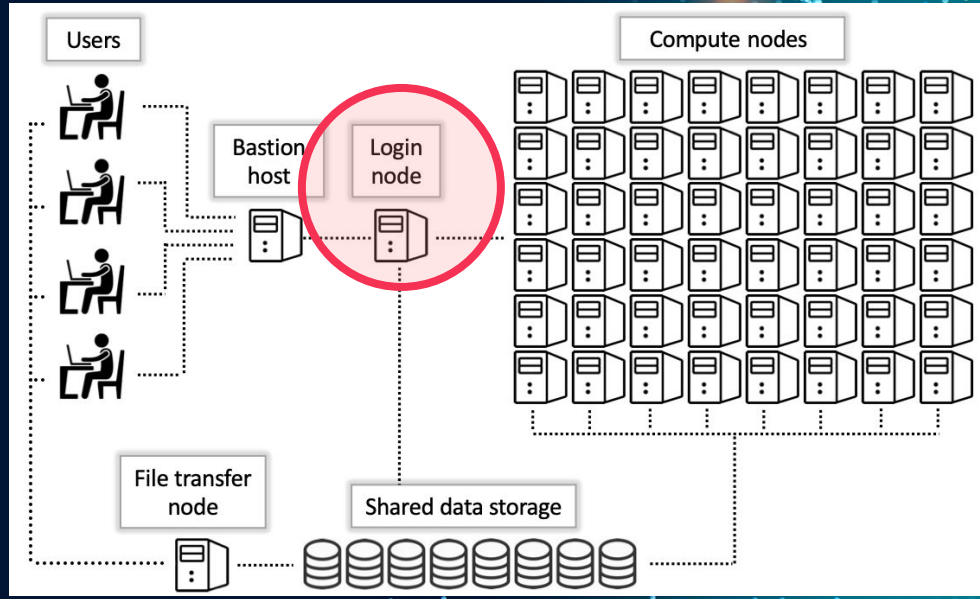


this method pairs nicely with SSH keys :)

Different Nodes - Different Purposes

Login Nodes - file management and job submission

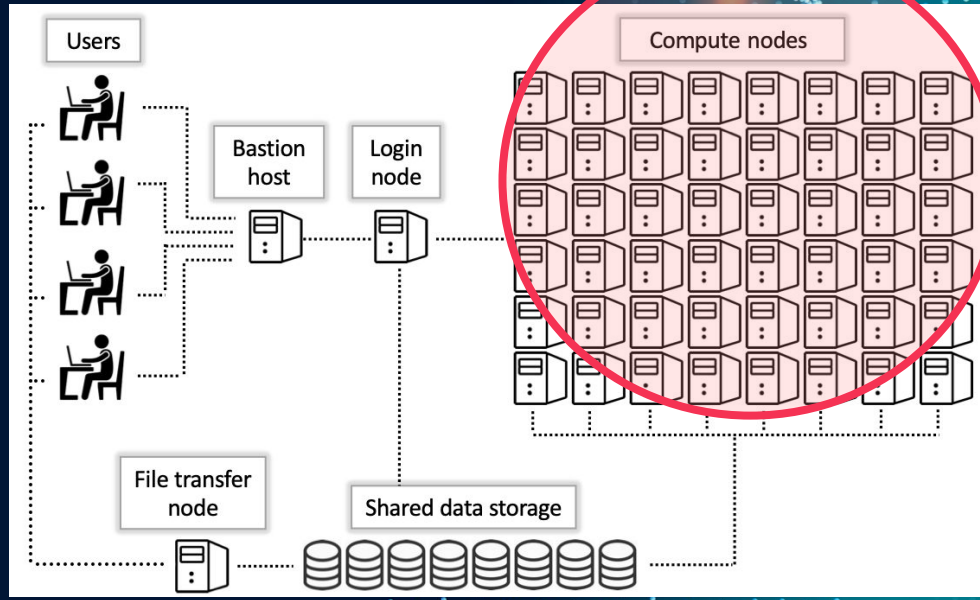
- hostnames: junonia, wentletrap
- domains: shell.hpc.arizona.edu
- ***access the main HPC filesystem!***
  - /home; /groups; /xdisk
- submit interactive or batch jobs to access compute nodes
- ***NO Compute***



Different Nodes - Different Purposes

Compute Nodes - HPC workhorses

- ***only accessible via job request!***
- use PI's CPU-time allocation to "pay" for jobs on standard partition (allocation itself is free)
- use windfall to run preemptable but free jobs
- interactive sessions provide shell access
- batch jobs enable remote, automatic running of scripts



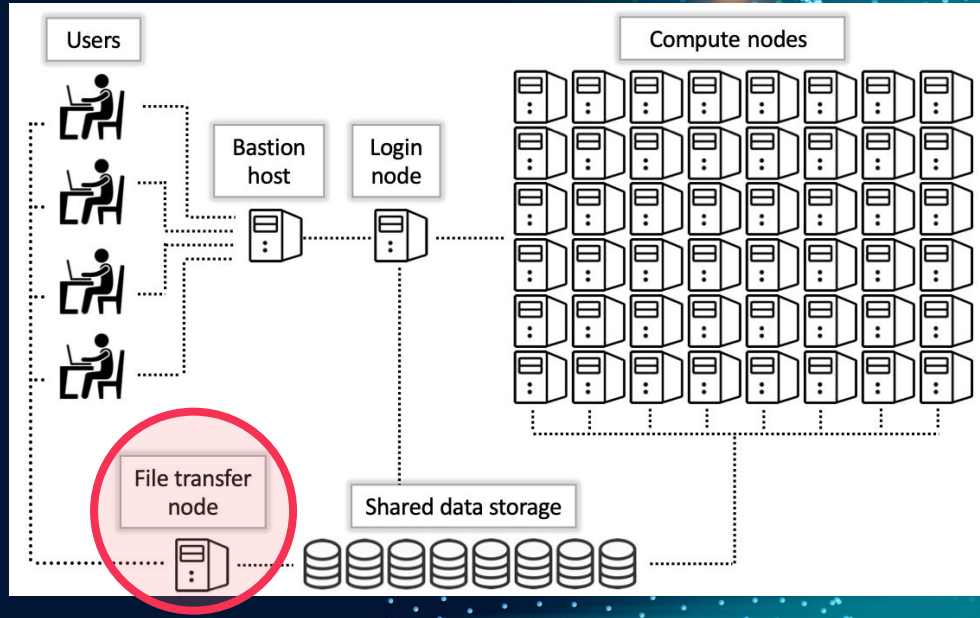
Different Nodes - Different Purposes

File Transfer Nodes - transfer your files :)

- domain: filexfer.hpc.arizona.edu
- access via ssh from login node:

```
ssh filexfer.hpc.arizona.edu
```

- transfer files from external machines
- access /rental file system



Check In!

Which type of node has access to the main filesystem, but should not be used to run jobs?

→ Login Node

Which type of node should be used when uploading data to the HPC?

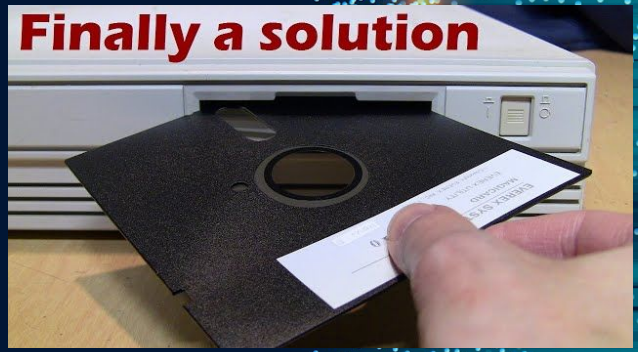
→ File Transfer Node

storage and transfers

Outline

1. HPC Overview
   a. What is HPC?
   b. Common Misconceptions
2. System Access
   a. Creating an account
   b. Browser Access
   c. Command Line Access
3. System Layout
   a. Node Types and Acceptable Use
4. **Storage and Transfers**
   a. File system overview
   b. Transfer Methods
5. Software
   a. System Modules
   b. User Installations
6. Accessing Compute
   a. GUI Jobs
   b. Interactive Jobs
   c. Batch Jobs

UA HPC Storage Resources



High Performance Storage

- mounted directly on HPC filesystem → *speed!*
- each user gets **`/home`** 50GB
- each PI gets **`/groups`** 500GB
- PIs can request **`/xdisk`** up to 20TB

Rental Storage

- PIs can purchase storage in **`/rental`**
- mounted to file transfer node (must be copied to HPC FS to use)

R-DAS

- PIs can request up to 5TB
- shares can be remotely mounted to workstations ***(not HPC!)***
- can be copied to HPC

Tier 2/AWS

- Long term backup options for archival data ***(not HPC!)*** – can be copied to HPC

Storage - File Permissions

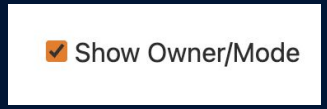
All computer systems have some form of permissions.

These are especially important on a shared system - we don't want other users (intentionally or accidentally) messing with your files!

→ *Let's briefly discuss linux file permissions*

See file permissions in terminal: command `ls -lha`

See file permissions in Open OnDemand:



Storage - File Permissions

Terminal Output:

```
(puma) [ejahn@wentletrap test]$ ls -lha
total 20K
-rw-r-----. 1 ejahn staff 0 May 14 15:57 file1
drwx------. 2 ejahn staff 0 May 14 15:57 folder1/
drwxr-x---. 2 ejahn staff 0 May 14 15:57 folder2/
drwxr-xr-x. 2 ejahn staff 0 May 14 15:57 folder3/
-rwxr-xr--. 1 ejahn staff 0 May 14 15:57 program*
```

item type

user

group

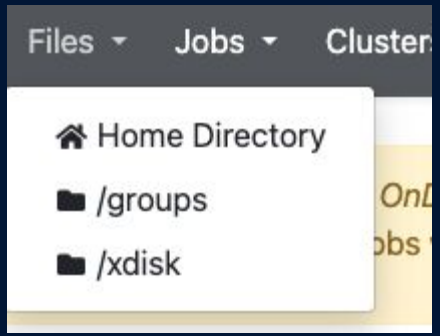
other

r = read
w = write
x = execute (run)

Transferring Data - Open OnDemand

navigate to ood.hpc.arizona.edu

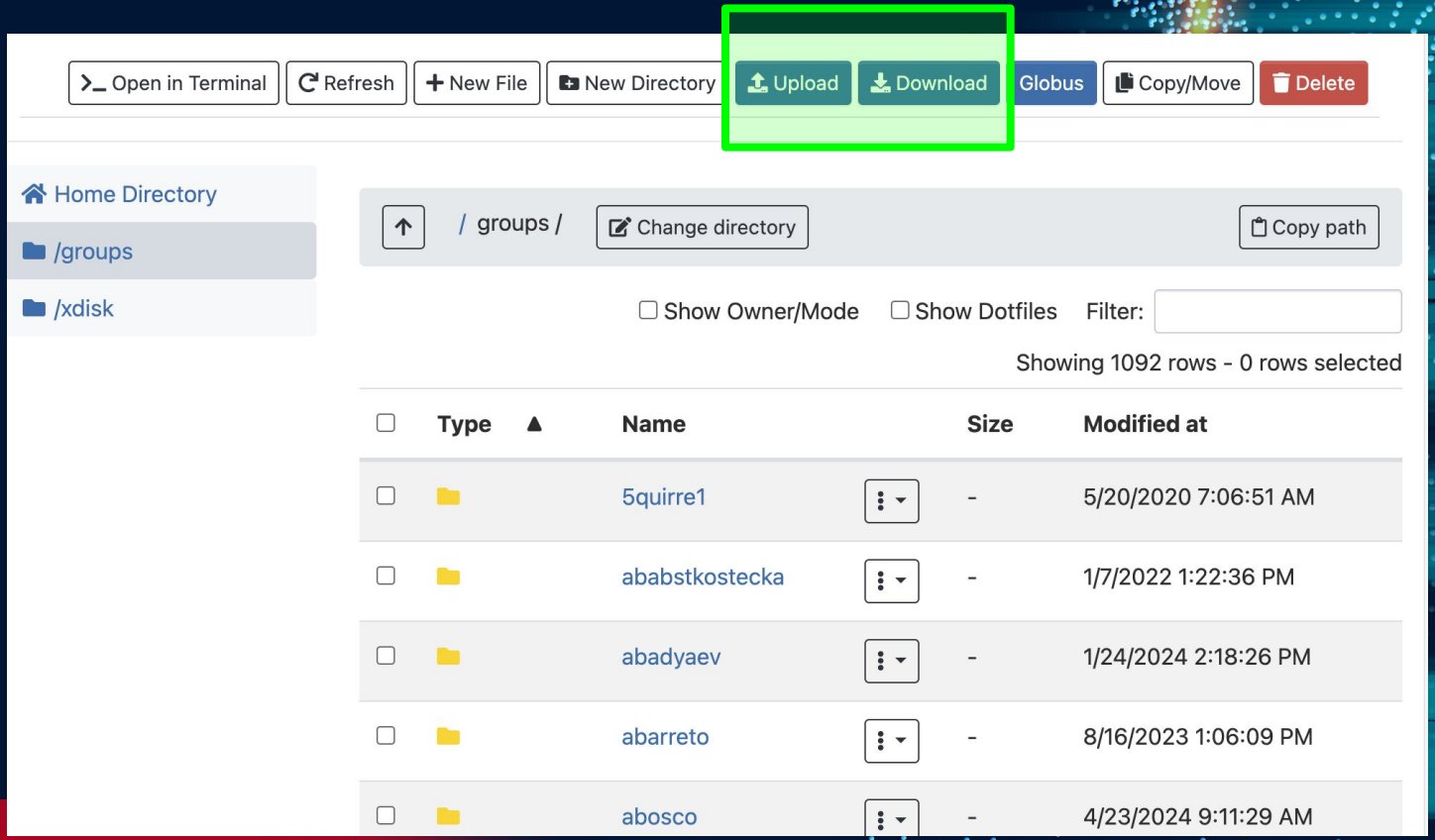
then, click Files from the top bar and select the share you want to access:



Transferring Data - Open OnDemand

OOD File Transfer Interface

Max File Size: 64 MB



Transferring Data

Transfers < 64 MB:

→ For small data transfers, the web portal offers the most intuitive method.

Transfers < 100 GB:

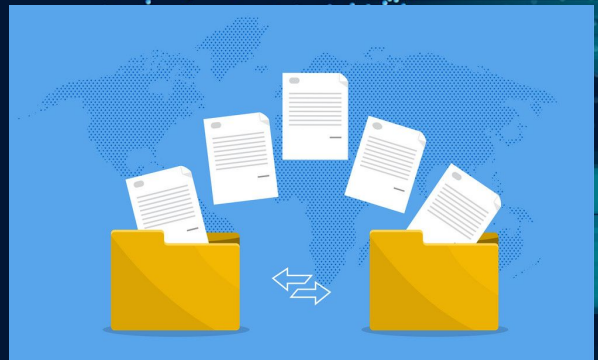
Command Line Utilities

→ we recommend SFTP, SCP or Rsync using `filexfer.hpc.arizona.edu`.

Transfers > 100 GB, transfers outside the university, and large transfers within HPC:

→ we recommend using Globus (GridFTP).

Graphical Application



Transferring Data

Rsync example (upload to hpc):

```
rsync -rva /path/to/local/file
        user@filexfer.hpc.arizona.edu:/path/to/destination
```

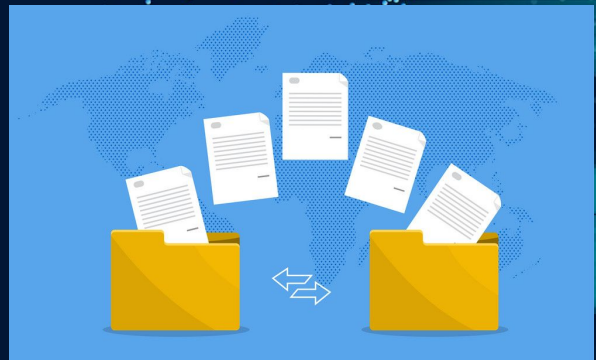
Note that source path is first and destination path is second

Remote host always has user@server

Options:

-r = recursive (include folders) -v = verbose (print more info)

-a = archive mode (preserve item information)



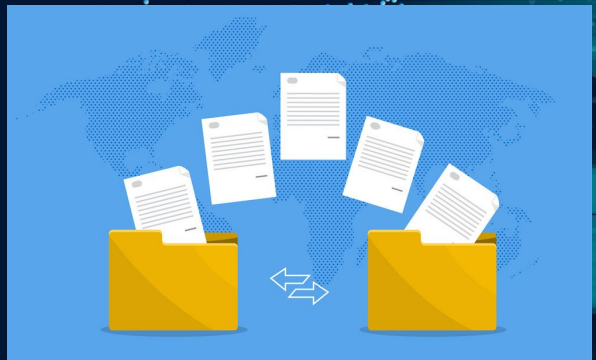
Transferring Data

There are tons of options for transferring data

We have instructions for all of them in our documentation:

docs.hpc.arizona.edu

Please check there for details on your method of choice!



Software on HPC

Outline

1. HPC Overview
   a. What is HPC?
   b. Common Misconceptions
2. System Access
   a. Creating an account
   b. Browser Access
   c. Command Line Access
3. System Layout
   a. Node Types and Acceptable Use
4. Storage and Transfers
   a. File system overview
   b. Transfer Methods
5. **Software**
   a. System Modules
   b. User Installations
6. Accessing Compute
   a. GUI Jobs
   b. Interactive Jobs
   c. Batch Jobs

Software Overview

Availability of software *depends on the node you are on!*

Login Node

- Bash, system utilities, & basic functions
- Slurm commands (squeue, sbatch, scontrol, etc)

Data Transfer Node

- some limitations

Compute Nodes

- “module” commands only available here
- used to access full suite of installed software

Module Commands

When you are on a compute node, the module commands are available to help you access the software you need.

Usage: `module <command> <arg>`

Commands:	Args:	Result:
avail	<search term>	print list of installed software
load	<software name>	add software to path
unload	<software name>	remove software from path
swap	<name1> <name2>	removes <name1> and adds <name2>
show	<name>	prints paths for loaded software
spider	<name>	prints versions and other info
list	--	prints all loaded modules

Installing Personal Software

Users cannot install system modules.

However, software can be downloaded and compiled within directories you own.

*Software should **always** be compiled using a **compute node!***

Do not compile software on the login node ! !

Compiling Personal Software (Only do this on a compute node!):

1. Download and unpack

```
[user@cpu39 make_example]$ wget https://ftp.gnu.org/gnu/hello/hello-2.10.tar.gz
[user@cpu39 make_example]$ tar xzvf hello-2.10.tar.gz
[user@cpu39 make_example]$ cd hello-2.10
```

2. Configure

```
[user@cpu39 hello-2.10]$ ./configure --prefix=$HOME/hello_install
```

3. Compile/Install

```
[user@cpu39 hello-2.10]$ make
[user@cpu39 hello-2.10]$ make install
```

4. Modify environment

```
[user@cpu39 hello-2.10]$ export PATH=$HOME/hello_install/bin:$PATH
```

and finally...

Running Jobs on HPC

Outline

1. HPC Overview
   a. What is HPC?
   b. Common Misconceptions
2. System Access
   a. Creating an account
   b. Browser Access
   c. Command Line Access
3. System Layout
   a. Node Types and Acceptable Use
4. Storage and Transfers
   a. File system overview
   b. Transfer Methods
5. Software
   a. System Modules
   b. User Installations
6. **Accessing Compute**
   a. GUI Jobs
   b. Interactive Jobs
   c. Batch Jobs

Overview

In order to access a compute node, you must make a **resource request**

This means specifying:

- how many **CPUs** you want for how much **time**
- what type of compute node do you want (standard, high mem, GPU)

IMPORTANT: number of CPUs determines memory (RAM)

In order to make a proper resource request, we need to know what kind of hardware we are working with

Hardware - Puma

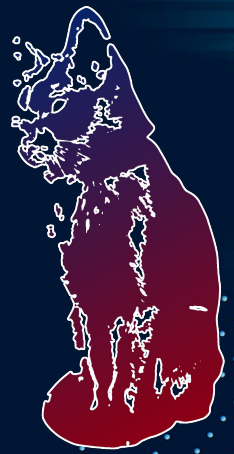
Node Type	Standard	High Memory	GPU
Number of Nodes	192 standard 108 buy-in	3 standard 2 buy-in	8 standard 7 buy-in
CPUs/Node	94	94	94
RAM/CPU	5 GB	32 GB	5 GB
CPU RAM/Node	470 GB	3008 GB	470 GB
GPUs/Node			4
RAM/GPU			32 GB (v100s) 20 GB (MIGs)
GPU RAM/Node			128 GB
Total GPUs			32 standard 28 buy-in

Hardware - Ocelote

Node Type	Standard	High Memory	GPU
Number of Nodes	400	1	46
CPUs/Node	28	48	28
RAM/CPU	6 GB	41 GB	8 GB
CPU RAM/Node	168 GB	1968 GB	224 GB
GPUs/Node			1
RAM/GPU			16 GB
GPU RAM/Node			16 GB
Total GPUs			46

Hardware - El Gato

Node Type	Standard
Number of Nodes	130
CPUs/Node	16
RAM/CPU	4 GB
CPU RAM/Node	64 GB



Choosing a **Cluster**

El Gato

- low wait times
- fewer resources
- good for **smaller jobs** or **testing**
- smallest CPU time allocation (7,000hr)

Ocelote

- more resources than El Gato
- **has many nodes with 1 GPU each**
- wait times *vary* depending on usage
- great tool for **many jobs**
- 100,000 hr CPU time

Puma

- most resources in terms of GPU and CPU
- **newest** hardware
- in high demand (*long wait times*)
- 150,000 hr CPU time
- best for **large** production **jobs**

Note on CPUs and Memory

Each compute node has a certain number of processors *physically mounted* on it

Each of those processors has a memory chip *physically connected* to it

That means

Number of CPUs determines Memory*

***think RAM, not disk space!**

Total Mem = Mem/CPU x N(CPU)

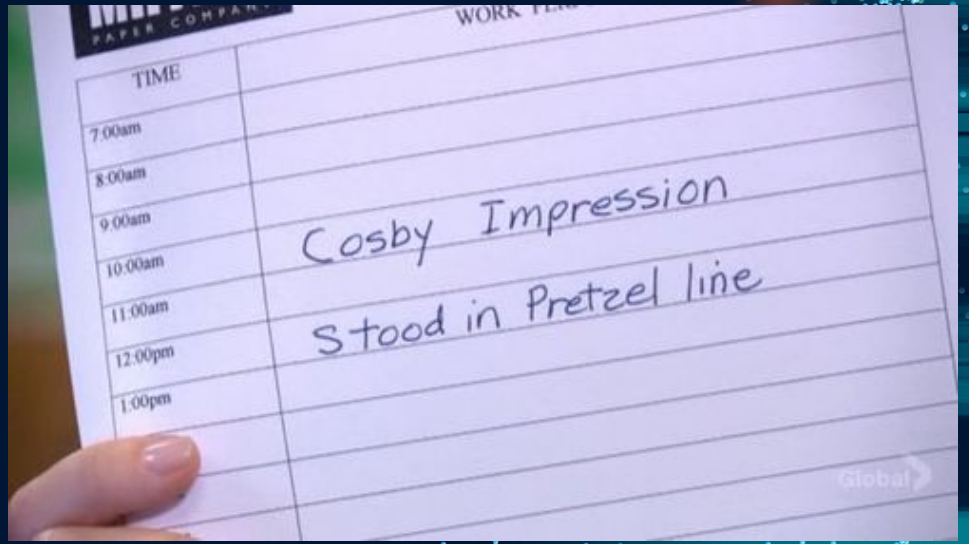
Memory is <u>not</u> a continuous scale or arbitrary number!!

Memory Takeaways

- Know the value for RAM per CPU for your machine!
- If using the total memory flag, make sure this equals N_{CPUs} x mem-per-CPU
- Only input allowed values for mem-per-CPU

job scheduling

slurm
workload manager



Job Scheduling

- Many users
- Many jobs
- Limited resources

→ ***task scheduler*** required to balance the requirements of hundreds of jobs

→ once your job has been requested, it goes to the scheduler, which places it in the ***queue*** according to *when* it was submitted and the *resources* it requested

→ jobs requesting **more resources** will generally take **longer** to start

→ jobs with **high time limits** will generally take **longer** to start

→ sometimes usage is low and jobs start soon

→ sometimes the queue is busy, and it takes a while

Job Scheduling

- Many user
- Many jobs
- Limit

→ ***task schedu*** hundreds of jobs

→ once your j er, which places it in the ***queue*** ac *esources* it requested

→ jobs requesting **m** ake **longer** to start

→ jobs with **high time limits** w erally take **longer** to start

→ sometimes usage is low and jobs start soon

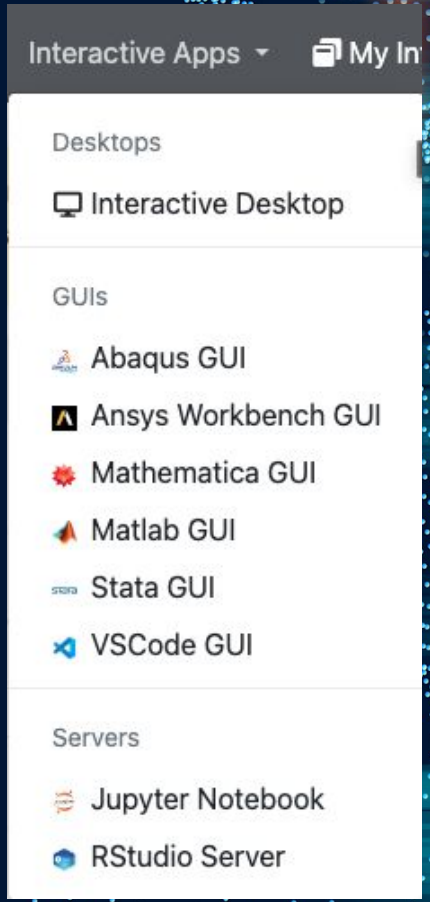
→ sometimes the queue is busy, and it takes a while

When the system is busy, it may take several hours for jobs on Puma to begin!

slurm workload manager

Graphical Jobs - Open OnDemand

1. Select your app from the list
2. Fill in the resource request form
3. Wait for the scheduler to allocate your resources!



Interactive Jobs

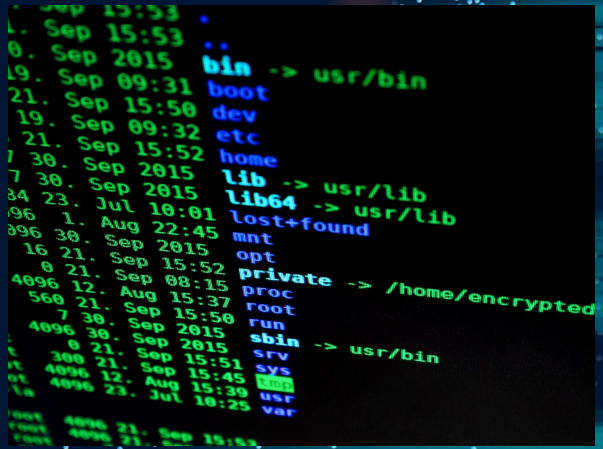
Terminal sessions that allow you to interact with compute resources in real time

1. Use your local machine to SSH into HPC
2. Navigate to the login node
3. Run the “interactive” command

Example:

```
interactive -a hpcteam -t 04:00:00 -n 16
```

What does this command do?

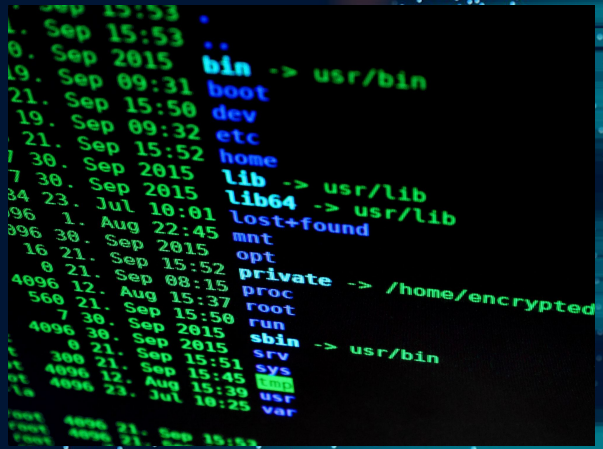


Interactive Jobs

What to do next?

Any ideas?

- load modules
- compile software
- run scripts
- testing
- debugging
- etc.



Interactive Jobs

What happens if my internet cuts out during an interactive session?

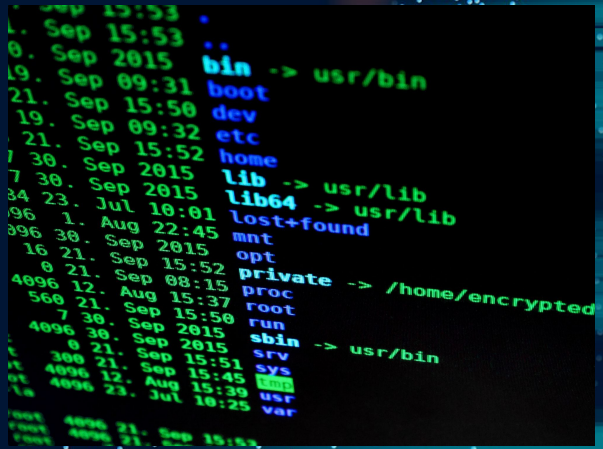
→ *it terminates*

What happens if I run some code and it takes so long my SSH connection is closed?

→ *it terminates*

How can I run longer jobs without fear of losing my work?

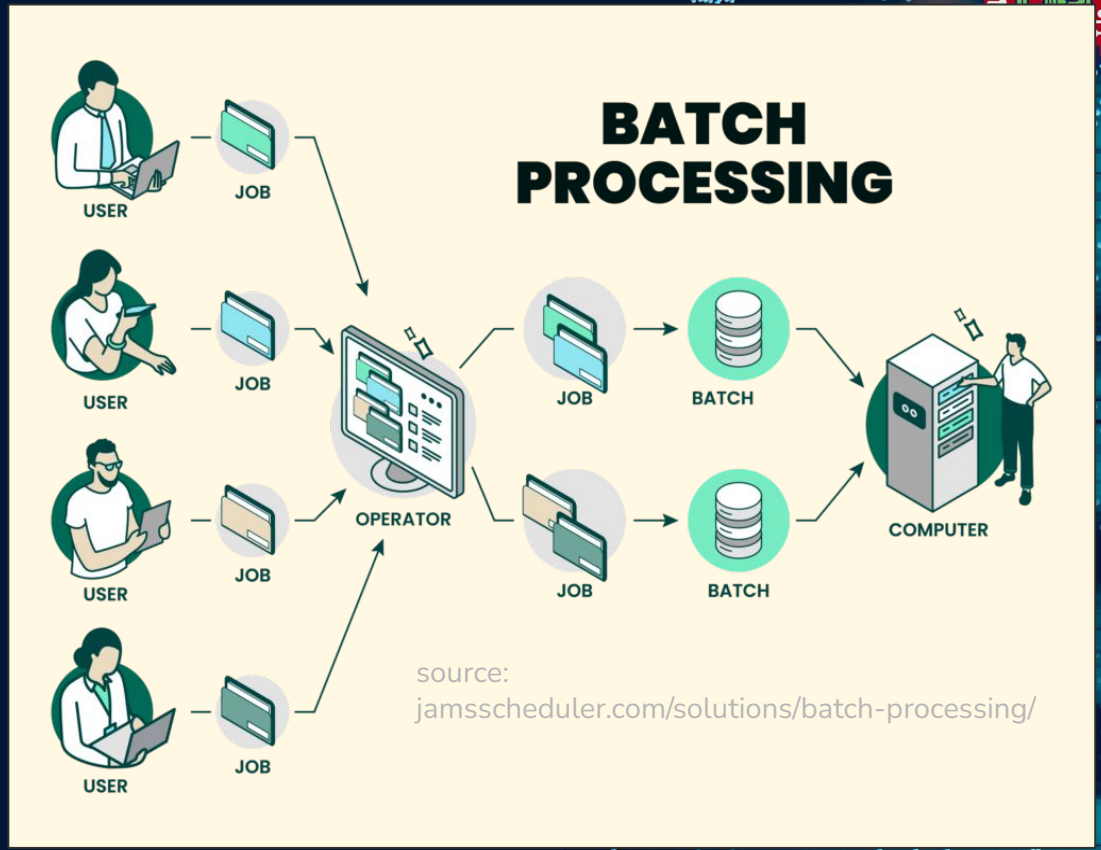
→ *batch jobs*



Batch Jobs

set-and-forget computing!

1. write a **script** with resource request and job commands
2. send to scheduler
3. log off and wait!



Batch Job Overview

Batch jobs are a way to **run code automatically**, without user oversight or even an active connection to HPC

Just like all other job types, batch jobs are **allocated compute resources** through the **task scheduler** Slurm

Unlike other job types, batch jobs require a **batch script** that determines **everything the job will do** *in advance*

The batch script is submitted using `sbatch <myscript.slurm>`, and will proceed through the queue *automatically*

Batch Script Overview

Example Batch Script:

```
#!/bin/bash

# --------------------
### Directives Section
# --------------------
#SBATCH --job-name=hello_world
#SBATCH --account=your_group
#SBATCH --partition=standard
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=00:01:00

# --------------------
### Code Section
# --------------------
module load python/3.9
cd ~/hello_world
python3 -c "print('hello world')"
### sleep is used for demonstration purposes
sleep 30
```

Sections:

“shebang”

use directives to request compute resources

write code to tell the computer what to run during the job

Slurm Commands

`sbatch`	submit your batch script
`squeue`	see jobs waiting or currently running
`sacct`	see jobs that have completed or failed
`salloc`	request an interactive session with more control
`scontrol`	get info on jobs
`scancel`	cancel a given job

Interactive vs. Batch Jobs

Interactive

- requested with 'interactive' or 'salloc' command
- resources described in **command**
- input commands and receive output in **real time**
- can *change* what it's doing after it starts
- job **cancelled** if connection lost

Both

- access compute nodes by submitting a request to **slurm**
- can run intense calculations
- subject to **time limit**
- access to standard and windfall partitions
- subject to **queue** *(wait times)*

Batch

- requested with 'sbatch'
- requires a **batch script**
  - describe resource request
  - load software
  - initiate the calculation
- saves **output to files**
- *cannot change* anything once submitted
- will run **without supervision** or active connection to HPC

Use cases

Interactive

- install and compile software
- test new code
- smaller scale computations
- iterative analyses
- debugging

Batch

- jobs that will take a long time to complete
- jobs that require significant resources
- jobs that may end up waiting in the queue for a while
- code that is known to work and requires minimal debugging

Both

- require use **BASH** in the terminal

Activity: Let's run a batch job!

Let's write a basic python script, then submit it to the queue

Process:

1. write a python script
2. write a batch script
3. submit the batch script

Python Script

You can write anything you want! It can be as simple as print(‘hello world’), or you can make it more interesting.. up to you ;)

Steps:

1. log onto the HPC
2. create a blank python file: ‘**`touch myscript.py`**’
3. edit the script: ‘**`nano myscript.py`**’
   a. feel free to use vim or emacs or any other text editor if you are more comfortable with one of those
   b. add something simple like

```
print(“hello world!”)
```

4. be sure to save it!

Batch Script

add the following to sections to a new file called myscript.slurm

1. Directives: tell the scheduler what resources you want

```
#!/bin/bash
#SBATCH --job-name=test
#SBATCH --output=test_%A.out
#SBATCH --error=test_%A.err
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --partition=standard
#SBATCH --account=<your_account>
#SBATCH --time=00:05:00
```

2. Load your modules

```
module load python/3.11
```

3. Call your command

```
python3 myscript.py
```

Submit!

send to scheduler

```
sbatch myscript.slurm
```

check the queue

```
squeue --user=<my netid>
```

view the output files

```
cat <name>.out
```

```
cat <name>.err
```

check the job-history report

```
job-history <job id>
```

Congrats! You just ran your first batch job!!!

UA HPC Resources and Help

The HPC Consult Team is here for you!

- **Documentation**: docs.hpc.arizona.edu
- **ServiceNow**: HPC Support and Consulting Request
- **Office Hours**: every Wednesday 2-4pm on GatherTown