ITRSS

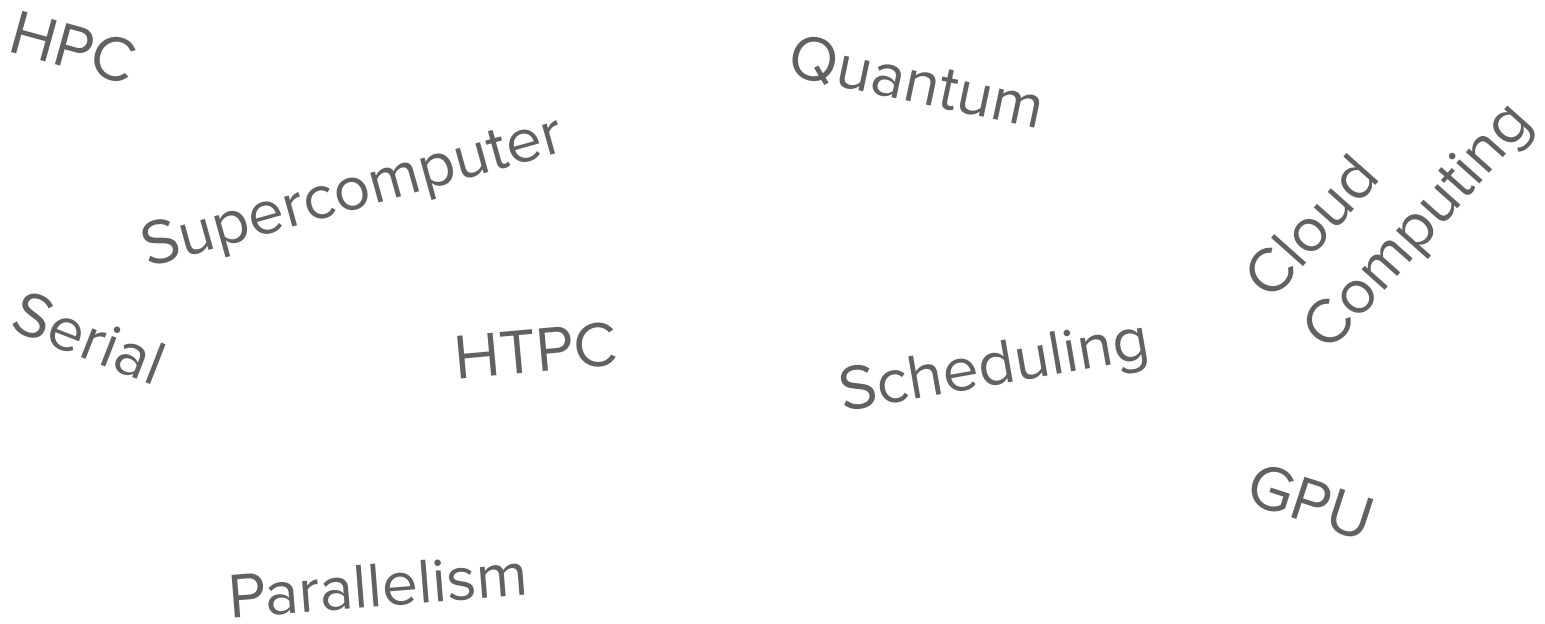
Research Support for Everyone

What is HPC?

Buddy Scharfenberg

April 2024

ITRSS


HPC
Quantum
Supercomputer
Cloud Computing
Serial
HTPC
Scheduling
GPU
Parallelism

What makes a supercomputer super?

The Power of Parallelism

Serial

Parallel

Which tasks can benefit?

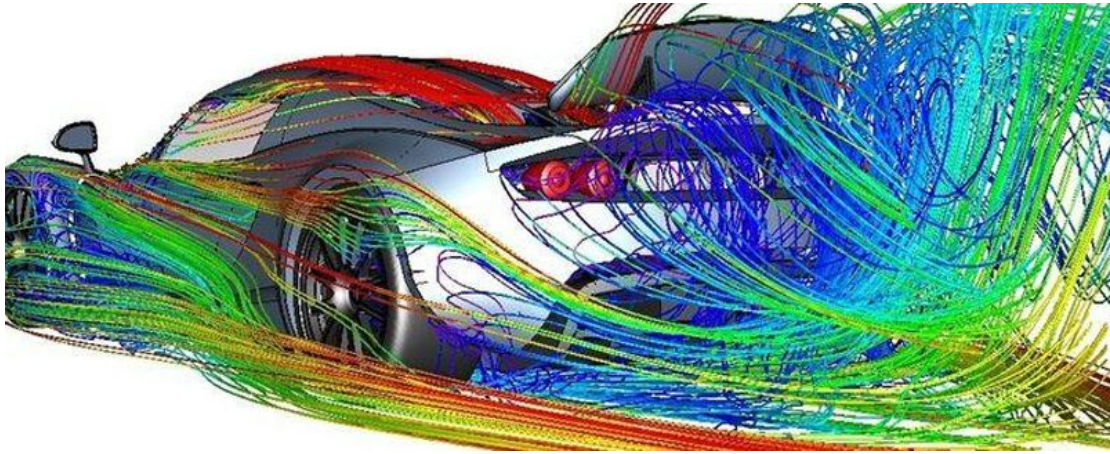
Molecular Dynamics

Fusion Energy

Weather prediction

Studying complex stress/strain

Nuclear technology development



Why These Tasks?

Steps do not require the result of a previous step “Data Independent”

Low amount of communication between processes

Many tasks of similar type, perhaps expressible as a loop.

$A(i) = B(i) + C(i)$ ✔

$A(i) = A(i-1) + 1$ ✘

Who Uses HPC?

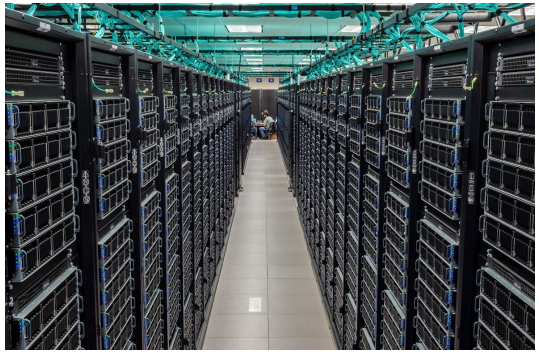
Academia

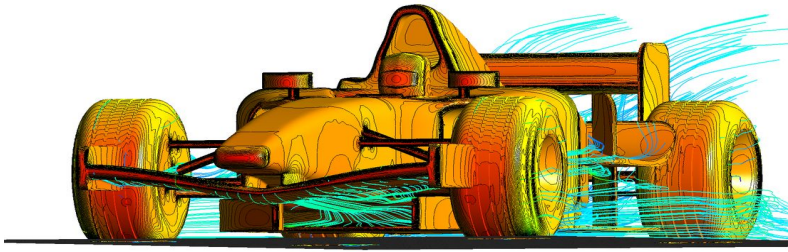
Research

Industry

Formula 1

NASA & Co.





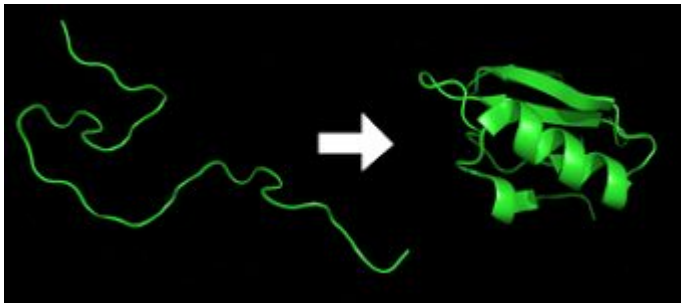
ITRSS

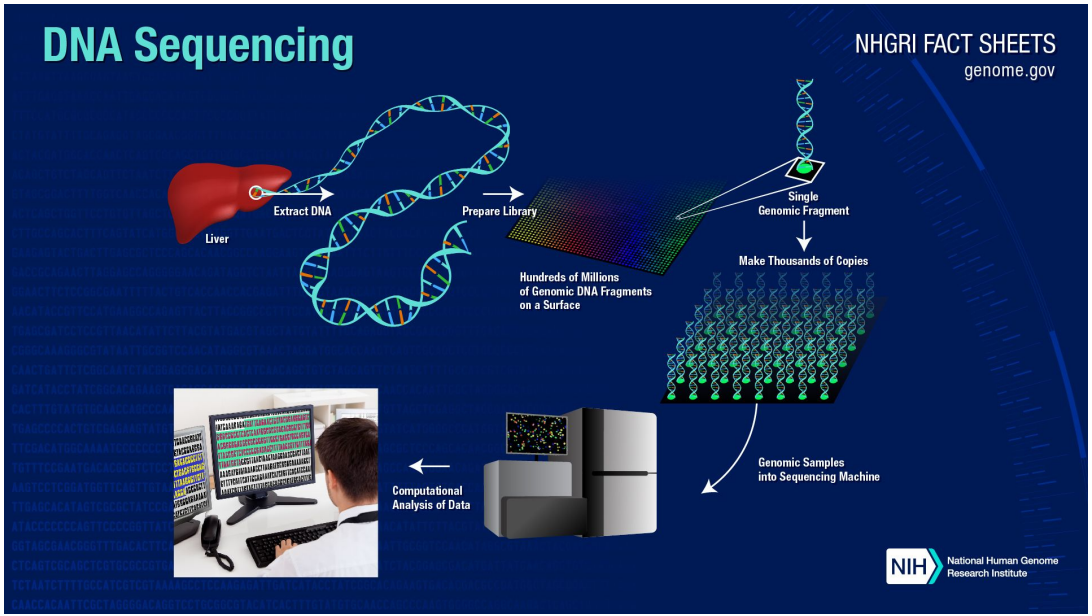
What about HTC?

High Throughput Computing

Gene/DNA sequencing

Protein folding





ITRSS

How Does a Supercomputer Work?

What do you need for HPC?

Staff

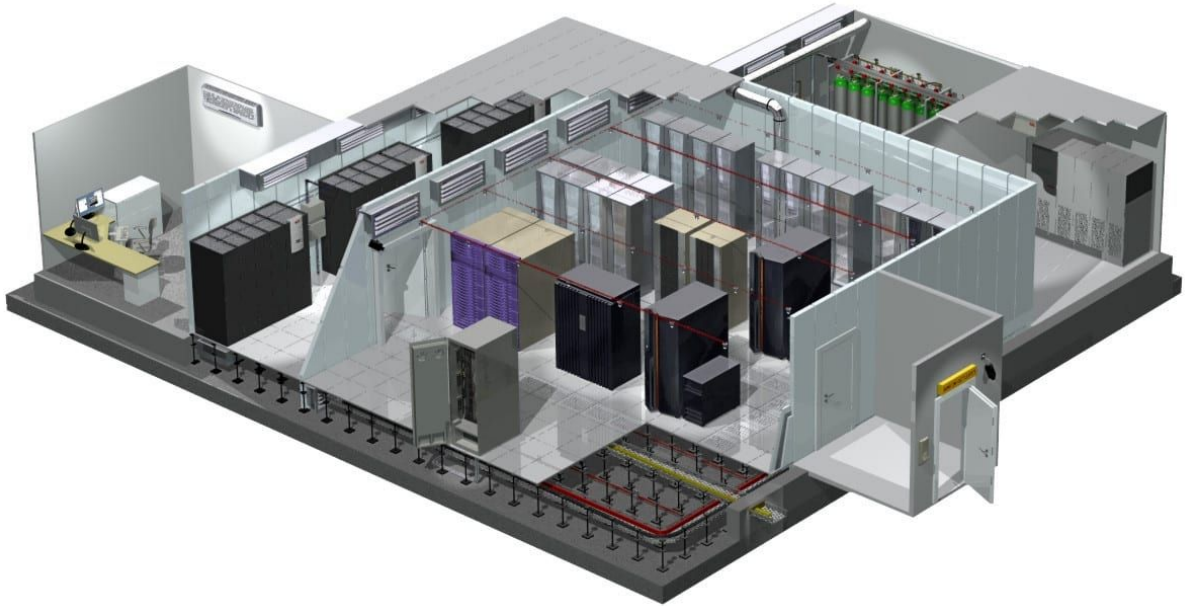
Space

Power

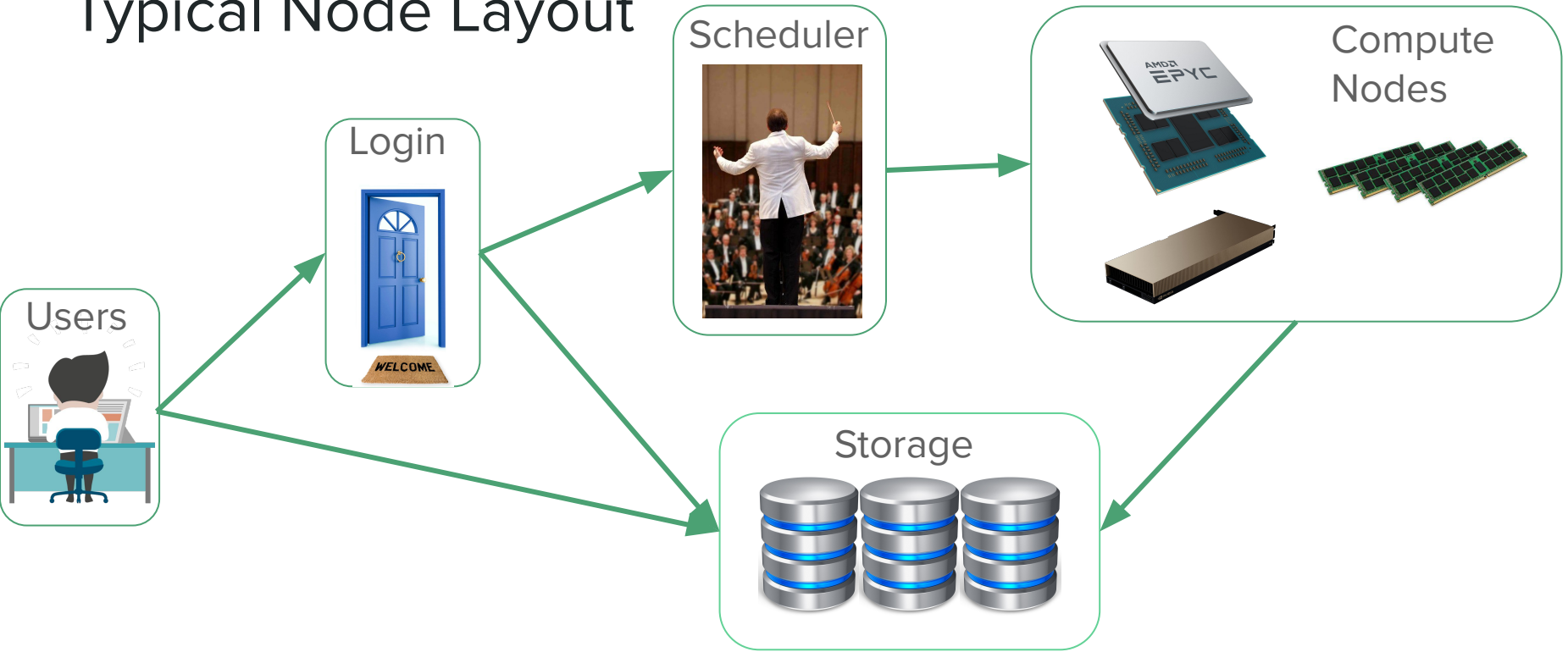
Cooling

Maintenance

Money



Typical Node Layout

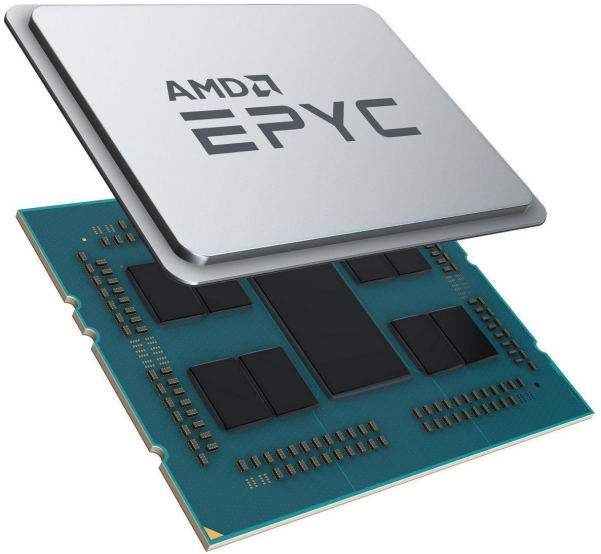


ITRSS

Basic Node Types

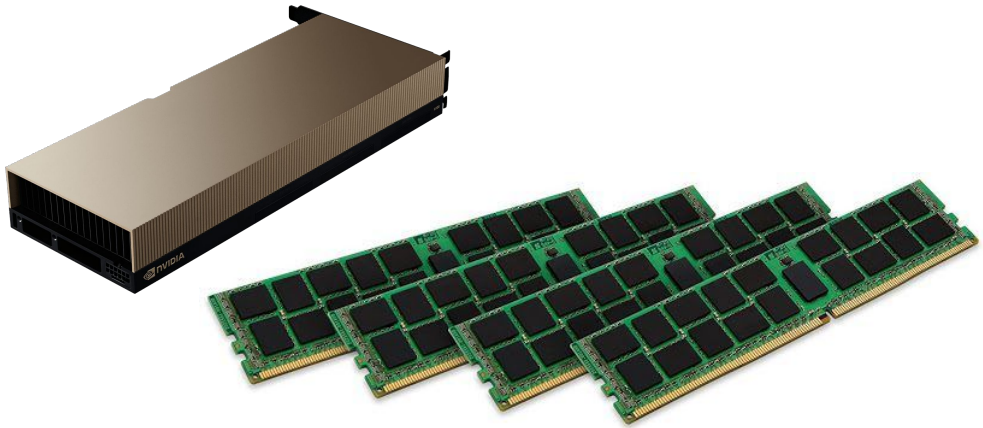
Login Node

Compute Node



Additional Node Types

GPU or High Memory



Storage or Data Transfer

Scheduler

Slurm

Partitions

Queues

Fairshare

Next up: Introduction to the Mill

ITRSS

Research Support for Everyone

Getting started with RSS & HPC

Buddy Scharfenberg

April 2024

ITRSS

Scope of the presentation

1. Introduction to RSS
2. Introduction to the Mill
3. Logging in
4. Storing files
5. Running Jobs
6. Software

Today's presentation will provide an introduction to the Mill and a high level overview of how it works.
We will also talk about other services ITRSS provides. At the end if time allows we will answer questions. "Getting started" documentation can also be found on our public wiki here: docs.itrss.umsystem.edu/pub/hpc/mill

ITRSS

Who is ITRSS?

ITRSS is short for Information Technology Research Support Solutions.

We provide technology solutions for researchers to be competitive with researchers nationwide. We are committed to helping UM researchers reduce their technology burden so they can focus on the scientific results of their research.

We have been providing dedicated campus HPC support since 2008.

Along with HPC support we also provide the following services:

- High Performance Workstations
- Linux Workstation builds
- Robotic data collection as a service
- Robotic machine design and manufacture
- Data presentation, extraction, cleaning
    - A.k.a data wrangling
- Assistance with data management plans
- Digital Signage for campus
- Instrumentation Support
- And More....

ITRSS

What Is the Mill, how did we get it?

The Mill is a large amount of computing and storage hardware made available through a handful of different funding sources as well as hardware migrated over from the Foundry (our previous HPC resource)

It has been designed and assembled into a computing cluster by ITRSS. It is also continuously maintained by ITRSS.

Please reference the Mill in all publications which use the resource.

General access is available to S&T students, faculty, and staff for research and coursework at no additional charge. A link to the application form can be found on our website at https://itrss.mst.edu/cluster/mill.

Priority access leases are available for research projects as well. Contact ITRSS for a cost estimate.

Thanks to Dr. Jeremy Maurer, and Perry Koob for winning grants to provide the new hardware for the Mill.

What is the Mill, what can it do?

Currently capable of a theoretical combined maximum compute capacity of around 380 TeraFlops.

We are expanding capability through migration of the existing Foundry hardware (800+ Teraflops).

- Each of the new compute nodes have 128 processors per node.
  - Older Foundry nodes have 64
  - Even older Forge nodes have 40.
- Each node is fully requestable by all users.
- All nodes can be requested for one job*.
- All nodes can intercommunicate with each other over a low latency Infiniband fabric.

ITRSS

What is the Mill, made from?

- Compute
  - 24 AMD (Ryzen 3)
  - 160 AMD (Ryzen 2) *
- GPU
  - 1 Nvidia H100 nodes with 8 GPUs NV-switch interconnected
  - 6 Nvidia V100 nodes with 4 GPUs each *
- Storage
  - Home: 50 TB of NFS connected Enterprise SAN based storage.
    - Longer term storage for building software etc.
    - 50Gb per user quota
    - Most Stable
    - **Not Backed up**
- Storage (continued)
  - Scratch: 61 TB of high speed Lustre based storage. *
    - Short term storage, cleaned when necessary systematically.
    - No user quota
    - Sometimes unstable
    - **Not Backed up**
  - Leased Storage: ~600TB usable Ceph based storage.
    - Longer term project storage for research groups etc.
    - Available in 1TB leased chunks
    - Mostly stable
    - **Not Backed up**

Logging In (More Info)

On Campus

- SSH access to login node
  - mill.mst.edu
- Putty on Windows or any other ssh client you prefer
- Linux and Mac have native ssh clients
- You can use ssh keys if you like

Off Campus

- VPN
  - If you connect to our campus VPN you will be able to use your SSO to log in.
- Without VPN
  - SSH keys are required

Both

- Web access through https://mill-ondemand.mst.edu **(NEW)**

ITRSS

Storing Files

1. Home Directories
2. Dedicated Storage
3. Other Storage
4. Upload and Download
5. Scratch

Files can be stored in various locations for different purposes. Be sure to backup all of your files whenever possible.

Home Directories

Each user gets a 50GB quota. If you have a specific need for more space than that feel free to ask. We may be able to increase it depending on the reason and if we have the extra space to do so.

You have full rights over your home directory. We recommend keeping it organized and removing old files that you don't need on the cluster anymore.

The Mill home directory storage is not backed up. Do not use it as a backup location and do not keep the only copy of important files there.

Dedicated Storage

Various researchers have leased storage that are available through the login nodes and are mounted to all systems on the cluster. These storage volumes are managed by their owners. These do not have individual user quotas unless the owner enforces them, however the whole volume will have a limit dependent on the amount leased.

Other Storage Access (More Info)

DFS, research, and web volumes are accessible from the Mill for transfering files. You cannot execute things from these directories or mount them on anywhere but the login node. Files must be copied to your home directory and run.

Running the command `"mountdfs"` will mount your dfs volume and you can access your sdrive and www drive from there with the aliases “`cds`” and “`cdwww`” respectively.

Scratch Space (Coming soon)

We have local scratch space on the nodes that can be accessed and we have a shared scratch space. These are for storing temporary files that are created while running and **should not be used for long term storage**. Files not accessed recently will be deleted periodically. Scratch is not guaranteed or backed up in any way.

Local scratch is located at /tmp/ <- this is only available on the node the file gets created on.

Global scratch is located: /lustre/scratch/$USER <- this is available cluster wide.

Leased Project Space

We have cluster attached project space which can be leased in 1TB chunks annually. These leased spaces are to allow research groups or individuals to have a large amount of space that has a similar use case to their home directory. The advantage of this space is that groups can share this space, and it can have much larger limits than the home directory space. **The project space is not backed up.**

Uploading and Downloading files

All of the standard scp, rsync, and sftp transfers will work on the Mill. We also have globus online file transfer for large data transfers which is fully integrated with our SSO, and file transfers are available with the web GUI (OnDemand) for smaller file transfers. There are reference guides and suggestions on which software you can use for file transfer on the Mill user wiki.

Git is also available tool for version control and downloading code. You can use git.mst.edu to store and version control your code. Please be sure to abide by the fair use policies on git.mst.edu. For classes/coursework please use git-classes.mst.edu.

ITRSS

Running Jobs

- Batch Jobs
- Interactive Jobs
- Basic Commands

Jobs are scheduled on the hardware or partition that you request as soon as the resource you request is available. You must request a certain amount of time for your job, though defaults are available. The more time you request the longer it may take to find that time in the scheduler. Only request as much as you need.

Batch Jobs (More Info)

- Create a <filename>.sub file
- Use the `sbatch` command
- Can specify partitions, processors, nodes, queues, time, and other resources.
- Everything can be specified either in the file or on the command line at submission time
- Command line arguments take precedence over in file arguments
- Output of command line is placed in a <filename>.out file
- Also files generated by the program you are running
- SLURM is now used instead of Torque (More Info)

Interactive Jobs (More Info)

- Uses the `sinteractive` command
- All options specified on command line
- Starts an interactive session on a back end node
- Dies once you exit the session
- Limited to one node at most
- X-forwarding is passed through to your local machine if you enabled it on your ssh session
- **GUI available through web interface https://mill-ondemand.mst.edu (NEW!)**

Basic Commands

- `squeue -u <username>` - Show your running jobs
- `scancel <jobnumber>` - Cancel a specific job
- `sbatch <file.sub>` - Submit jobs saved in submission files
- `scontrol` - View and modify Slurm configuration and state

ITRSS

Software

- Where it is and how to get to it
- Rolling your own software
- Requesting software

We have a large variety of software on the Mill. We can run most software that runs on 64 bit processors, and provides Linux support in some way. Parallel software is generally the most effective use of resources.

Where is the Software

- Most of the software on the Mill is installed in the /share/apps directory
- Software is loaded using modules.(More Info)
  - `module avail` - shows all available software titles.
  - `module load <software>` - loads a given software into your environment
  - `module whatis <software>` - shows basic information about a given software
  - `module show` - shows the script that runs when a module is loaded
- Modules temporarily modify your environment

Compiling Your Own Software

You can compile any software you like in your home directory. This software still counts towards your 50GB quota. Make sure that you have all modules loaded that you will need to compile the software.

Please ensure that you are authorized to use any software that you have and that you follow all S&T Acceptable Use Policies: https://it.mst.edu/policies/network-acceptable-use-policy/

Please also ensure that your software falls into either DCL1 or DCL2 as defined at https://www.umsystem.edu/ums/is/infosec/classification-definitions. If you have software which falls outside of these classifications contact us for advisement.

Requesting Software

If multiple people will potentially need the software we can install it in /share/apps and create a module for it. This will ensure that there are no compilation differences between different people and will not take up space in your home directory. Also unless otherwise license restricted, will be available to everyone on the cluster to use.

If you do require software with special licensing please let us know about the license and we can limit access to it.

Please send software requests via email to itrss-support@umsystem.edu

ITRSS

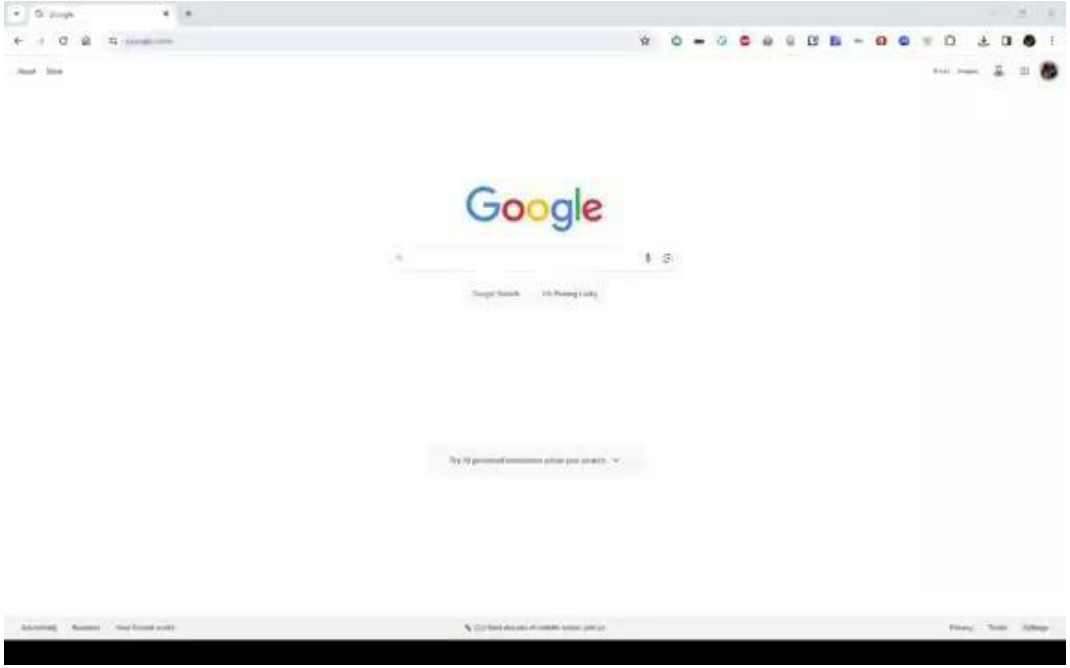
Information For Software Requests

Please include in an email to itrss-support@umsystem.edu:

- Software Name
- Where to get the software
- Where the documentation for compiling/installing the software is located
- Any license restrictions that the software may have
- Any special options, features, customizations, or addons that you need
- What you will be using the software for
- A known test case for you to verify that the software is working

ITRSS

OnDemand Demo Video



The Mill for coursework

Due to part of the funding for the Mill hardware coming from MoExcels (thanks again Perry!) we are happily providing dedicated nodes available for students to use for coursework. We have created a separate OnDemand web GUI for students to use to make use of this priority access partition from wherever they are. https://mill-classes.mst.edu

The same interface as the research facing ondemand, with some restrictions applied for ease of use.

Funding Opportunity for HPC! NSF 24-530 (Area 2) April and October deadlines
https://new.nsf.gov/funding/opportunities/campus-cyberinfrastructure-cc/nsf24-530/solicitation

Questions?

Next up: Getting Started With HPC

ITRSS

Research Support for Everyone

Getting Started with HPC

Danny Bylina

April 2024

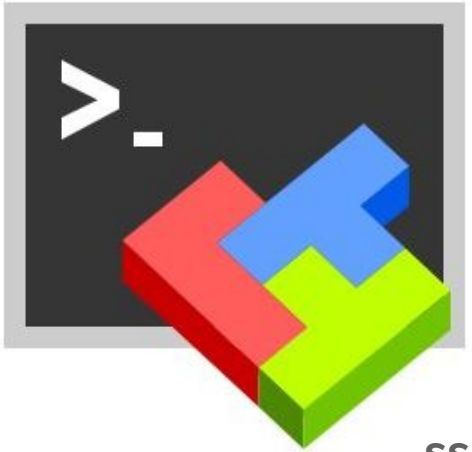
ITRSS

Getting Started With Linux

Connecting to the Cluster

Windows: MobaXterm

(Please grab portable version)



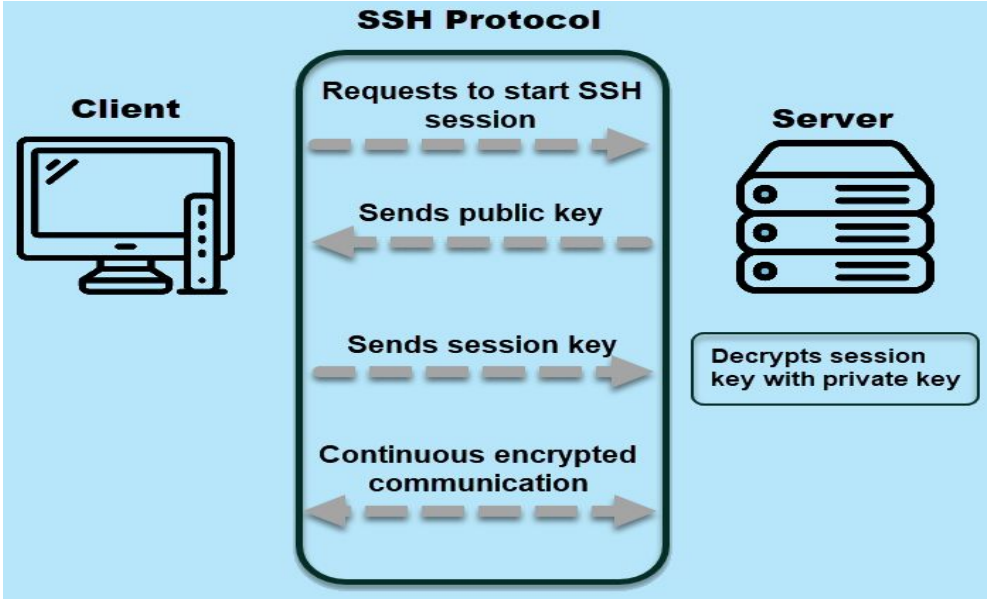
MacOS and Linux: Terminal

ssh (sso)@mill.mst.edu

SSH

What is ssh?

How does it work?



Using the Terminal

What happens when a terminal opens?

How do we navigate to are home directory?

How do you learn all of these commands?

```
command [-flag(s)][-option(s)[value]][argument(s)]
```

Useful Commands

cd : change directory
touch : create file
mv : move file
rm : remove file
ls : list contents
cp/scp : copy/network copy
mkdir : make directory
rsync : robust network copy

cat : print contents
less : read contents
man : manual pages
pwd : print working directory

Workshop Instructions

Create a file and a folder

List the contents of your current directory

Move the file into the folder and rename it

Make a copy of your file in the same folder

Delete the copy

cd	: change directory
touch	: create file
mv	: move/rename file
rm	: remove file
ls	: list contents
cp/scp	: copy/network copy
mkdir	: make directory
rsync	: robust network copy

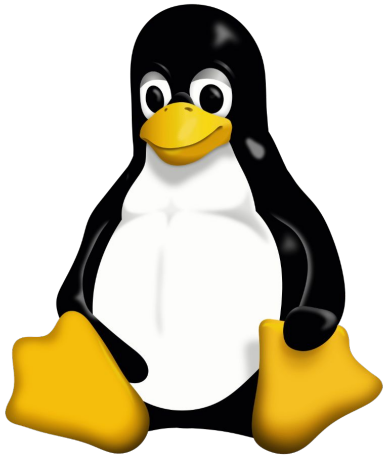
More Ways to Use Commands

Piping commands: |

Redirecting files: >, >>

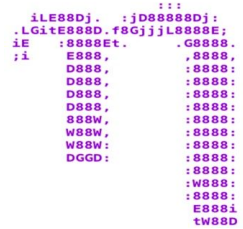
Chaining commands: &&

Writing commands into scripts



Editing Files

nano



- Simple
- Easy to use
- Always available

vi

- Learning curve
- Very powerful
- Gateway to vim

Honorable Mentions: emacs, vim

Advanced Topics

Multiple open windows with tmux/screen

Editing your .bashrc with aliases/scripts

Automating tasks with bash scripts

```
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
```

ITRSS

Cluster Specific Considerations

Using Slurm

sinfo : check state of partitions and nodes
squeue : monitor jobs or job steps
srun : submit a job for execution in real time
sbatch : submit job script for later execution
salloc : request interactive session

ITRSS

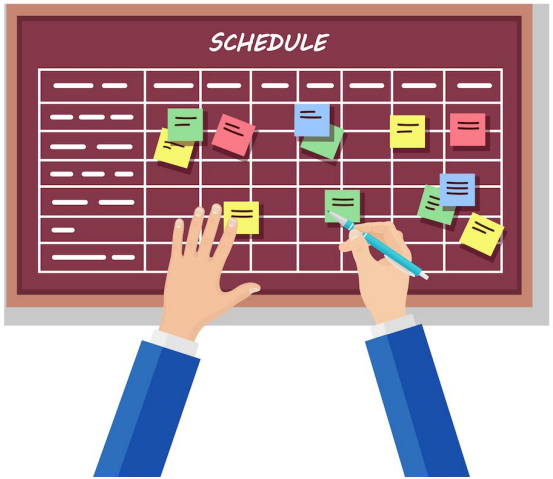
Writing Batch Files

Choosing a partition and queue

Requesting the correct resources

Adding commands to be run

Specifying output



https://docs.itrss.umsystem.edu/pub/hpc/mill

Slurm Workshop

sinfo	: check state of partitions and nodes
squeue	: monitor jobs or job steps
srun	: submit a job for execution in real time
sbatch	: submit job script for later execution
salloc	: request interactive session

Check partitions with sinfo

Request interactive session

See your job with ‘squeue -u <user>’

Exit your interactive session

Copy the example batch file from PATH and run it with sbatch

cp /share/apps/workshop/hostname.sbatch /home/(SSO)

Using Software

Modules:

module avail : list available modules
module load : load module from list
module list : list loaded modules
module purge : unload modules

Compiling Software

Check modules/conda first

Use your home directory

Follow instructions per software

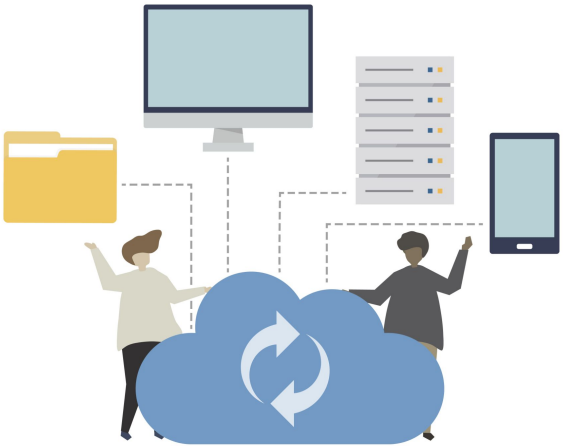
Use an interactive session

Transferring Data

SCP/RSYNC

Globus

Where to put it



Next up: Getting started with OpenOnDemand

ITRSS

Research Support for Everyone

Getting Started with Open OnDemand

Danny Bylina

April 2024

What if I don’t want to use a terminal?

ITRSS

Open OnDemand

Use cluster via remote desktop

Use gui applications easily

Only approved way to use VScode

Easiest way to use Jupyter Notebook

https://mill-ondemand-p1.itrss.mst.edu/

ITRSS

Managing Files

Can be done here

How to choose which way to manage files?

Why would I do it here?

https://mill-ondemand-p1.itrss.mst.edu/

ITRSS

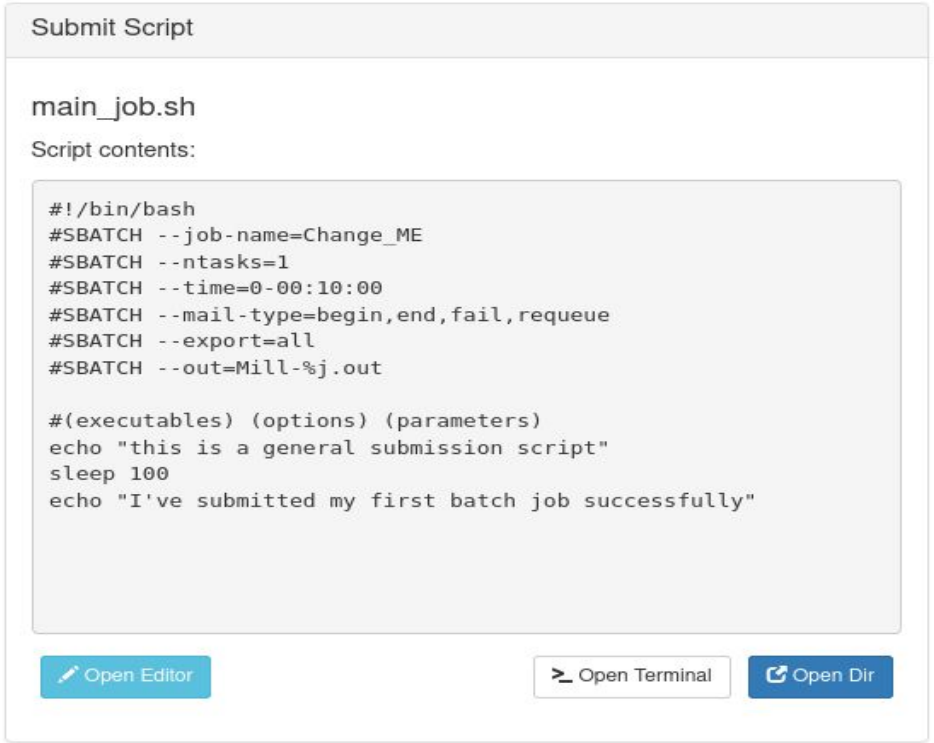
Job Composer

Use the default setting to save you jobs

Select open editor to modify your existing job

Open directory to check the results of your job

Run your job with the play button on top on the screen

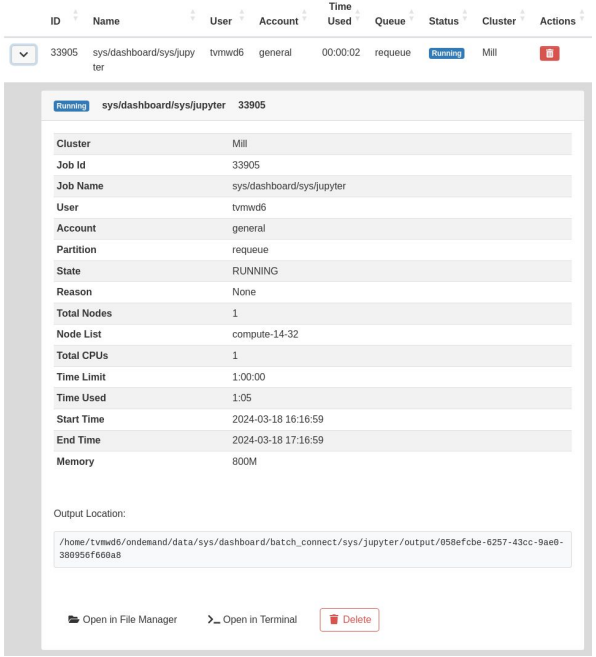


ITRSS

Monitoring Jobs

See your jobs or all jobs

Click expand to see more job results



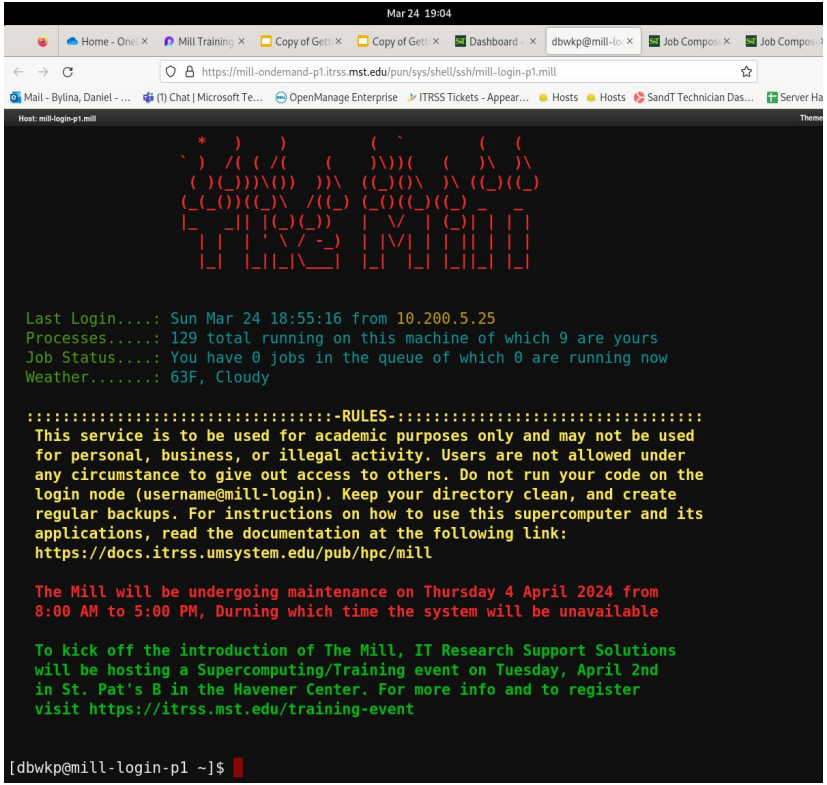
https://mill-ondemand-p1.itrss.mst.edu/

ITRSS

Using Command Line

Under the tab Clusters

Click on Mill Shell Access



ITRSS

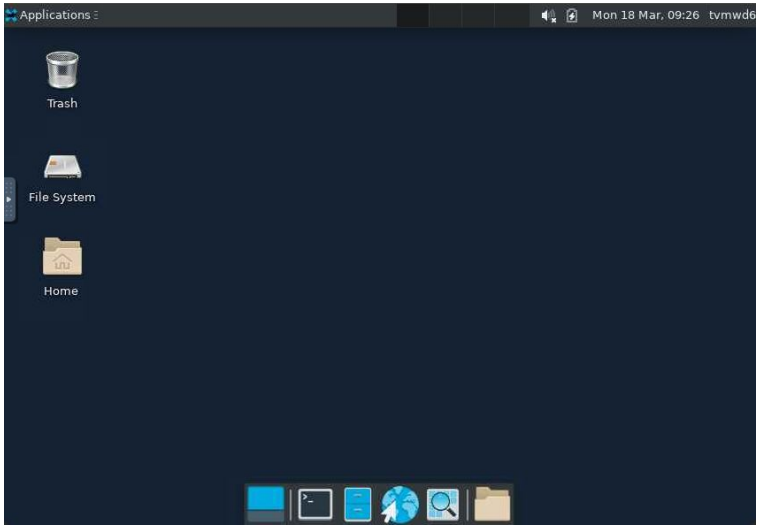
Desktop Mode

Remote desktop session

Managing files graphically

Open terminal to manage software/environment

Use browser to download data



https://mill-ondemand-p1.itrss.mst.edu/

ITRSS

Choosing a Program

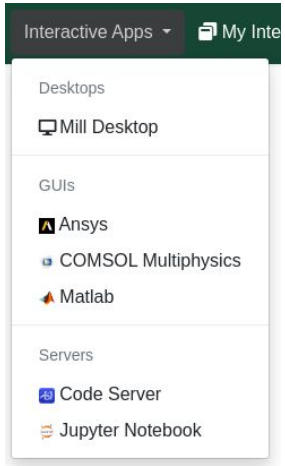
VSCode

Jupyter Notebook

Ansys

Comsol

Matlab



https://mill-ondemand-p1.itrss.mst.edu/

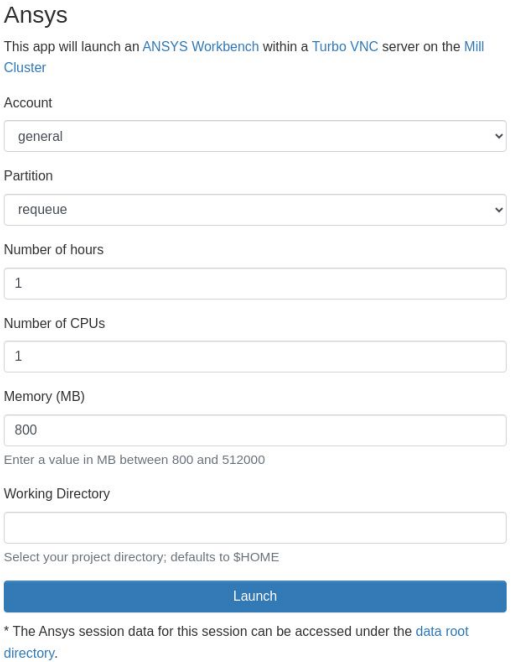
ITRSS

Requesting Resources

Notice similarities to batch scripts

Try not to request too much

Limits are in place



https://mill-ondemand-p1.itrss.mst.edu/

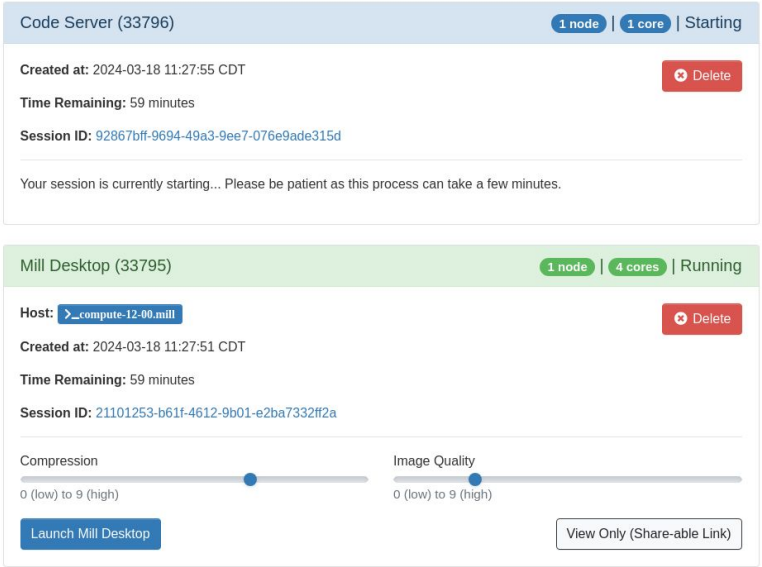
ITRSS

Managing Sessions

Join sessions here

Delete sessions to free resources

Sliders change video quality



https://mill-ondemand-p1.itrss.mst.edu/

ITRSS

Research Support for Everyone

Teaching With the Mill

Buddy Scharfenberg

April 2024

https://mill-classes.mst.edu

Questions?

Next up: Lunch and Speakers

Next up: Workflow Tools

ITRSS

Research Support for Everyone

Workflow Tools

Tim Maninger

April 2024

ITRSS

Workflow Tools

Anaconda

Jupyter Notebook

Git

Containers

docs.itrss.umsystem.edu

Anaconda (conda)

Create “virtual environments”

Contained set of packages

Allows tight control of versions

Sharable, reproducible, portable

docs.itrss.umsystem.edu

Using Conda

Centralized location for envs

Change envs for other workflows

Search and install packages

anaconda.org/anaconda/repo

bioconda.github.io/conda-package_index.html

docs.itrss.umsystem.edu

Jupyter Notebook

JN allows an interactive coding experience

Can use Python, R, etc.

Makes creating and interacting with visualizations much easier

ITRSS

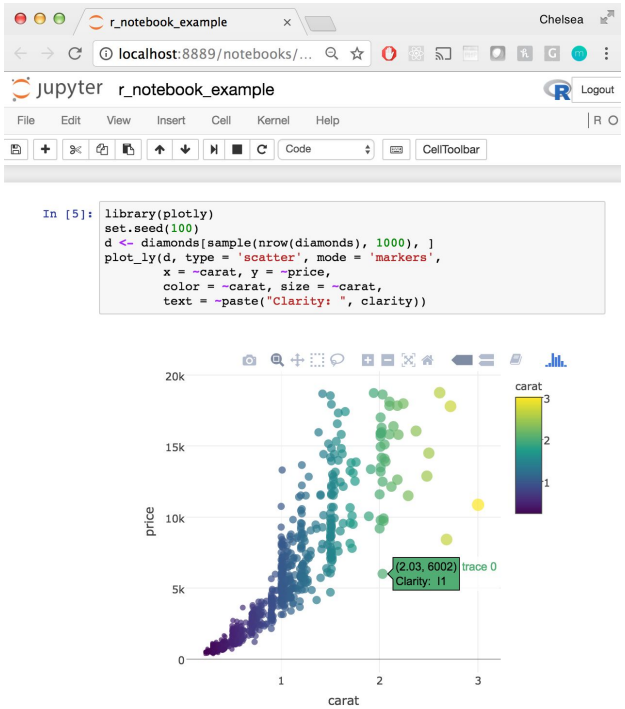
Jupyter Notebook

docs.itrss.umsystem.edu

OnDemand or SSH tunneling

Notebooks work well with GitLab for sharing/version control

git.mst.edu



Git

Primarily for code/text files

Keep track of changes

Revert to old versions if necessary

Branches for non-destructive changes

git-lfs allows larger files if necessary

git.mst.edu

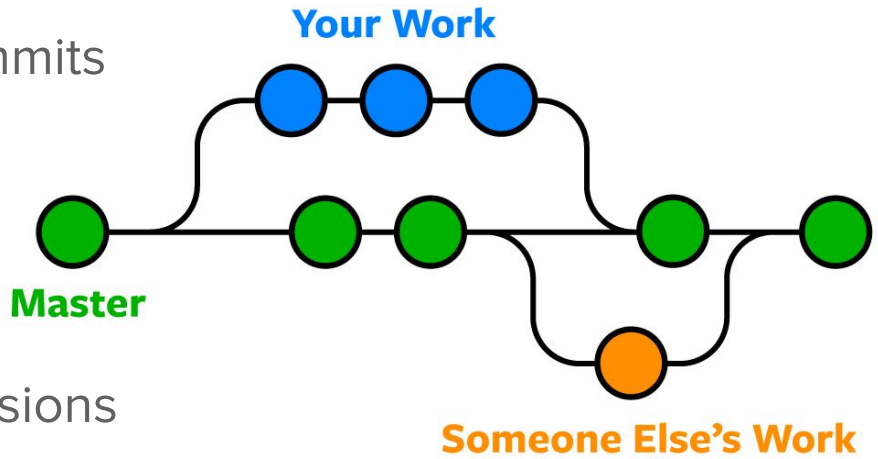
Git

Organized into repos, branches, and commits

Commits add your changes to branches

Branches can be merged once stable

Users can have different levels of permissions



docs.itrss.umsystem.edu

Containers

Make difficult to install software usable

Large library of containers available

Minimal performance impact

Apptainer (Singularity)

docs.itrss.umsystem.edu

Containers

Still able to use MPI and GPUs

Multiple containers can work together

Self contained for better reproducibility

Apptainer (Singularity)

Next up: Workshop Time until 4:30