

# Introduction to Open OnDemand and Best Practices

Center for Neurocomputation and Machine Intelligence  
Wu Tsai Institute  
Yale Center for Research Computing



- Have you heard of High Performance Computing?
- Have you used a High Performance Computing cluster?
- Have you heard of Open OnDemand?
- Have you used Open OnDemand?

[pollev.com/pingluo216](https://pollev.com/pingluo216)

# Outline

- What is High Performance Computing (HPC)
- What is Open OnDemand (OOD)?
- Basic use of OOD
- Advanced topics

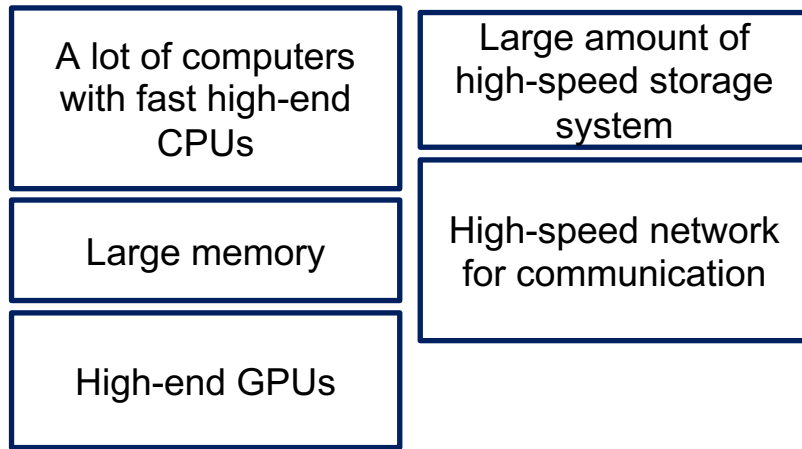
# What is High Performance Computing

- Technology that leverages the computing power of many high-end computers to solve problems that are beyond the capabilities of standard computers

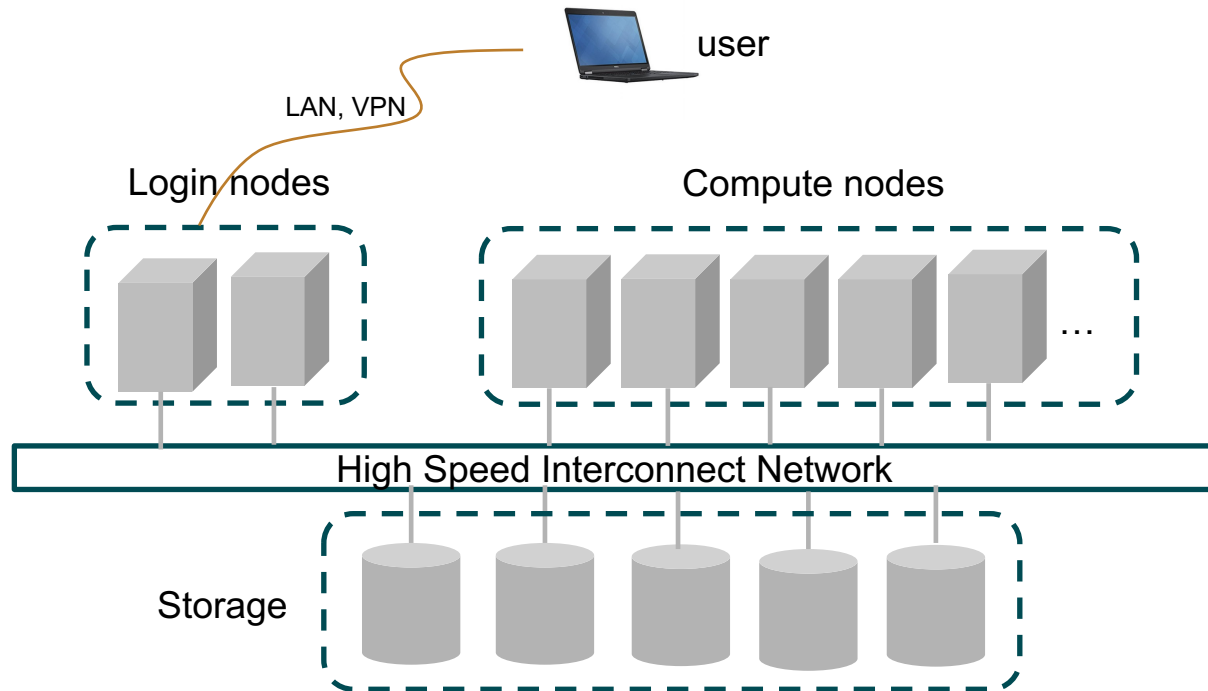


# What is an HPC Cluster

- An HPC cluster is a computing system that provides high performance computing capabilities to solve complex problems
- It is typically used for scientific computing, big data analysis, and AI modeling.



# Anatomy of an HPC Cluster



# Cluster Images



1U rackmount servers



12 servers in a 6U chassis



Cluster in 42U racks  
42U = 73.50"

# Top 500

1 PFlops =  $10^{15}$  flops.

- Website: <https://www.top500.org>
- Publishes the most powerful 500 HPC systems in the world every six months based on HPL benchmark results:

Rank	System	Cores	Rmax (PFlop/s)
1	Frontier Oak Ridge National Laboratory	8,699,904	1,194.00
2	Supercomputer Fugaku RIKEN Center for Computational Science, Japan	7,630,848	442.01
3	LUMI EuroHPC, Finland	2,220,288	309.10

June 2023 list



# Fundamental Software Components for HPC

- Linux operating system
- Batch jobs
- Software modules

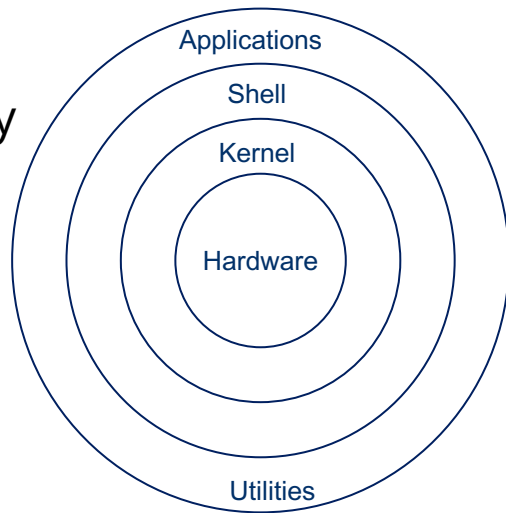
# Linux

- Free and open source
- Excellent flexibility, stability, extensibility
- Supports multiple users
- Highly secure
- Used by most HPC clusters

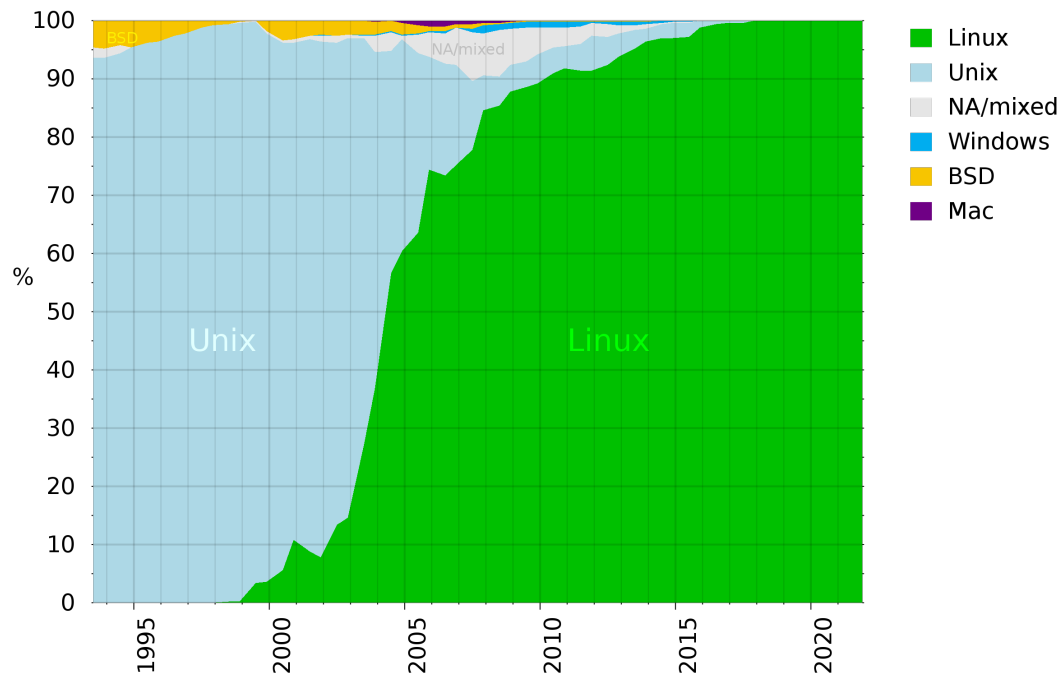
Shell programming

Introduction: <https://docs.ycrc.yale.edu/PIL/>

Advanced: <https://tldp.org/LDP/abs/abs-guide.pdf>



# Top 500 & Linux



[https://en.wikipedia.org/wiki/Supercomputer\\_operating\\_system#/media/File:Operating\\_systems\\_used\\_on\\_top\\_500\\_supercomputers.svg](https://en.wikipedia.org/wiki/Supercomputer_operating_system#/media/File:Operating_systems_used_on_top_500_supercomputers.svg)

# Batch Jobs

- A batch job is a script consists of resource specifications and user commands.
- A batch job is submitted to a batch scheduler which will schedule it to run on the cluster based on resource availability.
- Slurm is a cluster management and job scheduling system used by a lot of HPC clusters.
- Other batch schedulers include PBS Pro, SGE, LSF, Load Leveler, etc.

A sample `slurm` job script

```
#!/bin/bash
#SBATCH --job-name=test
#SBATCH --output=test_job.txt
#SBATCH --nodes=1
#SBATCH --cpus-per-task=1
#SBATCH --partition=day
#SBATCH --mem=10g
#SBATCH --time=10:00
# user commands
./hello.exe
```

Commonly used `slurm` commands

```
sbatch my_job.sh
queue
scancel job_id
```



[docs.ycrc.yale.edu/clusters-at-yale/job-scheduling](https://docs.ycrc.yale.edu/clusters-at-yale/job-scheduling)  
[slurm.schedmd.com](https://slurm.schedmd.com)

# Sample Job List

[pl543@login2.mccleary ~]\$ squeue

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	ODELIST(REASON)
9878984	admintest	/gpfs/gi	tl397	PD	0:00	1	(BeginTime)
9878985	admintest	/home/tl	tl397	PD	0:00	1	(BeginTime)
10398484	admintest	./master	root	PD	0:00	1	(BeginTime)
12358304	admintest	interact	ms725	R	5:03:55	1	r106u17n01
12358209	admintest	interact	rdb9	R	5:09:01	1	r208u27n01
12356748	admintest	interact	rdb9	R	6:18:36	1	r205u01n08
12355487	bigmem	interact	th722	R	6:25:58	1	r107u04n01
12364430	bigmem	ood-jupy	cl2553	R	2:43:49	1	r105u38n01
12368556	bigmem	merge_pr	kjj28	R	1:23:27	1	r202u06n01
12359710	bigmem	ood-rstu	as4388	R	4:20:26	1	r106u04n01
12357776	bigmem	ood-rstu	ek655	R	3:33:25	1	r202u12n01
12362307	bigmem	in75GSSn	ag662	R	3:27:55	1	r202u15n01
12366342	bigmem	ood-rstu	zb68	R	2:09:55	1	r107u04n01
12356723	bigmem	ood-rstu	dz286	R	6:20:17	1	r104u38n01
12359182	bigmem	ood-jupy	np593	R	4:31:06	1	r104u38n01
12365243	bigmem	ood-jupy	ah2428	R	2:29:55	1	r106u04n01
12366839	bigmem	ood-jupy	tl688	R	1:58:43	1	r104u38n01
12345316	bigmem	covid1	wl545	R	11:16:58	1	r104u01n01
12345151	bigmem	pbmc_cgg	wl545	R	11:35:55	1	r104u01n01
12345319	bigmem	covid2	wl545	R	11:16:58	1	r105u38n01
12362320	bigmem	archr_ne	yl2687	R	3:24:55	1	r202u06n01
12357411	bigmem	ood-rstu	yl2687	R	5:43:43	1	r104u01n01
12365366	bigmem	ood-rstu	jg2587	R	2:26:55	1	r106u04n01

# Software Modules

- HPC clusters usually have many software packages installed.
- Software modules are used to precisely control which software package and which version to use

Commonly used module commands

```
module avail  
module spider python  
module load Python  
module list  
module purge
```

# Open OnDemand

- An open-source web portal developed at the Ohio Supercomputer Center which provides an easy-to-use web interface to HPC resources



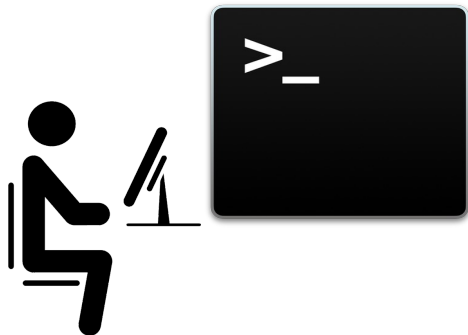
# Benefits of OOD

- Web-based and doesn't require installation of client software on your local machine (except for a modern browser)
- User-friendly, easy to learn, and greatly reduce the entry barrier to HPC.
- Provides a simple way to run GUI applications remotely on a cluster

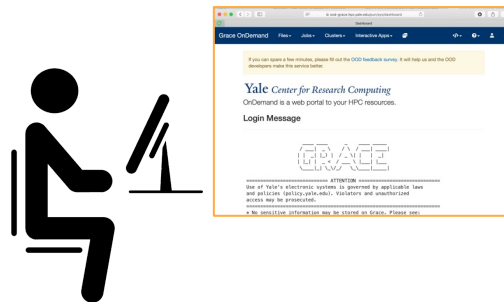


# Command-line vs Web-based Access

- Pros:
  - Very efficient for experienced users
  - Good for large scale job submission and data processing
- Cons:
  - High learning curve
  - Not GUI application friendly



- Pros:
  - Minimum learning curve
  - Simple to use
  - GUI application friendly
- Cons:
  - Not as efficient as the command line
  - Can be challenging to submit array jobs



# How to Access OOD

- Type the URL based on the type of account you have on the cluster into your browser
  - `ood-clustername.ycrc.yale.edu`
  - `coursename.ycrc.yale.edu`
  - `projectname.ycrc.yale.edu`
- Always **log in with your Yale NetID** and password
  - Use all lowercase letters for NetID
    - PL543, PI543 : **permission denied**
- VPN is required to access OOD off-campus



# Dashboard

Grace OnDemand Apps Files Jobs Interactive Apps Utilities Courses Clusters My Interactive Sessions Develop Help Logged in as pl543 Log Out

Welcome to OnDemand on Grace. Please report any issues or suggestions to [hpc@yale.edu](mailto:hpc@yale.edu)

## Yale Center for Research Computing

OnDemand is a web portal to your HPC resources.

**Pinned Apps** A featured subset of [all available apps](#)



Jupyter

System Installed App



MATLAB

System Installed App



Remote Desktop

System Installed App



RSTUDIO

RStudio Server

System Installed App



RSTUDIO

RStudio Desktop  
(with Conda R)

System Installed App

## Login Message

### ATTENTION

Use of Yale's electronic systems is governed by applicable [laws and policies](#). Violators and unauthorized access may be prosecuted.

- No sensitive information may be stored on Grace. Please see [cybersecurity.yale.edu/protectyourdata](https://cybersecurity.yale.edu/protectyourdata)
- Grace documentation: [docs.ycrc.yale.edu/clusters-at-yale/clusters/grace](https://docs.ycrc.yale.edu/clusters-at-yale/clusters/grace)
- Run [getquota](#) for your storage usage & limits
- Run [mydirectories](#) for absolute paths to your storage spaces
- Please send questions, comments or criticisms to [hpc@yale.edu](mailto:hpc@yale.edu)
- The next maintenance period on Grace will be Jun 6-8, 2023

10/04/2022 The scheduled maintenance on the Gibbs storage system has been completed. Gibbs

# Shell Access

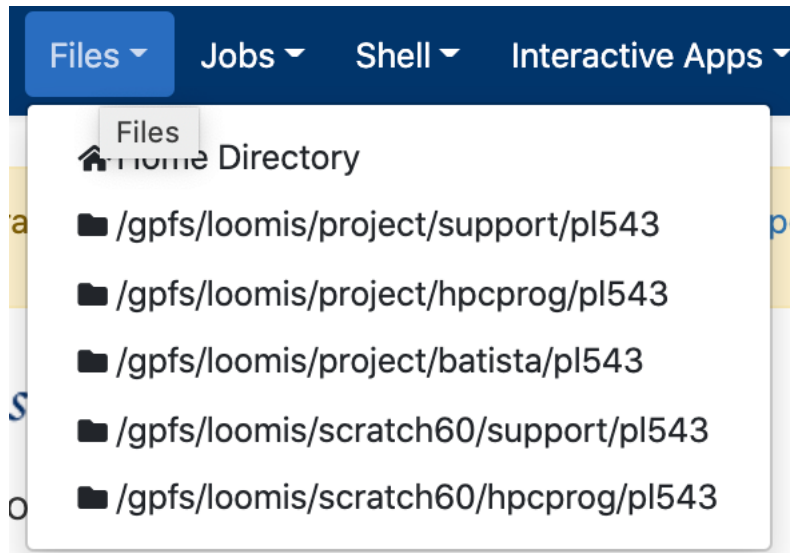


- Web-based secure shell access
- Similar to:
  - [ssh](#) on MacOS and Linux
  - [putty](#) and [MobaXterm](#) on Windows
- No need to set up public key authentication by yourself

A screenshot of a web browser window showing a terminal interface for 'Grace OnDemand'. The browser's address bar shows the URL 'ood-grace.hpc.yale.edu/pun/sys/shell/ssh/grace.hpc.yale.internal'. The terminal window has a black background with white text. It displays the 'GRACE' logo in a stylized font. Below the logo, there is a section titled 'ATTENTION' with a warning about Yale's electronic systems. This is followed by several bullet points providing information about storage usage, documentation, and maintenance. At the bottom, there is a 'PRO TIP' section and some command-line examples for submitting jobs. The terminal prompt shows the user is logged in as 'pl543@grace1'.

# File Manager

- Navigate files in home, project, and scratch
- Download/upload files
- View/edit files
- Copy/move files
- Rename files
- Select files
- New files
- Delete files



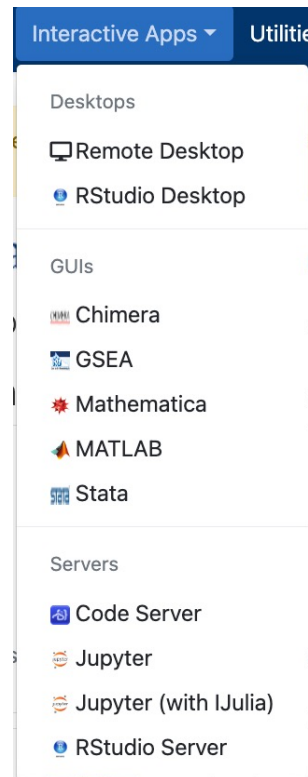
# Tips for Using File Manager

- Good for uploading/downloading files up to 10 GiB
- You can check results without downloading the files to your local machine
  - Select a file and click 'view' to display its content
- For Windows users:
  - Create a job script locally and upload it
    - Job may fail due to the DOS style line break in the job script
    - Fix the problem with '[dos2unix jobscript](#)'



# Interactive Apps

- VNC-based
  - Remote Desktop
  - Rstudio+Conda R
  - Matlab
  - Mathematica
  - Stata
- Servers
  - Jupyter
  - RStudio Server
  - Code Server (VS Code)



# Parameters for Interactive Apps

- Users can specify:
  - Number of hours `--time=hh:mm:ss`
  - Number of CPUs per node `--cpus-per-task=1`
  - Number of GPUs per node `--gres=gpu:2`
  - Memory in GiB per CPU core `--mem-per-cpu=10g`
  - Partition `--partition=partition`
  - Reservation `--reservation=reserv`
  - Check-box for email when job starts `--mail-type=BEGIN`
  - Slurm account (for non-default group) `--account=account`
  - Additional job options: any valid options from Slurm
  - App specific parameters: conda environment, additional modules, etc
- Toggle the “more options” checkbox to show or hide additional fields
- The default job parameters work fine in most cases

Remote Desktop version: v0.2.1

This app will launch an interactive desktop on a compute node.

Number of hours

Number of CPU cores per node

The equivalent slurm option is `--cpus-per-task`

Memory per CPU core in GiB

Partitions

Number of GPUs per node

Set number of GPUs per node. The minimal value is 1 for partition `gpu`, `gpu-devel`, and `scavenge-gpu`.

Reservation (optional)

☐ I would like to receive an email when the session starts

☐ Check the box to view more options

Launch



# Jupyter

- `ycrc_default` is a read-only Conda environment for testing and is stored in a central location
- Build your own Jupyter environment with

```
module load miniconda
conda create -n jupyter_env jupyter jupyterlab
```
- Make it available to OOD

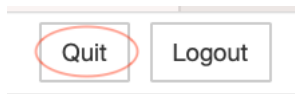
```
ycrc_conda_env.sh update
```
- By default, Conda packages are installed in sub-directories within your project directory. Customize the paths in `.bashrc` with:

```
export CONDA_ENV_PATH=/path/to/conda_envs
export CONDA_PKG_DIRS=/path/to/conda_pkgs
```
- By default, **Jupyter notebook** will be started, unless “start **jupyterlab** checkbox” is selected



# Stop Jupyter

- Stop your Jupyter session immediately after you are done to free any resources allocated to it
  - Jupyter Notebook: click 'Quit' from



- Jupyter Lab



- 'Logout' or 'Log Out' closes your current connection to the Jupyter server, allowing you to reconnect to the server later
  - Resources are are not freed

# RStudio Server

- Uses R from an R module that includes many preinstalled R packages and has undergone extensive testing
- Works seamlessly with a supporting module [R-bundle-Bioconductor](#), which provides many additional bioinformatics R packages
- Provides RStudio Server in an Apptainer (formerly Singularity) container
- Allows users to install additional R packages easily
- To stop an RStudio Server instance:
  - Step 1: save and close all files
  - Step 2: type 'q()' to exit the session
  - Step 3: click the 'delete' button to stop the RStudio server



# RStudio Desktop

- Use only if you need a customized R environment or need an R version not provided by the RStudio Server OOD app

- You must provide R and RStudio in a Conda environment

```
module load miniconda  
conda create -n r_env r-base r-essentials rstudio-desktop
```

- Mixing a Conda environment and modules on the cluster is not recommended
- Run this command to make your Conda environment available to OOD


```
ycrc_conda_env.sh update
```

- To customize the conda paths, add the following commands in your .bashrc

```
export CONDA_ENV_PATH=/path/to/conda_envs  
export CONDA_PKG_DIRS=/path/to/conda_pkgs
```

- Two ways to end RStudio Desktop cleanly

Click 'Files' -> 'Quit Session' or type 'q()' in RStudio consol.

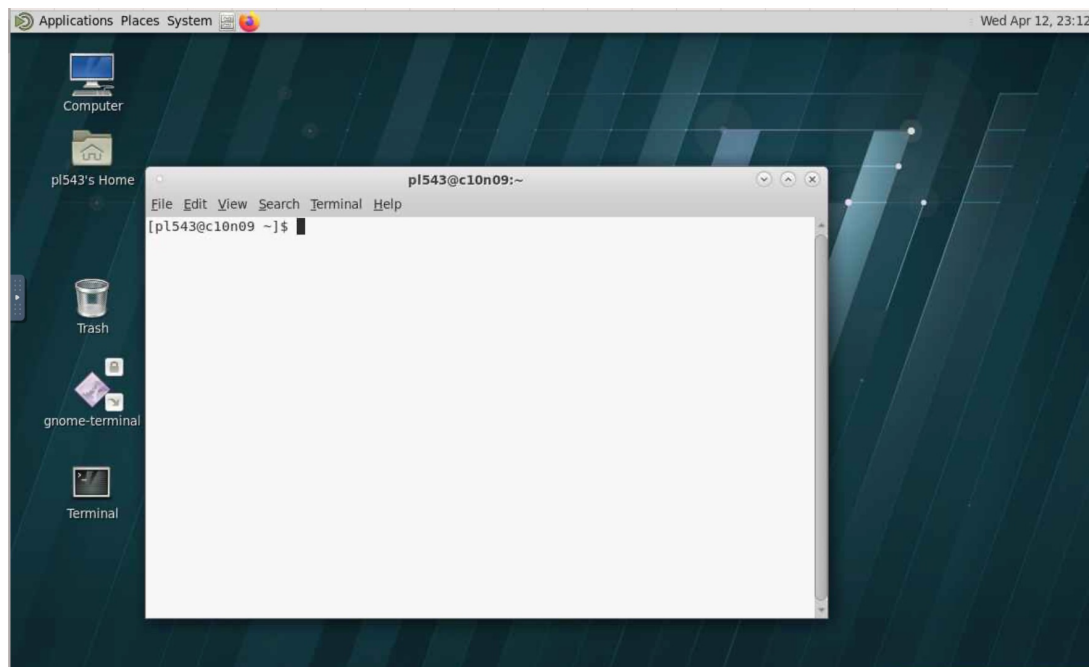
- Only click  Delete if the session is frozen.

# RStudio Server vs RStudio Desktop

- RStudio Server is preferred
  - Works well with modules on the cluster
  - Utilizes the CPU and memory more efficiently than RStudio Desktop
  - Provides better integration with the browser
  - Offers easy copy/paste operations
- RStudio Desktop is needed when
  - A specific R version that is not provided by RStudio Server
  - A special R environment

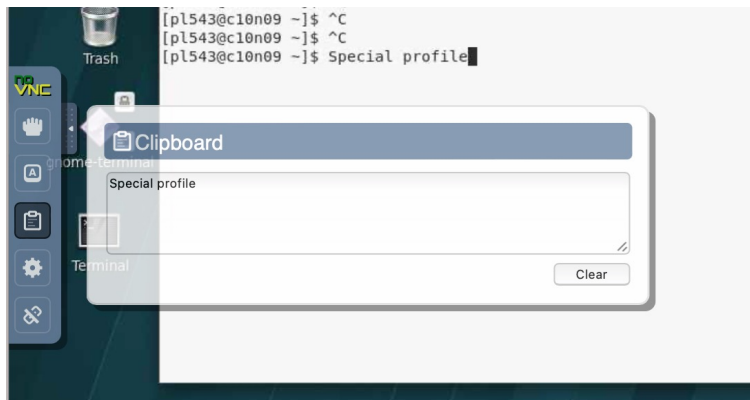
# Remote Desktop

- Remote Desktop allows you to run any GUI applications from a terminal within it



# Remote Desktop: Copy/Paste

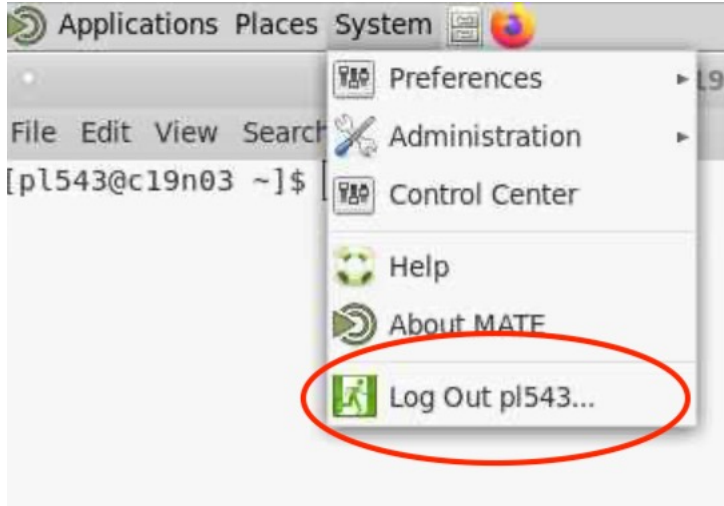
- Use the Clipboard to copy/paste between your local machine and the Remote Desktop



- Clipboard is required for Safari and Firefox, but not for [Google Chrome](#)

# Stop Remote Desktop

- First stop any applications you have launched within the remote desktop
- Then stop Remote Desktop itself by clicking 'System'-'Log Out'





# VNC-based Interactive Apps

- Besides RStudio Desktop, there are more VNC-based interactive apps




- The [Additional Modules](#) field is useful when you need to use other software packages on the cluster with the program
- After you are done, stop the OOD instance by terminating the current session of the program
  - type 'exit' or 'quit' in the command window, or select 'Exit' from the 'File' menu

# Best Practices for Interactive Apps

- For developing code and visualizing results; **Not for long running production jobs**
- For a single core application, requesting 2 CPU cores is helpful if your application takes up almost 100% of the CPU capacity
  - But no more than 2, unless your application is parallelized
- Use '**Additional Job Options**' to fine-tune the job options for interactive apps
  - EX: to request a specific GPU type `--gres=gpu:v100:2`
- Find all your running and ended sessions from the past seven days in 'My Interactive Sessions' page

 My Interactive Sessions

# Don't Forget to Stop Your Sessions !!!

- After you finish, stop the session immediately to avoid wasting the cluster resources
  - Simply **closing the tab** in the browser where you are connected to the session **WILL NOT TERMINATE** the session and **WILL NOT FREE** any CPUs, GPUs, and memory allocated to it
- Please stop a session properly as described earlier
- For the following cases, you must use  Delete
  - RStudio Server (run 'q()' first and then 'delete')
  - Code Server
  - A frozen session

# OOD Interactive App Resource Limits

- To reduce waste of resources on the cluster
- Specific to OOD interactive apps
- At most 4 simultaneous OOD interactive sessions per user
- Wall time limits are usually shorter than what is configured in the batch scheduler

Partitions	Slurm Wall Time Limit	OOD Interactive App Wall Time Limit
devel	4 hours	4 hours
public partitions (exclude week)	Varies	24 hours
week	7 days	48 hours
private	28 days	7 days

# Job Management

- Active Jobs: view running/queued jobs



- Job Composer: submit batch jobs through OOD
  - Jobs can be created from a template, an existing job, or a specified path
  - New jobs are created in a directory predetermined by OOD
  - You can re-run old jobs easily

A typical workflow



# Log Out Properly

- To log out of OOD properly, it is important to clear the cookies and close the browser
- Be sure to do so if you have accessed OOD from a public computer



# Advanced Topics

- Manage OOD Working Directories
- Data sharing and path shortcut
- Manage RStudio Server
- Launch Jupyter with Julia Kernel
- GitHub Copilot and Code Server
- Self-diagnoses

# Manage OOD Working Directories

- OOD uses your home directory to store configuration files, databases, and temporary files

`$HOME/ondemand`

- Files older than 60 days should be removed from the OOD App working directory

```
OODAPP_PATH=$HOME/ondemand/data/sys/dashboard/batch_connect/sys  
find $OODAPP_PATH -mtime +60 -print -exec rm -rf {} \;
```

- If you use OOD job composer for job submission, and it eats up a lot of your home space

```
OOD_PROJECTS=$HOME/ondemand/data/sys/myjobs/projects  
mv $OOD_PROJECTS /path/to/new/location  
ln -s /path/to/new/location $OOD_PROJECTS
```



# Data Sharing

fileshare

Shortcut usage:

```
fileshare add user1,user2,user3,... filename  
fileshare delete user1,user2,user3,... filename  
fileshare list filename
```

Full usage: (not implemented)

```
fileshare <add|delete|list|help> [-r] [-v] -u uname1[,uname2,uname3,...] -g  
gname1[,gname2,gname3,...] -p perm -f file
```

# Customize Your Path Shortcuts

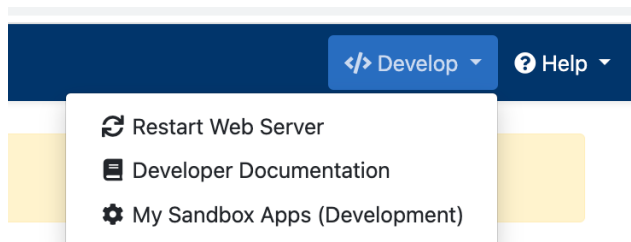
- Use the following commands to customize and view the list of directories as shown under 'Files' :

`ood_add_path:`      add a path


`ood_remove_path:` remove a user added path

`ood_list_path:`      list user added paths

- To make the changes reflect immediately in OOD, restart your web server



# Manage RStudio Server

- Start a clean R by default. The session will never be saved when you quit.
- Select “Start R from your last saved session” checkbox
  - Start R from the last saved sessions
  - Save the current session for next “non-clean” startup of an R session
- Always save and close your files before terminating any RStudio Server instance
- Remember to exit RStudio Server properly with ‘q()’ + 
- Run into a disk space issue? Symlinks may resolve the issue

```
mv $HOME/ondemand/data/sys/dashboard/batch_connect/sys/ycrc_rstudio_server /new/path/to/ycrc_rstudio_server  
ln -s /new/path/to/ycrc_rstudio_server $HOME/ondemand/data/sys/dashboard/batch_connect/sys/ycrc_rstudio_server
```

```
mv $HOME/.local/share/rstudio /new/path/to/rstudio  
ln -s /new/path/to/rstudio $HOME/.local/share/rstudio
```

# Having Issues with RStudio Server?

`ycrc_clean_rstudio.sh`

```
Usage: ycrc_clean_rstudio.sh [-n|-a|-f|-s|-h]
```

This script will cleanup cached files for RStudio Server and allow you to start it from anew.

no flag: make a copy of rstudio cached files to /home/pl543/.ycrc\_rstudio\_backup and then remove the cached files.

-n: do not make a copy of rstudio cached files and delete them directly.

-a: remove the backups in /home/pl543/.ycrc\_rstudio\_backup.

-f: remove all rstudio related files, including session files, backup files, and 00D files.

-s: list disk usage of rstudio cached files and /home/pl543/.ycrc\_rstudio\_backup.

-h: print the help message.

# GitHub Copilot and Code Server

- You will need a GitHub Copilot subscription
- Select Code Server version 4.16.1 in OOD
- Download and install the GitHub Copilot extension
- Register your Code Server session to the GitHub website and grant its access to GitHub Copilot

# Self-diagnoses

- Check disk quota
  - Use the OOD utility app “Quotas”
  - Run from the command line: `getquota`
- Check \$HOME/.bashrc file
  - Restore the original .bashrc by commenting out any changes made by you
  - Remove any Conda initialization
- Check the session output file



- What kind of training would you like to have in the future?

[pollev.com/pingluo216](https://pollev.com/pingluo216)

# Thank You