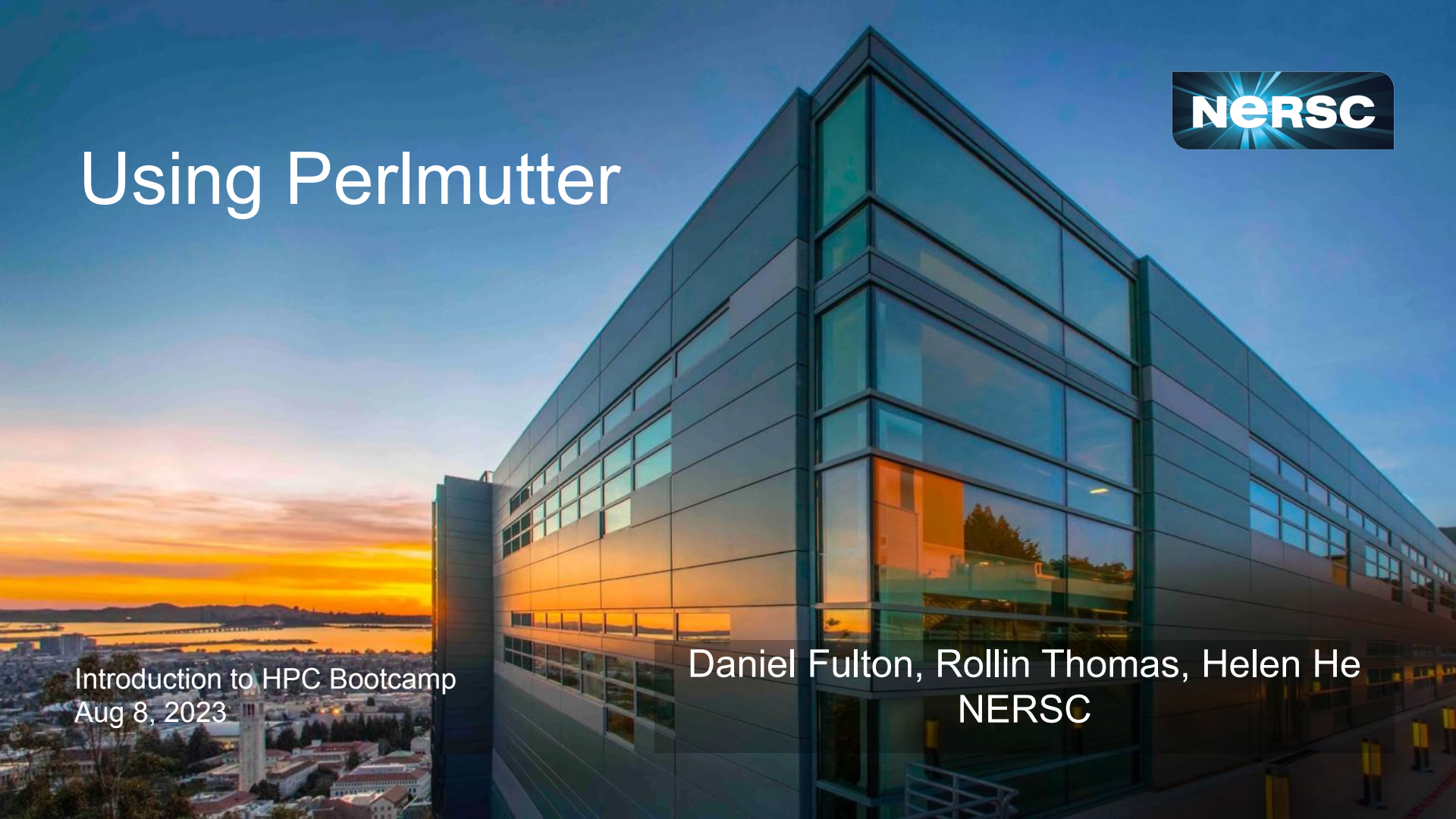


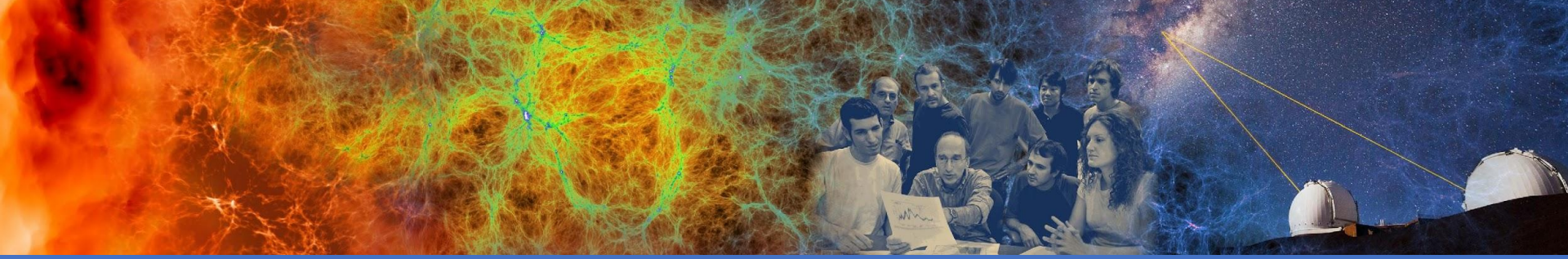
Using Perlmutter

Introduction to HPC Bootcamp
Aug 8, 2023

Daniel Fulton, Rollin Thomas, Helen He
NERSC

Outline

- **Connecting and transferring files to Perlmutter**
  - **Daniel Fulton**, HPC System Software Engineer, Computational Systems Group
- **Jupyter at NERSC**
  - **Rollin Thomas**, Senior Computing Engineer, Programming Environment and Models Group
- **File systems, compile and run jobs**
  - **Helen He**, HPC Consultant, User Engagement Group



Connecting and Transferring Files to Perlmutter

Connecting to Perlmutter

Users primarily interact with Perlmutter via text ***login shell*** (e.g. */bin/bash*)

- Terminal emulator + SSH
  - Windows: Native **Powershell** or 3rd party **putty**, **cmder**
  - Mac: Native **Terminal.app** or 3rd party **iTerm 2**
  - Linux: Any native terminal (**Gnome Terminal**, **Terminator**, **Konsole**)
- NERSC Jupyterhub web portal (https://jupyter.nersc.gov)
- NoMachine (NX) remote desktop client

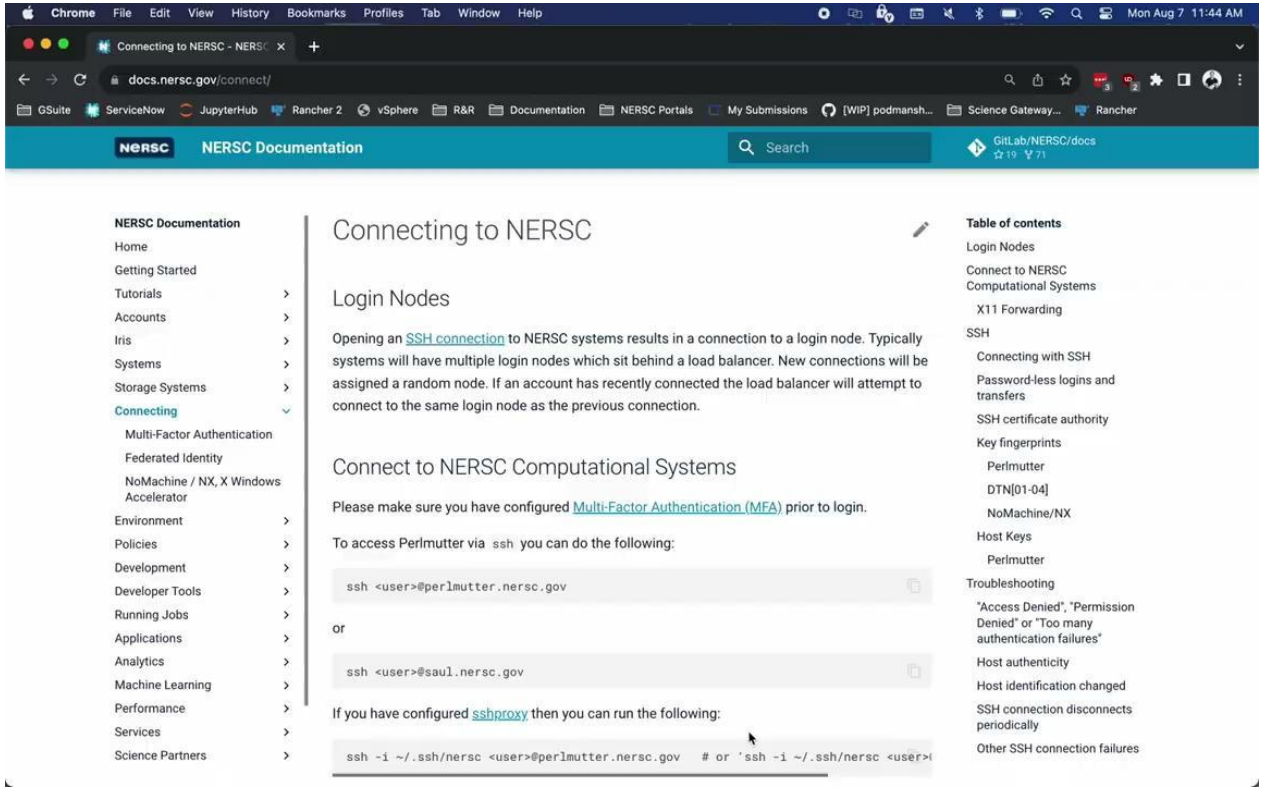
All connections to NERSC systems require an authorized NERSC user account and authentication via one of the following:

- Your NERSC user password + One Time Password
- 24-hour sshproxy key (obtained with password+OTP)

See https://docs.nersc.gov/connect/ and Slack #perlmutter_support

Connecting to Perlmutter with SSH

Connecting to Perlmutter: 24 hour keys

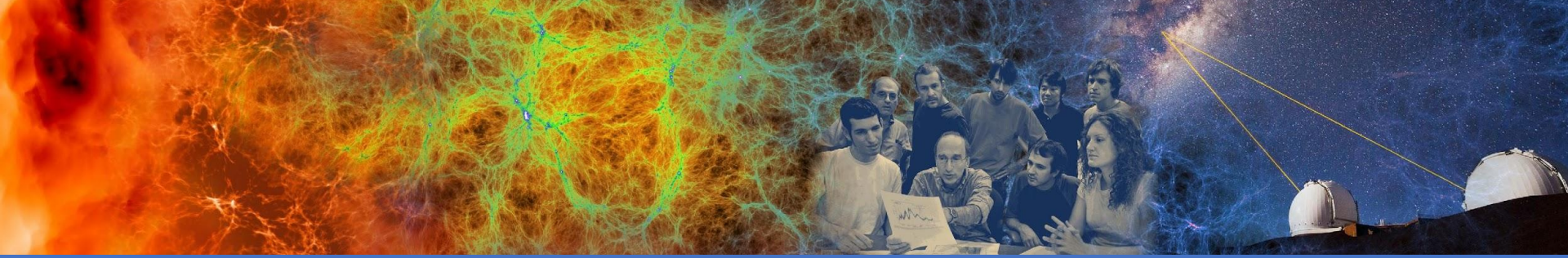


Connecting to Perlmutter: SSH Shortcuts

Transferring Data to Perlmutter

Great, now how do I get my files to onto the supercomputer?!

- For this workshop:
  - On NERSC internal filesystems (CFS): **`mv`**, **`cp`**, or **`rsync`**
    ```
    cp /path/to/original /path/to/new/copy
    ```
  - From your laptop: **`scp`**, **`rsync`**, drag and drop with Jupyter
    ```
    scp /path/on/laptop user@perlmutter.nersc.gov:/path/on/pm
    ```
  - From Github: **`git clone`**
    ```
    git clone https://www.github.com/ns/myrepo.git
    ```
- Other interesting use cases:
  - For large scientific data: Globus
  - When Globus doesn't work: **`rsync`**
  - Download from trusted URLs: **`wget`**, **`curl`**
  - Large, live, scientific data: come talk to us



Jupyter at NERSC

About Me (Rollin)

Grew up moving between Indiana, Tennessee, and Ohio

Wanted to work on computers and astronomy

Learned that physics would be a good foundation for that

Went to Purdue and got a Physics degree there

    Almost quit, but a computational physics class helped me understand physics better

Got rejected from 10+ internships but got one at University of Oklahoma working on supernovae

Went to OU, got a PhD doing supernova simulations w/Fortran using Seaborg at NERSC

    I kept avoiding writing my dissertation by optimizing my code

Got rejected from 12+ fellowships but got an interview at LBL w/Saul Perlmutter + friends

Worked on a few supernova cosmology experiments at LBL, more coding, less physics

Moved to NERSC to broaden my impact on science: New user communities, Python, Jupyter, …

About Jupyter

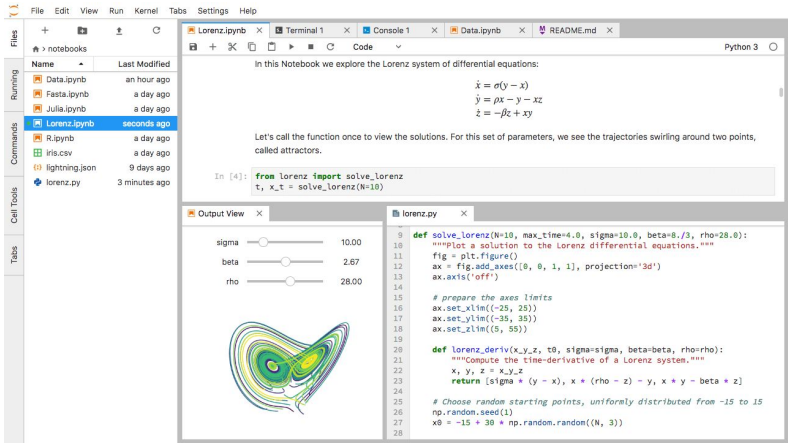
Open source, literate computing ecosystem built on “notebooks” and programming interfaces / user applications for interacting with and executing them

Notebooks (JSON documents) can contain:

- Live code
- Equations
- Visualizations
- Narrative text
- Interactive widgets

Jupyter is used for:

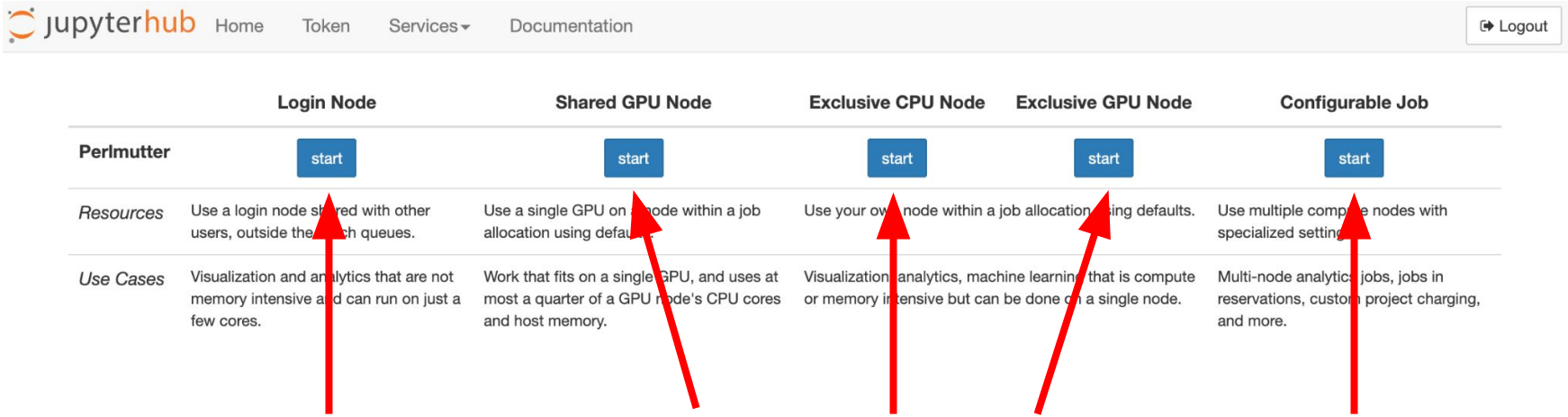
- Data cleaning and data transformation
- Numerical simulation
- Statistical modeling
- Data visualization
- Machine learning
- Workflows and analytics frameworks



Jupyter at NERSC

Hub Home Page or “Console”

All these options get you running Jupyter on Perlmutter but give you different ways to use its resources.



Use Jupyter on one of Perlmutter’s 40 login nodes

Immediate start, no charging, but more limited resources

Single-click launch of Jupyter on one of Perlmutter’s 4500+ compute nodes

Scheduled, charged, and time-limited, but with GPUs and/or a whole node just to yourself

Customized launch of Jupyter on Perlmutter compute nodes

Scheduled, charged, time-limited, but you control settings directly

About Using Jupyter This Week

We've *reserved* compute nodes for the bootcamp this week:

Day	Reservation Name	Start Time	End Time
Tuesday	`intro_hpc_aug8`	10:30 AM	5:00 PM
Wednesday	`intro_hpc_aug9`	1:30 PM	3:30 PM
Thursdsay	`intro_hpc_aug10`	1:00 PM	10:00 PM

This lets your notebook server launch faster on compute nodes.

But you have to tell Perlmutter to launch your notebook server on the reserved nodes. How do you do this…?

… use the "Configurable Job" option.

Outside of reservation hours, you can explore the other single-click options.

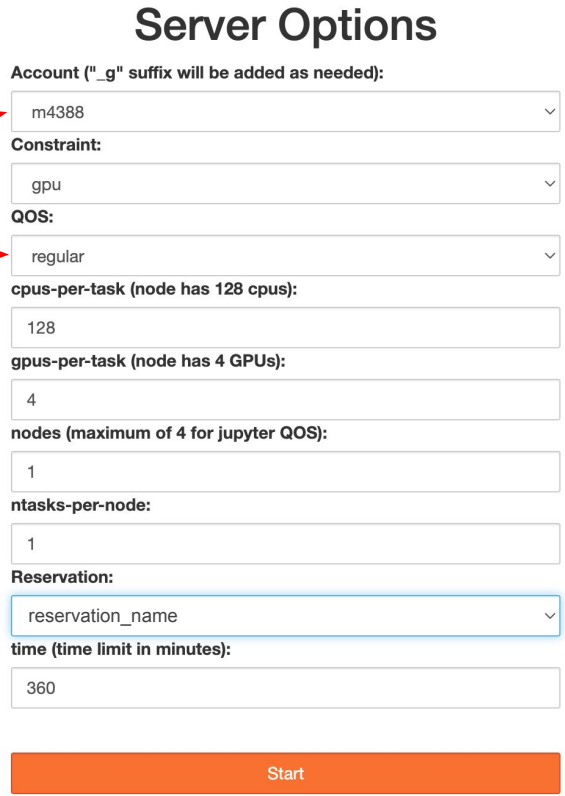
Running a Notebook Server Using One Whole Node

Select **Account**: m4388

Select **QOS**: regular

Select **Reservation**: ???

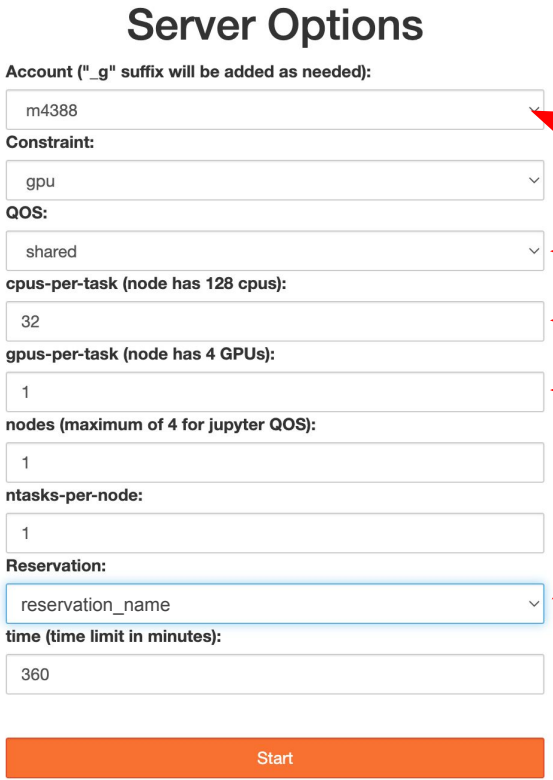
Leave everything else the same



Use this setup if you want either:

- To use all 4 GPUs on a node, and/or
- All of the CPU cores on a node

Running a Notebook Server Using a Single GPU



Select **Account**: m4388

Select **QOS**: shared

Lower **cpus-per-task** to: 32 (you can just type 32)

Lower **gpus-per-task** to: 1

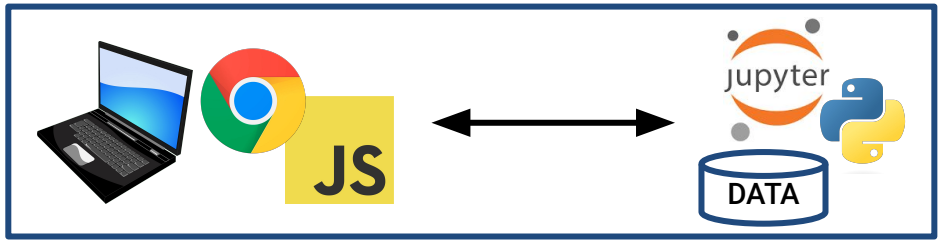
Select **Reservation**: ???

Leave everything else the same

Use this setup if you want just one GPU

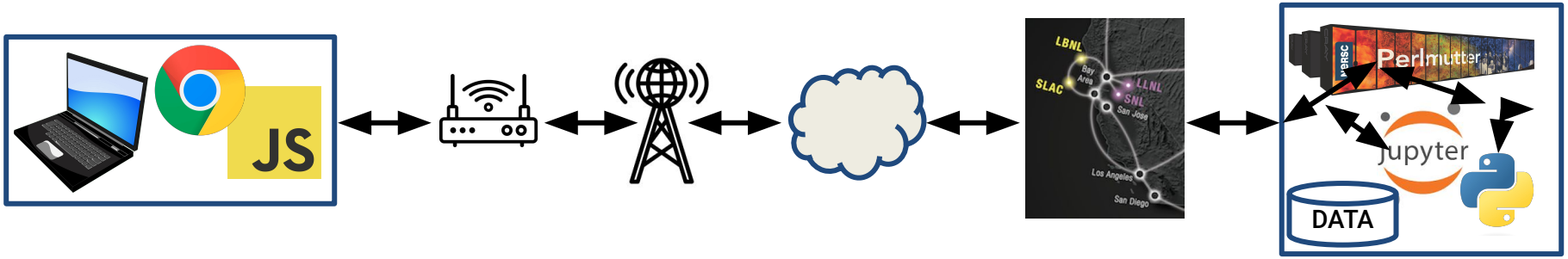
Laptop Jupyter vs HPC Jupyter (I)

A little compute
A little data
All in one place



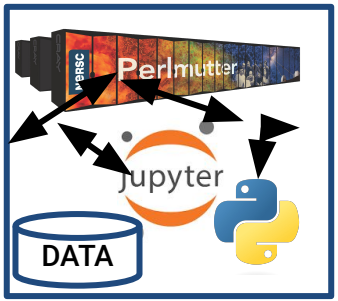
Laptop

Lots of compute
Lots of data
But way over here



Laptop Router ISP Internet ESnet NERSC

Laptop Jupyter vs HPC Jupyter (II)



NERSC

Laptop: Home directory is “right there” on your laptop
Perlmutter: Home directory is served over (an incredible) network
Many filesystems are served to Perlmutter over network

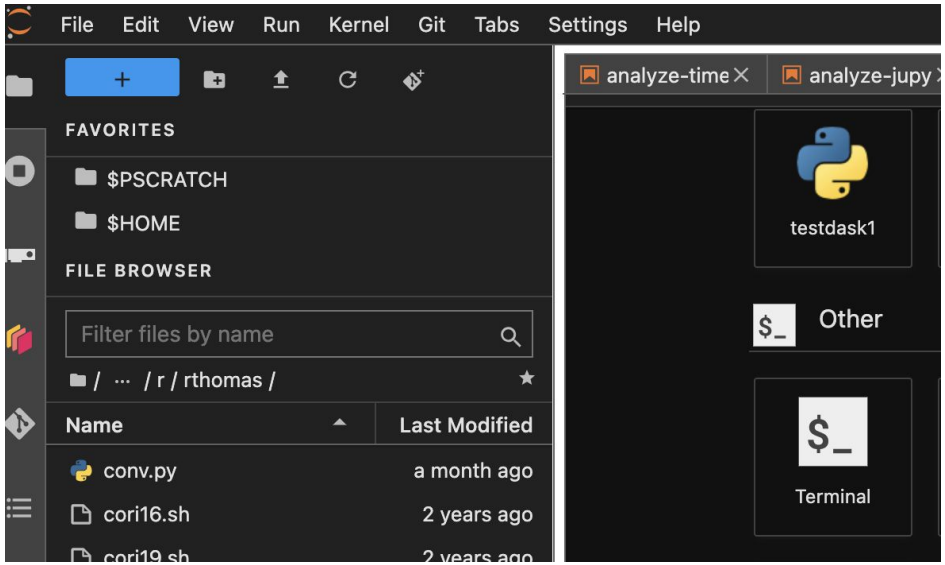
Great, but, … there’s no free lunch:

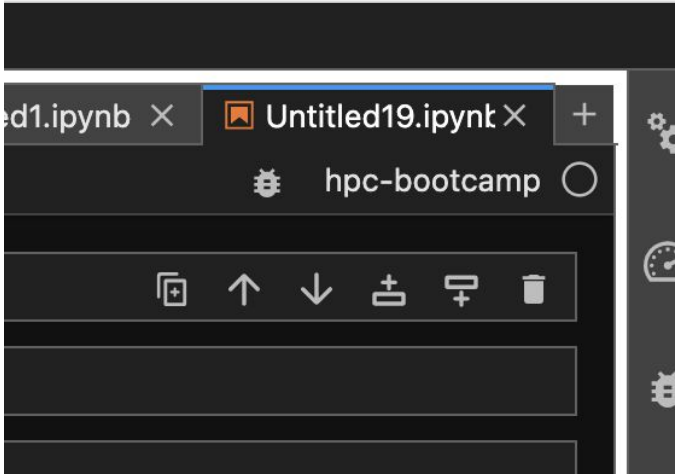
- File system has to look consistent across all those nodes
- I/O has to be coordinated from app to node to network to disk & back
- We use Cray’s “Data Virtualization Service,” and it’s being tuned

Sometimes another user’s file system usage pattern in a running job grabs ahold of DVS and won’t let go!

For Jupyter on compute nodes: Things can slow down, look sluggish, or you may get “gateway timeout” messages. Don’t panic, it’ll recover

Demo if Time Allows





To open a terminal panel:

- Click the “+” in the top left corner
- Scroll down
- Select “Terminal” from under “Other”

Select your kernel at the top right

- Most projects can use “hpc-bootcamp”
- Several others you may want to use

Things to Remember:

Remember to use the “Configurable Job” option during reservation hours:

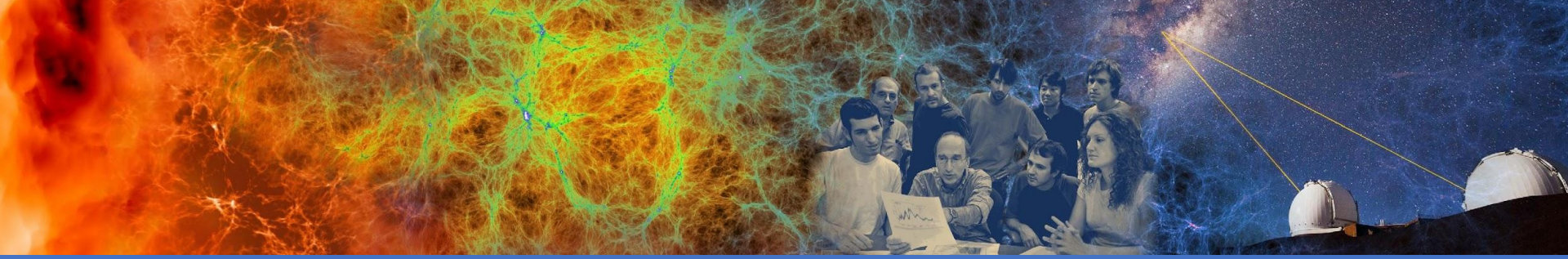
Day	Reservation Name	Start Time	End Time
Tuesday	`intro_hpc_aug8`	10:30 AM	5:00 PM
Wednesday	`intro_hpc_aug9`	1:30 PM	3:30 PM
Thursday	`intro_hpc_aug10`	1:00 PM	10:00 PM

Outside of reservation hours, you can explore the other single-click options.

NERSC Jupyter documentation: https://docs.nersc.gov/services/jupyter/

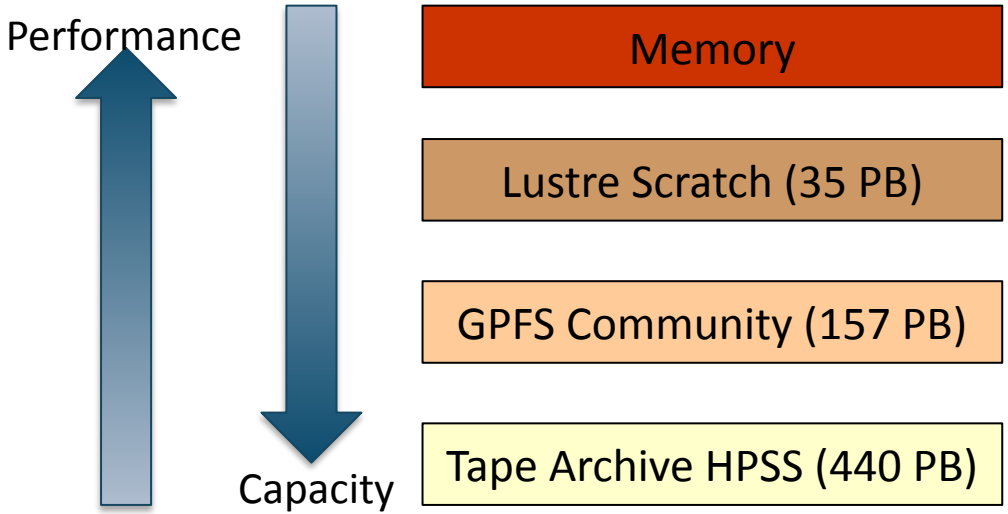
Have fun!

Any questions?



File Systems, Compile and Run Jobs

Simplified NERSC File Systems



Global Common (24 TB) — shared software

Global Home (408 TB) — individual home directories

Perlmutter File Systems

Global Home

- Permanent, relatively small storage
- NOT tuned to perform well for parallel jobs
- Snapshot backups
- **Perfect for storing data such as source codes, shell scripts**
- **cd $HOME**

Community File System (CFS)

- Permanent, larger storage
- Medium performance for parallel jobs
- Snapshot backups
- **Perfect for sharing data within research group**
- **cd $CFS**

Scratch

- Large, temporary storage
- Optimized for read/write operations, NOT storage
- Not backed up
- Purge policy (8 weeks)
- **Perfect for staging data and performing computations**
- **cd $SCRATCH**

Where should I work on my bootcamp project

- Materials for each project are available in the m4388 project area on Perlmutter CFS at **$CFS/m4388/Project*-<name>**, such as **Project2-ClimRR**
- Students who work on Project X Group Y will work in a shared directory in **$CFS/m4388/Prj*-Group***, such as **Prj2-GroupC**
- To copy over the entire project to your group working directory
  - cd $CFS/m4388/Prj2-GroupC
  - cp -r $CFS/m4388/Project2-ClimRR . (notice the last dot)
- Any student could also do individual work in their own scratch directory
  - cd $SCRATCH
  - cp -r $CFS/m4388/Project2-ClimRR . (notice the last dot)

Programming Environment and Compile

- Some users use JupyterHub to login and mostly using Python for data analytics. There is a “terminal” kernel in JupyterHub.
- Most users also directly login to Perlmutter with SSH from a terminal, and work on scientific applications written in C/C++ and Fortran
  - These codes need to be compiled first, then run the generated executable on compute nodes
- There are multiple compilers available on Perlmutter
  - The default is GCC compiler
- Compiler wrappers are used to compile, such as
  - cc -o mycode.exe mycode.c
  - CC -o mycode.exe mycode.cc
  - ftn -o mycode.exe mycode.f90

Jobs at NERSC

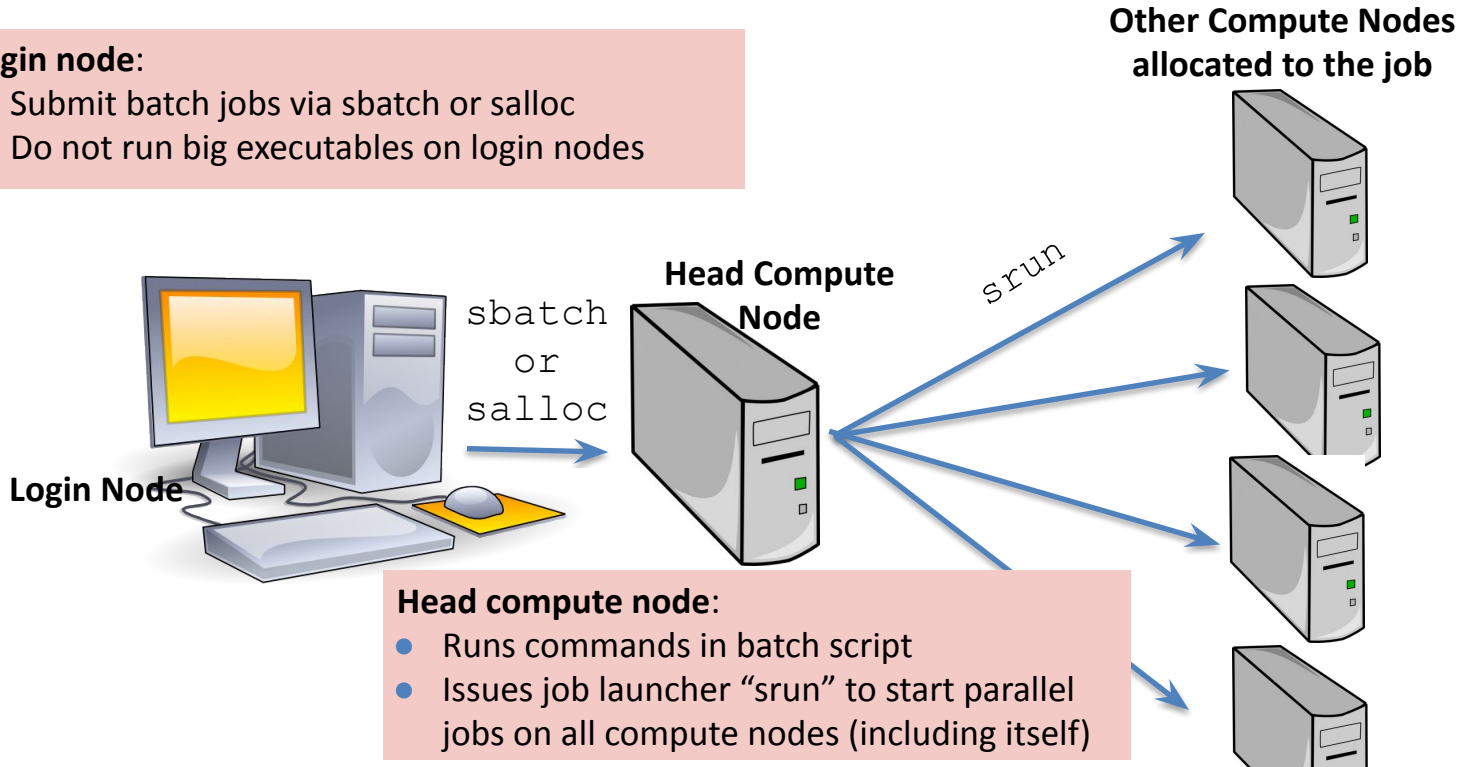
- Most are **parallel jobs** (10s to 100,000+ cores)
  - Meaning a job is run with multiple MPI tasks, each task tackle a subproblem, such as a subdomain
- Also a number of **“serial” jobs**
  - Typically “pleasantly parallel” simulation or data analysis
- Production runs execute in batch mode
- Our batch scheduler is SLURM
- Typical run times are a few to 10s of hours
  - Limits are necessary because of MTBF and the need to accommodate 9,000 users’ jobs

Login Nodes and Compute Nodes

- Login nodes
  - Edit files, compile codes, submit batch jobs, etc.
  - Run short, serial utilities and applications
- Compute nodes
  - Execute your application
  - Dedicated resources for your job
  - Perlmutter has CPU and GPU compute nodes

Launching Parallel Jobs with Slurm

Login node:
- Submit batch jobs via sbatch or salloc
- Do not run big executables on login nodes



Head compute node:
- Runs commands in batch script
- Issues job launcher "srun" to start parallel jobs on all compute nodes (including itself)

My First “Hello World” Program

```
/* C Example, mpi-hello.c */
#include <stdio.h>
#include <mpi.h>

int main (argc, argv)
     int argc;
     char *argv[];
{
  int rank, size;

  MPI_Init (&argc, &argv);    /* starts MPI */
  MPI_Comm_rank (MPI_COMM_WORLD, &rank);     /* get current process id */
  MPI_Comm_size (MPI_COMM_WORLD, &size);     /* get number of processes */
  printf( "Hello world from process %d of %d\n", rank, size);
  MPI_Finalize();
  return 0;
}
```

To compile:
% cc -o mpi-hello mpi-hello.c

Run “Hello World” Program

```
my_batch_script:
(request 2 CPU nodes for 10 min, run in debug queue)

#!/bin/bash
#SBATCH -q debug
#SBATCH -N 2
#SBATCH -t 10:00
#SBATCH -C cpu

# run with 8 MPI tasks (this is a comment)
srun -n 8 -c 64 –cpu-bind=cores ./mpi-hello
```

To run via batch queue

% sbatch my_batch_script

To run via interactive batch

login% salloc -N 2 -q interactive -C cpu -t 10:00

<Wait for session prompt. Land on a compute node>

compute% **srun -n 8** -c 64 –cpu-bind=cores **./mpi-hello**

Monitor Your Batch Jobs

- squeue
  - By default squeue displays jobs from all users
- sqs
  - sqs is a NERSC wrapper on squeue
  - By default sqs displays jobs from current user

Compile and Run Demo

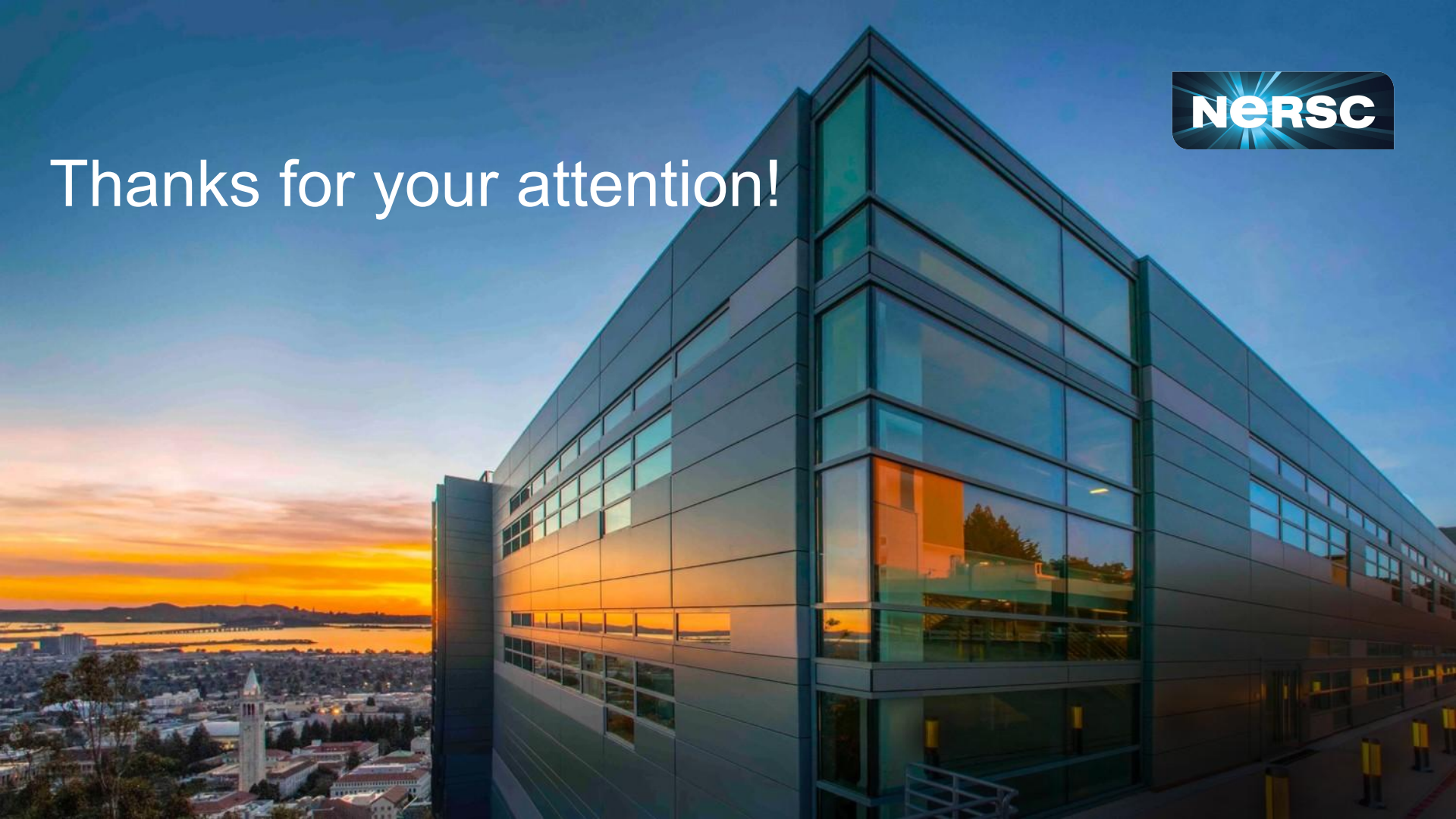
Commands Used in Compile and Run Demo

```
% pwd
% cd $SCRATCH
% cp -r $CFS/m4388/sample_compile_run .
% cd sample_compile_run
% ls
  mpi-hello.c mpi-hello.cc  mpi-hello.f90  submit_job.sh
% more mpi-hello.c
% cc -o mpi-hello mpi-hello.c
(or % CC -o mpi-hello mpi-hello.cc
 or % ftn -o mpi-hello mpi-hello.f90)
% more submit_job.sh
% sbatch submit_job.sh
% sqs
% squeue |more
% more slurm-*.out
% salloc -N 2 -C cpu -t 10:00 -q interactive
  <wait for allocation>
  % srun -n 8 -c 64 --cpu-bind=cores ./mpi-hello
```

If You Have Any Questions

- This week and next week:
  - Ask trainers, peer mentors, group members for help
  - Ask in Intro to HPC Bootcamp Slack channel

- Longer term (your NERSC account is valid through 01/16/2024):
  - Join NERSC user Slack channel
  - Submit a ticket via NERSC Help Portal
  - Check NERSC Docs: https://docs.nersc.gov/


NERSC
Thanks for your attention!