Introduction to NCAR HPC Resources

2023 CESM Tutorial

Rory Kelly

HPC Consultant, CISL

July 10, 2023

NCAR

Getting Help

- **Searchable Documentation**
- **Support Tickets for HPC Consultant Assistance**

Topics to cover

- Available systems and their uses
- Signing in and managing data
- Accessing and building software
- Managing jobs using Batch schedulers
- Customizing your user environment

This is only an introduction; for full documentation, see:
https://arc.ucar.edu/

- 4032 batch compute nodes
- 2-socket 18-core Intel Broadwell CPUs, 2.3 GHz
- 3168 nodes with 64 GB memory
- 864 nodes with 128 GB memory
- EDR Infiniband Network
- SUSE Enterprise Linux
- PBS job scheduler

- 22 nodes for data analysis and visualization jobs. Each node has 36 cores and up to 384 GB memory.
  - 9 of these nodes also feature an NVIDIA Quadro GP100 GPU.
- 18 nodes feature large-memory, dense GPU configurations to support explorations in machine learning (ML) and GPU computing
  - 4 of these nodes feature 4 NVIDIA Tesla V100 GPUs
  - 6 of these nodes feature 8 NVIDIA Tesla V100 GPUs
  - 8 of these nodes feature 4 NVIDIA Ampere A100 GPUs
- 64 high-throughput computing (HTC) nodes for small computing tasks using 1 or 2 CPUs.
  - 62 HTC nodes have 384 GB of available memory
  - 2 HTC nodes have 1.5 TB of available memory
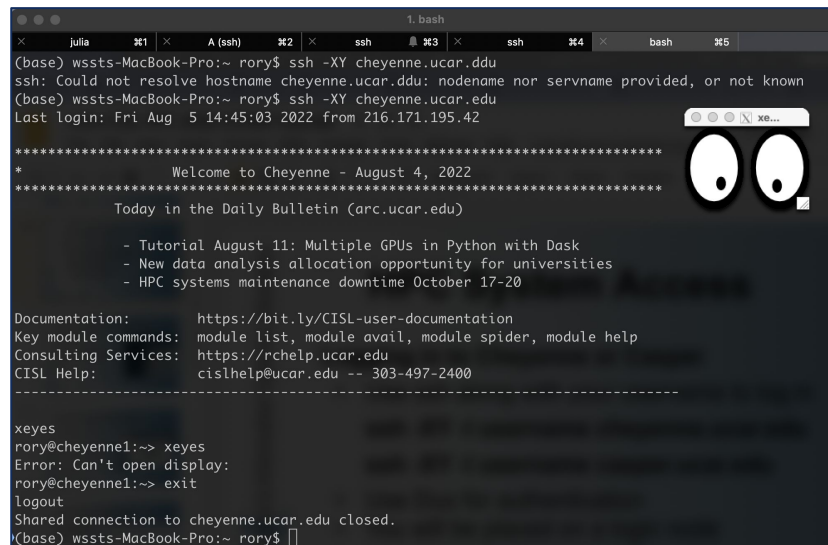- 4 nodes are reserved for Research Data Archive workflows

- HPC Cray EX, 19.87 petaflops, > 3.5x Cheyenne Performance
- Accepted and in production at NWSC, GA in August
- 2488 CPU Nodes
  - Dual socket, 64-core AMD 7763 "Milan" CPUs
  - 256 GB DDR4 Memory
  - 1 Cassini Slingshot-11 NIC
- 82 GPU Nodes
  - Single socket, 64-core AMD 7763 "Milan" CPUs
  - 512 GB DDR4 Memory
  - 4 NVIDIA A100 GPUs
  - NVLink GPU interconnect
  - 4 Cassini Slingshot-11 NICs

- Use ssh along with your username to log in

  ssh -XY -l username cheyenne.ucar.edu

  ssh -XY -l username casper.ucar.edu

- Use Duo for authentication
- Cheyenne - 6 login nodes
- Casper - 2 login nodes

Cheyenne and Casper are also accessible via the JupyterHub service at **jupyterhub.hpc.ucar.edu**
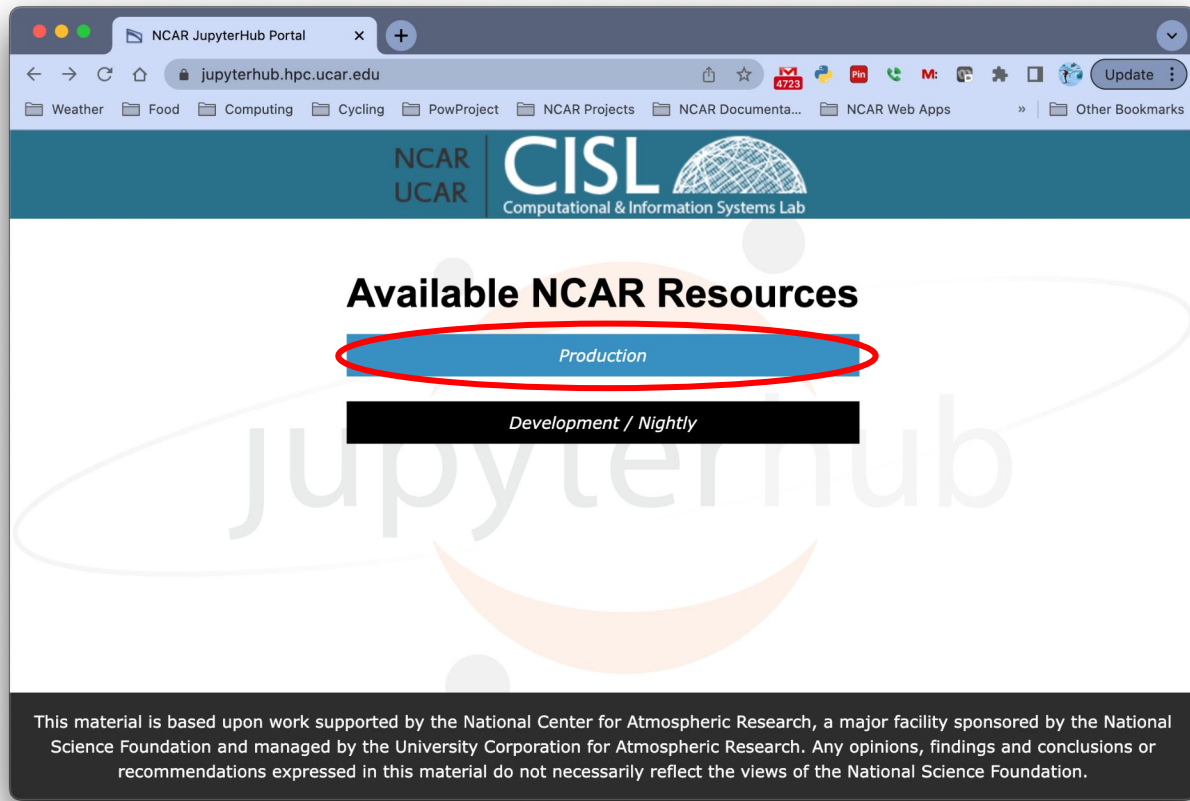
Log in with your standard Duo credentials and respond to the push notification

JupyterHub - Logging In

Start a server if you do not already have one

JupyterHub - Logging In

Select the **Casper PBS Batch** Resource, and fill in batch request details, including number of cores and project account

For this tutorial we'll be running in a reservation
R7873845

JupyterHub - Logging In

Many Notebooks and applications can be run from the main jupyterhub launcher screen.
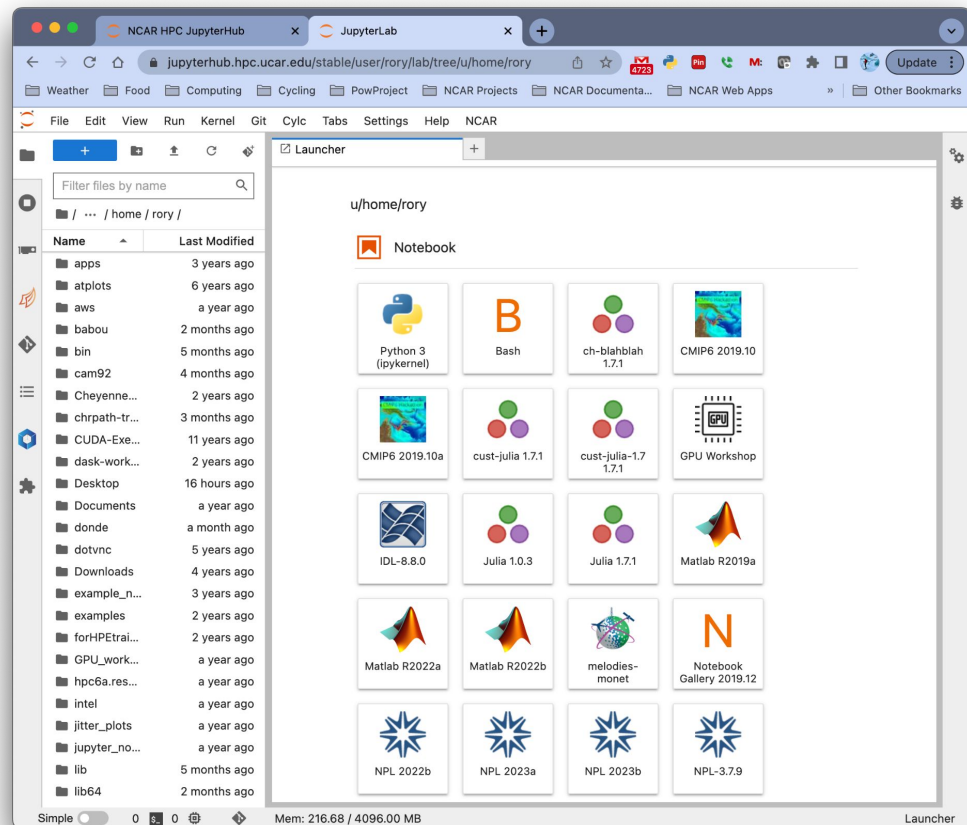
Run GUI Programs with TigerVNC

VNC can be used to run a remote GNOME/KDE desktop

Need to install a VNC client first - We recommend TigerVNC, but other VNC clients such as TurboVNC will also work

Usage:

`vncmgr create -A <project code>`

`vncmgr     (interactive)`
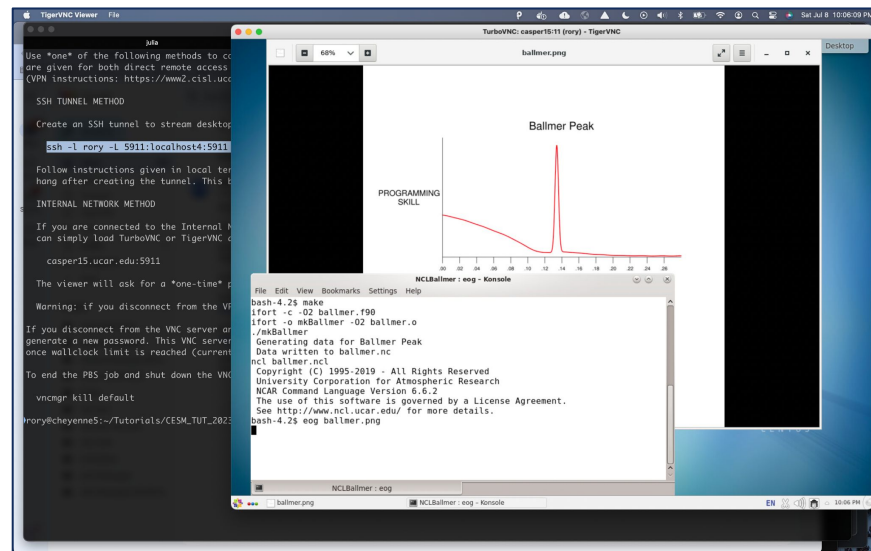
Run GUI Programs with FastX

FastX can also be used to run a remote desktop or terminal session

Can be run in a browser without a client

Connect to the NCAR VPN, then go to

https://fastx.ucar.edu:3300

Can also be setup using an SSH tunnel or a desktop client
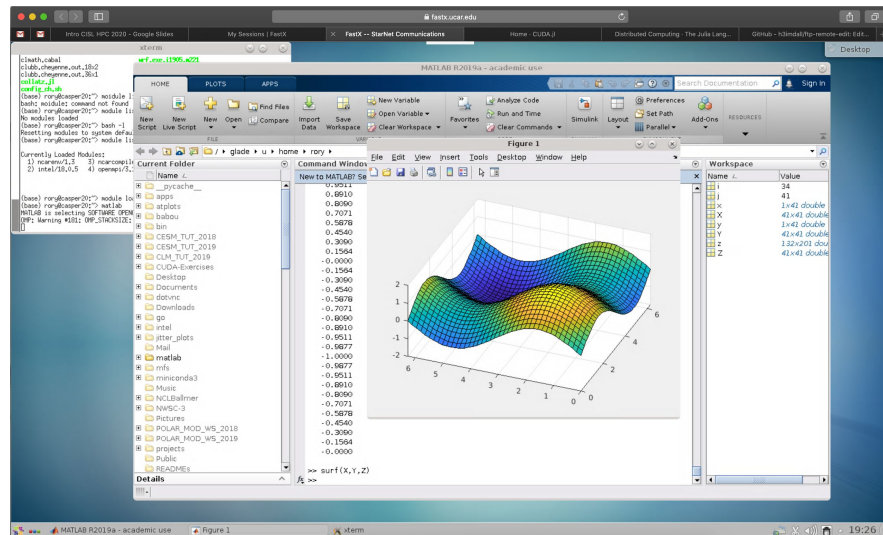
View images with jupyterhub

Jupyterhub is used to create sophisticated interactive computational notebooks for analysis, education, etc.

It can also be used for really unsophisticated things like viewing images
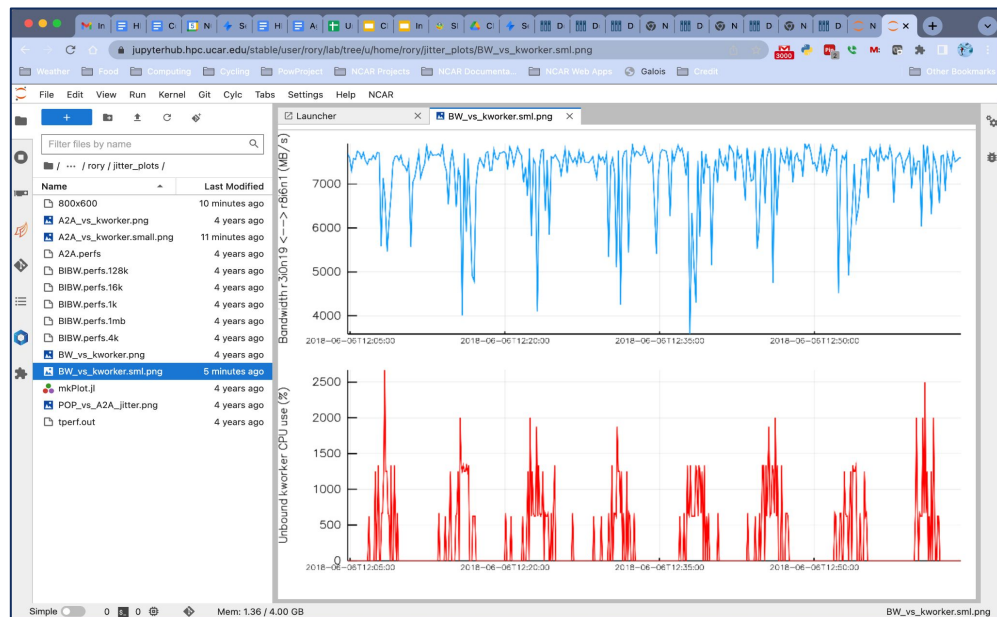
Be a good neighbor on shared resources

- Your activities coexists with those of other users
- CPUs and memory are shared on the login nodes
- Limit your usage to:
  - Reading and editing text/code
  - Compiling small programs
  - Performing data transfers
  - Interacting with the job scheduler
- Programs that use excessive resources on the login nodes will be terminated

Data storage - GLADE

File spaces accessible from all HPC systems

File space	Quota	Backup	Uses
Home **/glade/u/home/$USER**	50 GB	Yes	Settings, code, scripts
Work **/glade/work/$USER**	1 TB	No	Compiled codes, models
Scratch **/glade/scratch/$USER**	10 TB	**Purged!**	Run directories, temp output
Project **/glade/p/entity/project_code**	N/A	No	Project space allocations

*Keep track of usage with "**gladequota**"*

- Resource for storing data on project allocation time scales (3-5 years)
- Data expected to be migrated after 5 years.
- Multiple access methods
  - Globus (NCAR Campaign Storage)
  - Cheyenne / Casper nodes (/glade/campaign/)
  - Data access nodes (/glade/campaign/)
- Project space allocations (via allocation request)

Data storage - Collections

- Curated data collections available on Cheyenne and Casper to facilitate easy access to research data sets
- RDA
    - Research Data Archive
    - /glade/collections/rda/
    - https://www2.cisl.ucar.edu/data-portals/research-data-archive
- CMIP6
    - Coupled Model Intercomparison Project
    - /glade/collections/cmip/CMIP6/
    - https://www2.cisl.ucar.edu/resources/cmip-analysis-platform

- For short transfers use **scp/sftp** to transfer files
- For large transfers use **Globus**
  - To use Globus, create a Globus ID if you need an account, and search for **NCAR GLADE** or **NCAR Campaign Storage** endpoints
  - CISL endpoints currently can be activated for up to 30-days
  - Globus has a web interface and a command-line interface
  - **Globus Connect Personal** can manage transfers from your local workstation as well

Environment Modules

- CISL installed software is provided as modules
- Modules provide access to runnable applications (compilers, debuggers, ...) as well as libraries (NetCDF, MPI, ...)
- Modules prevent loading incompatible software into your environment
- **Note that Cheyenne and Casper each have independent collections of modules!**

- **module load/unload <software>**
- **module avail** - show all currently-loadable modules
- **module list** - show loaded modules
- **module purge** - remove all loaded modules
- **module save/restore <name>** create/load a saved set of software
- **module spider <software>** search for a particular module

Compiler
- Intel 17.0.1
- Intel 18.0.1
- GNU 6.3.0

Intel 17.0.1
- MKL
- netCDF

MPI
- SGI MPT 2.19
- Intel MPI 2017.1
- OpenMPI 3.0.1

Intel 17.0.1 MPT 2.19
- pnetCDF
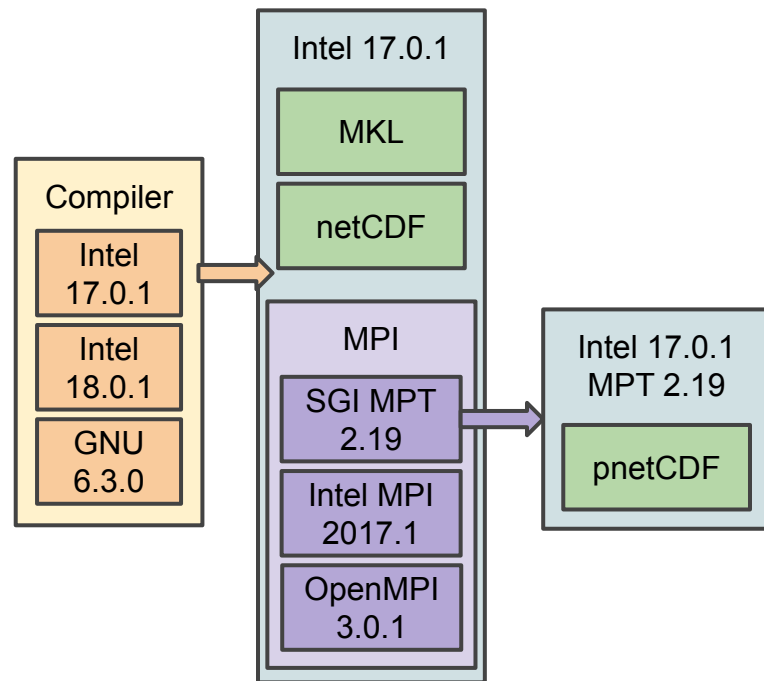
- Compilers (Intel, GNU, NVHPC)
- Debuggers / Performance Tools (Linar Forge, DDT, MAP)
- MPI Libraries (MPT, Intel MPI, OpenMPI)
- IO Libraries (NetCDF, PNetCDF, HDF5, Parallel IO)
- Analysis Languages (Python, Julia, R, IDL, Matlab)
- Convenience Tools (ncarcompilers, parallel, rclone)
- Many more ...

Considerations when compiling software

- Use **ncarcompilers** module along with library modules (e.g., netcdf) to simplify compiling and linking (*it adds include and link flags for you*)
- When using MPI, make sure you run with the same library with which you compiled your code
- **We strongly recommend you build code on the machine on which you will run**
  - Cheyenne and Casper have different CPUs and operating systems

Run large tasks on compute nodes using batch jobs

- Most tasks require too many resources to run on a login node
- Schedule these tasks to run on the Cheyenne compute nodes using the **PBS** batch system

ssh cheyenne.ucar.edu

Workstation → 6 login nodes → **4032** batch/compute nodes

2 DAV login nodes

108 data analysis and visualization, nodes

ssh casper.ucar.edu
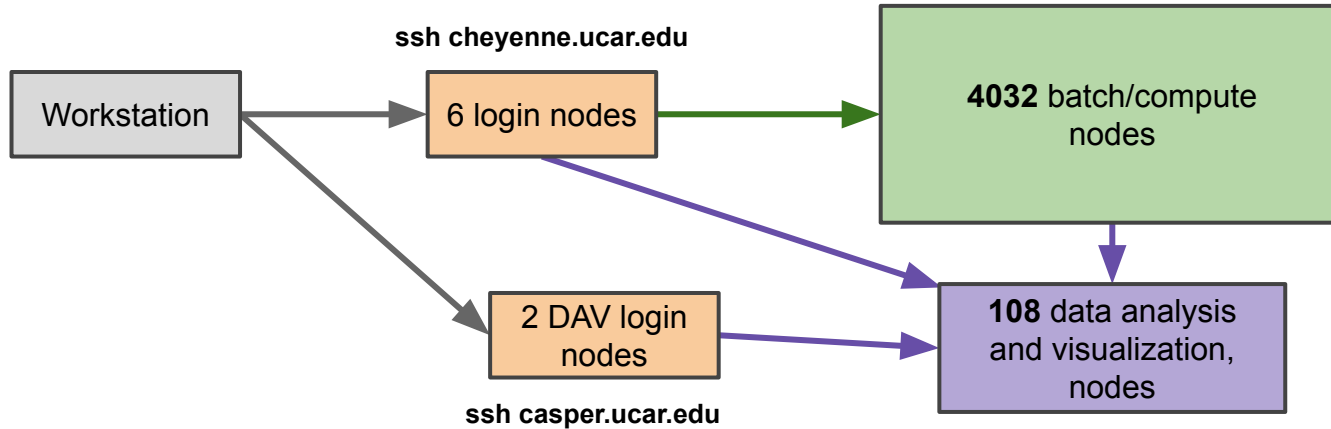
Cheyenne and Casper use separate allocations!

Run large tasks on compute nodes using batch jobs

- Most tasks require too many resources to run on a login node
- Schedule these tasks to run on the Cheyenne compute nodes using the **PBS** batch system
- Jobs request a given number of compute tasks for an estimated wall-time on specified hardware
- Jobs use core-hours, which are charged against your selected project/account
  - Remaining resources are viewable in SAM
- Temporary files are often written by programs - set TMPDIR variable to scratch space to avoid job failures

Example PBS job scripts

```
$ cat basic_mpi.pbs
#!/bin/tcsh
#PBS -N hello_pbs
#PBS -A <project_code>
#PBS -j oe
#PBS -o pbsjob.log
#PBS -q regular
#PBS -l walltime=00:05:00
#PBS -l select=2:ncpus=36:mpiprocs=36

### Set temp to scratch
setenv TMPDIR /glade/scratch/${USER}/temp
mkdir -p $TMPDIR

module load mpt/2.25

### Run MPT MPI Program
mpiexec_mpt ./hello_world
```

```
$ cat array_job.pbs
#!/bin/bash -l
#PBS -N job_array
#PBS -A project_code
#PBS -l select=1:ncpus=1:mem=4GB
#PBS -l walltime=00:10:00
#PBS -q casper
#PBS -J 2010-2020
#PBS -j oe

### Set temp to scratch
export TMPDIR=/glade/scratch/$USER/temp
mkdir -p $TMPDIR

module load mpt/2.25

### Run Array jobs program
./executable_name data.year-$PBS_ARRAY_INDEX
```

qsub <script> - submit batch job

qstat <jobid> - query job status

qdel <jobid> - delete/kill a job

qinteractive -A <project>

  Run an interactive job

qcmd -A <project> -- cmd.exe

  Run cmd.exe on a single compute node

Using OpenMP on Cheyenne with MPT

Only OpenMP

```
#!/bin/tcsh
#PBS -l select=1:ncpus=36:ompthreads=36

# Run program with 10 threads
./executable_name
```

Hybrid MPI/OpenMP

```
#!/bin/tcsh
#PBS -l select=2:ncpus=36:mpiprocs=12:ompthreads=3

module load mpt/2.19

# Run program with 12 MPI ranks per node
# and 3 threads / MPI rank
mpiexec_mpt omplace ./executable_name
```

For Running serial code on multiple data files

```
./cmd1.exe < input1 > output1
./cmd2.exe < input2 > output2
./cmd3.exe < input3 > output3
./cmd4.exe < input4 > output4
```

cmdfile contents

```
#!/bin/tcsh
#PBS -l select=1:ncpus=4:mpiprocs=4

module load mpt/2.19

# This setting is required to use command files
setenv MPI_SHEPHERD true

mpiexec_mpt launch_cf.sh cmdfile
```

PBS Job script

Optimal if commands have similar runtimes

Placing Casper jobs on specific resources

```
cat gpu_job.pbs
#!/bin/bash -l
#PBS -N mpi_job
#PBS -A project_code
#PBS -l select=1:ncpus=4:mpiprocs=4:ngpus=4:mem=40GB
#PBS -l gpu_type=v100
#PBS -l walltime=01:00:00
#PBS -q casper
#PBS -j oe

export TMPDIR=/glade/scratch/$USER/temp
mkdir -p $TMPDIR

### Provide CUDA runtime libraries
module load cuda

### Run program
mpirun ./gpu_code.exe
```

- This job can only run on a node with 40 GB of free memory and 4 V100 GPUs
- If multiple resources are specified, they must be compatible, otherwise, the job will be stuck in a pending state

PBS job queues on Cheyenne

PBS Queue	Priority	Wall clock	Details
premium	1	12 h	Jobs are charged at 150% of regular rate
regular	2	12 h	Most production compute jobs go here
economy	3	12 h	Jobs are charged at 70% of regular rate
share	N/A	6 h	Memory is shared among all users on a node Jobs are limited to 18 cores or less

Job charges depend on the queue:

Exclusive: wall-clock hours ✖ nodes ✖ 36 cores/node ✖ queue factor

Shared: core-seconds / 3600 (DAV jobs are shared as well)

Shell startup files - customizing your environment

tcsh/csh

```
$ cat ~/.tcshrc
alias rm "rm -i"

# Add programs built for each cluster
if ( $HOSTNAME =~ cheyenne* ) then
    setenv PATH ~/local/ch/bin:$PATH
else
    setenv PATH ~/local/dav/bin:$PATH
endif

# Settings for interactive shells
if ( $?prompt ) then
    set prompt = "%n@%m:%~"
endif
```

bash

```
$ cat ~/.profile
alias rm="rm -i"

# Add programs built for each cluster
if [[ $HOSTNAME == cheyenne* ]]; then
    export PATH=~/local/ch/bin:$PATH
else
    export PATH=~/local/dav/bin:$PATH
fi

# Settings for interactive shells
if [[ $- == *i* ]]; then
    PS1="\u@\h:\w> "
fi
```

Changing your default modules

- If you commonly load certain modules, you may wish to have them load automatically when logging onto a cluster
- The right way to do so is with saved module sets:

        module load ncl python nco mkl
        module save default

- Make multiple sets and load them using **module restore <set>**
- **Don't put module load commands in your shell startup files!**

SAM (Systems Accounting Manager)

- Web access: https://sam.ucar.edu
- Log in with Duo authentication
- Can change some user settings (default shell, etc)
- Get information about available projects and remaining allocation balance
- See history of jobs and charges

User Preferences: Edit Shell

USERNAME vanderwb
RESOURCE Cheyenne

Listed below are the shells that are available for this resource.

Shell
bash
ksh
nologin
tcsh

Cancel   Save

Getting Help

- **Searchable Documentation**
- **Support Tickets for HPC Consultant Assistance**