Intro to Computing at CARC with Crystallography Example

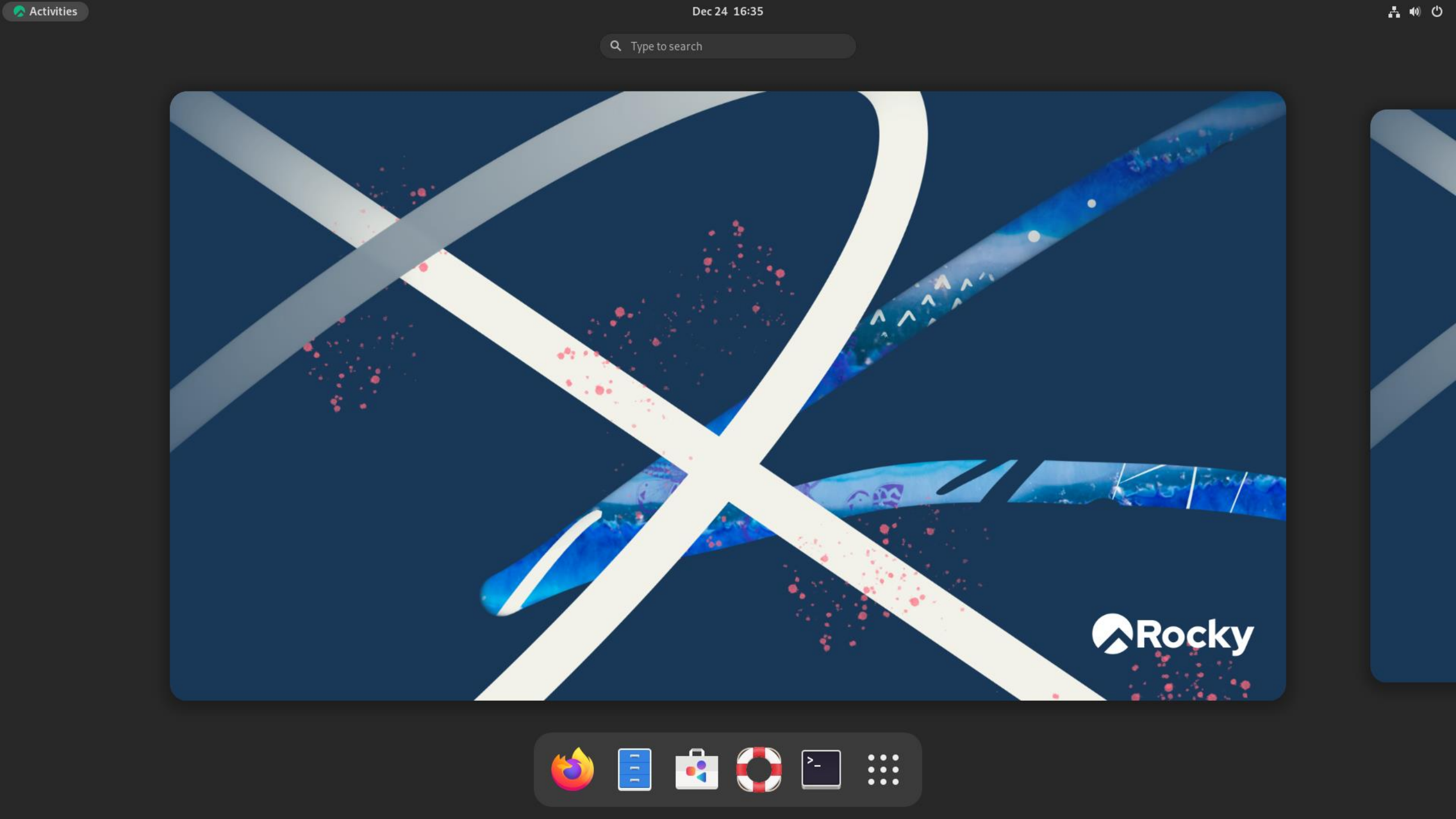
Matthew Fricke, Nick Porter,
and Sumaya Houssini Mohamed

Version 0.1

Topics

- 1) Linux BASH Shell
- 2) The purpose of High-Performance Computing (HPC)
- 3) Running Programs on an HPC system (Slurm)
- 4) Analyse Thaumatin

NOTE: All these topics are covered in our online video tutorials


Activities
Dec 24 16:35
Type to search
Rocky

Logging into Hopper


```
ssh vanilla@hopper.alliance.unm.edu
```

You may be prompted for a password…

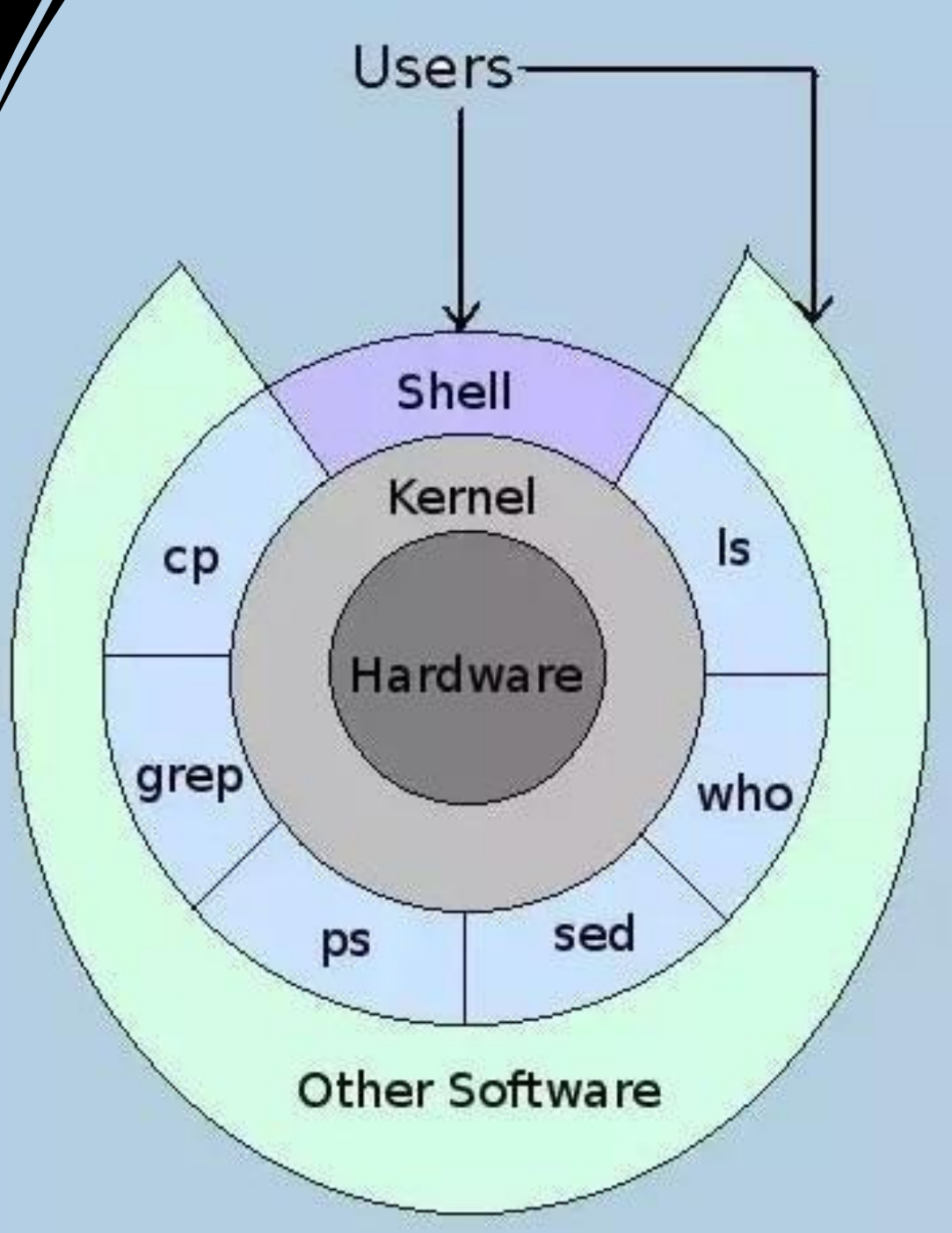
Don’t let me move on until you are able to login.

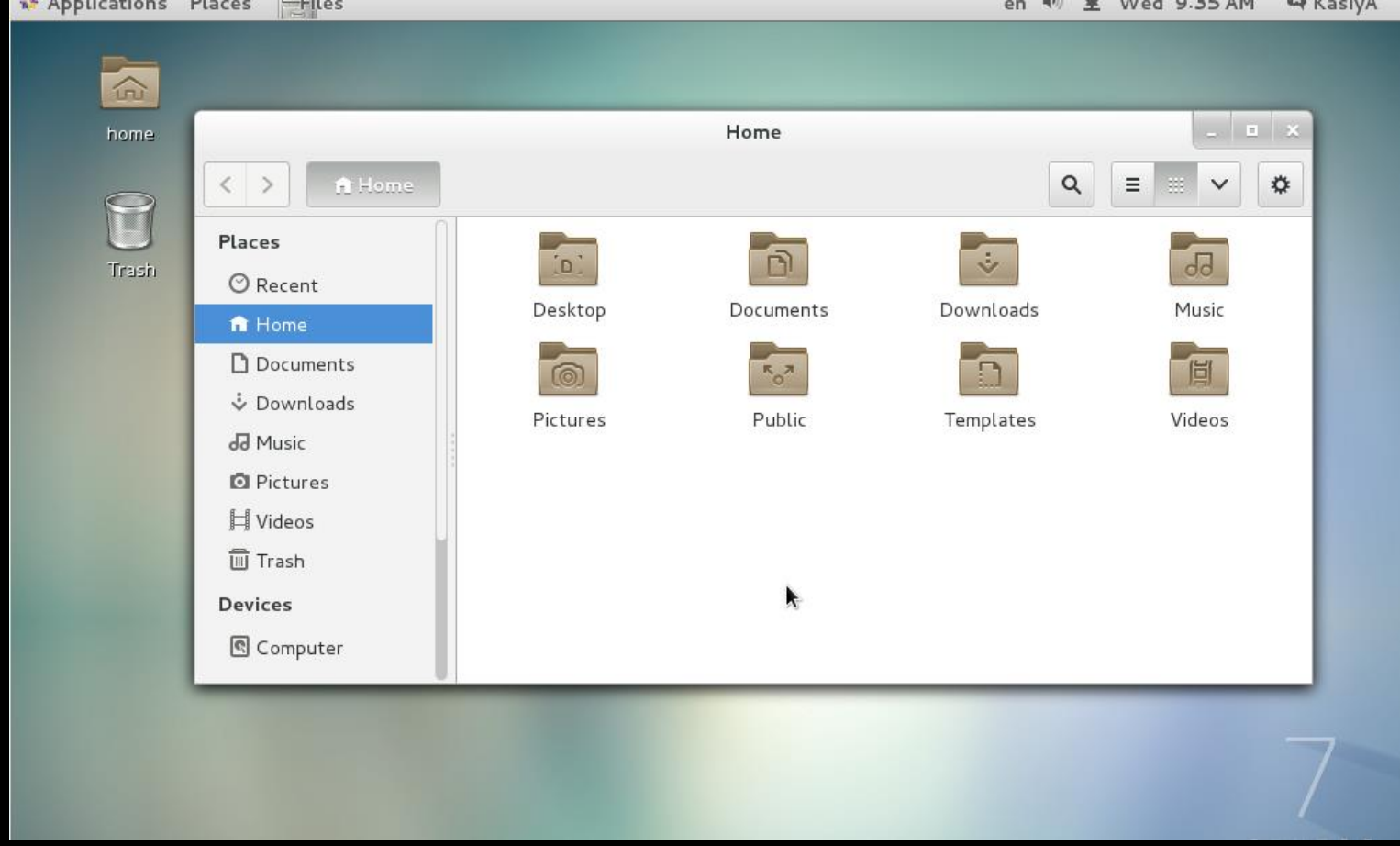
Logging into Hopper

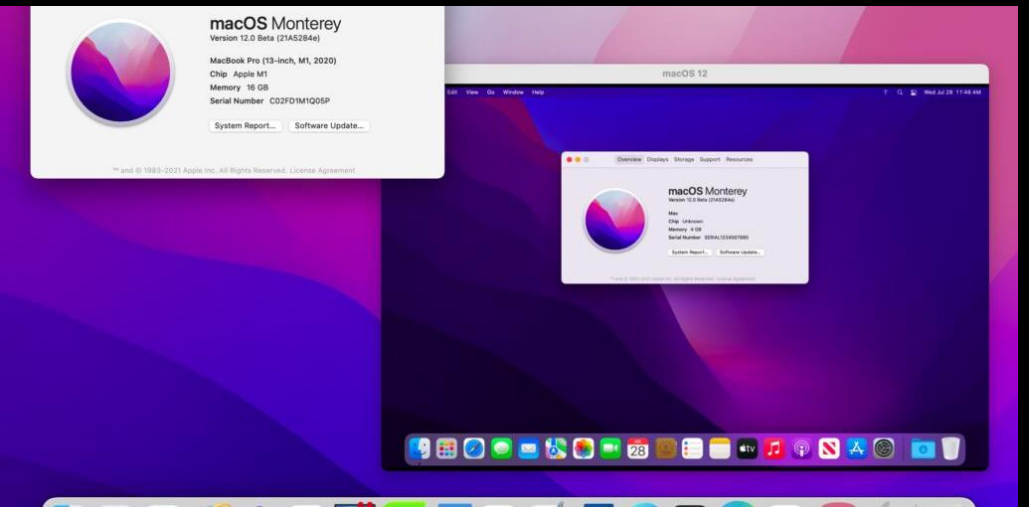
```
--------------------------------------------------------------------------------
  Welcome to Hopper

  Be sure to review the "Acceptable Use" guidelines posted on the CARC website.

  For assistance using this system email help@carc.unm.edu.

  Tutorial videos can be accessed through the CARC website: Go to
    http://carc.unm.edu, select the "New Users" menu and then click
      "Introduction to Computing at CARC".

  Warning: By default home directories are world readable. Use the chmod command
    to restrict access.

  Don't forget to acknowledge CARC in publications, dissertations, theses and
    presentations that use CARC computational resources:

  "We would like to thank the UNM Center for Advanced Research Computing,
  supported in part by the National Science Foundation, for providing the
  research computing resources used in this work."

  Please send citations to publications@carc.unm.edu.

  Hopper is our newest general purpose cluster and member of the Taos/Hopper condo.

  There are four types of slurm partitions on Hopper:
  1) General - this partition is accessible by all CARC users.
  2) Debug - for testing your code and interactive jobs. Short time limits so that nodes are usually available right away.
  3) Condo - this partition is accessible by users who are members of the Hopper/Taos condo. Users with access to this partition also have access to the
community partition on the Taos cluster. Jobs run here may be interrupted by the hardware owner, so we recommend that your software support
checkpointing so it can recover.
  4) Private partitions - these partitions are dedicated to the condo grant/lab/center that purchased the associated hardware.

  Type "qgrok" to get information about the partitions to which you have access.
  Enter "quotas" to see your storage usage and limits.
  For a list of software installed on Hopper enter "module spider".
--------------------------------------------------------------------------------
  ******************************************************************************
  * Logging in under another person's account is strictly forbidden and will *
  * result in the account being          locked.                          *
  ******************************************************************************
```

Linux and the BASH Shell

- The Kernel manages access to the hardware in a computer.
- An Operating System (OS) is the Kernel plus useful programs provided by the OS.
- The “shell” is the outermost layer of the OS.
- It is where the user interacts with the OS.

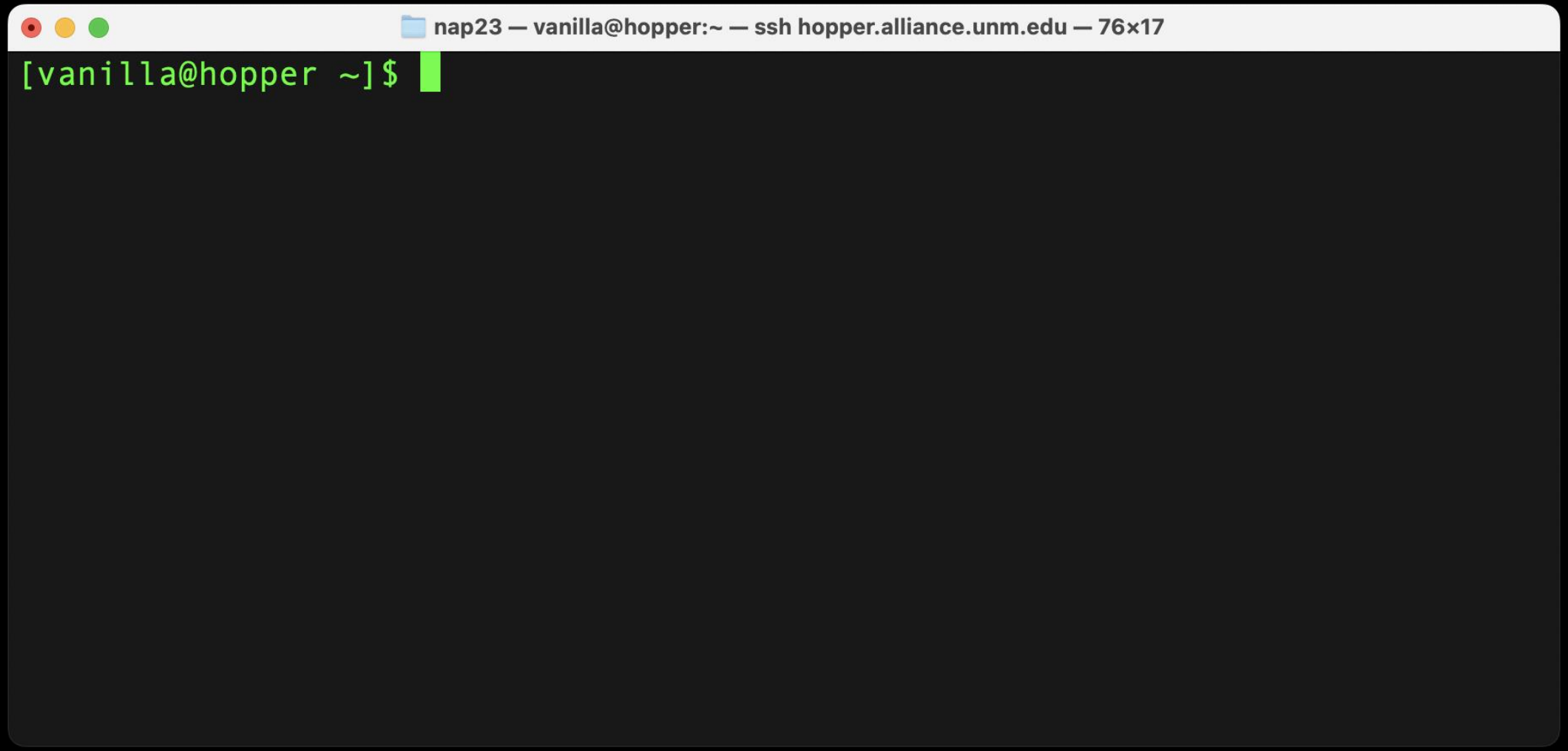






Graphical Shells (GUIs)

Logging into Hopper

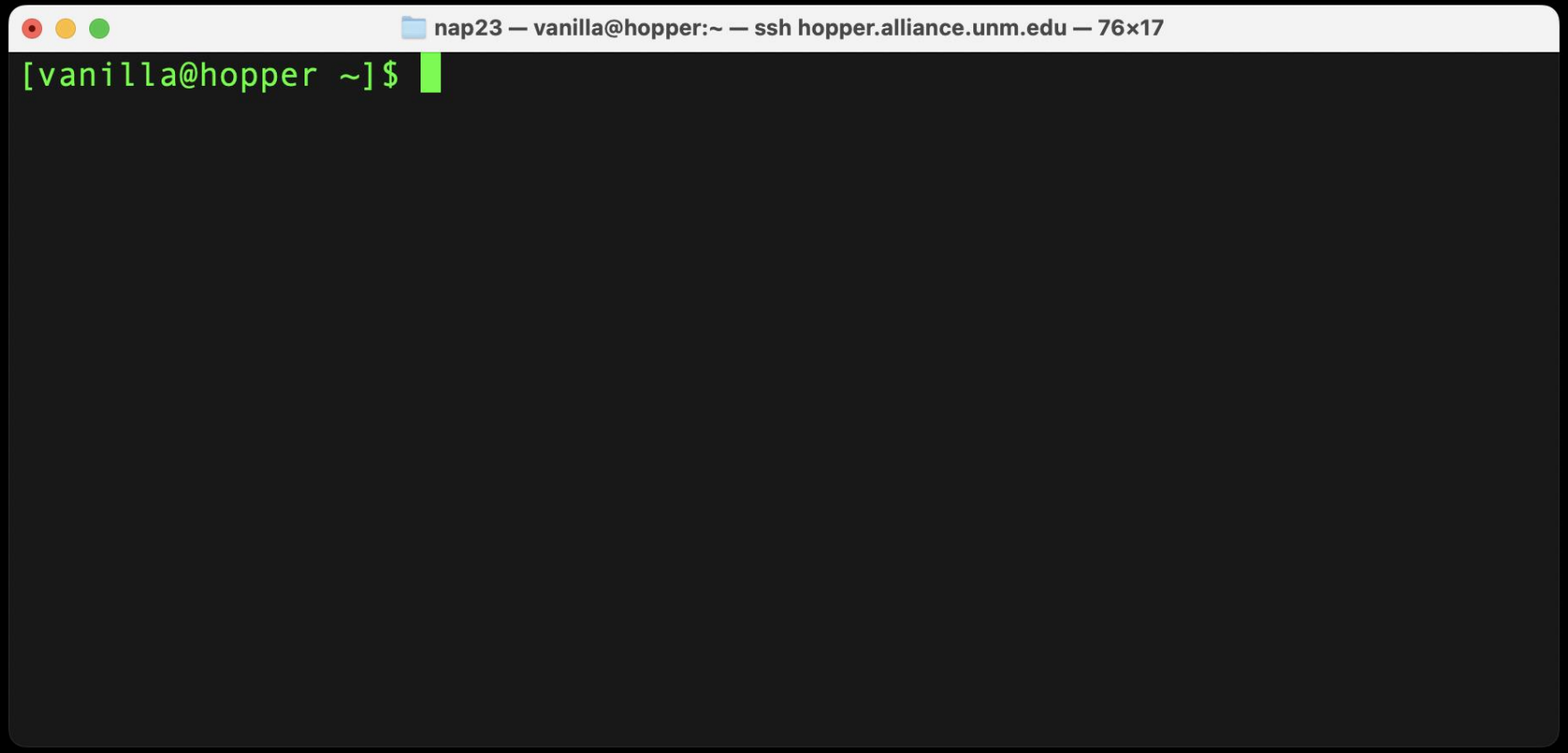


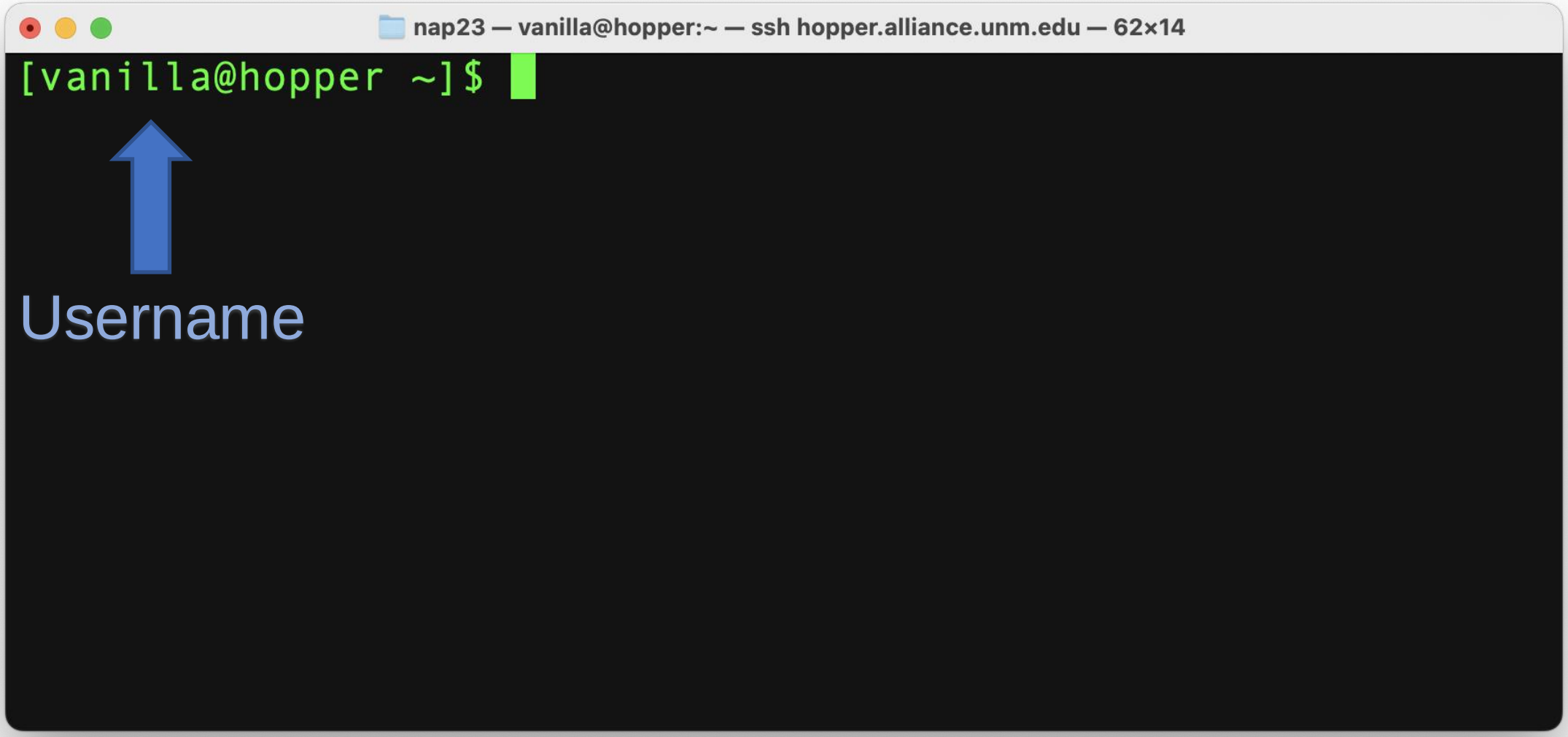

YOU ARE AT THE
RIGHT PLACE

Linux and the BASH Shell

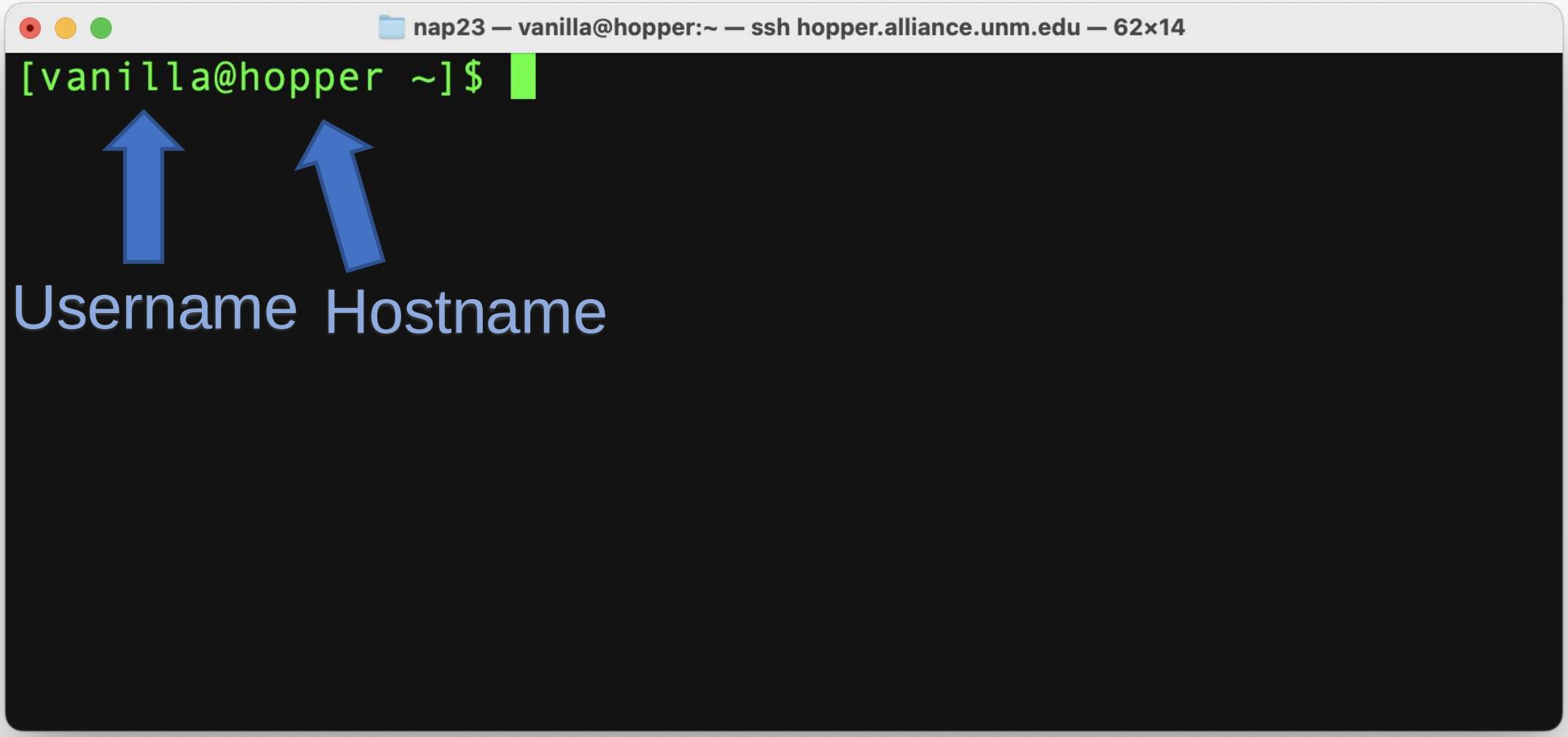
The Borne-Again Shell (BASH)

Written in 1976 by Stephen Bourne for UNIX version 7.

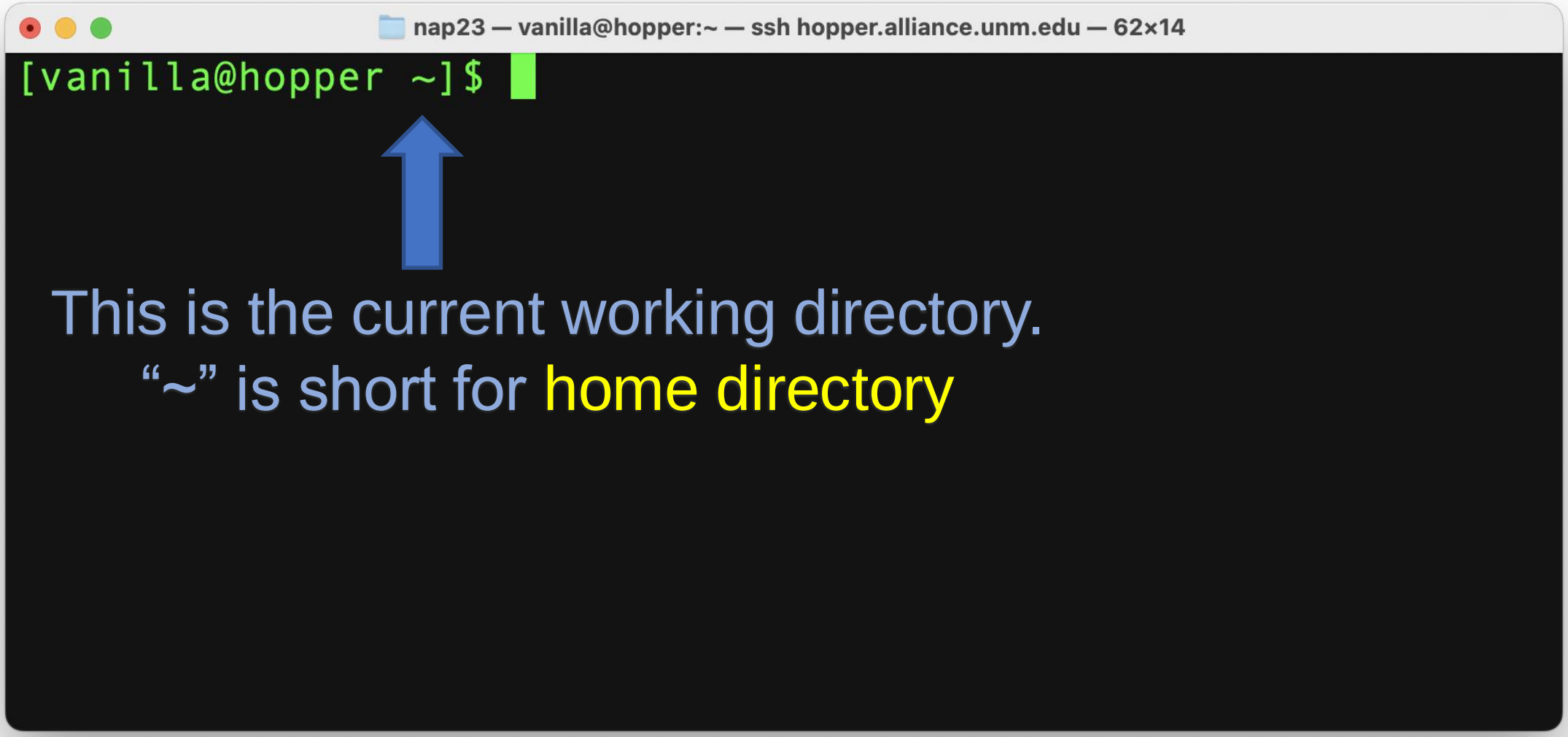




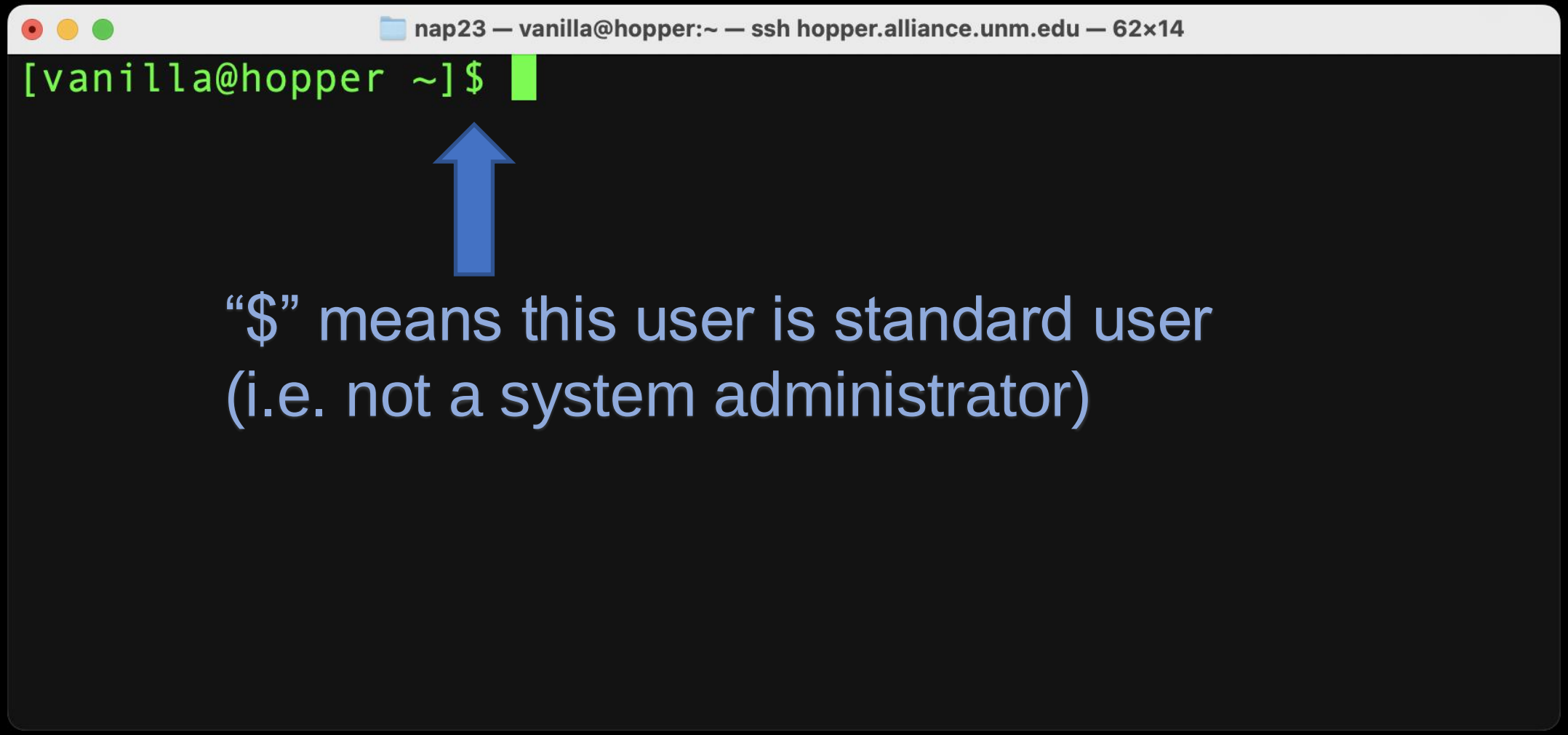
Understanding the BASH prompt…



Understanding the BASH prompt…



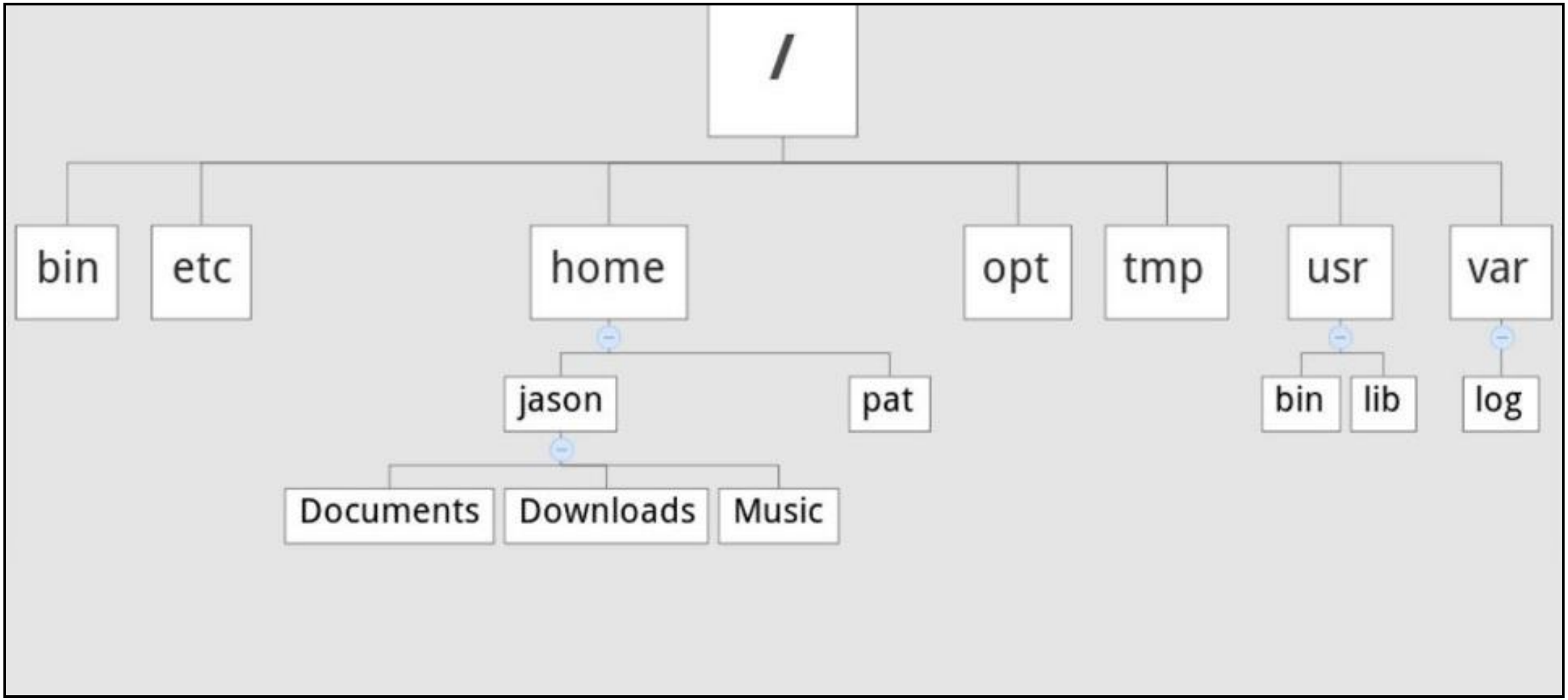
Understanding the BASH prompt…



Understanding the BASH prompt…

```
[vanilla@hopper ~]$ passwd
Changing password for user vanilla.
Current Password: {text will be hidden}
New password: {text will be hidden}
Retype new password:  {text will be hidden}
passwd: all authentication tokens updated successfully.
[vanilla@hopper ~]$
```

If you used the preset password set your own password now



Example Filesystem Tree

```
[vanilla@hopper ~]$ pwd
/users/vanilla
[vanilla@hopper ~]$
```

Figuring out where you are in the filesystem…

Figuring out where you are in the filesystem…

Please enter the following command:

```
cp -r /projects/shared/workshops/beginner/mystuff ~/
```

We will come help you if you have any trouble.

(Later I will go over what this command does)

```
[vanilla@hopper  ~]$ ls
mystuff  wheeler-scratch
[vanilla@hopper  ~]$
```

Figuring out where you are in the filesystem…

```
[vanilla@hopper  ~]$ tree
.
├── mystuff
│   ├── myfile1
│   └── myfile2
└── wheeler-scratch -> /wheeler/scratch/vanilla

2 directories, 2 files
[vanilla@hopper  ~]$
```

Figuring out where you are in the filesystem…

```
[vanilla@hopper  ~]$ tree
.
├── mystuff
│   ├── myfile1
│   └── myfile2
└── wheeler-scratch -> /wheeler/scratch/vanilla

2 directories, 2 files
[vanilla@hopper  ~]$
```

Figuring out where you are in the filesystem…

```
[vanilla@hopper  ~]$ tree
.
├── mystuff
│   ├── myfile1
│   └── myfile2
└── wheeler-scratch -> /wheeler/scratch/vanilla

2 directories, 2 files
[vanilla@hopper  ~]$
```

This . means the current directory

Figuring out where you are in the filesystem…

“Absolute” paths vs “relative” paths

- A path is a list of directories and/or files. It is a path through the directory tree that tells one how to get somewhere in the filesystem.

- An absolute path tells one how to get to the destination from starting from the root of the filesystem. E.g “/users/vanilla/mystuff/”

- A relative path specifies how to get there *starting from the current working directory*. E.g vanilla/mystuff/

```
[vanilla@hopper  ~]$ ls mystuff/
myfile1  myfile2
[vanilla@hopper  ~]$
```

Figuring out where you going…

```
[vanilla@hopper  ~]$ ls /users/vanilla/mystuff
myfile1  myfile2
 [vanilla@hopper  ~]$
```

Figuring out where you going…

```
[vanilla@hopper ~]$ ls ./mystuff/
myfile1  myfile2
[vanilla@hopper ~]$ ls ~/mystuff/
myfile1  myfile2
[vanilla@hopper ~]$
```

Figuring out where you going…

```
[vanilla@hopper ~]$ ls -a
.               .modulesbeginenv
..              mystuff
.addressbook    .oracle_jre_usage
.addressbook.lu .pinerc
.bashrc         .pki
.cache          .rhosts
.comsol         .shosts
.config         .spack
.flexlmrc       .ssh
```

Figuring out where you going…

```
[vanilla@hopper ~]$ ls -l
total 4
drwxr-xr-x 2 vanilla users 4096 Jun 14 22:05 mystuff
lrwxrwxrwx 1 vanilla users   24 Jun 14 21:20 wheeler-scratch -> /wheeler/scratch/vanilla
[vanilla@hopper ~]$
```

Figuring out where you going…

```
[vanilla@hopper ~]$ ls -l mystuff/
total 473704
-rw-r--r-- 1 vanilla users 483165473 Jun 14 23:20 myfile1
-rw-r--r-- 1 vanilla users         0 Jun 14 22:05 myfile2
[vanilla@hopper ~]$
```

Figuring out where you going…

```
[vanilla@hopper ~]$ ls -lh mystuff/
total 463M
-rw-r--r-- 1 vanilla users 461M Jun 14 23:20 myfile1
-rw-r--r-- 1 vanilla users    0 Jun 14 22:05 myfile2
[vanilla@hopper ~]$
```

Figuring out where you going…

- Now you know how to view the filesystems using bash
- Let's see how to move around and modify the filesystem.

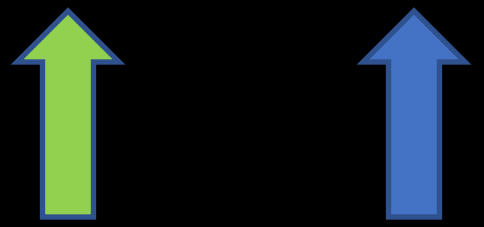
- To move we used the cd (change directory) command.
- In bash to move a file we use the mv command.
- To copy a file it is cp.
- To copy files from CARC to a personal computer use scp or rsync.

For more information and graphical tools see the "transferring data" video tutorial in in this playlist.

```
[vanilla@hopper ~]$ cd mystuff
[vanilla@hopper ~/mystuff]$ mv myfile1 myfile0
[vanilla@hopper ~/mystuff]$ ls
myfile0  myfile2  myfile3
[vanilla@hopper ~/mystuff]
```

Modifying the filesystem…
moving a file.

```
[vanilla@hopper ~/mystuff]$ cp myfile0 myfile1
[vanilla@hopper ~/mystuff]$
```

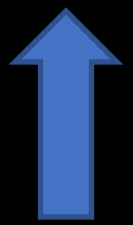


Source Destination

```
[vanilla@hopper ~/mystuff]$ ls
myfile0  myfile1  myfile2  myfile3
[vanilla@hopper ~/mystuff]$
```

Modifying the filesystem…
copying a file.

```
[vanilla@hopper ~/mystuff]$ mkdir mynewdir
[vanilla@hopper ~/mystuff]$
```



New directory name

Modifying the filesystem…
create a new directory.

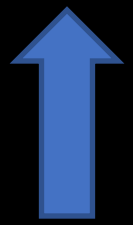
```
[vanilla@hopper ~]$ cp -r mystuff mystuff2
[vanilla@hopper ~]$
```


```
[vanilla@hopper ~]$ ls
mystuff  mystuff2  wheeler-scratch
```

Copying a whole directory tree…

```
[vanilla@hopper ~]$ exit
CARCWS-01:~ vanilla$ scp vanilla@hopper.alliance.unm.edu:~/mystuff/myfile3 Desktop/
```

Source



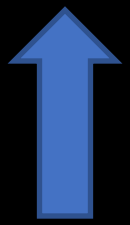
Destination

```
(vanilla@hopper.alliance.unm.edu) Password:
myfile3                   100%   40     2.0KB/s   00:00
```

Copying data to a personal computer from CARC…

```
CARCWS-01:~ vanilla$ scp -r vanilla@hopper.alliance.unm.edu:~/mystuff Desktop/
```

Source



Destination

```
(vanilla@hopper.alliance.unm.edu) Password:
myfile1                 100% 1024KB   6.5MB/s   00:00
myfile2                 100% 2048KB 382.5KB/s   00:05
myfile3                 100%   40     3.2KB/s   00:00
myfile0                 100% 1024KB   8.8MB/s   00:00
```

Copying data to a personal computer from CARC…

```
CARCWS-01:~ vanilla$ scp -r Desktop/mystuff vanilla@hopper.alliance.unm.edu:~/
```

Source

Destination

```
(vanilla@hopper.alliance.unm.edu) Password:
myfile1                  100% 1024KB 591.5KB/s   00:01
myfile0                  100% 1024KB   2.0MB/s   00:00
myfile2                  100% 2048KB   2.1MB/s   00:00
myfile3                  100%   40     2.1KB/s   00:00
```

To copy from a personal computer to CARC…

```
ssh vanilla@hopper.alliance.unm.edu
```

Log back into hopper…

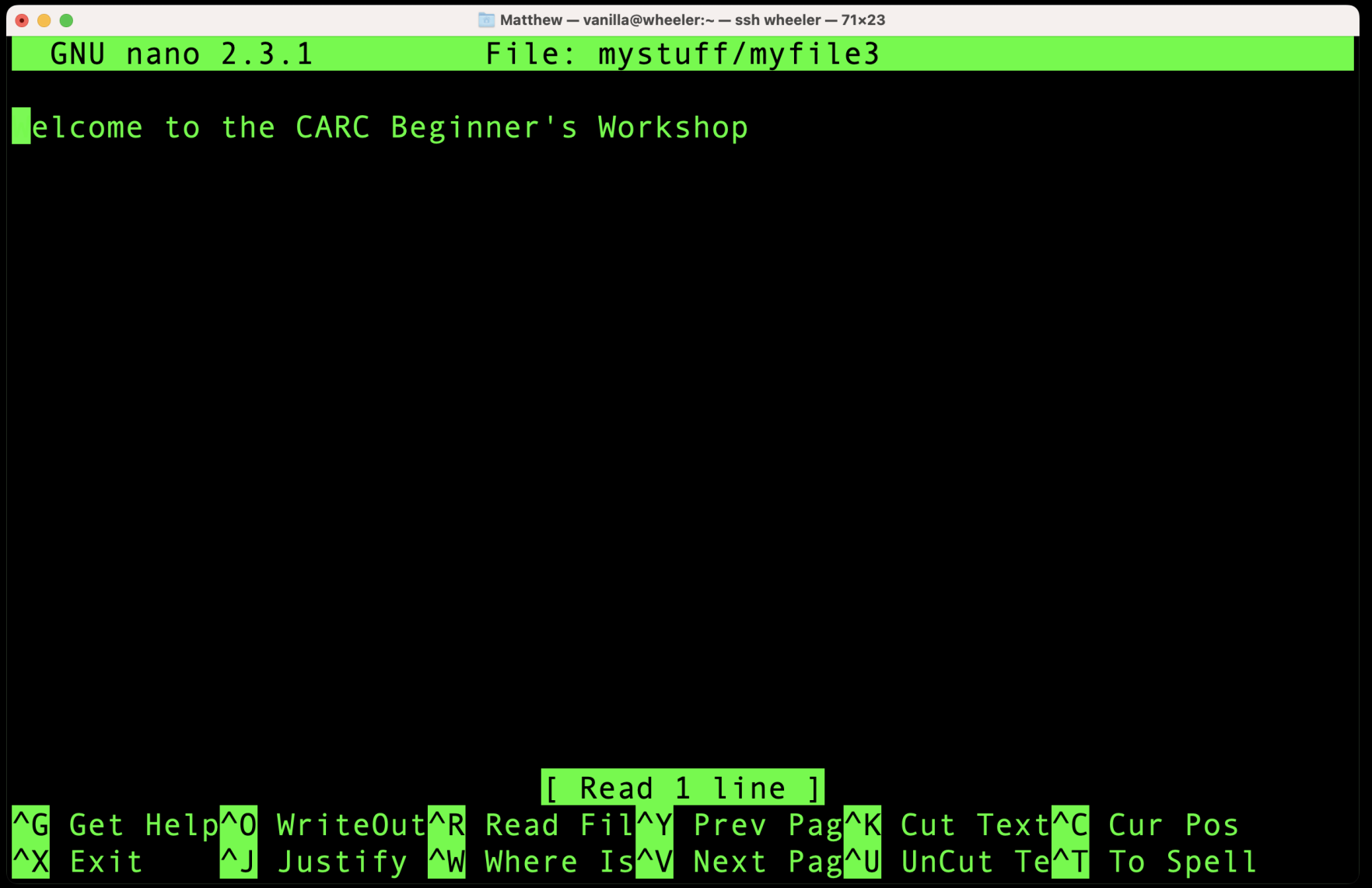
```
[vanilla@hopper ~]$ file mystuff/myfile0
mystuff/myfile0: data

[vanilla@hopper ~]$ file mystuff/myfile3
mystuff/myfile3: ASCII text
```

Figuring out file types …

```
[vanilla@hopper ~]$ cat mystuff/myfile3
Welcome to the CARC Beginner's Workshop
```

Text files …

[vanilla@hopper ~]$ nano mystuff/myfile3

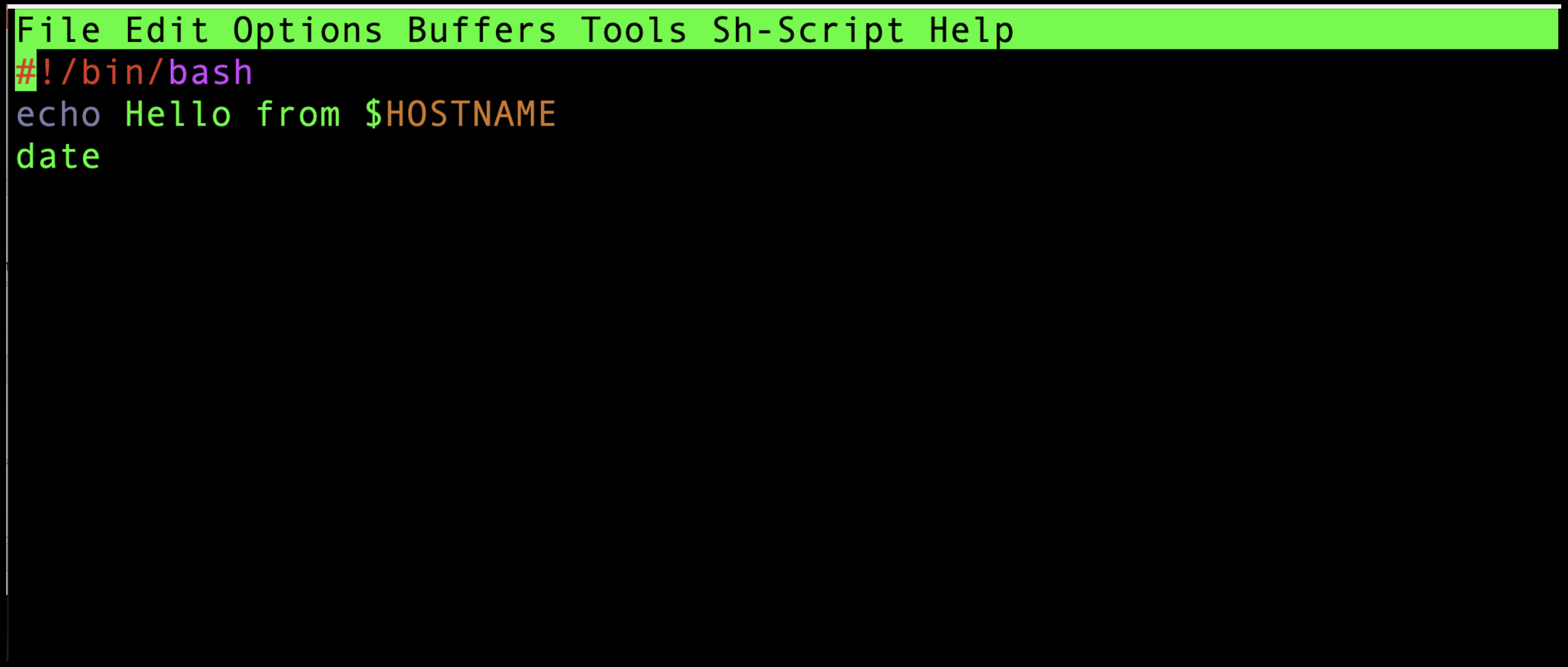


```
[vanilla@hopper ~]$ date
Wed Jun 15 03:08:15 MDT 2022

[vanilla@hopper ~]$ echo Hello from $HOSTNAME
Hello from hopper

[vanilla@hopper ~]$ hostname
hopper
```

Programs we will use as examples…

```
[vanilla@hopper ~]$ nano myscript.bash
```

BASH Script



Shell Scripts

```
[vanilla@hopper ~]$ nano myscript.bash
[vanilla@hopper ~]$ bash myscript.bash
Hello from hopper
Wed Mar 26 16:17:52 MDT 2025
```

BASH Script

Software Access

Lmod
Modules

Installed Systemwide by CARC personnel
Email help@carc.unm.edu

Conda

Installed in your home dir yourself

```
[vanilla@hopper ~]$ module spider matlab

----------------------------------------------------------------------------
  matlab:
----------------------------------------------------------------------------
     Versions:
        matlab/R2017a
        matlab/R2018b
        matlab/R2019a
        matlab/R2020a
        matlab/R2021a
        matlab/R2021b
        matlab/R2022a
        matlab/R2023a
        matlab/R2023b
```

Lmod Modules

Getting access to software…

```
[vanilla@hopper ~]$ module load matlab/R2021a
Lmod has detected the following error: Matlab may only be run on compute
nodes. hopper is not a compute node. Exiting...
While processing the following module(s):
   Module fullname  Module Filename
   ---------------  ---------------
   matlab/R2021a    /opt/local/modules/matlab/R2021a.lua
```

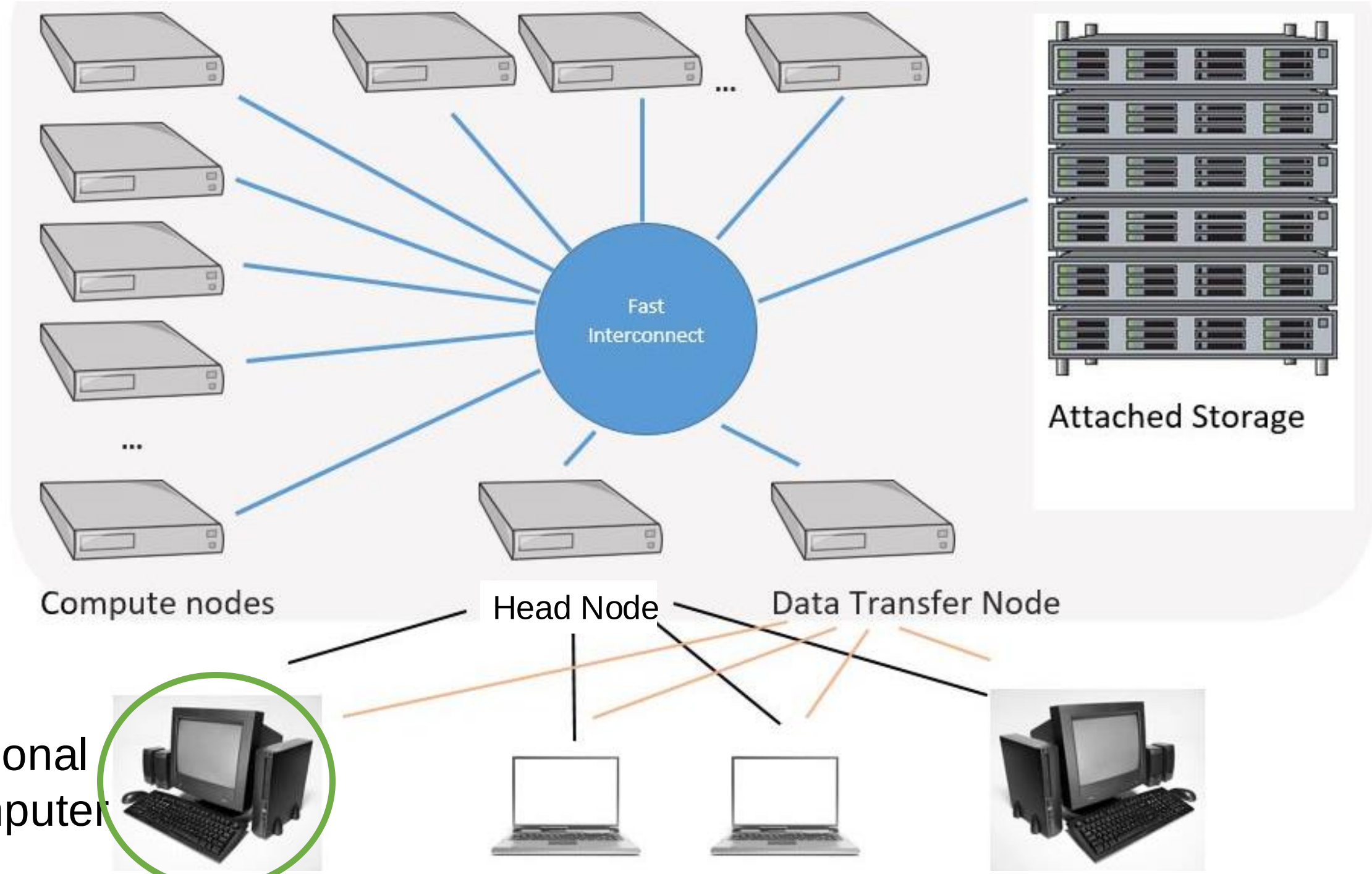
Getting access to software…

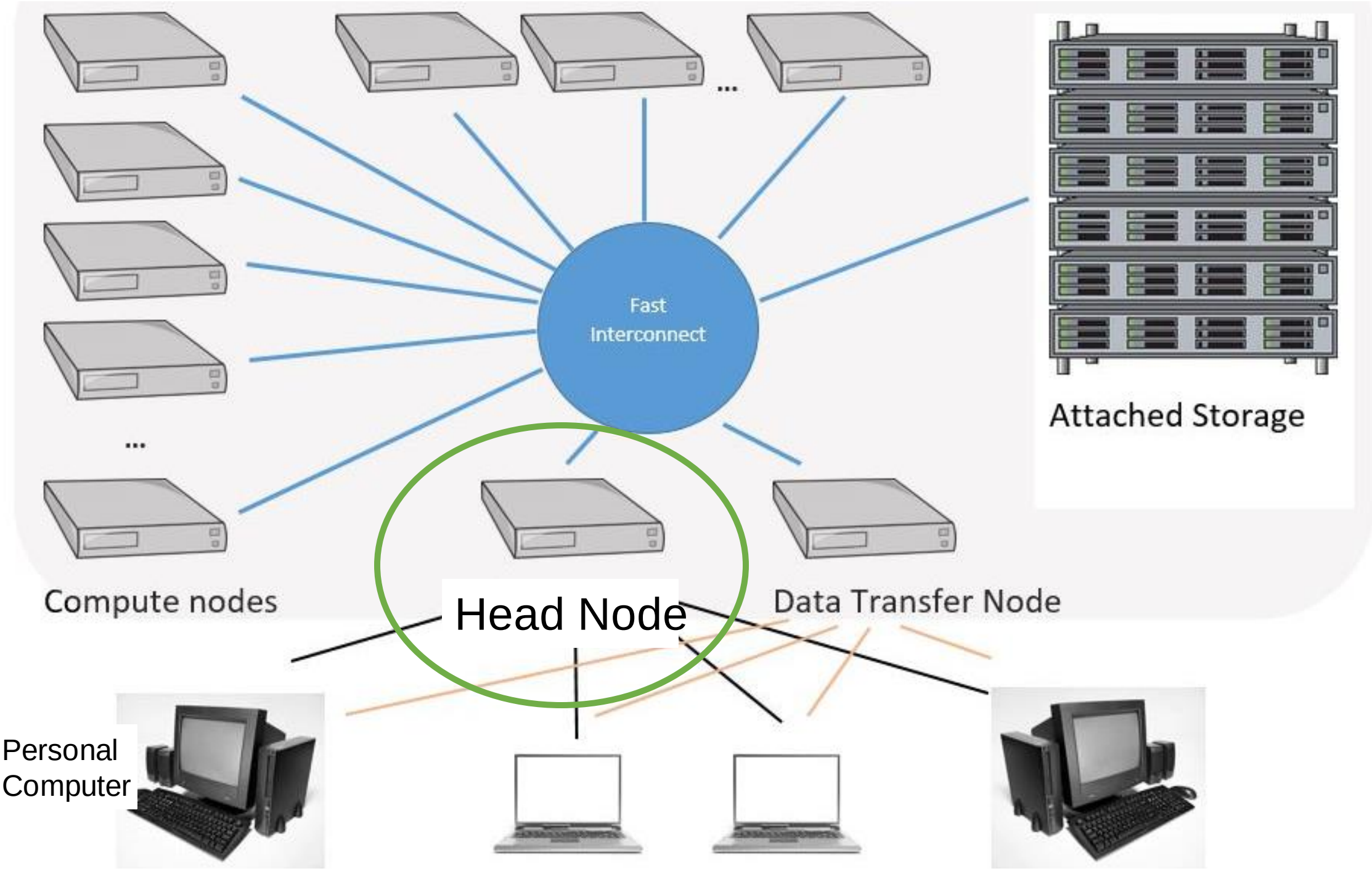
```
[vanilla@hopper ~]$ module load matlab/R2021a
Lmod has detected the following error: Matlab may only be run on compute
nodes. hopper is not a compute node. Exiting...
While processing the following module(s):
   Module fullname  Module Filename
   ---------------  ---------------
   matlab/R2021a    /opt/local/modules/matlab/R2021a.lua
```

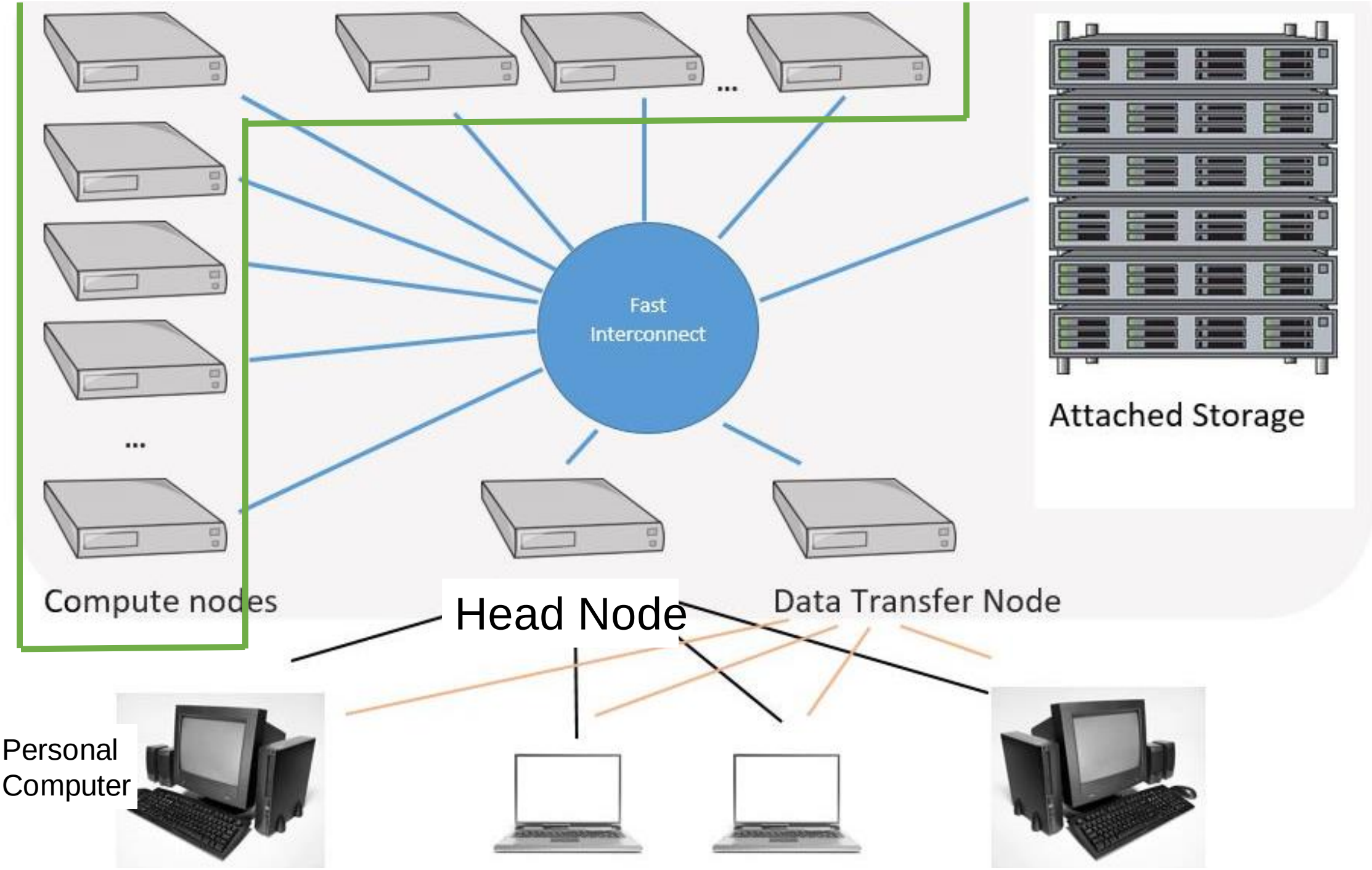
What is a compute node?

Getting access to software…


HPC
Cluster


Fast
Interconnect
Attached Storage
Compute nodes
Head Node
Data Transfer Node
Personal
Computer

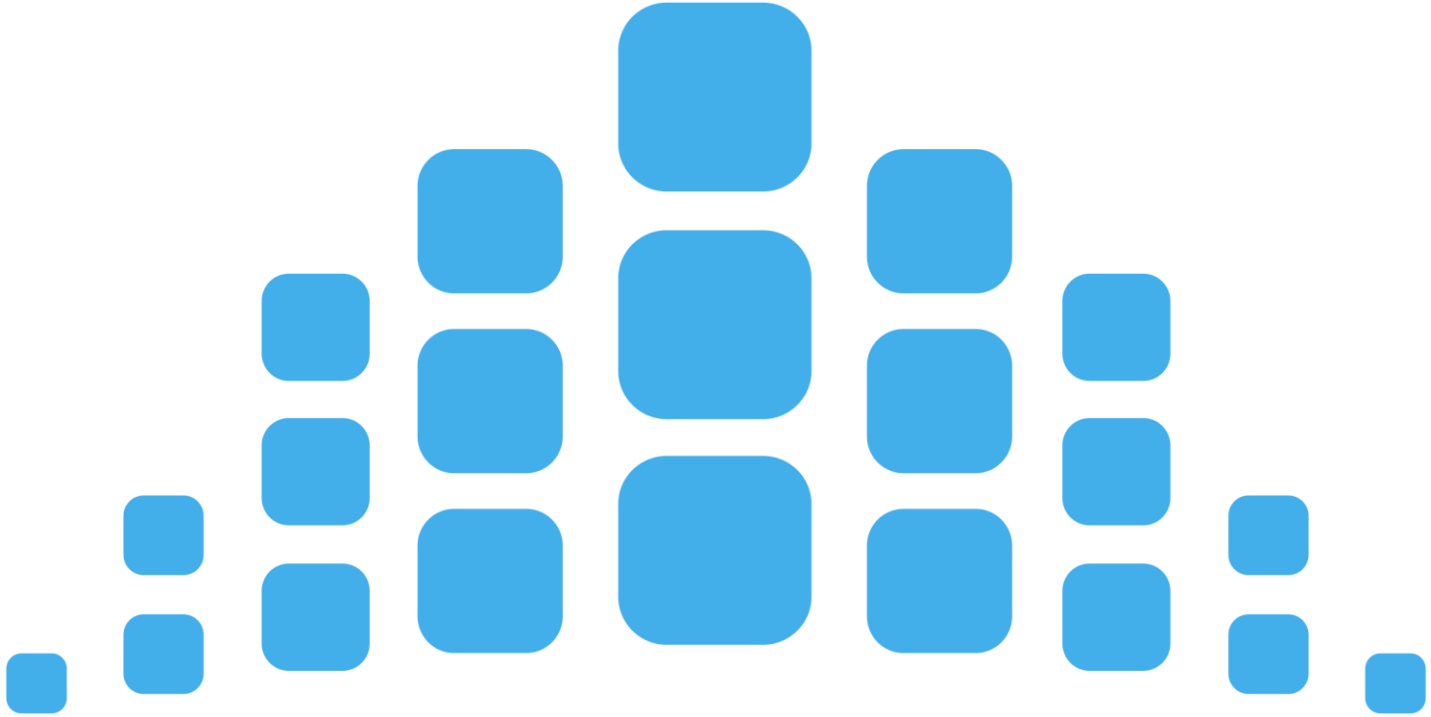

Fast
Interconnect
Attached Storage
Compute nodes
Head Node
Data Transfer Node
Personal
Computer


Fast
Interconnect
Attached Storage
Compute nodes
Head Node
Data Transfer Node
Personal
Computer

Never run computations on the head node

(This can make it so no one else can login)

Always use compute nodes



slurm

workload manager

Simple Linux Utility for Resource Management

Simple Linux

workload manager

```
[vanilla@hopper ~]$ qgrok
partition      nodes nodes nodes total total free  total free  CPUs  RAM/node  time  CPU   GPU   RAM
name     jobs  free  busy  down  nodes CPUs  CPUs  GPUs  GPUs  /node       limit limit limit limit
---------- ----- ----- ----- ----- ----- ----- ----- ----- ----- --------- ----- ----- ----- -----
general   4    4    6    0    10   320  156  0    0    32   93G     2d   128   0    372G
debug     1    1    1    0    2    64   63   0    0    32   93G     4h   8     0    25G
totals:   5    5    7    0    12   384  219  0    0
```

```
[vanilla@hopper ~]$ qgrok
partition      nodes nodes nodes total total free  total free  CPUs  RAM/node  time  CPU   GPU   RAM
name     jobs  free  busy  down  nodes CPUs  CPUs  GPUs  GPUs  /node       limit limit limit limit
---------- ----- ----- ----- ----- ----- ----- ----- ----- ----- --------- ----- ----- ----- -----
general   4    4    6    0    10   320  156  0    0    32   93G     2d   128   0    372G
debug     1    1    1    0    2    64   63   0    0    32   93G     4h   8     0    25G
condo     29   30   25   0    55   1760 1060 37   8    32   94G-1.5T 2d   512   4    1.5T
bugs      0    0    2    0    2    64   0    0    0    32   93G     1w   64    0
pcnc      0    0    2    0    2    64   0    0    0    32   93G     1w   64    0
pathogen  0    1    0    0    1    32   32   0    0    32   93G     1w   32    0
tc        8    1    9    0    10   320  104  0    0    32   93G-1.5T 1w   320   0
gold      0    2    0    0    2    64   64   0    0    32   93G     1w   64    0
fishgen   1    0    1    0    1    32   12   0    0    32   377G    1w   32    0
neuro-hsc 0    14   0    0    14   448  448  0    0    32   93G     1w   448   0
pna       0    1    0    0    1    32   32   0    0    32   93G     1w   32    0
geodef    0    4    0    0    4    128  128  0    0    32   504G    1w   128   0
cup-ecs   0    0    2    0    2    64   0    13   8    32   188G    1w   64    13
tid       0    0    1    0    1    32   0    2    0    32   188G    1w   32    2
biocomp   0    1    0    0    1    32   32   1    0    32   188G    1w   32    1
chakra    0    0    1    0    1    32   0    1    0    32   188G    1w   32    1
quark     0    3    7    0    10   320  112  20   0    32   377G    1w3d 320   20
toadpole  0    1    0    0    1    32   32   0    0    32   94G     1w   32    0
insar     0    2    0    0    2    64   64   0    0    32   504G    1w   64    0
totals:   34   35   32   0    67   2144 1279 37   8
```

```
[vanilla@hopper ~]$ qgrok
partition      nodes nodes nodes total total free  total free  CPUs  RAM/node  time  CPU   GPU   RAM
name     jobs  free  busy  down  nodes CPUs  CPUs  GPUs  GPUs  /node       limit limit limit limit
---------- ----- ----- ----- ----- ----- ----- ----- ----- ----- --------- ----- ----- ----- -----
general   4    4    6    0    10   320  156  0    0    32   93G     2d   128   0    372G
debug     1    1    1    0    2    64   63   0    0    32   93G     4h   8     0    25G
condo     29   30   25   0    55   1760 1060 37   8    32   94G-1.5T 2d   512   4    1.5T
bugs      0    0    2    0    2    64   0    0    0    32   93G     1w   64    0
pcnc      0    0    2    0    2    64   0    0    0    32   93G     1w   64    0
pathogen  0    1    0    0    1    32   32   0    0    32   93G     1w   32    0
tc        8    1    9    0    10   320  104  0    0    32   93G-1.5T 1w   320   0
gold      0    2    0    0    2    64   64   0    0    32   93G
fishgen   1    0    1    0    1    32   12   0    0    32   377
neuro-hsc 0    14   0    0    14   448  448  0    0    32
pna       0    1    0    0    1    32   32   0    0    32   93
geodef    0    4    0    0    4    128  128  0    0    32
cup-ecs   0    0    2    0    2    64   0    13   8    32   1
tid       0    0    1    0    1    32   0    2    0    32   188
biocomp   0    1    0    0    1    32   32   1    0    32
chakra    0    0    1    0    1    32   0    1    0    32   18
quark     0    3    7    0    10   320  112  20   0    32
toadpole  0    1    0    0    1    32   32   0    0    32
insar     0    2    0    0    2    64   64   0    0    32   50
totals:   34   35   32   0    67   2144 1279 37   8
```

Open partitions for use by everyone with a CARC account.

Purchased by the Office for the Vice President for Research.

```
[vanilla@hopper ~]$ sinfo --partition debug
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug        up    4:00:00      1    mix hopper011
debug        up    4:00:00      1   idle hopper012
```

sinfo reports information about partitions

```
[vanilla@hopper ~]$ sinfo --partition debug
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug        up    4:00:00      1    mix hopper011
debug        up    4:00:00      1   idle hopper012
```

The debug queues are intended for testing your programs.

And for interactive jobs.

```
[vanilla@hopper ~]$ sinfo --partition debug
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug        up    4:00:00      1    mix hopper011
debug        up    4:00:00      1   idle hopper012
```

Name

```
[vanilla@hopper ~]$ sinfo --partition debug
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug        up    4:00:00      1    mix hopper011
debug        up    4:00:00      1   idle hopper012
```

You can run a “job” for up to 4 hrs.

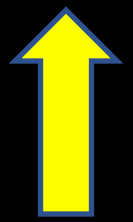
```
[vanilla@hopper ~]$ sinfo --partition debug
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug        up    4:00:00      1    mix hopper011
debug        up    4:00:00      1   idle hopper012
```

There are two nodes in this partition.

```
[vanilla@hopper ~]$ sinfo --partition debug
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug        up    4:00:00      1    mix hopper011
debug        up    4:00:00      1   idle hopper012
```

The state of the nodes in the partition

```
[vanilla@hopper ~]$ sinfo --partition debug
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug       up    4:00:00      1    mix hopper011
debug       up    4:00:00      1   idle hopper012
```



The name of the nodes in the partition

```
[vanilla@hopper ~]$ sinfo --partition general
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
general*     up 2-00:00:00      5  alloc hopper[001-009]
general*     up 2-00:00:00      4   idle hopper010
```

Hopper001 through 009 are running jobs.

Hopper010 is waiting to be used.

```
[vanilla@hopper ~]$ hostname
hopper
[vanilla@hopper ~]$
```

Running on the Head Node.
The head node's name is "hopper".

```
[vanilla@hopper ~]$ srun --partition debug hostname
```

Tell slurm to run a program on a compute node…

```
[vanilla@hopper ~]$ srun --partition debug hostname
```

Run the program on a compute node in the debug partition.

```
[vanilla@hopper ~]$ srun --partition debug hostname
```

The program to run.

```
[vanilla@hopper ~]$ srun --partition debug hostname
srun: Account not specified in script or ~/.default_slurm_account, using latest project
You have not been allocated GPUs. To request GPUs, use the -G option in your submission script.
hopper011
```

```
[vanilla@hopper ~]$ squeue
```

[vanilla@hopper ~]$ squeue

```
  JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
   4314   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4315   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4317   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4318   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4319   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4320   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4321   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4322   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4323   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4324   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4325   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4326   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4328   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4329   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4330   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4331   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4332   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4333   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4334   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4335   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4336   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
   4337   general      PRE erowland PD       0:00      2 (QOSMaxCpuPerUserLimit)
```

[vanilla@hopper ~]$ squeue

```
  JOBID PARTITION     NAME     USER ST       TIM
   4314   general      PRE erowland PD      0:00
   4315   general      PRE erowland PD      0:00
   4317   general      PRE erowland PD      0:00
   4318   general      PRE erowland PD      0:00
   4319
   4320
   4321
   4322
   4323
   4324
   4325
   4326
   4328
   4329
   4330
   4331
   4332
   4333
   4334
   4335
   4336
   4337   general      PRE erowland PD      0:00
```

PD means programs that are waiting their turn.

Shows you what the slurm scheduler is doing right now.

Here we can see that user ’erowland’ has a lot of programs waiting to run.

```
[vanilla@hopper ~]$ squeue -t R --all
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
              4405     condo    2ndMA  mfricke  R 1-07:48:30      6 hopper[031-036]
              5208     condo       NN     kgu  R    5:48:49      1 hopper015
              5210     condo       NN     kgu  R    6:30:13      1 hopper014
              5209     condo       NN     kgu  R    6:31:13      1 hopper013
              5206     condo       NN     kgu  R    6:32:13      1 hopper051
              5207     condo       NN     kgu  R    6:32:13      1 hopper052
              5205     condo       NN     kgu  R    6:32:43      1 hopper028
              4595   cup-ecs golConfi aalasand  R 2-06:51:59      1 hopper050
              4594   cup-ecs golConfi aalasand  R 2-06:52:03      1 hopper049
              5120   general jupyterh   jacobm  R   11:45:47      1 hopper007
              4313   general      PRE erowland  R    1:17:29      2 hopper[003-004]
              5111   general    1stMA  mfricke  R   11:15:28      2 hopper[005-006]
              5025   general      c2n    jxzuo  R      1:50      1 hopper001
              5024   general      c2n    jxzuo  R     31:28      1 hopper002
              5203   general       NN     kgu  R    6:37:50      1 hopper009
              5201   general       NN     kgu  R    6:38:14      1 hopper008
              4390        tc UCsTpCyd lepluart  R 2-15:18:18      3 hopper[018-020]
              5198        tc       NN     kgu  R    6:40:19      1 hopper030
              5196        tc       NN     kgu  R    6:40:31      1 hopper029
```

```
[vanilla@hopper ~]$ squeue -t R --all
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
              4405     condo    2ndMA  mfricke  R 1-07:48:30      6 hopper[031-036]
              5208     condo       NN      kgu  R    5:48:49      1 hopper015
              5210     condo       NN      kgu  R    6:30:13      1 hopper014
              5209     condo       NN      kgu  R    6:31:13      1 hopper013
              5206     condo       NN      kgu  R    6:32:13      1 hopper051
              5207     condo       NN      kgu  R    6:32:13      1 hopper052
              5205     condo       NN      kgu  R    6:32:43      1 hopper028
              4595   cup-ecs golConfi aalasand  R 2-06:51:59      1 hopper050
              4594   cup-ecs golConfi aalasand  R 2-06:52:03      1 hopper049
              5120   general jupyterh   jacobm  R   11:45:47      1 hopper007
              4313   general      PRE erowland  R    1:17:29      2 hopper[003-0
              5111   general    1stMA  mfricke  R   11:15:28      2 hopper[005-006]
              5025   general      c2n    jxzuo  R      1:50      1 hopper001
              5024   general      c2n    jxzuo  R     31:28      1 hopper002
              5203   general       NN      kgu  R    6:37:50      1 hopper009
              5201   general       NN      kgu  R    6:38:14      1 hopper008
              4390        tc UCsTpCyd lepluart  R 2-15:18:18      3 hopper[018-020]
              5198        tc       NN      kgu  R    6:40:19      1 hopper030
              5196        tc       NN      kgu  R    6:40:31      1 hopper029
```

Only one job running

```
[vanilla@hopper ~]$ srun --partition debug --ntasks 8 hostname
srun: Account not specified in script or ~/.default_slurm_account, using latest project
hopper011
hopper011
hopper011
You have not been allocated GPUs. To request GPUs, use the -G option in your submission
script.
hopper011
hopper011
hopper011
hopper011
hopper011
hopper011
```

You ran eight copies of your program.

ntasks is the number of copies to run.

```
[vanilla@hopper ~]$ srun --partition debug --ntasks 8 hostname
srun: Account not specified in script or ~/.default_slurm_account, using latest project
hopper011
hopper011
hopper011
You have not been allocated GPUs. To request GPUs, use the -G option in your submission
script.
hopper011
hopper011
hopper011
hopper011
hopper011
hopper011
```

By default, each task (copy of your program) is allowed to use one CPU.

Many programs are able to use more than one CPU at a time.

```
[vanilla@hopper ~]$ srun --partition debug --ntasks 2 --cpus-per-task 2 hostname
srun: Account not specified in script or ~/.default_slurm_account, using latest project
You have not been allocated GPUs. To request GPUs, use the -G option in your submission script.
hopper011
hopper011
```

Here we are telling SLURM to run 2 copies of our program and let each copy of our program use 2 CPUs.

```
[vanilla@hopper ~]$ srun --partition debug --mem 4G
--ntasks 2 --cpus-per-task 2 hostname
srun: Account not specified in script or ~/.default_slurm_account, using latest
project
hopper012
hopper012
You have not been allo                                                    r
submission script.
```

And we can specify how much memory we want.

--mem 4G means give me 4 gigabytes of memory per node.

```
[vanilla@hopper ~]$ srun --partition debug --mem 4G
--ntasks 2 --cpus-per-task 2 hostname
srun: Account not specified in script or ~/.default_slurm_account, using latest
project
hopper012
hopper012
You have not been allo                                                    r
submission script.
```

Why does all this matter?

The purpose of SLURM is to provide you the hardware your programs need.

So you have to understand what those requirements are really well.

```
[vanilla@hopper ~]$ srun --partition debug --mem 4G
--ntasks 2 --cpus-per-task 2 hostname
srun: Account not specified in script or ~/.default_slurm_account, using latest
project
hopper012
hopper012
You have not been all                                                    r
submission script.
```

1) **Can my program use multiple CPUs?**
2) **How much memory does my program need?**
3) **Can my program use multiple compute nodes (MPI*, GNU Parallel*)?**
4) **Can my program use GPUs?**

Interactive vs Batch Mode

Interactive Mode

- Everything so far has been interactive. You request hardware, run your program, and get the output on your screen right away.

Batch Mode

- Most programs at an HPC center are run in “batch” mode.
- Batch mode means we write a shell script that the SLURM scheduler runs for us. The script requests hardware just like we did with salloc and then runs the commands in the script.
- Whatever would have been written to the screen is saved to a file instead.

```
[vanilla@hopper ~]$ git clone https://lobogit.unm.edu/CARC/workshops.git
Cloning into 'workshops'...
remote: Enumerating objects: 132, done.
remote: Counting objects: 100% (75/75), done.
remote: Compressing objects: 100% (43/43), done.
remote: Total 132 (delta 33), reused 74 (delta 32), pack-reused 57
Receiving objects: 100% (132/132), 57.58 KiB | 3.60 MiB/s, done.
Resolving deltas: 100% (51/51), done.
```

Rather than make you write shell scripts lets just download some we wrote for this workshop…

```
[vanilla@hopper ~]$ tree workshops
workshops
├── crystallography
│   ├── data
│   │   └── thaumatin.pdb
│   └── slurm
│       ├── integrate_xray_images.slurm
│       └── solve_structure.slurm
├── escape
│   ├── fit_insar_data.ipynb
│   └── kml_functions.py
├── gaussian
│   ├── AcetylChloride.com
│   └── gaussian16_linda_acetylchloride.sh
├── intro_workshop
│   ├── code
│   │   ├── calc_pi_mpi.f90
│   │   ├── calcPiMPI.py
│   │   ├── calc_pi_serial.f90
│   │   ├── calcPiSerial.py

...

6 directories, 30 files
```

Run tree to see how the workshops directories are organized…

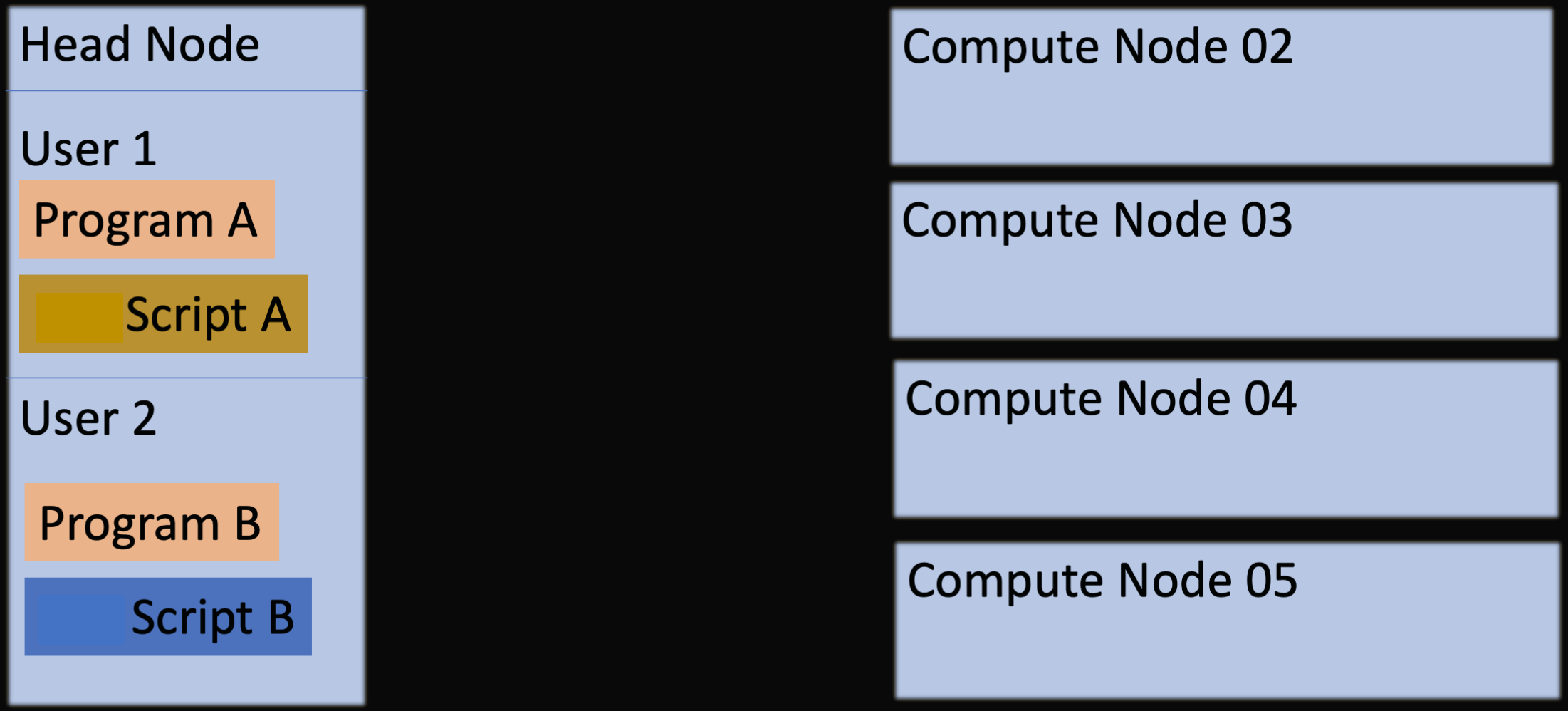
```
[vanilla@hopper intro_workshop]$ pwd
/users/vanilla/workshops/intro_workshop
[vanilla@hopper intro_workshop]$ cat slurm/workshop_example1.sh
#!/bin/bash
#SBATCH --partition debug
#SBATCH --ntasks 4
#SBATCH --time 00:05:00
#SBATCH --job-name ws_example
#SBATCH --mail-user your_username@unm.edu
#SBATCH --mail-type ALL

srun hostname
```

Let's take a look at the workshop_example.sh script in the slurm directory…

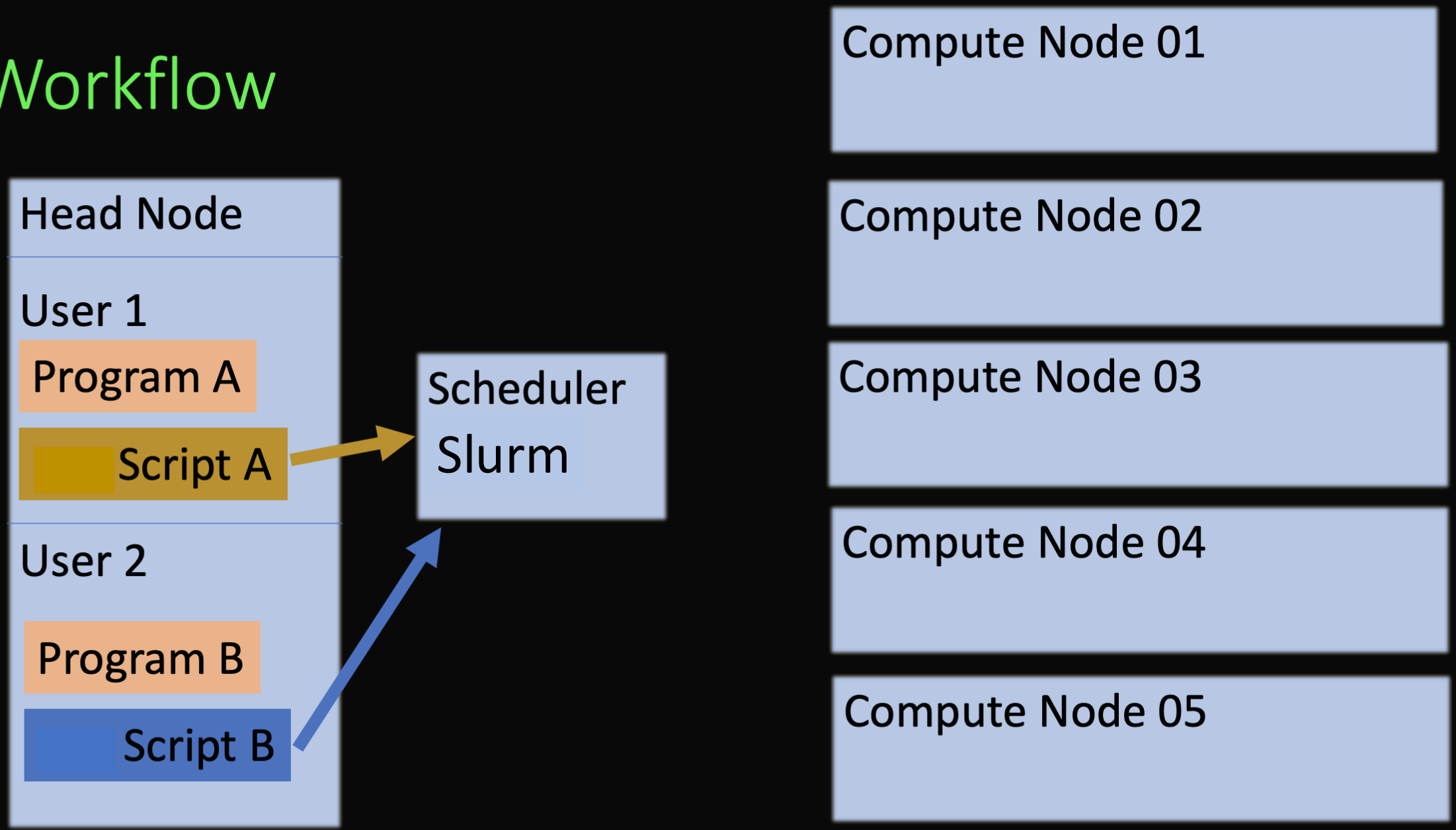
```
[vanilla@hopper intro_workshop]$ sbatch slurm/workshop_example1.sh
sbatch: Account not specified in script or ~/.default_slurm_account, using latest project
Submitted batch job 5252
[vanilla@hopper intro_workshop]$
```

We submit our slurm shell script with the sbatch command.

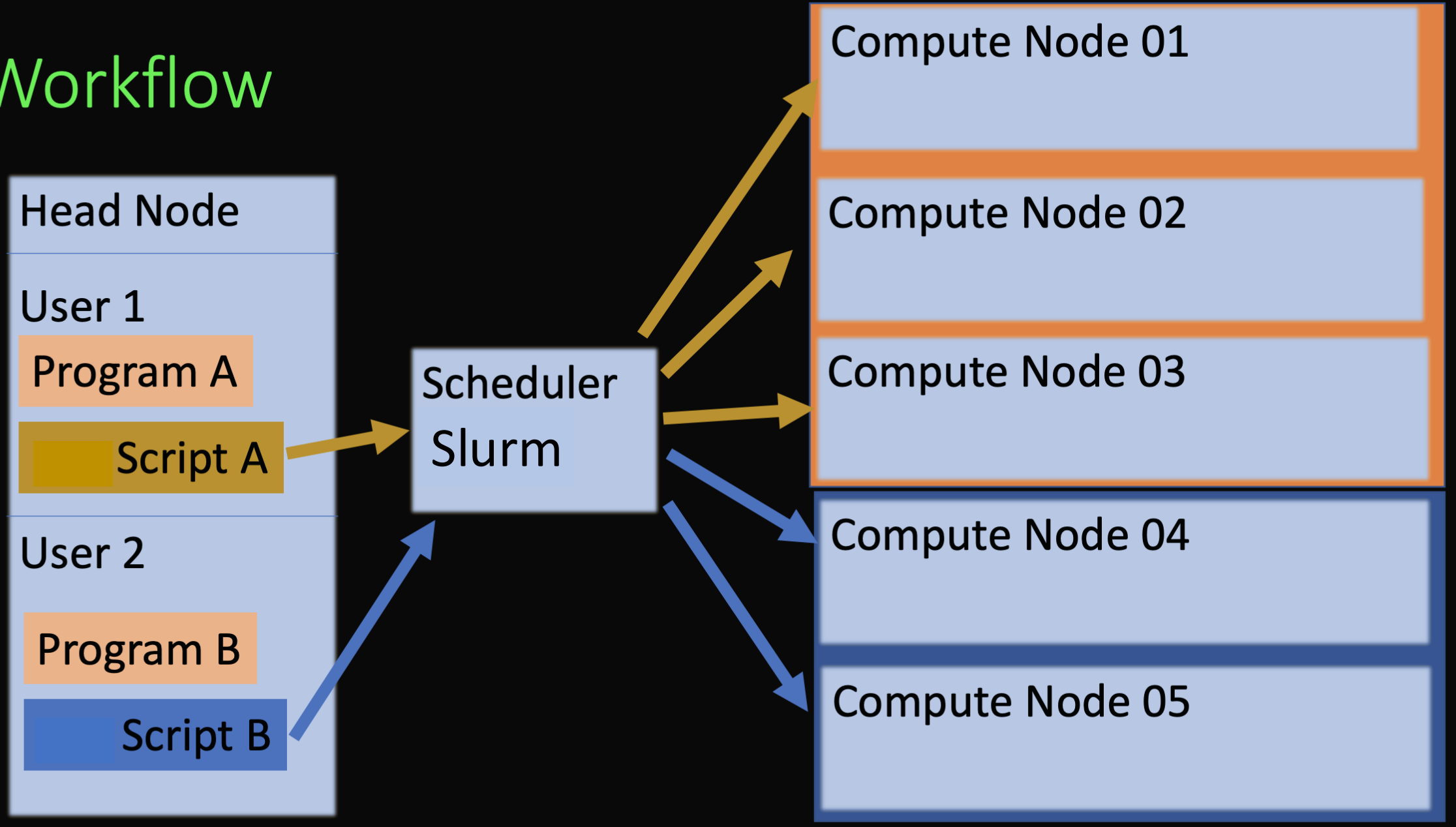
Workflow


Head Node
User 1
Program A
Script A
User 2
Program B
Script B
Compute Node 01
Compute Node 02
Compute Node 03
Compute Node 04
Compute Node 05
Shared filesystems – All nodes can access the same programs and write output

Workflow

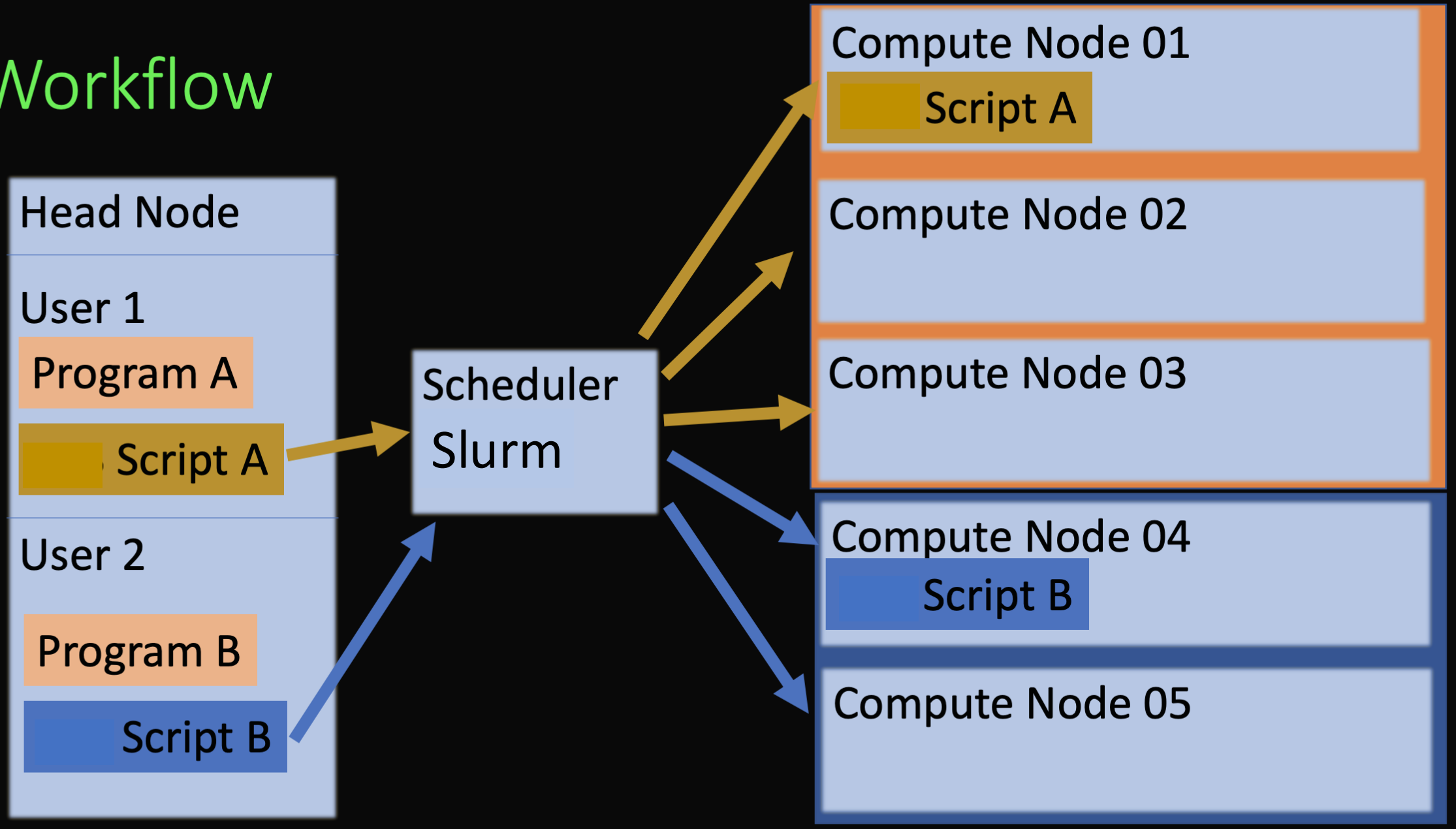

Head Node
User 1
Program A
Script A
User 2
Program B
Script B
Scheduler
Slurm
Compute Node 01
Compute Node 02
Compute Node 03
Compute Node 04
Compute Node 05
Shared filesystems – All nodes can access the same programs and write output

Workflow



Shared filesystems – All nodes can access the same programs and write output

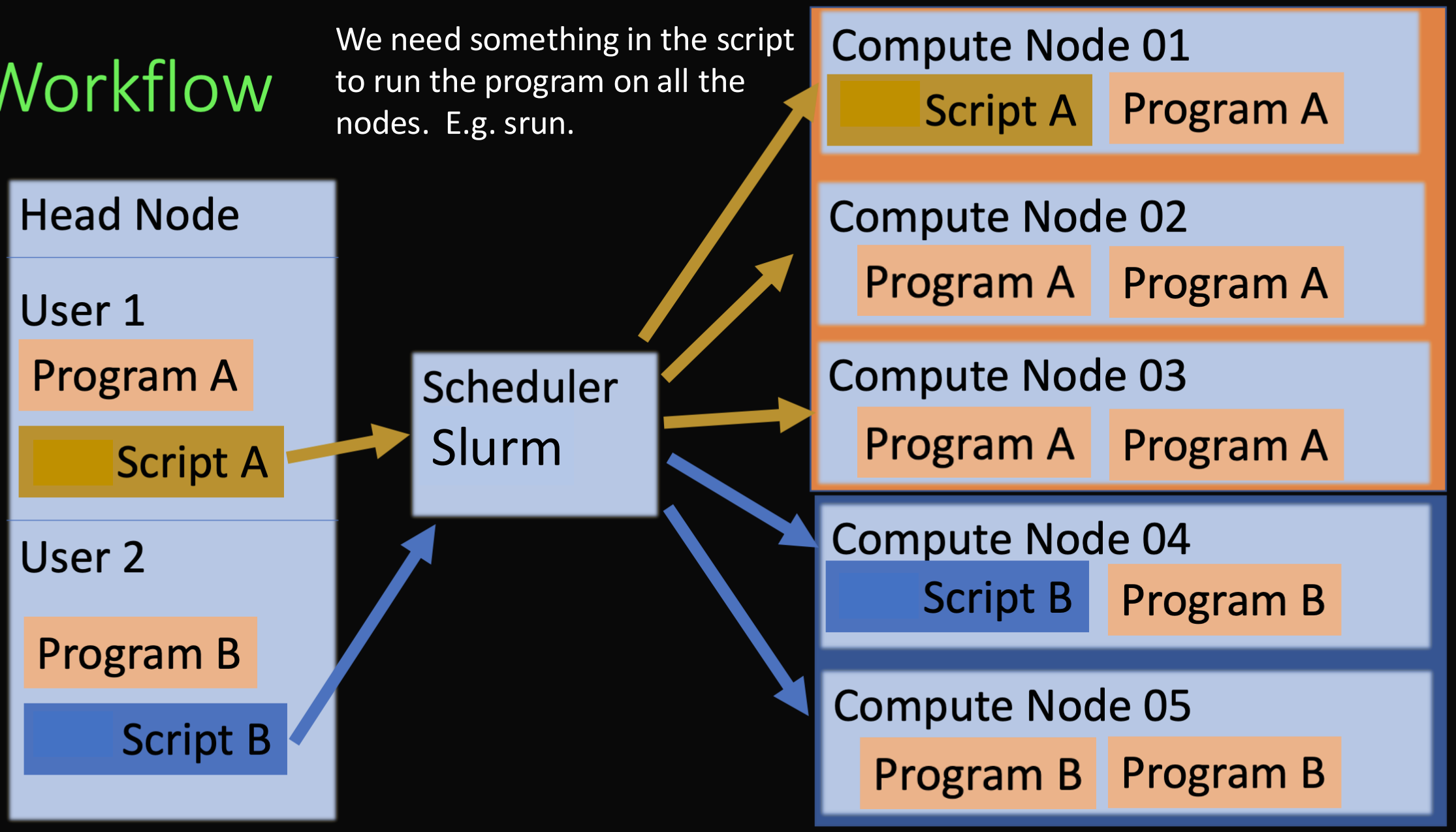
Workflow



Shared filesystems – All nodes can access the same programs and write output

Workflow

We need something in the script to run the program on all the nodes. E.g. srun.



Shared filesystems – All nodes can access the same programs and write output

```
[vanilla@hopper intro_workshop]$ ls
code  data  pbs  slurm  slurm-5252.out
```

The hostname command is very fast so everyone's job should finish in a few seconds.

When it is finished you will have a new file named slurm-{your job id}.out.

```
[vanilla@hopper intro_workshop]$ ls
code  data  pbs  slurm  slurm-5252.out
```

When it is finished you will have a new file named slurm-{your job id}.out.

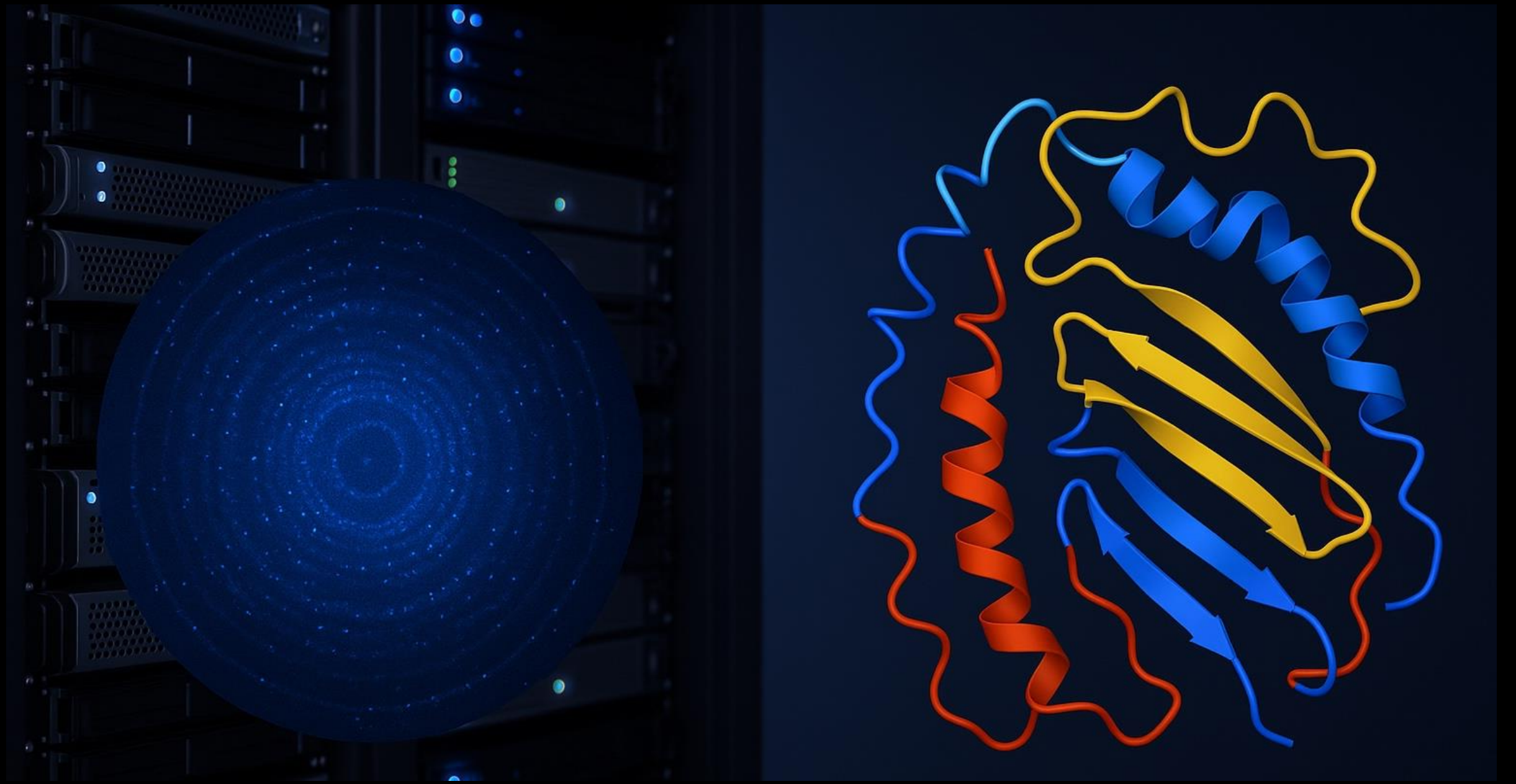
```
[vanilla@hopper intro_workshop]$ cat slurm-5252.out
What do you see?
```

```
[vanilla@hopper intro_workshop]$ ls
code  data  pbs  slurm  slurm-5252.out
```

When it is finished you will have a new file named slurm-{your job id}.out.

```
[vanilla@hopper intro_workshop]$ cat slurm-5252.out
To request GPUs, add --gpus-per-node X or --gpus X, where X is the desired number of GPUs.
Job 3687623 running on hopper011
hopper011
hopper011
hopper011
hopper011
```

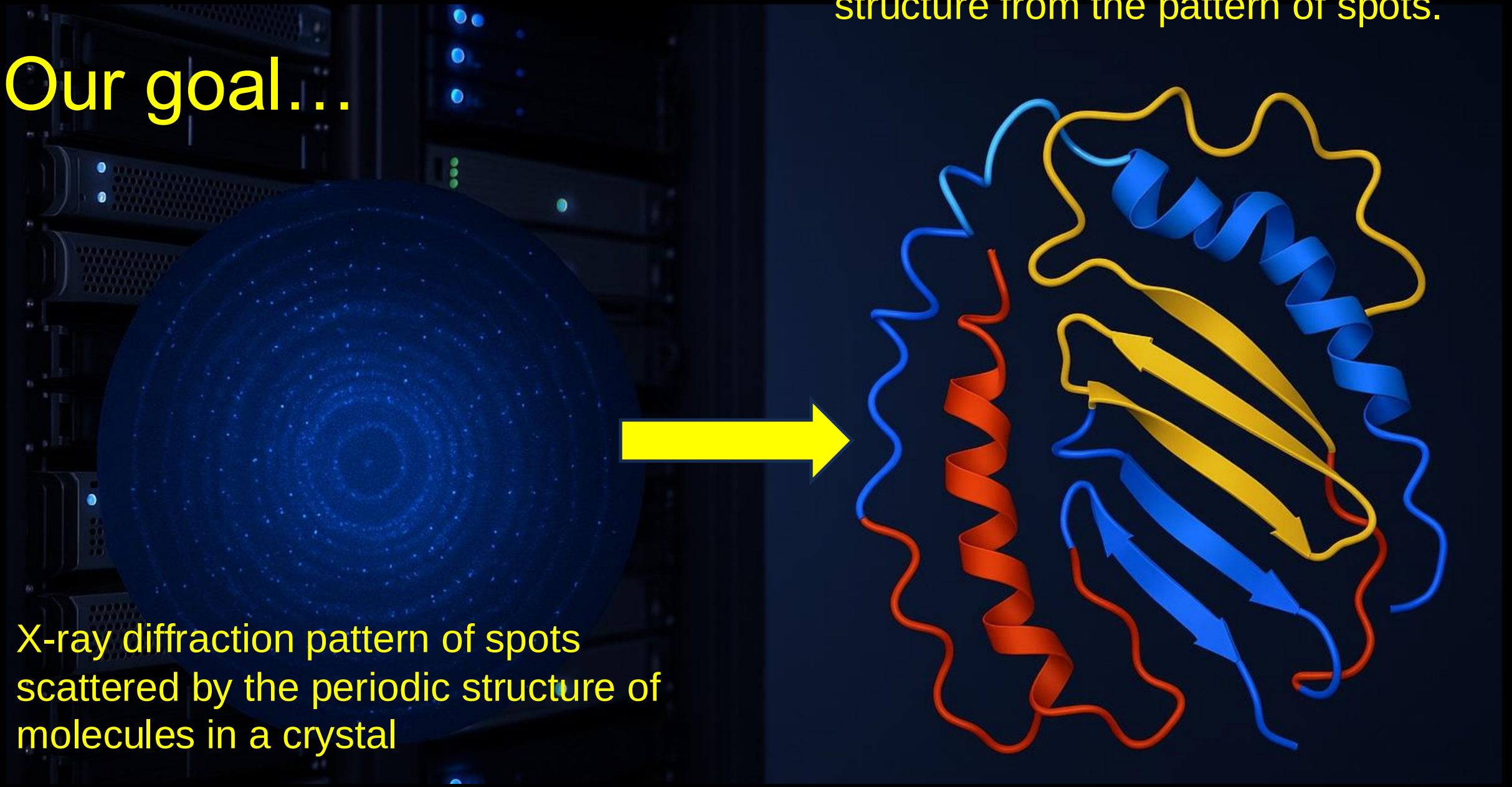
Crystallography



Crystallography

Our goal…

Computationally Inferred molecular structure from the pattern of spots.



X-ray diffraction pattern of spots scattered by the periodic structure of molecules in a crystal

Collaborative Computational Project Number 4

A collection of programs designed to help crystallographers determine the 3D structures of biological macromolecules (proteins, DNA, etc.) from X-ray diffraction data.

It’s been under active development since the 1970s, primarily in the UK, with global contributions from structural biology labs.

Collaborative Computational Project Number 4


```
vanilla@hopper:~$ cd ~/workshops/crystallography

vanilla@hopper:crystallography$ tree
.
├── data
│   └── thaumatin.pdb
└── slurm
    ├── integrate_xray_images.slurm
    └── solve_structure.slurm
```

Collaborative Computational Project Number 4


```
vanilla@hopper:~$ module spider ccp4

----------------------------------------------------------------------------------------------------------------
  ccp4: ccp4/9.0.006
----------------------------------------------------------------------------------------------------------------

    This module can be loaded directly: module load ccp4/9.0.006

    Help:
      an interactive molecular graphics program for macromolecular model building, validation, and visualization, primarily used
in crystallography and cryo-EM.
```

Collaborative Computational Project Number 4


```
vanilla@hopper:~$ module spider ccp4

----------------------------
  ccp4: ccp4/9.0.006
----------------------------

    This module can b

    Help:
      an interactive molecular graphics program for macromolecular model building, validation, and visualization, primarily used
in crystallography and cryo-EM.
```

Search for software by name
(if not installed ask us to install it)

Collaborative Computational Project Number 4


```
vanilla@hopper:~$ module spider ccp4

----------------------------------------------------------------------------------------------------
  ccp4: ccp4/9.0.006
----------------------------------------------------------------------------------------------------

    This module can be lo  led directly: module load ccp4/9.0.006

    H
    a                                                                    , validation, and visualization, primarily used
in c
```

Version of the software Sumaya installed for you

Collaborative Computational Project Number 4


```
vanilla@hopper:~$ module load ccp4
Executing ccp4 environment script: /opt/local/ccp4/ccp4-9/bin/ccp4.setup-sh
Environment variables set by this script persist after the module is unloaded.
You are on the hopper head node. Be sure to run ccp4 tools on a compute node, e.g. "srun --x11 --pty coot”
```

Collaborative Computational Project Number 4


```
vanilla@hopper:~$ module load ccp4
Executing ccp4 environment script: /opt/local/ccp4/ccp4-9/bin/ccp4.setup-sh
Environment variables set by this script persist after the module is unloaded.
You are on the hopper head node. Be sure to run ccp4 tools on a compute node, e.g. "srun --x11 --pty coot"
```

Reminder to run these tools on the compute nodes with srun or sbatch

Collaborative Computational Project Number 4


```
vanilla@hopper:~$ module load ccp4
Executing ccp4 environment script: /opt/local/ccp4/ccp4-9/bin/ccp4.setup-sh
Environment variables set by this script persist after the module is unloaded.
You are on the hopper head node. Be sure to run ccp4 tools on a compute node, e.g. "srun --x11 --pty coot”
```

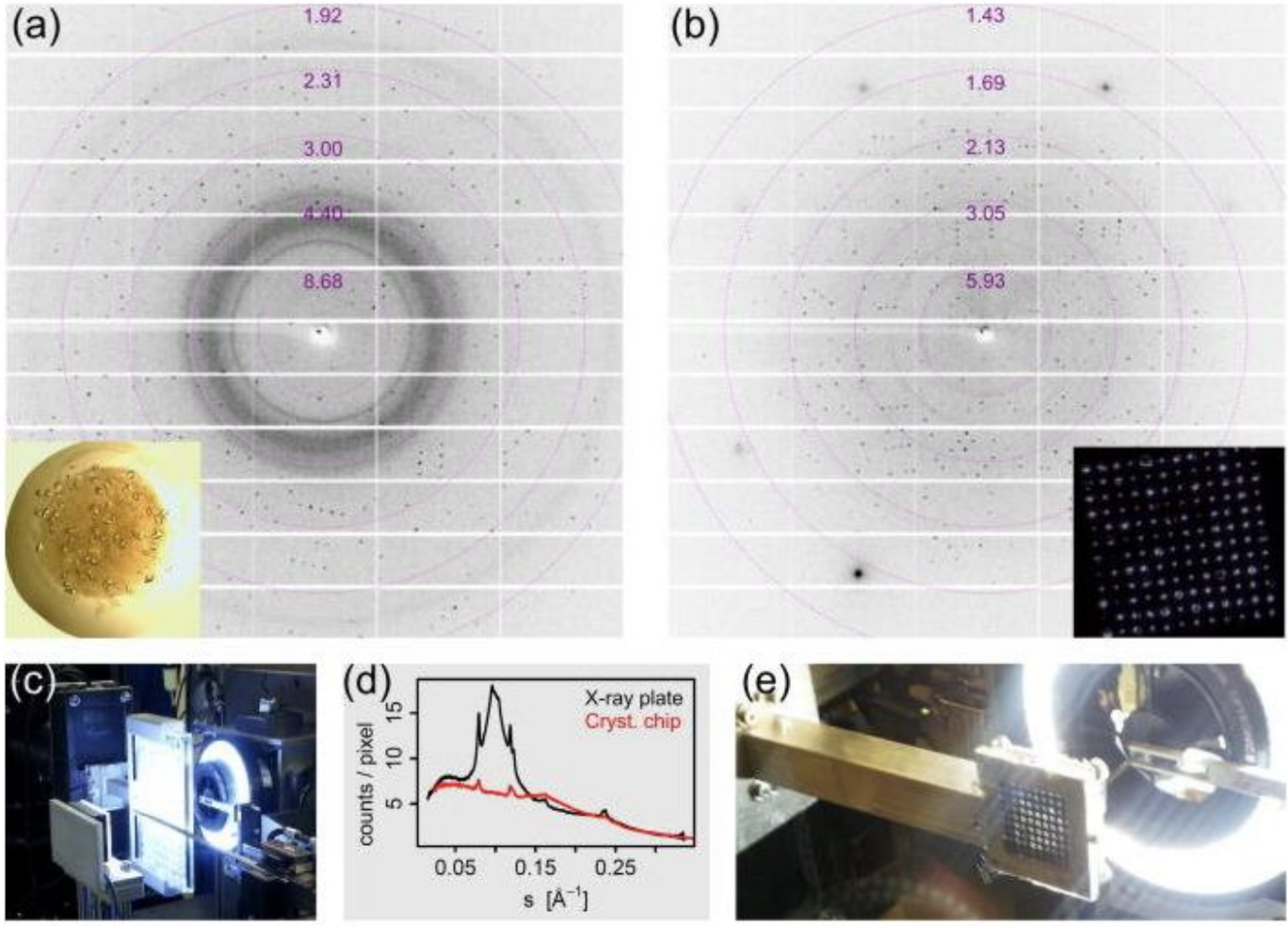
--x11 allows us to run graphical programs like coot. See the video tutorial on X11 forwarding in in this playlist for more information.

Image set for Thaumatin

```
vanilla@hopper: crystallography $ ls –lh
/projects/shared/workshops/crystallography/vmxi_thaumatin/
total 514M
-rw-r--r-- 1 mfricke users 512M Mar 25 22:13 image_15799_data_000001.h5
-rw-r--r-- 1 mfricke users  44K Mar 25 22:07 image_15799_master.h5
-rw-r--r-- 1 mfricke users 119K Mar 25 22:07 image_15799_meta.hdf5
-rw-r--r-- 1 mfricke users  49K Mar 25 22:07 image_15799.nxs
```

Katemfe fruit (*Thaumatococcus daniellii*), a tropical West African plant known for producing **thaumatin**—a protein that's thousands of times sweeter than sugar.

X-ray diffraction pattern of Crystalised Thaumatin



(a) X-ray diffraction pattern of a 40–50 µm T. daniellii thaumatin crystal located on a commercially available 96-well X-ray plate. The inset shows a photograph of the 1 µl crystallization droplet on the plate. (b) X-ray diffraction pattern of a 40–50 µm thaumatin crystal mounted on a crystallography chip with 50 µm sized features. The inset is a photograph of one compartment with 144 features at prescribed positions. The photograph in (c) shows the X-ray plate mounted at I24, Diamond Light Source. (d) Background scatter, radially averaged, as function of $s = \sin\theta/\lambda$. The photograph in (e) shows the crystallography chip mounted in its holder at beamline I24.

Mueller, C. & Marx, Alexander & Epp, S. & Zhong, Yinpeng & Kuo, A. & Balo, Aidin & Soman, J. & Schotte, F. & Lemke, H. & Owen, R. & Pai, Emil & Pearson, Arwen & Olson, John & Anfinrud, Philip & Ernst, O. & Miller, R.. (2015). Fixed target matrix for femtosecond time-resolved and in situ serial micro-crystallography. Structural Dynamics. 2. 10.1063/1.4928706.

X-ray Databank

Image set for Thaumatin

```
vanilla@hopper: crystallography $ ls –lh
/projects/shared/workshops/crystallography/vmxi_thaumatin/
total 514M
-rw-r--r-- 1 mfricke users 512M Mar 25 22:13 image_15799_data_000001.h5
-rw-r--r-- 1 mfricke users  44K Mar 25 22:07 image_15799_master.h5
-rw-r--r-- 1 mfricke users 119K Mar 25 22:07 image_15799_meta.hdf5
-rw-r--r-- 1 mfricke users  49K Mar 25 22:07 image_15799.nxs
```

(PS I used this command to get the data, so you don't have to)

```
dials.data get -q vmxi_thaumatin
Downloading https://zenodo.org/record/2547566/files/image_15799.nxs
Downloading https://zenodo.org/record/2547566/files/image_15799_data_000001.h5
Downloading https://zenodo.org/record/2547566/files/image_15799_master.h5
Downloading https://zenodo.org/record/2547566/files/image_15799_meta.hdf
```

DIALS (Diffraction Integration for Advanced Light Sources)

Integration of Thaumatin Images

vanilla@hopper: crystallography $ cat slurm/integrate_xray_images.slurm

```
#!/bin/bash
#SBATCH --job-name xray_int
#SBATCH --partition debug
#SBATCH --output xray_int_%j.out
#SBATCH --error xray_int_%j.out
#SBATCH --time 0:05:00
#SBATCH --nodes 1
#SBATCH --ntasks-per-node 1
#SBATCH --cpus-per-task 8
#SBATCH --mem 8G
#SBATCH --mail-type ALL
#SBATCH --mail-user your.email@unm.edu

# Load CCP4 and configure environment
module load ccp4
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
cd $SLURM_SUBMIT_DIR

# Define input dataset path
INPUT_PATH=/projects/shared/workshops/crystallography/vmxi_thaumatin/image_15799_master.h5
OUTPUT_PATH=~/workshops/crystallography/data/integration_output


# Create output directory
mkdir -p $OUTPUT_PATH
cd $OUTPUT_PATH


# Run xia2 with input and working directory
xia2 pipeline=dials \
    image=$INPUT_PATH \
    multiprocessing.mode=serial \
    multiprocessing.nproc=$SLURM_CPUS_PER_TASK \
    trust_beam_centre=False \
    read_all_image_headers=False
```

Integration of Thaumatin Images

```
vanilla@hopper: crystallography $ cat slurm/integrate_xray_images.slurm
#!/bin/bash
#SBATCH --job-name xray_int
#SBATCH --partition debug
#SBATCH --output xray_int_%j.out
#SBATCH --error xray_int_%j.out
#SBATCH --time 0:05:00
#SBATCH --nodes 1
#SBATCH --ntasks-per-node 1
#SBATCH --cpus-per-task 8
#SBATCH --mem 8G
#SBATCH --mail-type ALL
#SBATCH --mail-user your.email@unm.edu
```

Integration of Thaumatin Images

```
vanilla@hopper: crystallography $ cat slurm/integrate_xray_images.slurm

...

# Load CCP4 and configure environment
module load ccp4
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
cd $SLURM_SUBMIT_DIR
```

Integration of Thaumatin Images

```
vanilla@hopper: crystallography $ cat slurm/integrate_xray_images.slurm

...
# Define input dataset path
INPUT_PATH=/projects/shared/workshops/crystallography/vmxi_thaumatin/image_15799_master.h5
OUTPUT_PATH=~/workshops/crystallography/data/integration_output



# Create output directory
mkdir -p $OUTPUT_PATH
cd $OUTPUT_PATH
```

Integration of Thaumatin Images

vanilla@hopper: crystallography $ cat slurm/integrate_xray_images.slurm

```
...
# Run xia2 with input and working directory
xia2 pipeline=dials \
    image=$INPUT_PATH \
    multiprocessing.mode=serial \
    multiprocessing.nproc=$SLURM_CPUS_PER_TASK \
    trust_beam_centre=False \
    read_all_image_headers=False
```

xia2 is data reduction pipeline that wraps around **DIALS**

Integration of Thaumatin Images

```
vanilla@hopper: crystallography $ cat slurm/integrate_xray_images.slurm

...
# Run xia2 with input and working directory
xia2 pipeline=dials \
    image=$INPUT_PATH \
    multiprocessing.mode=serial \
    multiprocessing.nproc=$SLURM_CPUS_PER_TASK \
    trust_beam_centre=False \
    read_all_image_headers=False
```

This BASH variable contains the number of CPUs Slurm allocated

Integration of Thaumatin Images

```
vanilla@hopper: crystallography $ sbatch slurm/integrate_xray_images.slurm
sbatch: Account not specified in script or ~/.default_slurm_account, using latest project
Submitted batch job 3687646


vanilla@hopper: crystallography $ watch squeue --me
```

1) Submit the batch job to the Slurm scheduler
2) Monitor the progress of your job with squeue --me

Integration of Thaumatin Images

```
Every 2.0s: squeue --me hopper: Wed Mar 26 10:49:35 2025

JOBID   PARTITION     NAME     USER     ST TIME  NODES   NODELIST(REASON)
3687646 debug       xray_int vanilla  R  1:24     1   hopper011
```

Check the job status every 2 seconds

Once your job is in the R state check the name of the node on which it's running

Integration of Thaumatin Images

```
Every 2.0s: squeue --me hopper: Wed Mar 26 10:49:35 2025

JOBID   PARTITION     NAME     USER     ST TIME  NODES   NODELIST(REASON)
3687646 debug       xray_int vanilla  R  1:24     1   hopper011
```

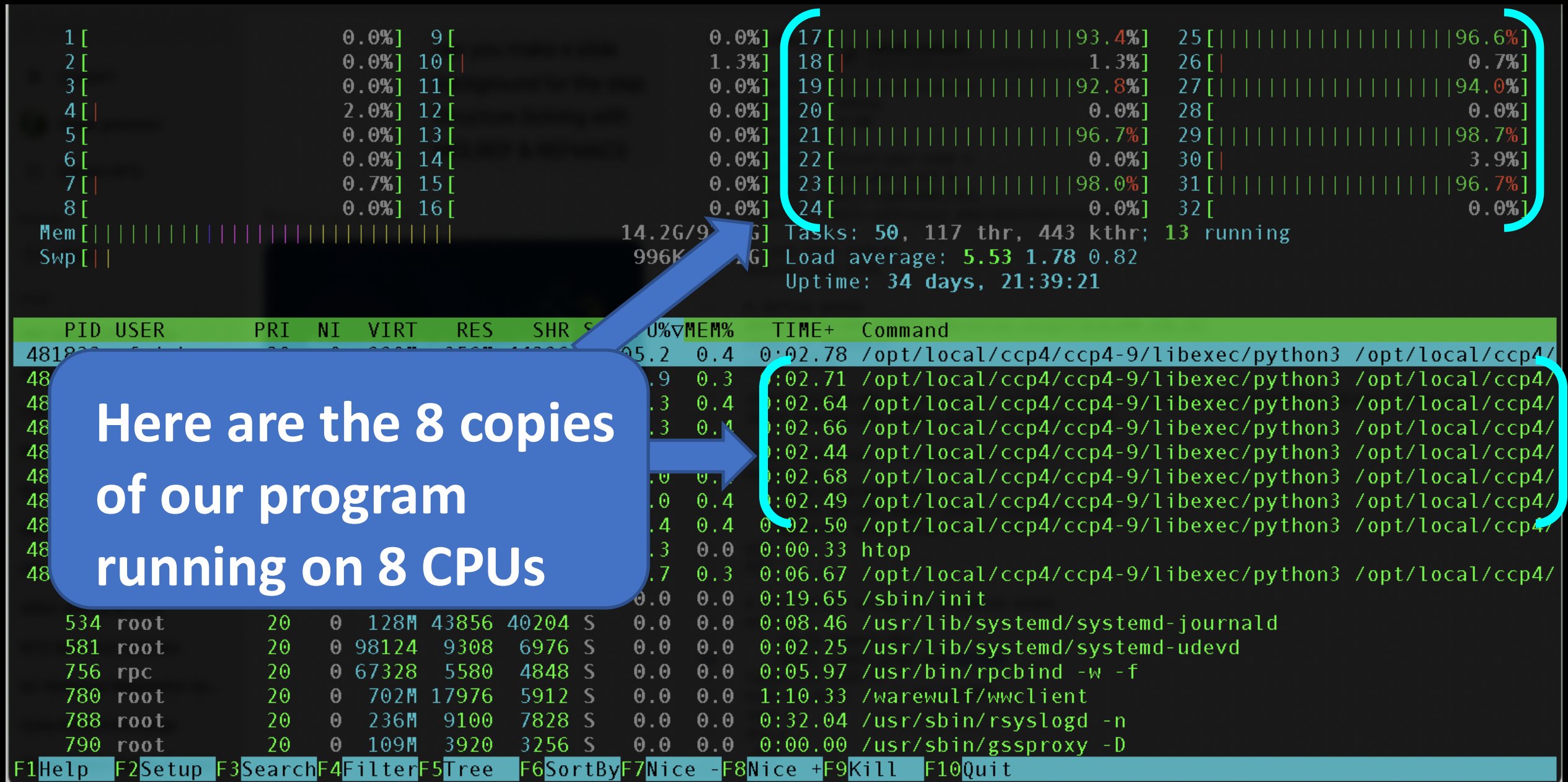
Check the job status every 2 seconds

Once your job is in the R state check the name of the node on which it's running

```
vanilla@hopper: crystallography $ ssh hopper011
vanilla@hopper011: crystallography $ htop
```

```
    1[                     0.0%]   9[                     0.0%]  17[||||||||||||||||||||93.4%]  25[||||||||||||||||||||96.6%]
    2[                     0.0%]  10[|                    1.3%]  18[|                    1.3%]  26[|                    0.7%]
    3[                     0.0%]  11[                     0.0%]  19[||||||||||||||||||||92.8%]  27[||||||||||||||||||||94.0%]
    4[|                    2.0%]  12[                     0.0%]  20[                     0.0%]  28[                     0.0%]
    5[                     0.0%]  13[                     0.0%]  21[||||||||||||||||||||96.7%]  29[||||||||||||||||||||98.7%]
    6[                     0.0%]  14[                     0.0%]  22[                     0.0%]  30[|                    3.9%]
    7[|                    0.7%]  15[                     0.0%]  23[||||||||||||||||||||98.0%]  31[||||||||||||||||||||96.7%]
    8[                     0.0%]  16[                     0.0%]  24[                     0.0%]  32[                     0.0%]
  Mem[|||||||||||||||||||||||||||||         14.2G/92.9G]  Tasks: 50, 117 thr, 443 kthr; 13 running
  Swp[||                                    996K/85.2G]  Load average: 5.53 1.78 0.82
                                                         Uptime: 34 days, 21:39:21

    PID USER       PRI  NI  VIRT   RES   SHR S  CPU%▽MEM%   TIME+  Command
 481822 mfricke     20   0  990M  356M 44296 R 105.2  0.4  0:02.78 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
 481795 mfricke     20   0  964M  331M 44168 R 101.9  0.3  0:02.71 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
 481823 mfricke     20   0 1003M  369M 44296 R  97.3  0.4  0:02.64 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
 481797 mfricke     20   0  990M  356M 44292 R  93.3  0.4  0:02.66 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
 481806 mfricke     20   0  991M  358M 44296 R  92.7  0.4  0:02.44 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
 481796 mfricke     20   0  990M  356M 44228 R  92.0  0.4  0:02.68 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
 481814 mfricke     20   0  990M  356M 44296 R  90.0  0.4  0:02.49 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
 481798 mfricke     20   0  990M  356M 44292 R  87.4  0.4  0:02.50 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
 481623 mfricke     20   0 35964  4952  3488 R   1.3  0.0  0:00.33 htop
 481713 mfricke     20   0  442M  330M  101M S   0.7  0.3  0:06.67 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
      1 root        20   0  232M 11612  8660 S   0.0  0.0  0:19.65 /sbin/init
    534 root        20   0  128M 43856 40204 S   0.0  0.0  0:08.46 /usr/lib/systemd/systemd-journald
    581 root        20   0 98124  9308  6976 S   0.0  0.0  0:02.25 /usr/lib/systemd/systemd-udevd
    756 rpc         20   0 67328  5580  4848 S   0.0  0.0  0:05.97 /usr/bin/rpcbind -w -f
    780 root        20   0  702M 17976  5912 S   0.0  0.0  1:10.33 /warewulf/wwclient
    788 root        20   0  236M  9100  7828 S   0.0  0.0  0:32.04 /usr/sbin/rsyslogd -n
    790 root        20   0  109M  3920  3256 S   0.0  0.0  0:00.00 /usr/sbin/gssproxy -D
F1Help  F2Setup  F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

vanilla@hopper011: crystallography $ htop



Here are the 8 copies of our program running on 8 CPUs

vanilla@hopper011: crystallography $ htop

```
  1[                      0.0%]   9[                      0.0%]  17[||||||||||||||||||93.4%]  25[||||||||||||||||||96.6%]
  2[                      0.0%]  10[|                     1.3%]  18[|                    1.3%]  26[|                    0.7%]
  3[                      0.0%]  11[                      0.0%]  19[||||||||||||||||||92.8%]  27[||||||||||||||||||94.0%]
  4[|                     2.0%]  12[                      0.0%]  20[                    0.0%]  28[                    0.0%]
  5[                      0.0%]  13[                      0.0%]  21[||||||||||||||||||96.7%]  29[||||||||||||||||||98.7%]
  6[                      0.0%]  14[                      0.0%]  22[                    0.0%]  30[|                    3.9%]
  7[|                     0.7%]  15[                      0.0%]  23[||||||||||||||||||98.0%]  31[||||||||||||||||||96.7%]
  8[                      0.0%]  16[                      0.0%]  24[                    0.0%]  32[                    0.0%]
Mem[||||||||||||||||||||||||||||||||||||14.2G/92.9G] Tasks: 50, 117 thr, 443 kthr; 13 running
Swp[||                                    996K/85.2G] Load average: 5.53 1.78 0.82
                                                      Uptime: 34 days, 21:39:21

    PID USER       PRI  NI  VIRT   RES   SHR S  CPU%▽MEM%   TIME+  Command
 481                                                 05.2  0.4  0:02.78 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
 48                                                   .9  0.3  0:02.71 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
 48                                                   .3  0.4  0:02.64 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
 48                                                   .3  0.4  0:02.66 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
 48                                                   .7  0.4  0:02.44 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
 48                                                   .0  0.4  0:02.68 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
 48                                                   .0  0.4  0:02.49 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
 48                                                   .4  0.4  0:02.50 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
 48                                                   .3  0.0  0:00.33 htop
 48                                                   .7  0.3  0:06.67 /opt/local/ccp4/ccp4-9/libexec/python3 /opt/local/ccp4/
                                                      0.0  0.0  0:19.65 /sbin/init
    534 root       20   0  128M 43856 40204 S   0.0  0.0  0:08.46 /usr/lib/systemd/systemd-journald
    581 root       20   0 98124  9308  6976 S   0.0  0.0  0:02.25 /usr/lib/systemd/systemd-udevd
    756 rpc        20   0 67328  5580  4848 S   0.0  0.0  0:05.97 /usr/bin/rpcbind -w -f
    780 root       20   0  702M 17976  5912 S   0.0  0.0  1:10.33 /warewulf/wwclient
    788 root       20   0  236M  9100  7828 S   0.0  0.0  0:32.04 /usr/sbin/rsyslogd -n
    790 root       20   0  109M  3920  3256 S   0.0  0.0  0:00.00 /usr/sbin/gssproxy -D
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

‘q’ when you want to exit.

vanilla@hopper011: crystallography $ exit

Enter exit to log out of the compute node.

Be careful to not log out of the head node.

If you your prompt is shows hopper don’t type exit

vanilla@hopper: crystallography $ watch squeue --me

Check on our job status again

Integration of Thaumatin Images

Integration of Thaumatin Images

```
Every 2.0s: squeue --me hopper: Wed Mar 26 10:49:35 2025

JOBID   PARTITION     NAME     USER     ST TIME  NODES   NODELIST(REASON)
```

When your job finishes exit watch with *ctrl-c* or ‘q’

Integration of Thaumatin Images

```
vanilla@hopper: crystallography $ ls
data  slurm  xray_int_{your job ID}.out
```

You will have a new slurm output file.

Integration of Thaumatin Images

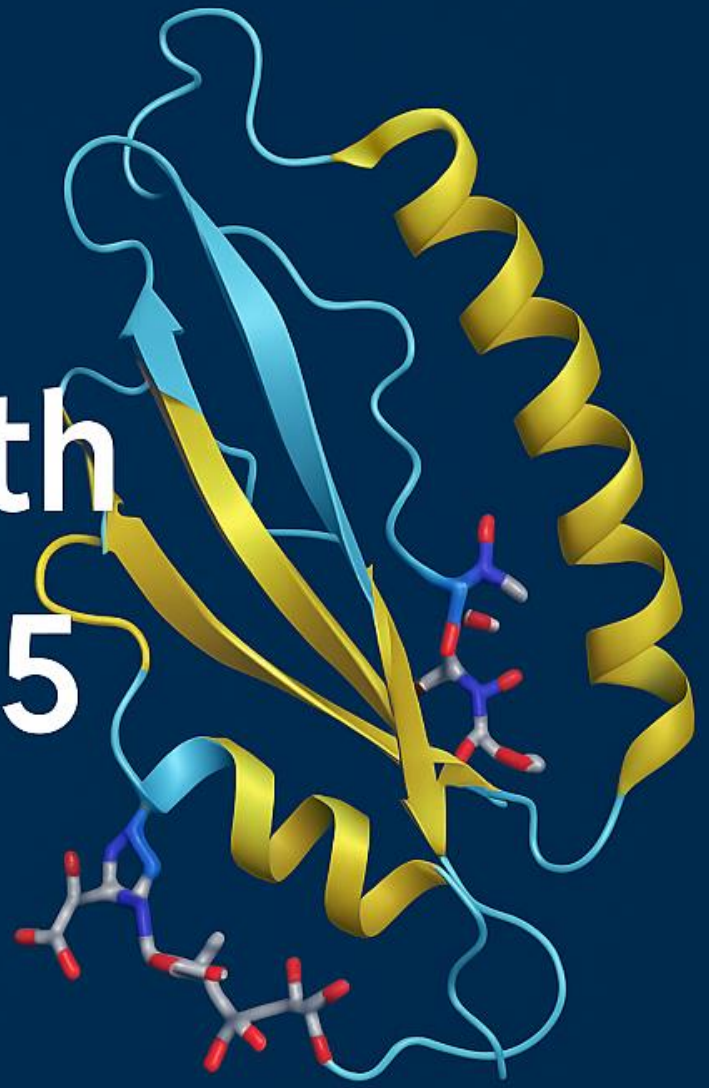
vanilla@hopper: crystallography $ cat xray_int_{your job ID}.out

```
Unit cell (with estimated std devs):
58.6134(2) 58.6134(2) 151.3821(8)
90.0       90.0       90.0
mtz_unmerged format:
Scaled reflections (NATIVE): /users/vanilla/workshops/crystallography/data/integration_output/DataFiles/AUTOMATIC_DEFAULT_scaled_unmerged.mtz
mtz format:
Scaled reflections: /users/vanilla/workshops/crystallography/data/integration_output/DataFiles/AUTOMATIC_DEFAULT_free.mtz
Processing took 00h 02m 59s
XIA2 used... ccp4, dials, dials.scale, xia2
Here are the appropriate citations (BIBTeX in xia2-citations.bib.)
Beilsten-Edmands, J. et al. (2020) Acta Cryst. D76, 385-399.
Winn, M. D. et al. (2011) Acta Cryst. D67, 235-242.
Winter, G. (2010) J. Appl. Cryst. 43, 186-190.
Winter, G. et al. (2018) Acta Cryst. D74, 85-97.
Status: normal termination
```

Integration of Thaumatin Images

```
vanilla@hopper: crystallography $ tree data
data
├── integration_output
│   ├── automatic.xinfo
│   ├── DataFiles
│   │   ├── AUTOMATIC_DEFAULT_free.mtz
│   │   ├── AUTOMATIC_DEFAULT_NATIVE_SWEEP1.expt
│   │   ├── AUTOMATIC_DEFAULT_NATIVE_SWEEP1_INTEGRATE.mtz
│   │   ├── AUTOMATIC_DEFAULT_NATIVE_SWEEP1.refl
│   │   ├── AUTOMATIC_DEFAULT_scaled.expt
│   │   ├── AUTOMATIC_DEFAULT_scaled.mtz
│   │   ├── AUTOMATIC_DEFAULT_scaled.refl
│   │   ├── AUTOMATIC_DEFAULT_scaled.sca
│   │   ├── AUTOMATIC_DEFAULT_scaled_unmerged.mtz
│   │   ├── AUTOMATIC_DEFAULT_scaled_unmerged.sca
│   │   ├── xia2.cif
│   │   └── xia2.mmcif.bz2
│   ├── DEFAULT
│   │   ├── NATIVE
│   │   │   └── SWEEP1
12 directories, 154 files
```

Structure Solving with MOLREP & REFMAC5



molrep

```
+-------------------------------------+
|                                     |
|           --- MOLREP ---            |
|      /Vers 11.9.02; 28.02.2022/     |
|                                     |
+-------------------------------------+
```

Use experimental data and model to place the molecule in the unit cell.

molrep HKLIN data.mtz MODEL model.pdb
labin F=FP SIGF=SIGFP
mode MR_AUTO
nmon 1
doc y
_end

You can run this interactively and enter the commands you want to run…

Solving for Thaumatin Structure

```
molrep \
  HKLIN $INT_PATH/DataFiles/AUTOMATIC_DEFAULT_free.mtz \
  MODEL $PDB_PATH << EOF
labin F=FP SIGF=SIGFP
mode MR_AUTO
nmon 1
doc y
_end
EOF
```

… but we will using the batch system not interactive sessions. So, we will send the commands to molrep in a script by wrapping them in an EOF (same with refmac5)

refmac5

```
##############################################################
##############################################################
##############################################################
### CCP4 9.0.006: Refmac          version 5.8.0430 : 05/20/24##
##############################################################
User: vanilla  Run date: 26/ 3/2025 Run time: 11:41:25
```

REFMAC5 refines atomic models in crystallography, improving fit to X-ray data while preserving chemical geometry.

Solving for Thaumatin Structure

vanilla@hopper: crystallography $ less slurm/solve_structure.slurm

```
#!/bin/bash
#SBATCH --job-name solve_structure
#SBATCH --partition debug
#SBATCH --output solve_structure_%j.out
#SBATCH --error solve_structure_%j.out
#SBATCH --time 0:05:00
#SBATCH --nodes 1
#SBATCH --ntasks-per-node 1
#SBATCH --cpus-per-task 8
#SBATCH --mem 8G
#SBATCH --mail-type ALL
#SBATCH --mail-user your.email@university.edu
# Load CCP4 environment
module load ccp4
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
cd $SLURM_SUBMIT_DIR
INPUT_PATH=~/workshops/crystallography/data
OUTPUT_PATH=$INPUT_PATH/solver_output

# Path to the output of the image integrator (xia2)
INT_PATH=$INPUT_PATH/integration_output
```

Solving for Thaumatin Structure

```
# Create output directory for this job
mkdir -p $OUTPUT_PATH
cd $OUTPUT_PATH

# Molecular replacement using molrep
molrep \
  HKLIN $INT_PATH/DataFiles/AUTOMATIC_DEFAULT_free.mtz \
  MODEL $PDB_PATH << EOF
labin F=FP SIGF=SIGFP
mode MR_AUTO
nmon 1
doc y
_end
EOF
```

```
# Refinement using refmac5
refmac5 \
    hklin $INT_PATH/DataFiles/AUTOMATIC_DEFAULT_free.mtz \
    hklout $OUTPUT_PATH/refined.mtz \
    xyzin $OUTPUT_PATH/molrep.pdb \
    xyzout $OUTPUT_PATH/refined.pdb << EOF >
$OUTPUT_PATH/refmac.log
make hydrogen ALL
ncyc 10
weight AUTO
monitor MEDIUM
scales PEAK
labout  FP=F SIGFP=SIGF FREE=FreeR_flag
end
EOF
```

Solving for Thaumatin Structure

```
# Create output directory for this job
mkdir -p $OUTPUT_PATH
cd $OUTPUT_PATH

# Molecular replacement using molrep
molrep \
  HKLIN $INT_PATH/DataFiles/AUTOMATIC_DEFAULT_free.mtz \
  MODEL $PDB_PATH << EOF
labin F=FP SIGF=SIGFP
mode MR_AUTO
nmon 1
doc y
_end
EOF
```

This PDB gives us a starting point for the refinement

Solving for Thaumatin Structure

```
# Refinement using refmac5
refmac5 \
    hklin $INT_PATH/DataFiles/AUTOMATIC_DEFAULT_free.mtz \
    hklout $OUTPUT_PATH/refined.mtz \
    xyzin $OUTPUT_PATH/molrep.pdb \
    xyzout $OUTPUT_PATH/refined.pdb << EOF > $OUTPUT_PATH/refmac.log
make hydrogen ALL
ncyc 10
weight AUTO
monitor MEDIUM
scales PEAK
labout  FP=F SIGFP=SIGF FREE=FreeR_flag
end
EOF
```

Solving for Thaumatin Structure

```
vanilla@hopper: crystallography $ sbatch slurm/solve_structure.slurm
vanilla@hopper: crystallography $ watch squeue --me

Every 2.0s: squeue --me                    hopper: Wed Mar 26 11:54:15 2025

 JOBID     PARTITION    NAME     USER    ST TIME  NODES NODELIST(REASON)
 3687680    debug       solve_st  vanilla  R  1:29     1 hopper011
```

Solving for Thaumatin Structure

```
vanilla@hopper:crystallography $ ls
data  slurm  solve_structure_3687680.out  xray_int_3687646.out

vanilla@hopper:crystallography $ cat solve_structure_3687680.out
```

--- Summary (V2) ---

RF	TF	theta	phi	chi	tx	ty	tz	TF/sg	Tcoef	Score
1	1	1	0.00	0.00	0.09	0.212	0.413	0.210	30.04	1.000	0.72199
2	13	1	11.43	88.85	155.77	0.330	0.425	0.339	3.68	1.000	0.29457
3	12	2	13.34	91.75	134.42	0.010	0.301	0.066	3.26	1.000	0.29332
4	4	2	23.97	87.94	129.70	0.407	0.112	0.230	3.39	1.000	0.28992
5	14	2	8.49	70.24	140.71	0.378	0.224	0.120	3.11	1.000	0.28702
6	7	1	19.89	51.50	179.63	0.241	0.071	0.398	4.11	1.000	0.27902
7	9	14	11.53	92.47	156.62	0.960	0.410	0.029	2.58	0.900	0.25990
8	10	12	42.11	44.11	156.36	0.863	0.392	0.497	2.81	0.900	0.25988
9	8	11	92.72	136.00	81.81	0.874	0.228	0.070	3.01	0.900	0.25872
10	5	13	23.90	86.48	129.03	0.375	0.062	0.379	2.86	0.900	0.25821
11	2	14	73.81	89.88	71.61	0.854	0.448	0.480	2.71	0.900	0.25792
12	3	13	12.31	81.24	142.53	0.043	0.294	0.326	2.59	0.900	0.25447
13	6	11	19.59	50.07	178.92	0.710	0.046	0.370	3.34	0.900	0.25107
14	11	8	19.08	48.47	178.54	0.982	0.339	0.178	3.23	0.900	0.24901

Summary table showing top molecular replacement solutions ranked by their rotation and translation functions, with associated orientation angles (theta, phi, chi), shifts (tx, ty, tz), signal-to-noise (TF/σ), and final placement scores.

Solving for Thaumatin Structure

```
vanilla@hopper:crystallography $ tree data

...

├── solver_output
│   ├── molrep.btc
│   ├── molrep.doc
│   ├── molrep_mtz.cif
│   ├── molrep.pdb
│   ├── molrep.xml
│   ├── refined.mmcif
│   ├── refined.mtz
│   ├── refined.pdb
│   └── refmac.log
└── thaumatin.pdb
```

Visualise the Thaumatin Structure

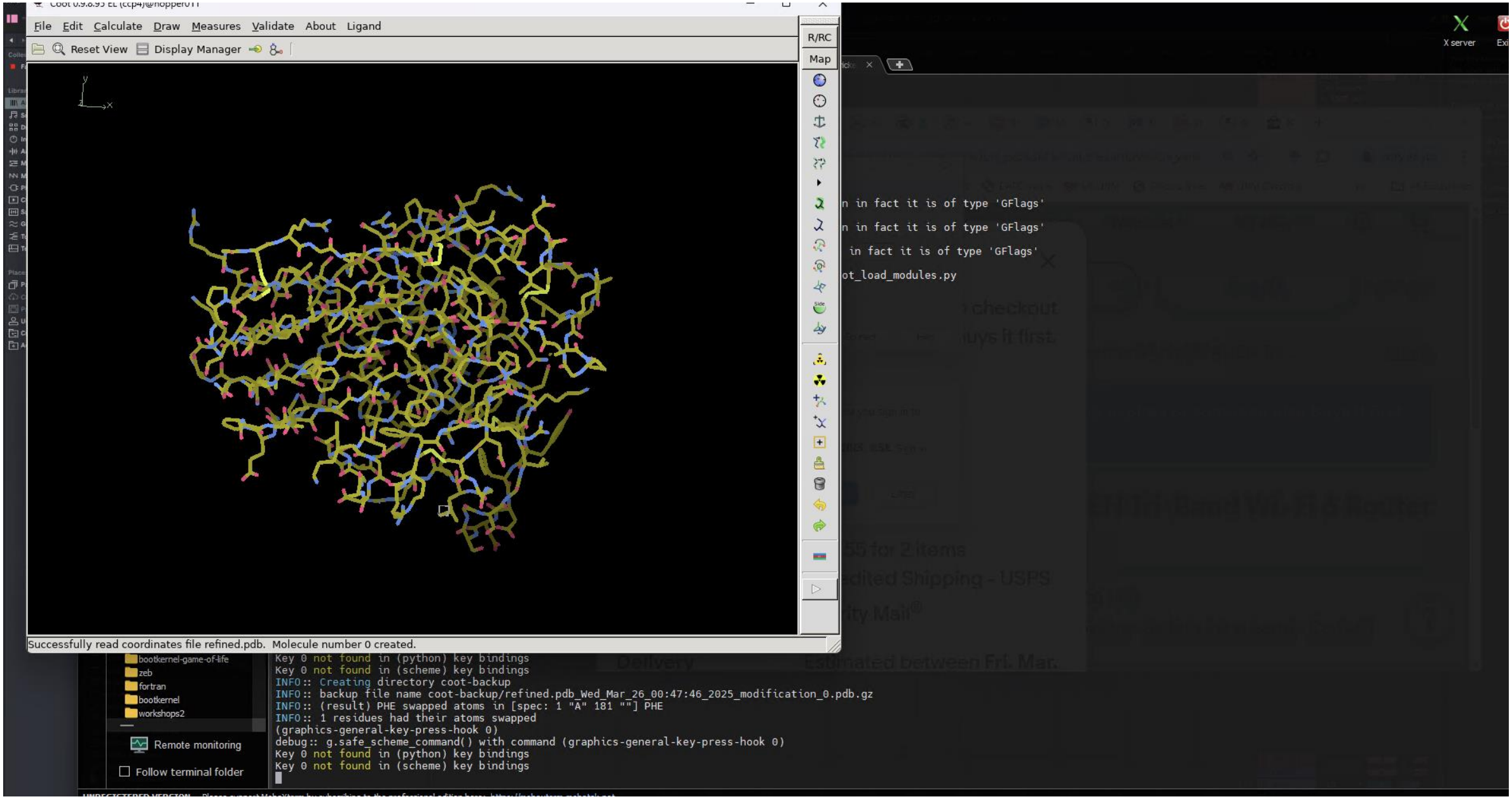
```
vanilla@hopper:crystallography $ exit


CARCWS-01:~ vanilla$ ssh –Y vanilla@hopper.alliance.unm.edu
```

-Y allows us to run graphical programs remotely. See the video tutorial on X11 forwarding in in this playlist for more information.

Visualise the Thaumatin Structure

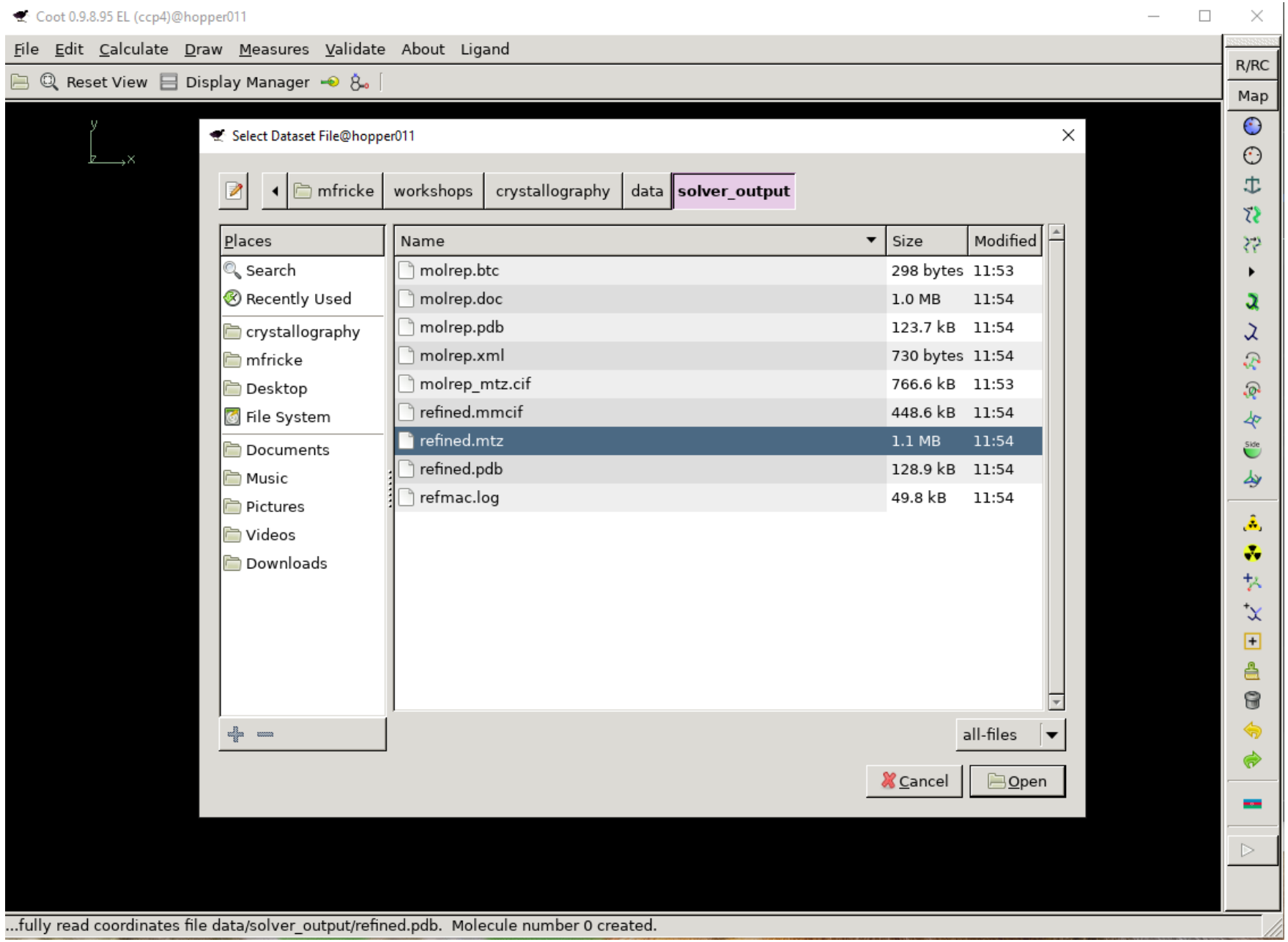
```
module load ccp4
cd ~workshops/crystallography
srun --partition debug --x11 --pty coot --pdb data/solver_output/refined.pdb
```

--x11 allows us to run graphical programs remotely. See the video tutorial on X11 forwarding in in this playlist for more information.

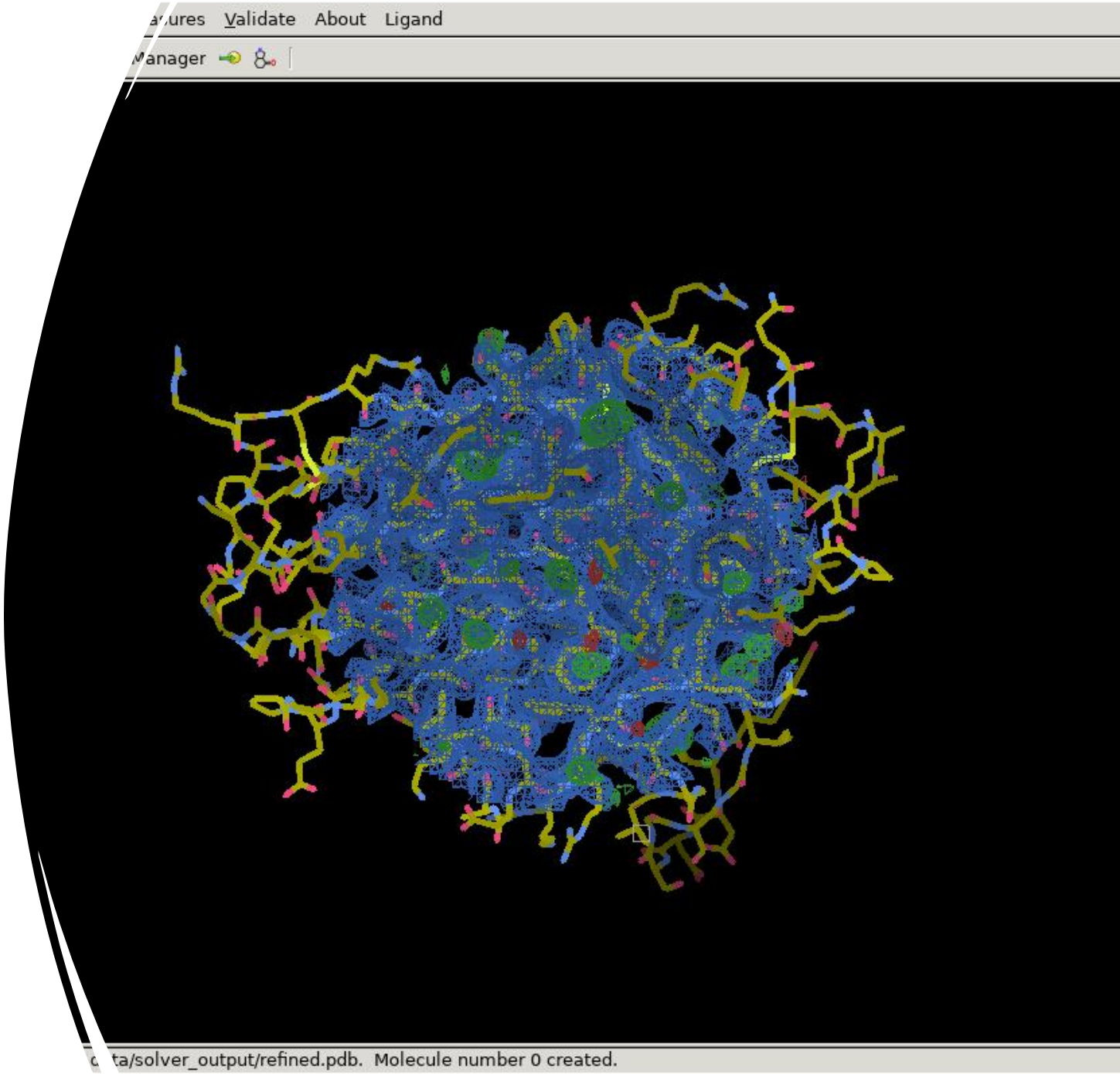


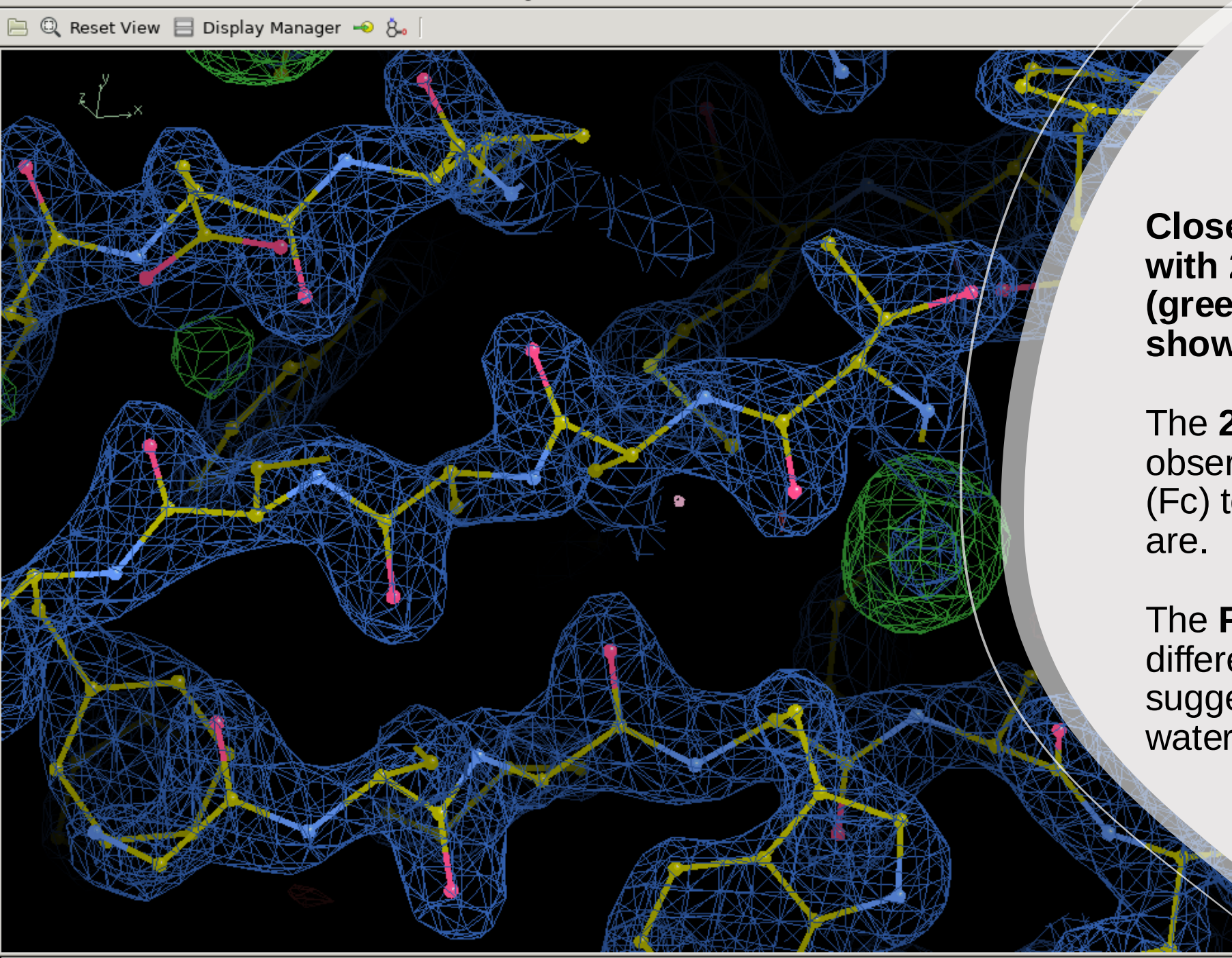
Refined thaumatin structure visualised in Coot following molecular replacement and refinement with MOLREP and REFMAC5.

Loading the refined structure and map in Coot: click 'File' → 'Auto Open MTZ' and select the appropriate .pdb and .mtz files from the solver_output directory.



Refined thaumatin model displayed with electron density in Coot, showing fit between atomic coordinates and experimental data.





Close-up of the refined structure with 2Fo–Fc (blue) and Fo–Fc (green) electron density maps, showing model fit.

The **2Fo–Fc map** combines observed data (Fo) and the model (Fc) to indicate where atoms likely are.

The **Fo–Fc map** highlights differences — green density suggests features like unmodelled water molecules.

Hardware Utilisation Performance

```
vanilla@hopper:crystallography $ ls
data  slurm  solve_structure_3687680.out  xray_int_3687646.out
vanilla@hopper:crystallography $ seff 3687680
Job ID: 3687680
Cluster: hopper
User/Group: mfricke/users
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 8
CPU Utilized: 00:00:42
CPU Efficiency: 12.50% of 00:05:36 core-walltime
Job Wall-clock time: 00:00:42
Memory Utilized: 207.62 MB
Memory Efficiency: 2.53% of 8.00 GB
```

How efficiently did our programs use the resources we allocated?
Not great. We should request fewer CPUs and less memory next time.

Hardware Utilisation Performance

```
vanilla@hopper:crystallography $ ls
data  slurm  solve_structure_3687680.out  xray_int_3687646.out
vanilla@hopper:crystallography $ seff 3687646
Job ID: 3687646
Cluster: hopper
User/Group: mfricke/users
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 8
CPU Utilized: 00:08:34
CPU Efficiency: 34.92% of 00:24:32 core-walltime
Job Wall-clock time: 00:03:04
Memory Utilized: 1.33 GB
Memory Efficiency: 16.58% of 8.00 GB
```

How efficiently did our programs use the resources we allocated?
Better than the solve job, but we should still request fewer CPUs and less RAM.

Useful Slurm Commands

squeue --me --long	shows information about jobs you submitted
squeue --me --start	shows when slurm expects your job to start
scancel jobid	cancels a job
scancel --u $USER	cancels all your jobs
sacct	shows your job history
seff jobid	shows how efficiently the hardware was used