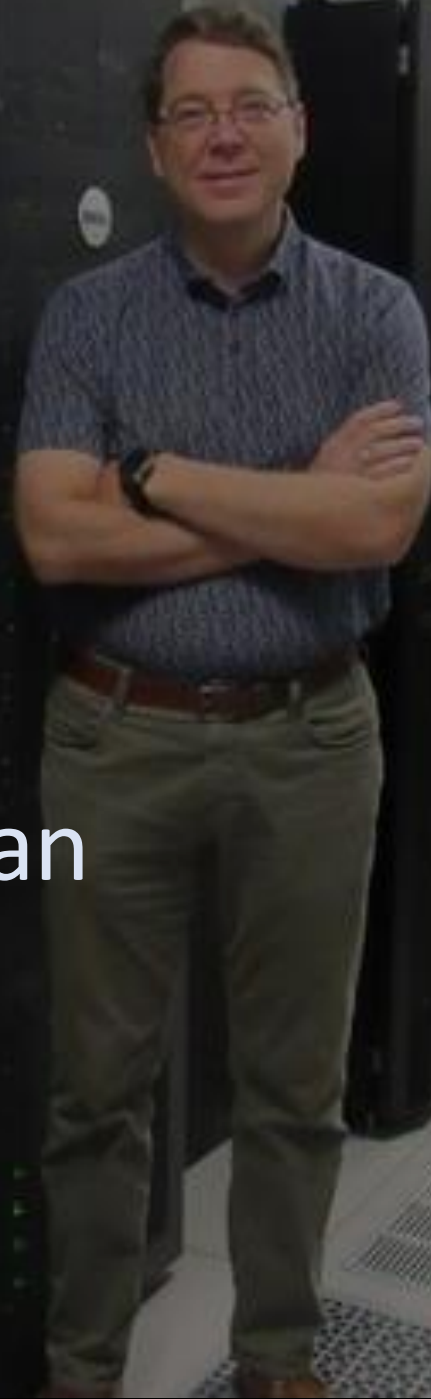


# Intermediate Level Introduction to Computing at CARC

1 hour version with Gaussian

Matthew Fricke

Version 0.1



# Goals

- 1) SLURM scheduler literacy
- 2) Example Gaussian
- We wont cover file transfer, storage systems, module system, or conda. (These are all covered in depth in the video tutorials)

# Logging into CARC



First login to the Linux **workstation** in front of you.

Use the password you chose when you created your account.

If you forgot your password go to:

<https://mokey.alliance.unm.edu/auth/forgotpw>

This is an “important step” so don’t let me move on until you have logged in

Read all updates

# Welcome to the Center for Advanced Research Computing

The UNM Center for Advanced Research Computing is the hub of computational research at UNM and one of the largest computing centers in the State of New Mexico. It is an interdisciplinary community that uses computational resources to create new research insights. The goal is to lead and grow the computational research community at UNM..

CARC provides not just the computing resources but also the expertise and support to help the university’s researchers. This service is available to faculty, staff, and student researchers free of charge through support from the UNM Office of the Vice President for Research.



- Getting started at CARC
- Good Neighbor Use Policy
- Video: Introduction to Computing at CARC tutorials

Get started at CARC

Create a help ticket

Downtime notices



ever...  
New users - Video: Introduction to Computing at CARC tutorials

**UNM launches search for the next director of the CARC**

**UNM awarded prestigious grant to lead national research on high-performance computing**

**Genome sequencing reveals a newly identified virus**

**Undergraduate researcher leverages CARC for molecular modeling breakthrough**



# CARC Video Tutorials: Logging in

Home

Shorts

Subscriptions

You

YouTube

Search

Q

Sign in

QuickBytes

by UNMCARC

Playlist • 19 videos • 8,136 views

▶ Play all

3

UNMCARC • 469 views • 1 year ago

4

Annual Project Review

UNMCARC • 205 views • 1 year ago

5

Logging into CARC Systems

UNMCARC • 1.3K views • 5 years ago

6

Slurm Job Scheduler

UNMCARC • 700 views • 10 months ago

7

Storage Systems

UNMCARC • 674 views • 4 years ago





# Logging into Easley



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

Should prompt you for a password...

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

Replace vanilla with your name (unless your last name is Ice)

# Logging into Hopper

Welcome to Easley

- ▶ By using CARC systems you agree to the CARC Good Neighbor Guide
- ▶ Tutorials and the Good Neighbor Guide are at

<https://goto.unm.edu/carc-start>

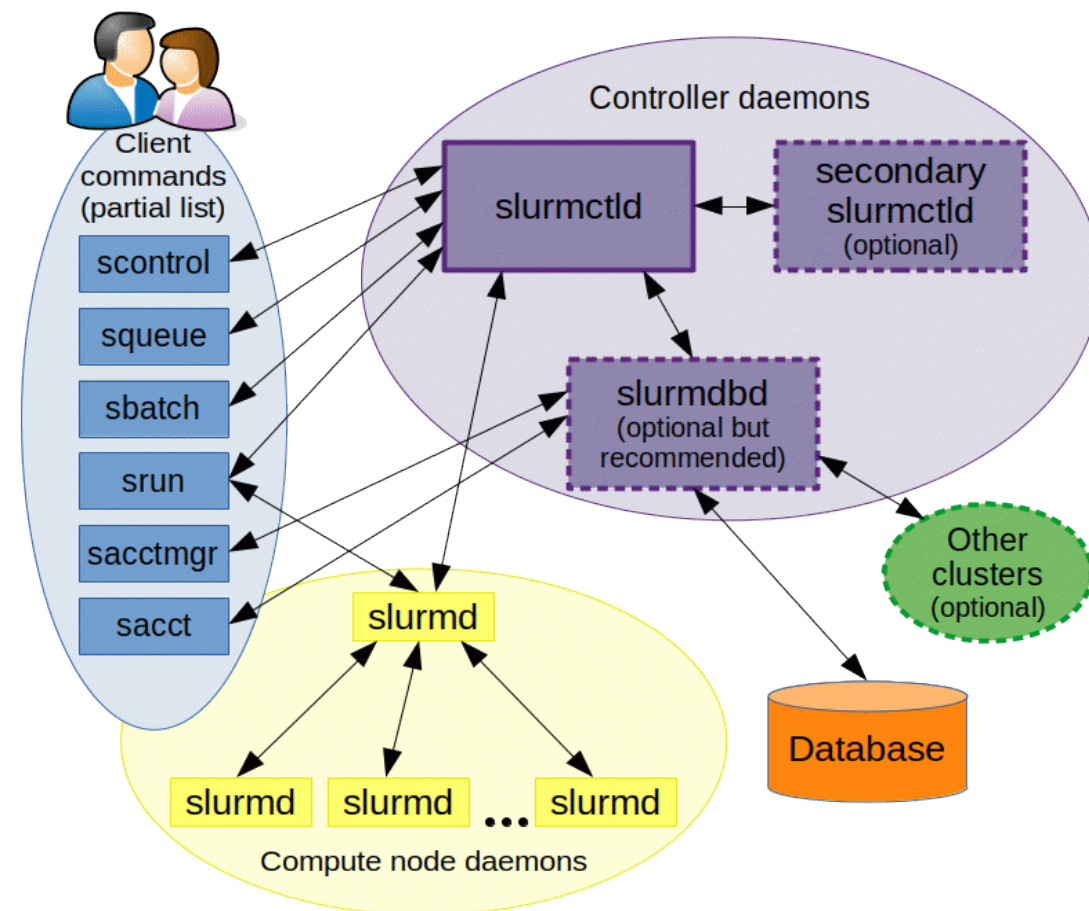
- ▶ Need help? Email: [help@carc.unm.edu](mailto:help@carc.unm.edu)

▶ Please include an acknowledgement of CARC in any publications that make use of computations performed here.

Last login: Tue Feb 10 17:03:07 2026 from 50.188.169.22

Warning: Your home directory (~) usage is 187G\*/100G. Please reduce your usage. Use the `ncdu` command to see details.

# Simple Linux Utility for Resource Management





VOTED **#1** SOFT DRINK OF THE 31<sup>ST</sup> CENTURY!



SLURPEE'S MAKEOVER

[vanilla@easley ~]\$ qgrok

| queues      | free | busy | offline | jobs | nodes | CPUs | idle_CPUS | GPUs | CPUs/node | memory/node | time_limit | CPU_limit | GPU_limit | RAM_limit |
|-------------|------|------|---------|------|-------|------|-----------|------|-----------|-------------|------------|-----------|-----------|-----------|
| general     | 3    | 31   | 14      | 17   | 48    | 3072 | 427       | 0    | 64        | 250G        | 2-00:00:00 | 512       | 0         | 2059472M  |
| bigmem      | 2    | 0    | 0       | 0    | 2     | 128  | 128       | 0    | 64        | 1.9T        | 2-00:00:00 | 64        | 0         | 2063765M  |
| h100        | 0    | 4    | 0       | 1    | 4     | 256  | 63        | 8    | 64        | 1006G       | 2-00:00:00 | 64        | 2         | 1031578M  |
| l40s        | 0    | 3    | 0       | 0    | 3     | 192  | 189       | 12   | 64        | 250G        | 2-00:00:00 | 96        | 3         | 171624M   |
| interactive | 0    | 7    | 0       | 0    | 7     | 448  | 252       | 20   | 64        | 250G-1006G  | 4:00:00    | 64        | 1         |           |
| debug       | 5    | 38   | 14      | 0    | 57    | 3648 | 807       | 20   | 64        | 250G-1006G  | 1:00:00    | 128       | 2         |           |
| scavenger   | 9    | 40   | 14      | 3    | 63    | 4032 | 1186      | 44   | 64        | 250G-1006G  | 2-00:00:00 | 4032      | 44        |           |
| liulab      | 4    | 2    | 0       | 1    | 6     | 384  | 379       | 24   | 64        | 250G        | 7-00:00:00 | 384       | 24        |           |
| totals:     | 5    | 38   | 14      | 18   | 57    | 3648 | 807       | 20   | 64        |             |            |           |           |           |

General:

For longer jobs

Debug:

For short small jobs to test code

```
[vanilla@easley ~]$ qgrok
```

| queues      | free | busy | offline | jobs | nodes | CPUs | idle_CPUS | GPUs | CPUs/node | memory/node | time_limit | CPU_limit | GPU_limit | RAM_limit |
|-------------|------|------|---------|------|-------|------|-----------|------|-----------|-------------|------------|-----------|-----------|-----------|
| general     | 3    | 31   | 14      | 17   | 48    | 3072 | 427       | 0    | 64        | 250G        | 2-00:00:00 | 512       | 0         | 2059472M  |
| bigmem      | 2    | 0    | 0       | 0    | 2     | 128  | 128       | 0    | 64        | 1.9T        | 2-00:00:00 | 64        | 0         | 2063765M  |
| h100        | 0    | 4    | 0       | 1    | 4     | 256  | 63        | 8    | 64        | 1006G       | 2-00:00:00 | 64        | 2         | 1031578M  |
| l40s        | 0    | 3    | 0       | 0    | 3     | 192  | 189       | 12   | 64        | 250G        | 2-00:00:00 | 96        | 3         | 171624M   |
| interactive | 0    | 7    | 0       | 0    | 7     | 448  | 252       | 20   | 64        | 250G-1006G  | 4:00:00    | 64        | 1         |           |
| debug       | 5    | 38   | 14      | 0    | 57    | 3648 | 807       | 20   | 64        | 250G-1006G  | 1:00:00    | 128       | 2         |           |
| scavenger   | 9    | 40   | 14      | 3    | 63    | 4032 | 1186      | 44   | 64        | 250G-1006G  | 2-00:00:00 | 4032      | 44        |           |
| liulab      | 4    | 2    | 0       | 1    | 6     | 384  | 379       | 24   | 64        | 250G        | 7-00:00:00 | 384       | 24        |           |
| totals:     | 5    | 38   | 14      | 18   | 57    | 3072 | 427       | 24   | 64        |             |            |           |           |           |

## “scavenger” partition

- Allows you to use idle nodes from any partition even special hardware
- May be interrupted at any time someone decides to use that special hardware

[vanilla@easley ~]\$ **quotas**

Home Directory (/users/vanilla):

quota: Cannot resolve mountpoint path /root/.spack: Permission denied

Disk quotas for user vanilla (uid 659):

| Filesystem        | space | quota | limit | grace | files | quota | limit | grace |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| chama:/home/homes | 1527M | 100G  | 200G  |       | 14913 | 4295m | 4295m |       |

-----

Centerwide user scratch (/carc/scratch/users/mfricke)

Quota information for storage pool Default (ID: 1):

| user/group |       | size       |             | chunk files |           |
|------------|-------|------------|-------------|-------------|-----------|
| name       | id    | used       | hard        | used        | hard      |
| -----      | ----- | -----      | -----       | -----       | -----     |
| mfricke    | 1512  | 592.71 GiB | 1024.00 GiB | 32784       | unlimited |

Centerwide scratch quota for project mfricke2016174 (/carc/scratch/projects/mfricke2016174)

Quota information for storage pool Default (ID: 1):

| user/group     |         | size       |             | chunk files |           |
|----------------|---------|------------|-------------|-------------|-----------|
| name           | id      | used       | hard        | used        | hard      |
| -----          | -----   | -----      | -----       | -----       | -----     |
| mfricke2016174 | 2016142 | 190.97 GiB | 1024.00 GiB | 23704       | unlimited |

```
[vanilla@easley ~]$ sinfo --partition debug  
PARTITION AVAIL TIMELIMIT  NODES  STATE NODELIST  
debug      up    1:00:00    2  idle easley[011-012]
```

sinfo reports information about  
partitions

```
[vanilla@easley ~]$ sinfo --partition debug  
PARTITION AVAIL TIMELIMIT  NODES  STATE NODELIST  
debug      up    1:00:00    2  idle easley[011-012]
```

The debug queues are intended  
for testing your programs.

```
[vanilla@easley ~]$ sinfo --partition debug
```

```
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
```

```
debug      up    1:00:00    2 idle easley[011-012]
```



Name

```
[vanilla@easley ~]$ sinfo --partition debug
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug      up      1:00:00    2  idle easley[011-012]
```



You can run a “job” for up to 4 hrs.

```
[vanilla@easley ~]$ sinfo --partition debug
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug      up    1:00:00    2  idle easley[011-012]
```



There are two nodes in this partition.

```
[vanilla@easley ~]$ sinfo --partition debug
PARTITION AVAIL TIMELIMIT  NODES  STATE NODELIST
debug      up    1:00:00    2  idle easley[011-012]
```



The names of the nodes in the  
partition

```
[vanilla@easley ~]$ sinfo --partition debug  
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST  
debug      up    1:00:00    2 idle easley[011-012]
```



The names of the nodes in the  
partition

```
[vanilla@easley ~]$ sinfo --partition general
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
general*   up 2-00:00:00    9  alloc easley[001-009]
general*   up 2-00:00:00    1  idle easley010
```



Easley001 through 009 are running jobs.

Easley010 is waiting to be used.

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



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

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

```
[vanilla@easley ~]$ man hostname
```

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

```
[vanilla@easley ~]$ man hostname  
(‘q’ to quit)
```

```
[vanilla@easley ~]$ man man  
(‘q’ to quit)
```

```
[vanilla@easley ~]$ man sinfo
```

sinfo(1)

Slurm Commands

sinfo(1)

## NAME

sinfo - View information about Slurm nodes and partitions.

## SYNOPSIS

sinfo [OPTIONS...]

## DESCRIPTION

sinfo is used to view partition and node information for a system running Slurm

## OPTIONS

-a, --all

Display information about all partitions. This causes information to be displayed about partitions that are configured as hidden and partitions that are unavailable to the user's group.

# [vanilla@easley ~]\$ sinfo --all

| PARTITION   | AVAIL | TIMELIMIT  | NODES | STATE | NODELIST                                                |
|-------------|-------|------------|-------|-------|---------------------------------------------------------|
| general*    | up    | 2-00:00:00 | 5     | down* | easley[002,023,027,037,048]                             |
| general*    | up    | 2-00:00:00 | 8     | drain | easley[005,007-008,017,035-036,039-040]                 |
| general*    | up    | 2-00:00:00 | 3     | idle  | easley[009-010,042]                                     |
| general*    | up    | 2-00:00:00 | 1     | down  | easley016                                               |
| bigmem      | up    | 2-00:00:00 | 2     | idle  | easley[049-050]                                         |
| h100        | up    | 2-00:00:00 | 1     | mix   | easley054                                               |
| h100        | up    | 2-00:00:00 | 3     | alloc | easley[051-053]                                         |
| l40s        | up    | 2-00:00:00 | 3     | mix   | easley[055-057]                                         |
| interactive | up    | 4:00:00    | 4     | mix   | easley[054-057]                                         |
| interactive | up    | 4:00:00    | 3     | alloc | easley[051-053]                                         |
| debug       | up    | 1:00:00    | 5     | idle  | easley[009-010,042,049-050]                             |
| debug       | up    | 1:00:00    | 1     | down  | easley016                                               |
| scavenger   | up    | 2-00:00:00 | 5     | down* | easley[002,023,027,037,048]                             |
| scavenger   | up    | 2-00:00:00 | 8     | drain | easley[005,007-008,017,035-036,039-040]                 |
| scavenger   | up    | 2-00:00:00 | 17    | mix   | easley[001,003-004,006,011-013,018,024-026,054-059]     |
| scavenger   | up    | 2-00:00:00 | 23    | alloc | easley[014-015,019-022,028-034,038,041,043-047,051-053] |
| scavenger   | up    | 2-00:00:00 | 9     | idle  | easley[009-010,042,049-050,060-063]                     |
| scavenger   | up    | 2-00:00:00 | 1     | down  | easley016                                               |

# [vanilla@easley ~]\$ sinfo --all

| PARTITION   | AVAIL | TIMELIMIT  | NODES | STATE | NODELIST                                                |
|-------------|-------|------------|-------|-------|---------------------------------------------------------|
| general*    | up    | 2-00:00:00 | 5     | down* | easley[002,023,027,037,048]                             |
| general*    | up    | 2-00:00:00 | 8     | drain | easley[005,007-008,017,035-036,039-040]                 |
| general*    | up    | 2-00:00:00 | 3     | idle  | easley[009-010,042]                                     |
| general*    | up    | 2-00:00:00 | 1     | down  | easley016                                               |
| bigmem      | up    | 2-00:00:00 | 2     | idle  | easley[049-050]                                         |
| h100        | up    | 2-00:00:00 | 1     | mix   | easley054                                               |
| h100        | up    | 2-00:00:00 | 3     | alloc | easley[051-053]                                         |
| l40s        | up    | 2-00:00:00 | 3     | mix   | easley[055-057]                                         |
| interactive | up    | 4:00:00    | 4     | mix   | easley[054-057]                                         |
| interactive | up    | 4:00:00    | 3     | alloc | easley[051-053]                                         |
| debug       | up    | 1:00:00    | 5     | idle  | easley[009-010,042,049-050]                             |
| debug       | up    | 1:00:00    | 1     | down  | easley016                                               |
| scavenger   | up    | 2-00:00:00 | 5     | down* | easley[002,023,027,037,048]                             |
| scavenger   | up    | 2-00:00:00 | 8     | drain | easley[005,007-008,017,035-036,039-040]                 |
| scavenger   | up    | 2-00:00:00 | 17    | mix   | easley[001,003-004,006,011-013,018,024-026,054-059]     |
| scavenger   | up    | 2-00:00:00 | 23    | alloc | easley[014-015,019-022,028-034,038,041,043-047,051-053] |
| scavenger   | up    | 2-00:00:00 | 9     | idle  | easley[009-010,042,049-050,060-063]                     |
| scavenger   | up    | 2-00:00:00 | 1     | down  | easley016                                               |

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



Tell slurm to run a program  
on a compute node...

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



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

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



The program  
to run.

```
[vanilla@easley ~]$ 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.
```

```
easley011
```

```
[vanilla@easley ~]$ queue
```

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

| JOBID | PARTITION | NAME | USER     | ST | TIME |
|-------|-----------|------|----------|----|------|
| 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 |



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.

|      |         |     |          |    |      |                           |
|------|---------|-----|----------|----|------|---------------------------|
| 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) |
| 4327 | 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@easley ~]$ queue
```

| 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) |
| 4327  | 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) |

The reason these jobs are not running is that 'erowland' is already using the maximum number of CPUs they are allowed.

```
[vanilla@easley ~]$ squeue -t R --all
```

| JOBID | PARTITION | NAME     | USER     | ST | TIME       | NODES | NODELIST(REASON) |
|-------|-----------|----------|----------|----|------------|-------|------------------|
| 4405  | condo     | 2ndMA    | mfricke  | R  | 1-07:48:30 | 6     | easley[031-036]  |
| 5208  | condo     | NN       | kgu      | R  | 5:48:49    | 1     | easley015        |
| 5210  | condo     | NN       | kgu      | R  | 6:30:13    | 1     | easley014        |
| 5209  | condo     | NN       | kgu      | R  | 6:31:13    | 1     | easley013        |
| 5206  | condo     | NN       | kgu      | R  | 6:32:13    | 1     | easley051        |
| 5207  | condo     | NN       | kgu      | R  | 6:32:13    | 1     | easley052        |
| 5205  | condo     | NN       | kgu      | R  | 6:32:43    | 1     | easley028        |
| 4595  | cup-ecs   | golConfi | aalasand | R  | 2-06:51:59 | 1     | easley050        |
| 4594  | cup-ecs   | golConfi | aalasand | R  | 2-06:52:03 | 1     | easley049        |
| 5120  | general   | jupyterh | jacobm   | R  | 11:45:47   | 1     | easley007        |
| 4313  | general   | PRE      | erowland | R  | 1:17:29    | 2     | easley[003-004]  |
| 5111  | general   | 1stMA    | mfricke  | R  | 11:15:28   | 2     | easley[005-006]  |
| 5025  | general   | c2n      | jxzuo    | R  | 1:50       | 1     | easley001        |
| 5024  | general   | c2n      | jxzuo    | R  | 31:28      | 1     | easley002        |
| 5203  | general   | NN       | kgu      | R  | 6:37:50    | 1     | easley009        |
| 5201  | general   | NN       | kgu      | R  | 6:38:14    | 1     | easley008        |
| 4390  | tc        | UCsTpCyd | lepluart | R  | 2-15:18:18 | 3     | easley[018-020]  |
| 5198  | tc        | NN       | kgu      | R  | 6:40:19    | 1     | easley030        |
| 5196  | tc        | NN       | kgu      | R  | 6:40:31    | 1     | easley029        |

```
[vanilla@easley ~]$ srun --partition debug --ntasks 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.
```

```
easley011
```

```
easley011
```

```
[vanilla@easley ~]$ srun --partition debug --ntasks 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.
```

```
easley011
```

```
easley011
```

You ran two **copies** of your program.

**ntasks** is the number of copies to run.

```
[vanilla@easley ~]$ srun --partition debug --ntasks 8 hostname
```

```
srun: Account not specified in script or ~/.default_slurm_account, using latest project
```

```
easley011
```

```
easley011
```

```
easley011
```

```
You have not been allocated GPUs. To request GPUs, use the -G option in your submission script.
```

```
easley011
```

```
easley011
```

```
easley011
```

```
easley011
```

```
easley011
```

You ran eight **copies** of your program.

**ntasks** is the number of copies to run.

```
[vanilla@easley ~]$ srun --partition debug --ntasks 8 hostname
```

```
srun: Account not specified in script or ~/.default_slurm_account, using latest project
```

```
easley011
```

```
easley011
```

```
easley011
```

```
You have not been allocated GPUs. To request GPUs, use the -G option in your submission script.
```

```
easley011
```

```
easley011
```

```
easley011
```

```
easley011
```

```
easley011
```

**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@easley ~]$ 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.
```

```
easley011
```

```
easley011
```

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

```
[vanilla@easley ~]$ srun --partition debug --nodes 2 --ntasks-per-node 4 hostname  
srun: Account not specified in script or ~/.default_slurm_account, using latest project  
easley012
```

You have not been allocated GPUs. To request GPUs, use the -G option in your submission script.

```
easley012
```

```
easley011
```

```
easley011
```

```
easley012
```

```
easley012
```

```
easley011
```

```
easley011
```

**Here we are telling SLURM to run 4  
copies of our program on 2 different  
compute nodes.**

**This is useful when our programs need  
a bigger share of the compute node.**

```
[vanilla@easley ~]$ srun --partition debug --nodes 2  
--ntasks-per-node 2 --cpus-per-task 2 hostname  
srun: Account not specified in script or ~/.default_slurm_account, using latest  
project
```

```
easley011
```

You have not been allocated GPUs. To request GPUs, use the -G option in your submission script.

```
easley011
```

```
easley012
```

```
easley012
```

**And we can combine all three.**

```
[vanilla@easley ~]$ srun --partition debug --mem 4G  
--nodes 2 --ntasks-per-node 2 --cpus-per-task 2 hostname  
srun: Account not specified in script or ~/.default_slurm_account, using latest  
project
```

```
easley012
```

```
easley012
```

```
You have not been allocated  
submission script.
```

```
easley011
```

```
easley011
```

**And we can specify how much  
memory we want.**

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

```
[vanilla@easley ~]$ srun --partition debug --mem 4G  
--nodes 2 --ntasks-per-node 2 --cpus-per-task 2 hostname  
srun: Account not specified in script or ~/.default_slurm_account, using latest  
project
```

```
easley012
```

```
easley012
```

```
You have not been allocated  
submission script.
```

```
easley011
```

```
easley011
```

**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@easley ~]$ srun --partition debug --mem 4G  
--nodes 2 --ntasks-per-node 2 --cpus-per-task 2 hostname  
srun: Account not specified in script or ~/.default_slurm_account, using latest  
project
```

```
easley012
```

```
easley012
```

```
You have not been allocated  
submission script.
```

```
easley011
```

```
Hopper011
```

- 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?

```
[vanilla@easley ~]$ srun --partition debug --mem 4G  
--nodes 2 --ntasks-per-node 2 --cpus-per-Easley hostname  
srun: Account not specified in script or ~/.default_slurm_account, using latest  
project  
easley012  
easley012  
You have not been allocated a  
submission script.  
easley011  
easley011
```

This command is getting pretty long.

We can use **shell scripts** to automate  
all this in **batch mode**.

# 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@easley ~]$ 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...**

**If, and only if, you already have a workshops directory:**

```
cd workshops
git pull
```

```
[vanilla@easley ~]$ cd workshops  
[vanilla@easley ~]$ cd intro_workshop  
[vanilla@easley intro_workshop]$ pwd  
/users/vanilla/workshops/intro_workshop  
[vanilla@easley intro_workshop]$ cat slurm/workshop_example.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
```

```
hostname
```

Let's take a look at the **workshop\_example.sh** script in the slurm directory...

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

We **submit** our slurm  
shell script with the  
sbatch command.

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

We **submit** our slurm  
shell script with the  
sbatch command.

Easley 250K jobs (6 mo beta, 1 mo production)

Hopper: 4 million+ jobs (4 yr)

Xena: 1.6 million+ jobs (5 yrs, retiring)

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

Notice that the only output we get is a job id.

This indicates that the script was successfully sent to the scheduler.

The commands in the script will run as soon as the hardware requested is available.

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

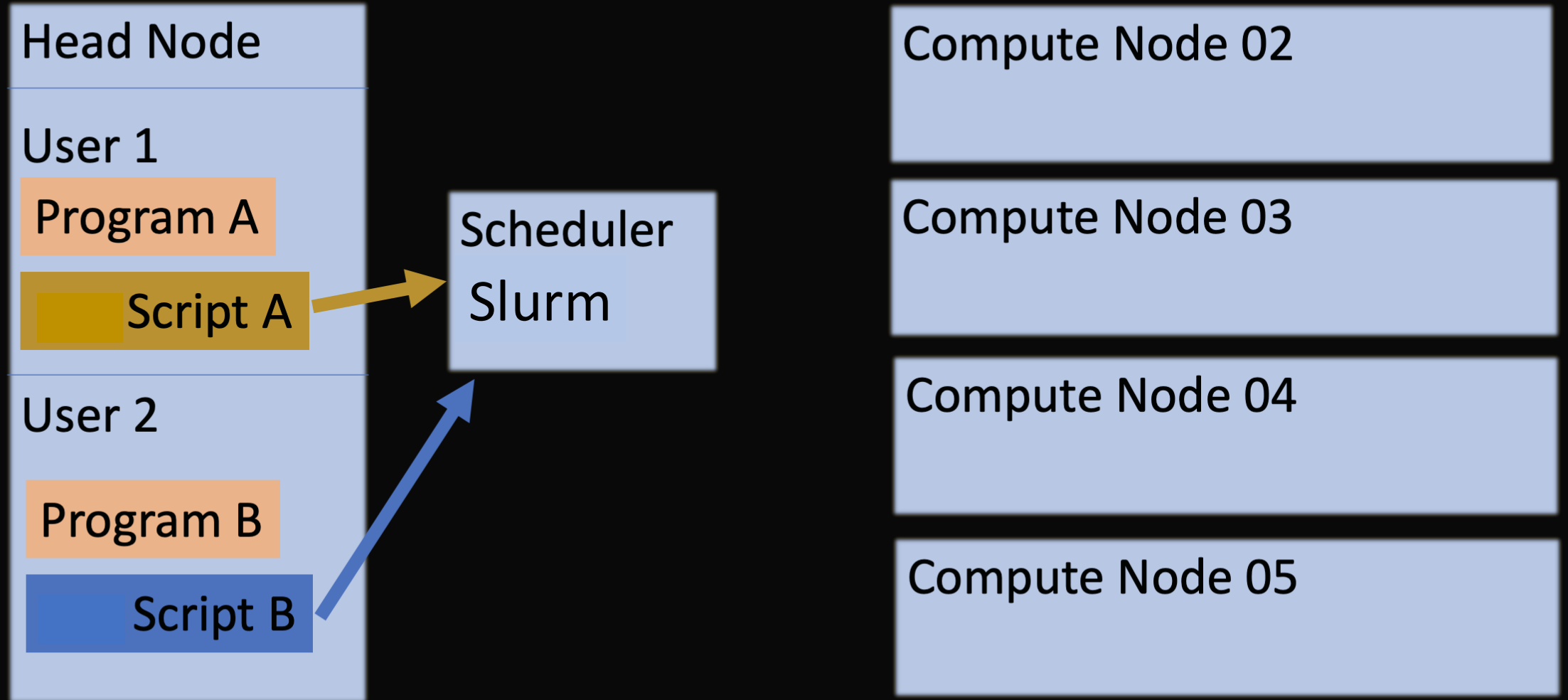
Compute Node 03

Compute Node 04

Compute Node 05

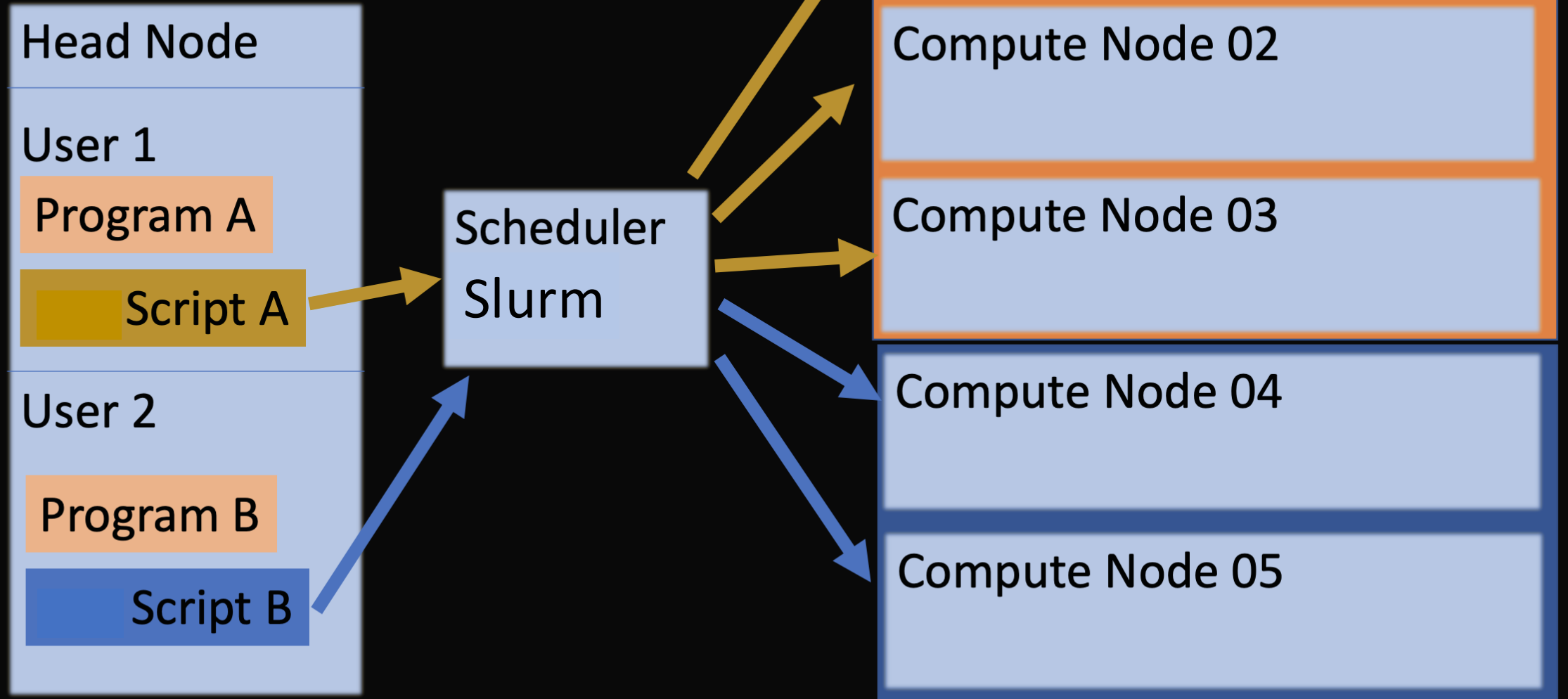
Shared filesystems – All nodes can access the same programs and write output

# Workflow



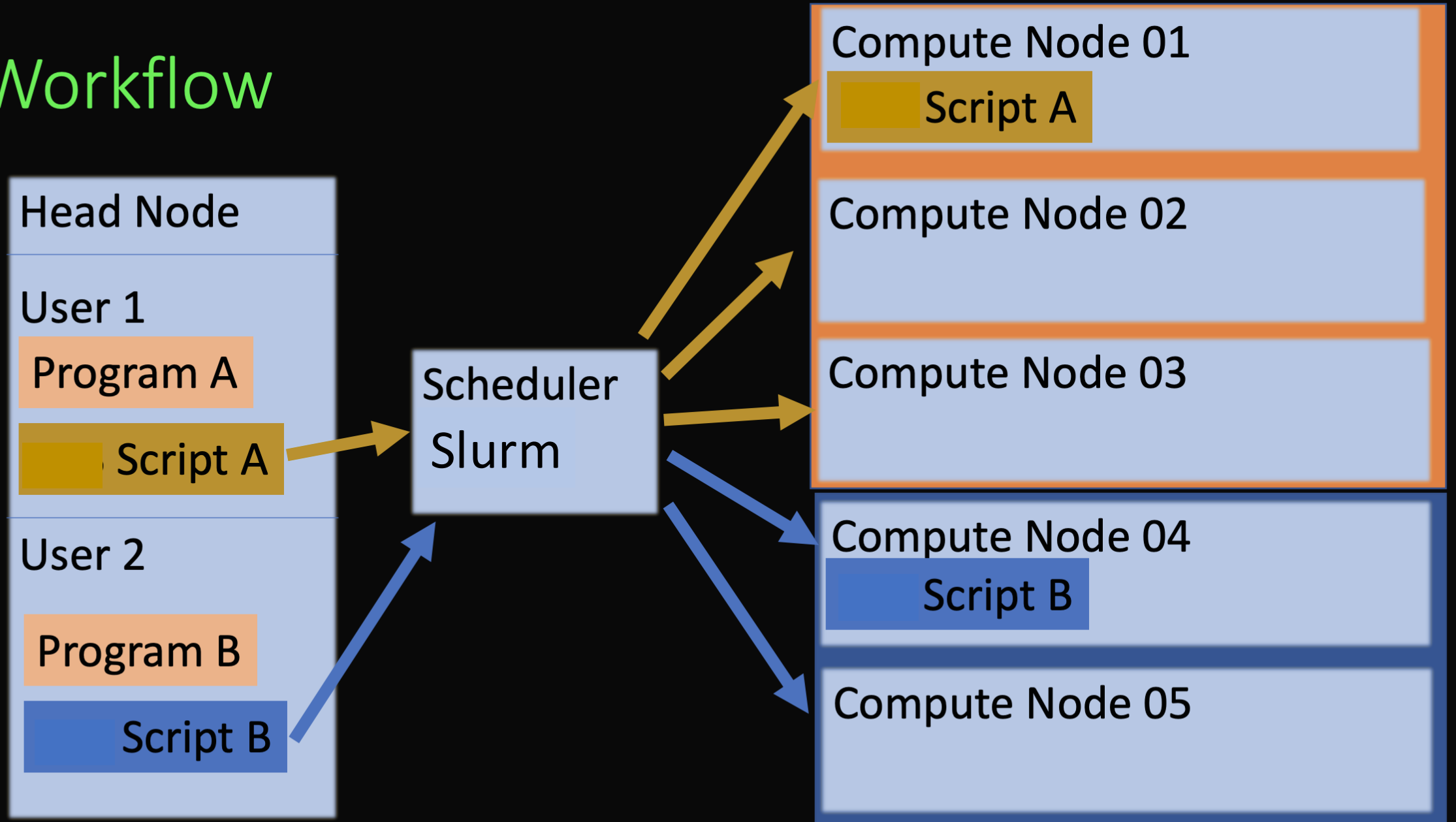
Shared filesystems – All nodes can access the same programs and write output

# Workflow



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

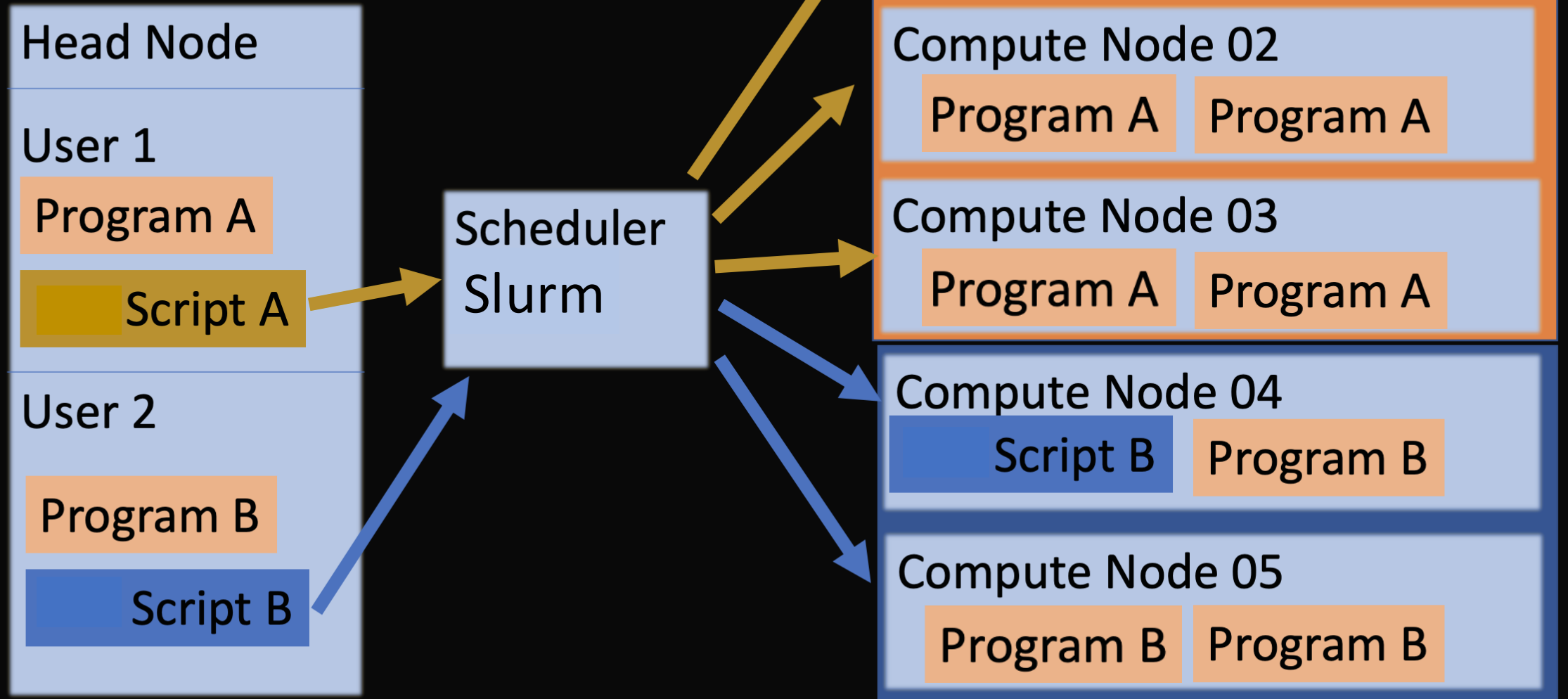
# Workflow



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@easley 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@easley 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@easley intro_workshop]$ cat slurm-5252.out  
easley011
```

```
[vanilla@easley 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@easley intro_workshop]$ cat slurm-5252.out  
easley011
```

Why did it only run the program once instead of 4 times?

```
[vanilla@easley 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@easley intro_workshop]$
```

**Take a look at the  
output file.**

General  
Purpose  
Computational  
Chemistry  
Package



We will design a molecule using Avogadro and then optimize its geometry with Gaussian

Favorites



Avogadro2

Accessories



IcedTea-Web Policy Editor

Documentation



OpenJDK 1.8.0 for x86\_64 Policy ...

Internet

Office

Programming

Sound &amp; Video

Sundry

System Tools

Utilities

Other



7  
CENTOS



[HPC Docs: Slurm Environmental Va...



[mfricke@hopper:~/workshops]

File Edit View Crystal Extensions Quantum Help



Draw

Element: Carbon (6)

Bond Order: Automatic

☒ Adjust Hydrogens

Display Types

- ☒ Crystal Lattice
- ☒ Ball and Stick
- ☐ Licorice
- ☐ Van der Waals
- ☐ Wireframe
- ☐ Meshes
- ☒ Reference Axes Overlay
- ☐ Van der Waals (AO)

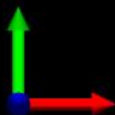
View Configuration

Display Types

Molecules

Untitled ()

Split Horizontal Split Vertical Close



CENTOS

File Edit View Crystal Extensions Quantum Help



Draw

Element: Carbon (6)

Bond Order: Automatic

☒ Adjust Hydrogens

Display Types

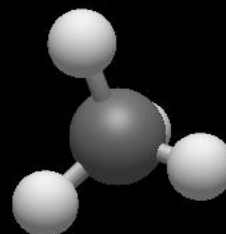
- ☒ Crystal Lattice
- ☒ Ball and Stick
- ☐ Licorice
- ☐ Van der Waals
- ☐ Wireframe
- ☐ Meshes
- ☒ Reference Axes Overlay
- ☐ Van der Waals (AO)

View Configuration

Display Types

Molecules

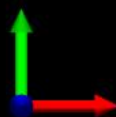
Untitled (CH4)



Split Horizontal

Split Vertical

Close



CENTOS

File Edit View Crystal Extensions Quantum Help



Draw



\*

Element: Carbon (6)



Bond Order: Automatic

☒ Adjust Hydrogens

Display Types



- ☒ Crystal Lattice
- ☒ Ball and Stick
- ☐ Licorice
- ☐ Van der Waals
- ☐ Wireframe
- ☐ Meshes
- ☒ Reference Axes Overlay
- ☐ Van der Waals (AO)

View Configuration

Display Types

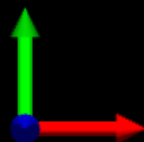
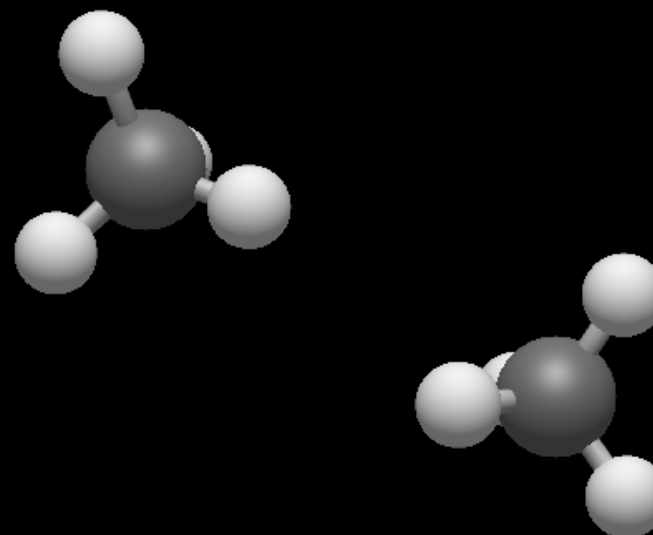
Molecules



Untitled (C2H8)



Split Horiz





Element: Carbon (6)

Order: Automatic

☒ Adjust Hydrogens

Types

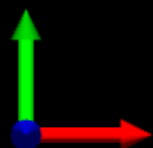
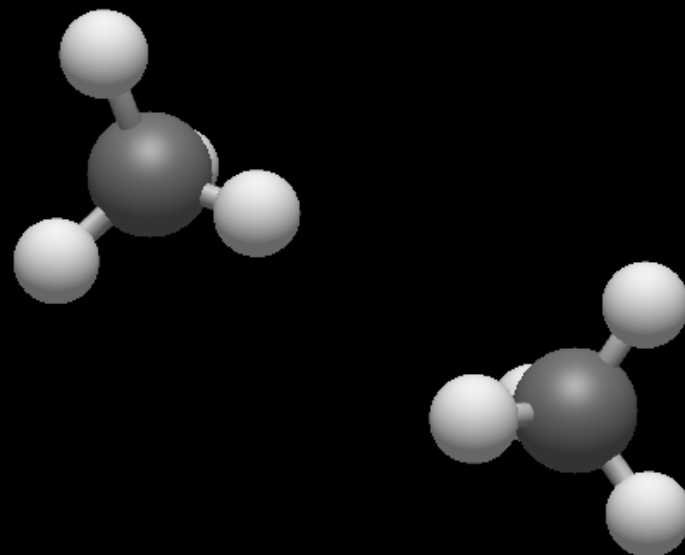
- Crystal Lattice
- Ball and Stick
- Space-Filling
- Reduced Representation
- Wireframe
- Surface
- Reference Axes Overlay
- Reduced Representation (AO)

Configuration Display Types

es

(C2H8)

Split Horizontal





ent: Oxygen (8) ▼

der: Automatic ▼

☒ Adjust Hydrogens

types

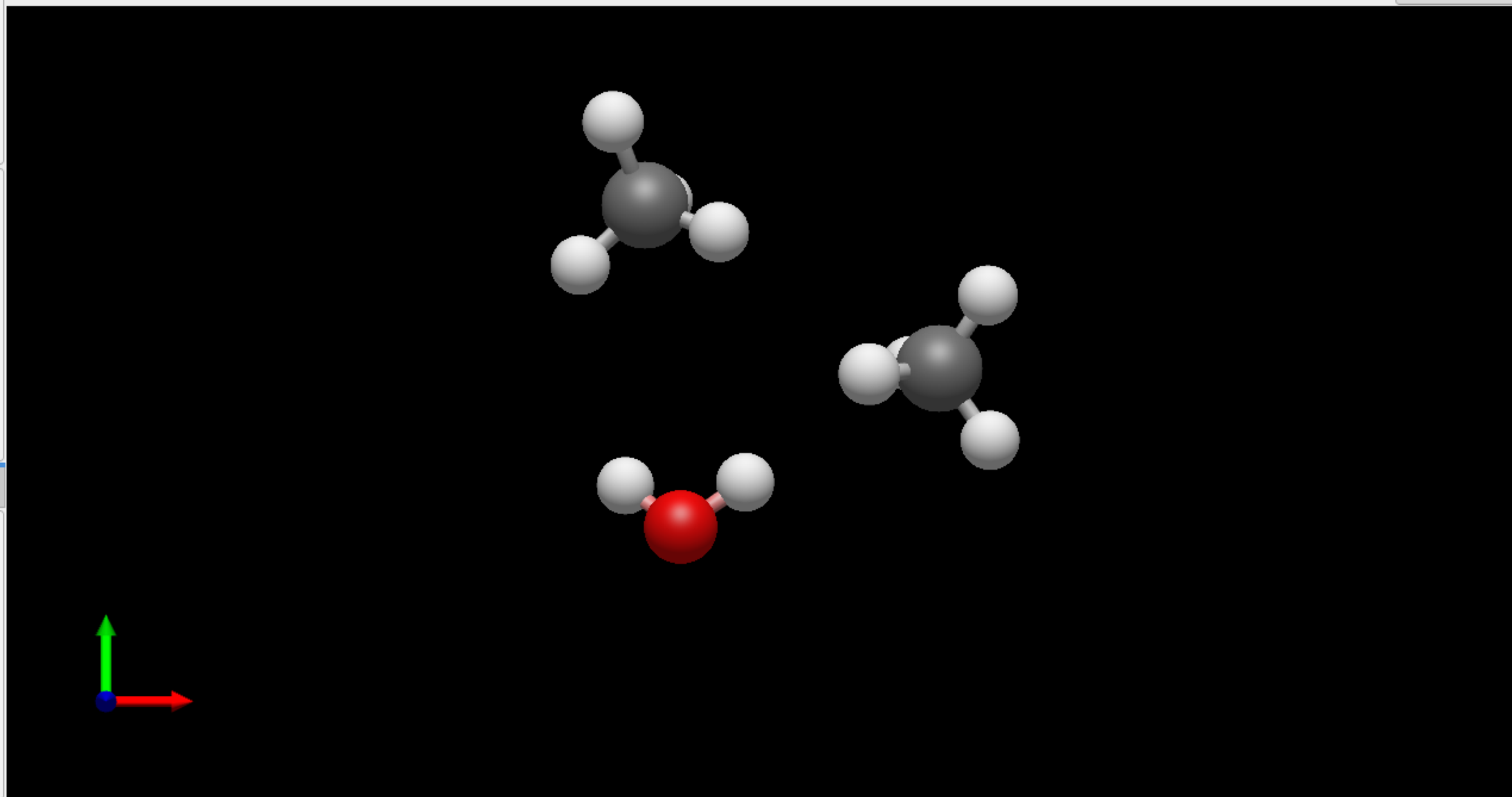
- al Lattice
- and Stick
- ice
- ler Waals
- frame
- es
- ence Axes Overlay
- ler Waals (AO)

onfiguration Display Types

s

(C2H10O) ✕

Split Horizontal





ent: Chlorine (17) ▾

ler: Automatic ▾

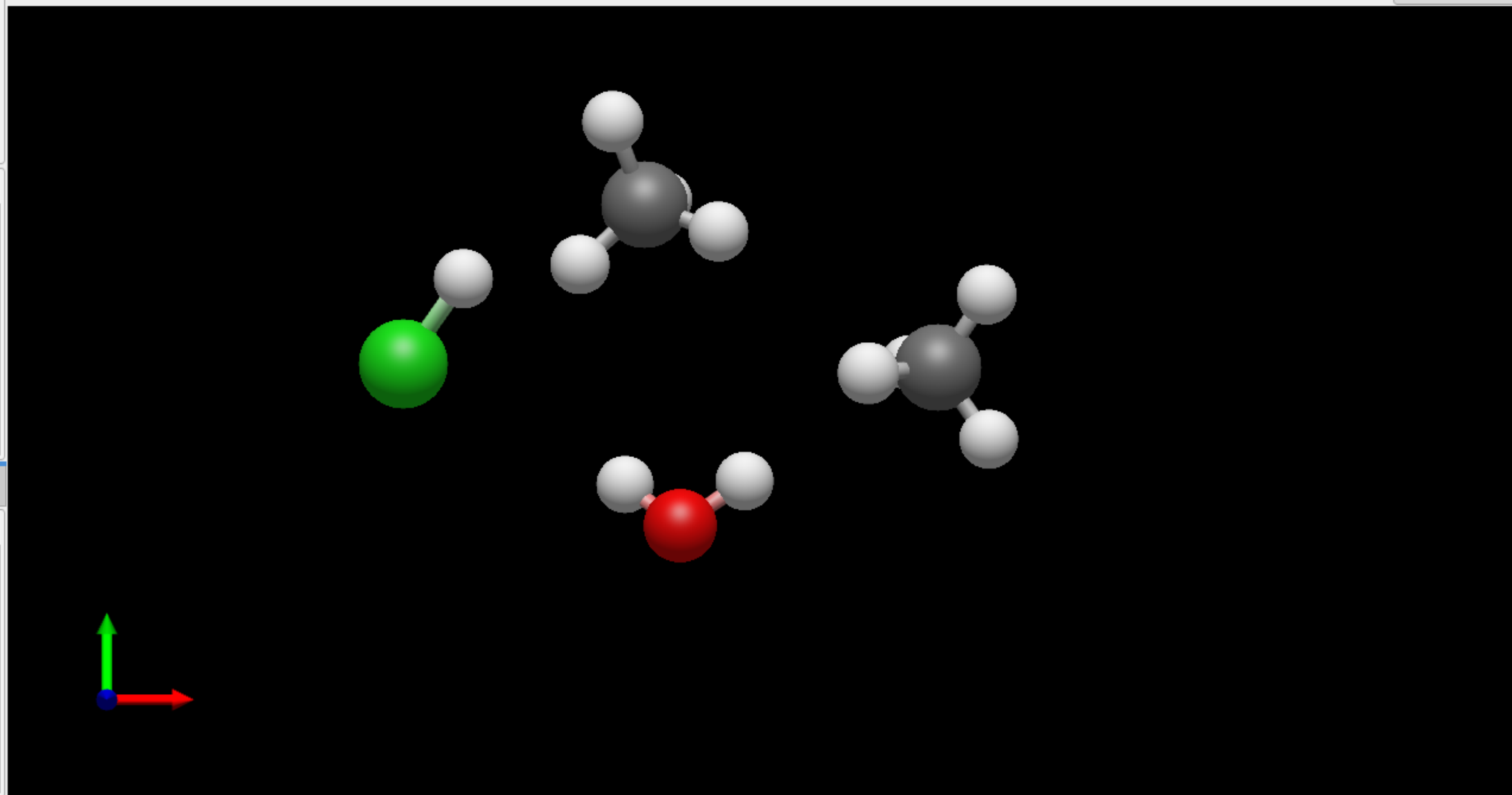
☒ Adjust Hydrogens

types

- al Lattice
- nd Stick
- ce
- er Waals
- frame
- es
- ence Axes Overlay
- er Waals (AO)

Configuration Display Types

Chemical formula: C2H11OCl





New



Open



Save



Save As

Molecule



Draw



Element: Chlorine (17)



Bond Order: Automatic

☒ Adjust Hydrogens

Display Types

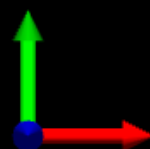
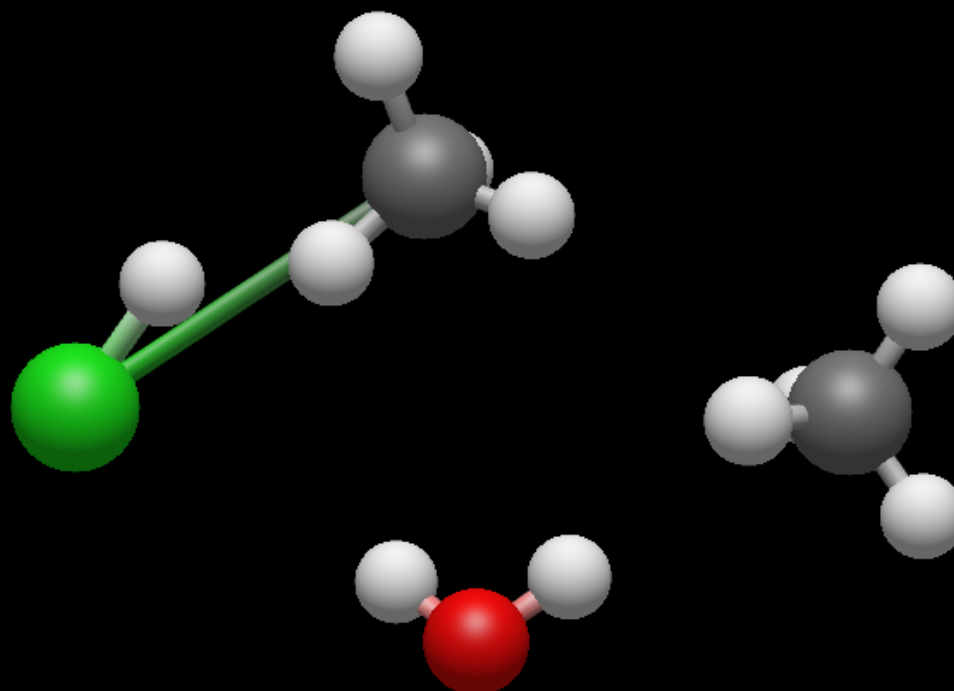


- ☒ Crystal Lattice
- ☒ Ball and Stick
- ☐ Licorice
- ☐ Van der Waals
- ☐ Wireframe
- ☐ Meshes
- ☒ Reference Axes Overlay
- ☐ Van der Waals (AO)

View Configuration

Display Types

Molecules

Untitled (C<sub>2</sub>H<sub>11</sub>OCl)

Distance: 2.934 Å

lorine (17)

Automatic

Automatic

gle

uble

ble

s

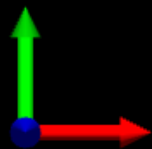
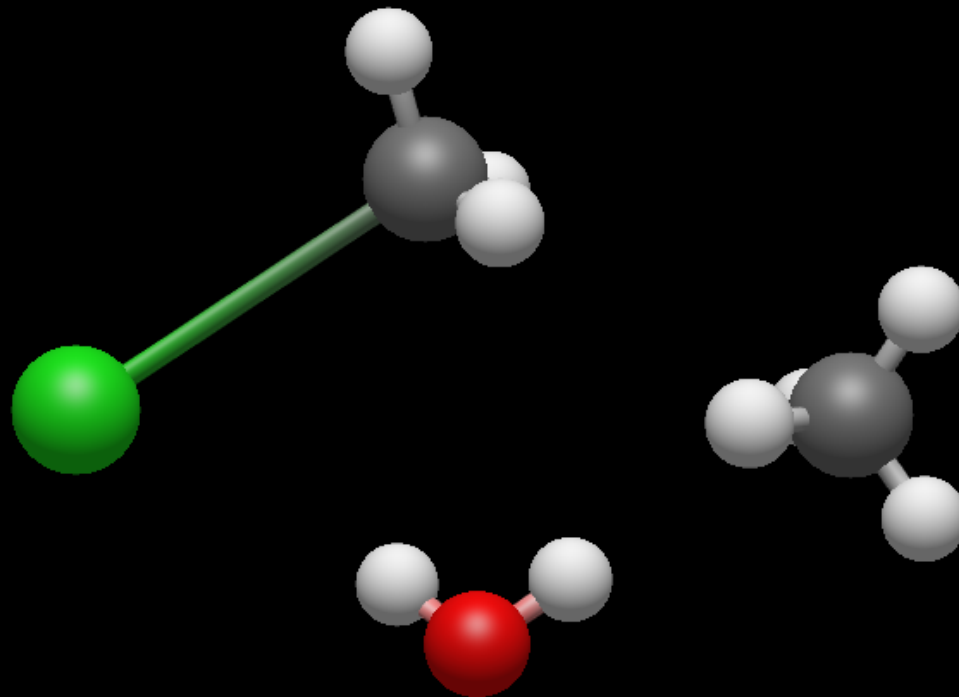
es Overlay

s (AO)

tion

Display Types

Cl)



Split Horizontal

Split

File Edit View Crystal Extensions Quantum Help



Draw



Element: Chlorine (17)

Bond Order: Double

☒ Adjust Hydrogens

Display Types

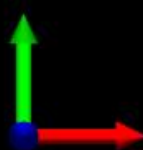
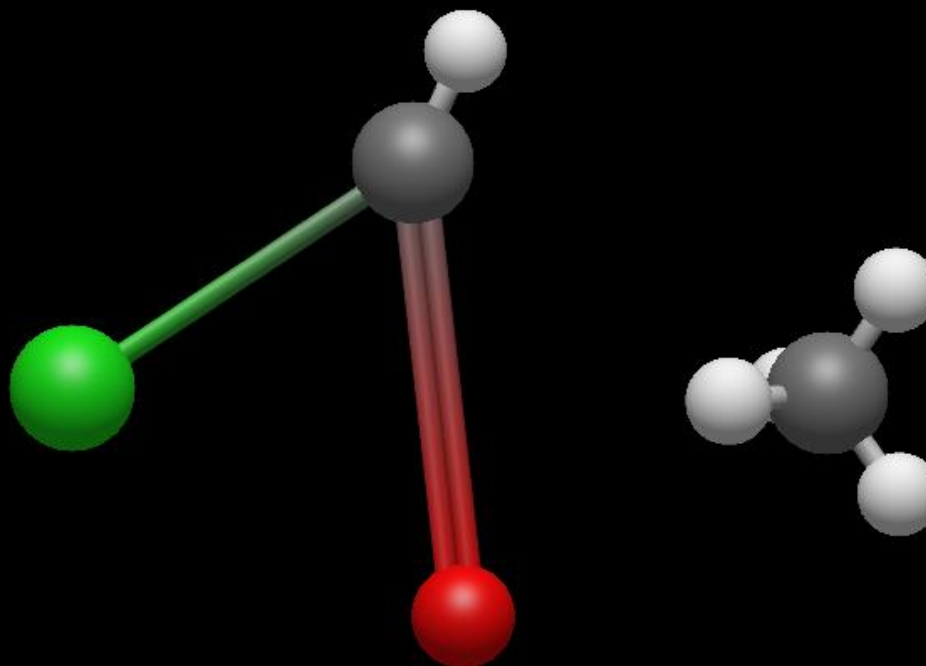


- ☒ Crystal Lattice
- ☒ Ball and Stick
- ☐ Licorice
- ☐ Van der Waals
- ☐ Wireframe
- ☐ Meshes
- ☒ Reference Axes Overlay
- ☐ Van der Waals (AO)

View Configuration

Display Types

Molecules

Untitled (C<sub>2</sub>H<sub>5</sub>OCl)

Draw  

Element: Chlorine (17) 

Bond Order: Single 

☒ Adjust Hydrogens

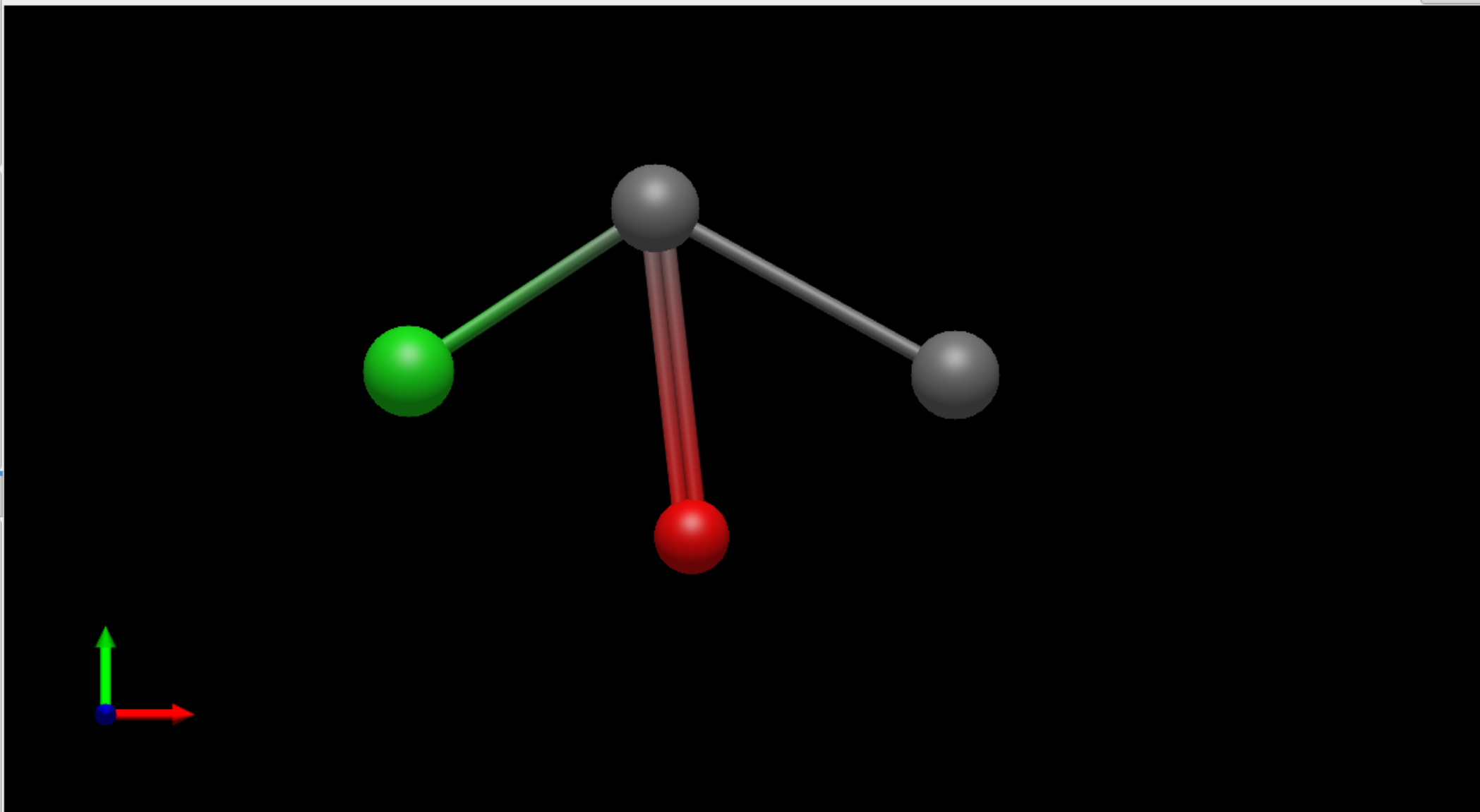
Display Types  

- ☒ Crystal Lattice
- ☒ Ball and Stick
- ☐ Licorice
- ☐ Van der Waals
- ☐ Wireframe
- ☐ Meshes
- ☒ Reference Axes Overlay
- ☐ Van der Waals (AO)

View Configuration Display Types

Molecules  

Untitled (C2OCl) 



File Edit View Crystal Extensions

New Open Save

Draw

Element: Chlorine (17)

Bond Order: Single

☒ Adjust Hydrogens

Display Types

☒ Crystal Lattice

☒ Ball and Stick

☐ Licorice

☐ Van der Waals

☐ Wireframe

☐ Meshes

☒ Reference Axes Overlay

☐ Van der Waals (AO)

View Configuration Display Types

Molecules

Untitled (C2OCl)

Cancel Name AcetylChloride.cml Save

Home mfricke

Documents

Downloads

Music

Pictures

Videos

+ Other Locations

| Name                          | Size      | Modified    |
|-------------------------------|-----------|-------------|
| AcetylChloride.cml            | 794 bytes | 12:20       |
| admin_scripts                 |           | 5 Aug 2019  |
| AIPSDATA                      |           | 24 Oct 2022 |
| AIPSDATA_BAK                  |           | 24 Oct 2022 |
| alphafold                     |           | 3 Aug       |
| armpl                         |           | 16 Jun 2022 |
| assembly_index                |           | 29 Jul 2022 |
| astrobiology                  |           | 18 Feb      |
| backups                       |           | 11 Mar 2022 |
| bad.local                     |           | 18 Feb      |
| bad_crypto                    |           | 19 Jul 2022 |
| batchspawner                  |           | 9 Aug 2019  |
| bin                           |           | 9 Sep 2021  |
| bin_bak                       |           | 16 Oct 2019 |
| carpentry                     |           | 9 Mar       |
| casa_env                      |           | 18 Feb      |
| casa_example                  |           | 19 Feb      |
| CATSettings                   |           | 9 Jun 2022  |
| certs                         |           | 28 Aug 2019 |
| clblast                       |           | 16 Jun 2022 |
| climate_change_jupyterhub     |           | 15 Sep 2022 |
| climate_change_jupyterhub_old |           | 13 Sep 2022 |
| Complmmunology                |           | 27 Apr      |
| conf.d                        |           | 12 Oct 2022 |
| configs                       |           | 9 Jun 2022  |
| Coursera                      |           | 31 Aug 2019 |
| crest                         |           | 16 Jun 2022 |
| CS499_HPC                     |           | Sat         |
| cudnn-training                |           | 5 Nov 2020  |
| Desktop                       |           | 9 Mar       |

Chemical Markup Language

File Edit View Crystal Extensions Quantum Help

New Ctrl+N  
Open... Ctrl+O  
Open Recent  
Save Ctrl+S  
Save As... Ctrl+Shift+S

Export Graphics...  
Molecule...  
3DMol HTML Snippet...  
Quit Ctrl+Q

☐ Licorice  
☐ Van der Waals  
☐ Wireframe  
☐ Meshes  
☒ Reference Axes Overlay  
☐ Van der Waals (AO)

View Configuration

Display Types

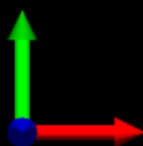
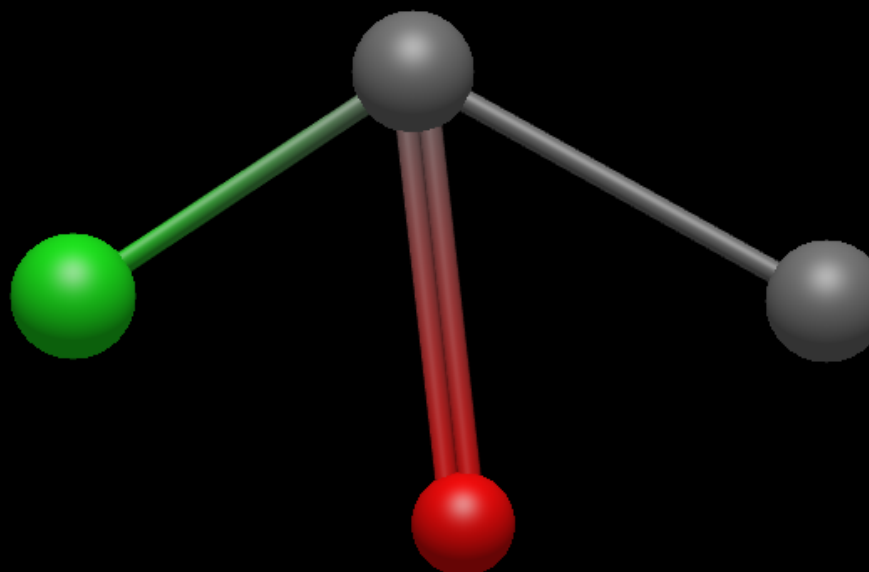
Molecules

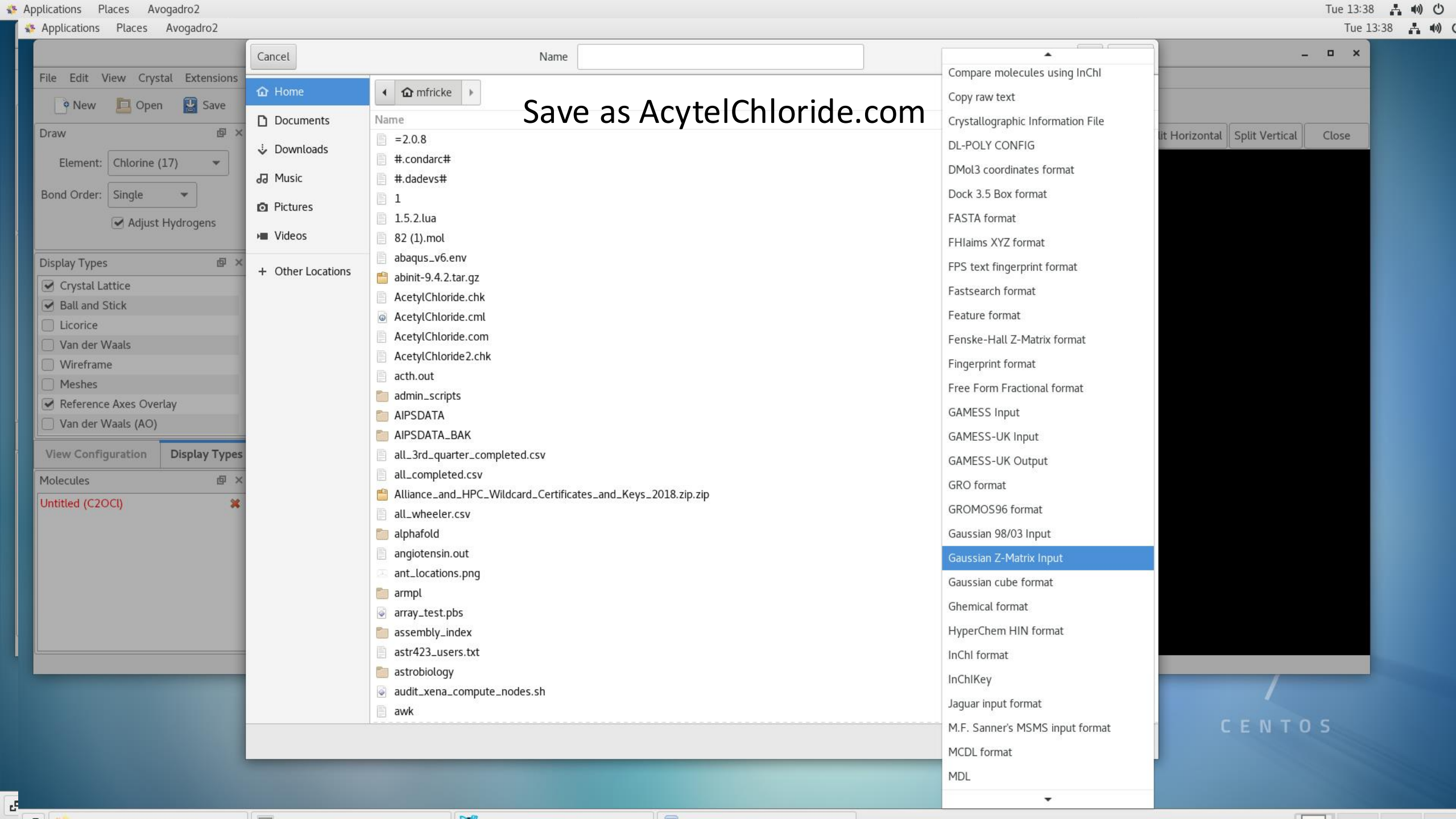
Untitled (C2OCl)

Save Save As Molecule



Split H





Save as AcytelChloride.com

- Compare molecules using InChI
- Copy raw text
- Crystallographic Information File
- DL-POLY CONFIG
- DMol3 coordinates format
- Dock 3.5 Box format
- FASTA format
- FHlaims XYZ format
- FPS text fingerprint format
- Fastsearch format
- Feature format
- Fenske-Hall Z-Matrix format
- Fingerprint format
- Free Form Fractional format
- GAMESS Input
- GAMESS-UK Input
- GAMESS-UK Output
- GRO format
- GROMOS96 format
- Gaussian 98/03 Input
- Gaussian Z-Matrix Input**
- Gaussian cube format
- Chemical format
- HyperChem HIN format
- InChI format
- InChIKey
- Jaguar input format
- M.F. Sanner's MSMS input format
- MCDL format
- MDL

```
scp AcetylChloride.com vanilla@easley.alliance.unm.edu:~
```

```
AcetylChloride.com          100% 269  114.7KB/s  00:00
```

# ssh vanilla@easley.alliance.unm.edu:~

-----  
Welcome to Easley

- ▶ By using CARC systems you agree to the CARC Good Neighbor Guide
- ▶ Tutorials and the Good Neighbor Guide are at <https://goto.unm.edu/carc-start>
- ▶ Need help? Email: [help@carc.unm.edu](mailto:help@carc.unm.edu)
- ▶ Please include an acknowledgement of CARC in any publications that make use of computations performed here.

vanilla@easley:~ \$

```
cd workshops/gaussian/
```

```
ls -l
```

```
total 1192
```

```
-rw-r--r-- 1 mfricke users 500 Sep 12 13:22 AcetylChloride.com
```

```
-rw-r-xr-- 1 mfricke users 795 Sep 12 14:15 gaussian16_linda_acetylchloride.sh
```

```
cd workshops/gaussian/
```

```
ls -l
```

```
total 1192
```

```
-rw-r--r-- 1 mfricke users 500 Sep 12 13:22 AcetylChloride.com
```

```
-rw-r-xr-- 1 mfricke users 795 Sep 12 14:15 gaussian16_linda_acetylchloride.sh
```

```
sbatch gaussian16_linda_acetylchloride.sh
```

```
sbatch: Using account 2016199 from ~/.default_slurm_account
```

```
Submitted batch job 2047647
```

```
cd workshops/gaussian/
```

```
ls -l
```

```
total 1192
```

```
-rw-r--r-- 1 mfricke users 500 Sep 12 13:22 AcetylChloride.com
```

```
-rw-r-xr-- 1 mfricke users 795 Sep 12 14:15 gaussian16_linda_acetylchloride.sh
```

```
cat gaussian16_linda_acetylchloride.sh
```

```
#!/bin/bash
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=4
#SBATCH --time=00:10:00
#SBATCH --job-name=g16_acetylchloride
#SBATCH --mail-user yourusername@unm.edu
```

```
module load gaussian/g16
```

```
# To make linda print verbose messages
export GAUSS_LFLAGS="-v"
```

```
INPUT_MOLECULE=$SLURM_SUBMIT_DIR/AcetylChloride.com
OUTPUT_FILE=$SLURM_SUBMIT_DIR/AcetylChloride.log
```

```
# Reformat the SLURM provided list of compute nodes to a format Gaussian can understand
# i.e. remove duplicates and replace newlines with commas
export GAUSS_WDEF=$(cat $CARC_NODEFILE | uniq | sed -z 's/\n/,/g;s/,,$/\n/')
```

```
# Tell Linda to use as many nodes as requested by the user
export GAUSS_PDEF=$SLURM_CPUS_PER_TASK
echo "Parallelizing $GAUSS_PDEF processes across $GAUSS_WDEF nodes."
```

```
# Set the memory Gaussian variable
export GAUSS_MDEF=4GB
```

```
# Run Gaussian
g16 $INPUT_MOLECULE $OUTPUT_FILE
```

Gaussian uses a helper program called Linda to run multiple copies on multiple nodes. Linda does not understand SLURM so we have to spend some time converting SLURM information into a format Linda understand. That's most of this script.

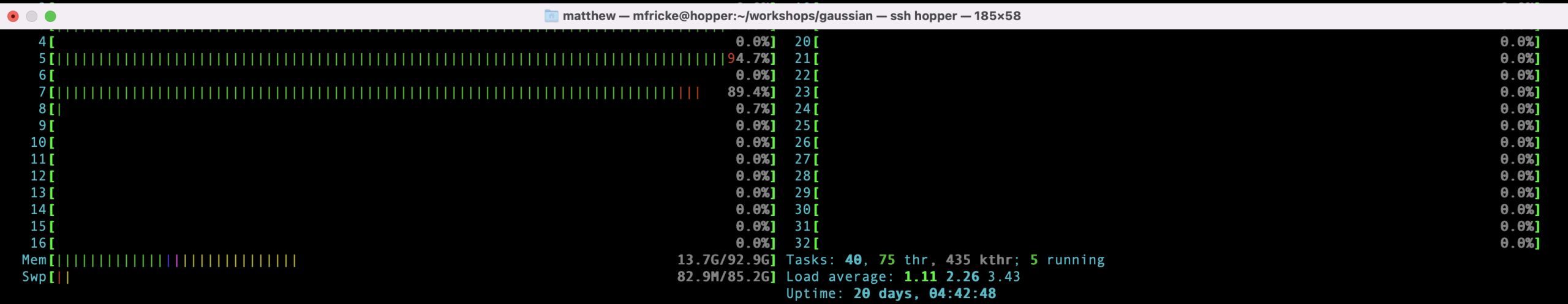
# watch squeue --me

Every 2.0s: squeue --me

easley: Tue Sep 12 14:37:13 2023

| JOBID   | PARTITION | NAME     | USER    | ST | TIME | NODES | NODELIST(REASON) |
|---------|-----------|----------|---------|----|------|-------|------------------|
| 2047652 | general   | g16_acet | mfricke | R  | 1:05 | 2     | easley[003-004]  |

Output will be written to AcetylChloride.log and slurm-<jobid>.out



\$ ssh easley<number>  
\$ htop

# cat AcetylChloride.log

```
...  
EnCoef did 31 forward-backward iterations  
EnCoef did 100 forward-backward iterations  
EnCoef did 5 forward-backward iterations  
EnCoef did 100 forward-backward iterations  
>>>>>>>> Convergence criterion not met.  
SCF Done: E(RB3LYP) = -612.652222413 A.U. after 33 cycles  
NFOck= 32 Conv
```

Output will be written to AcetylChloride.log and slurm-<jobid>.out

Gaussian completed the run, but the SCF did not fully meet the convergence criteria, so the output should be interpreted with caution.

# You have learned

- how to run programs using the SLURM scheduler
- the difference between interactive and batch jobs
- how to check the status of your jobs
- how to select debug vs general SLURM partitions
- how to ask for the hardware resources you need
- you ran the Gaussian code using SLURM

