

# Blanca and the BioKEM environment.

Shawn Laursen and John Heumann  
2022.06.13

# Outline.

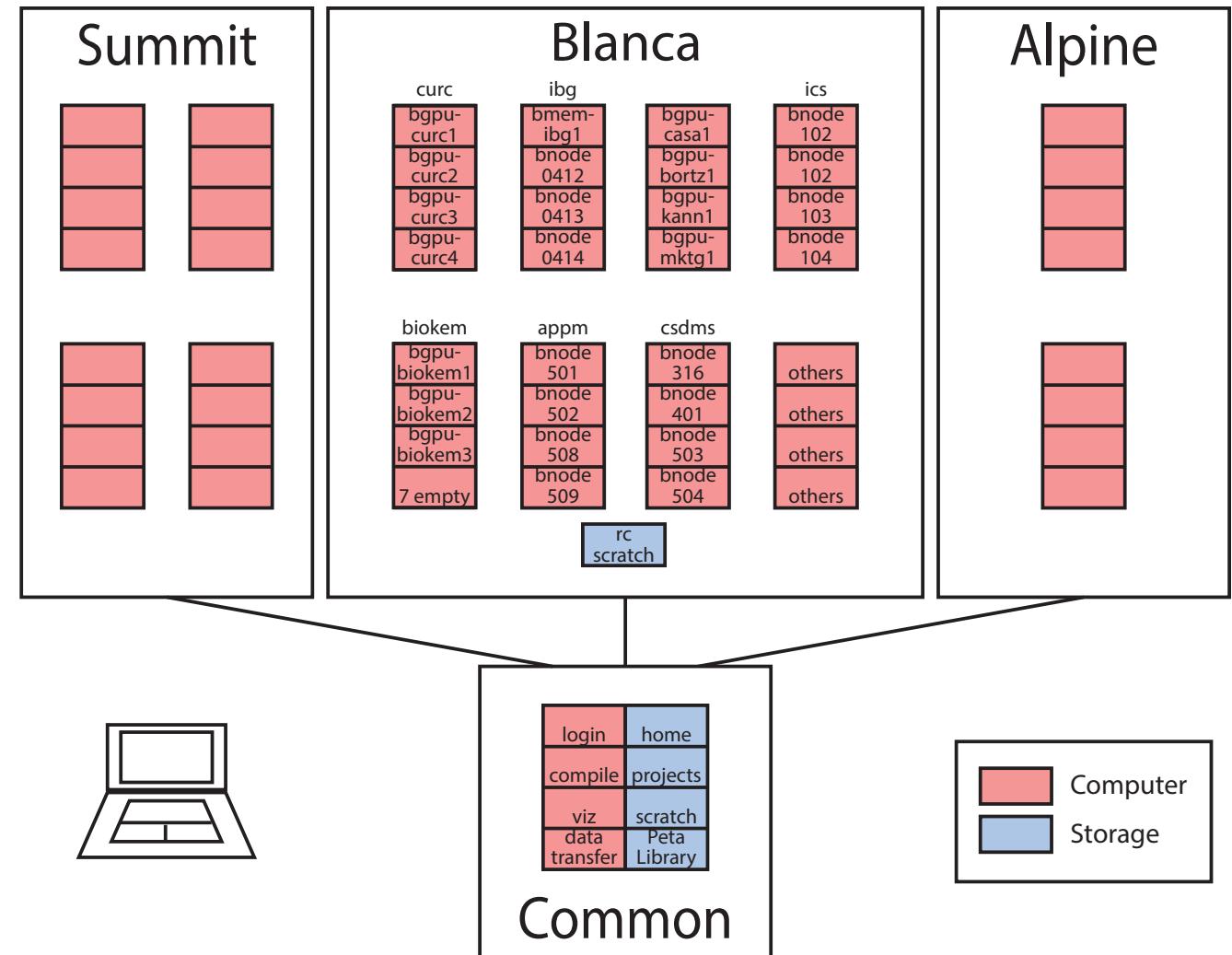
1. Cluster architecture
2. Logging on
3. SLURM
4. Data management
5. Software
6. Practical examples
7. Summary workflow



# Cluster architecture

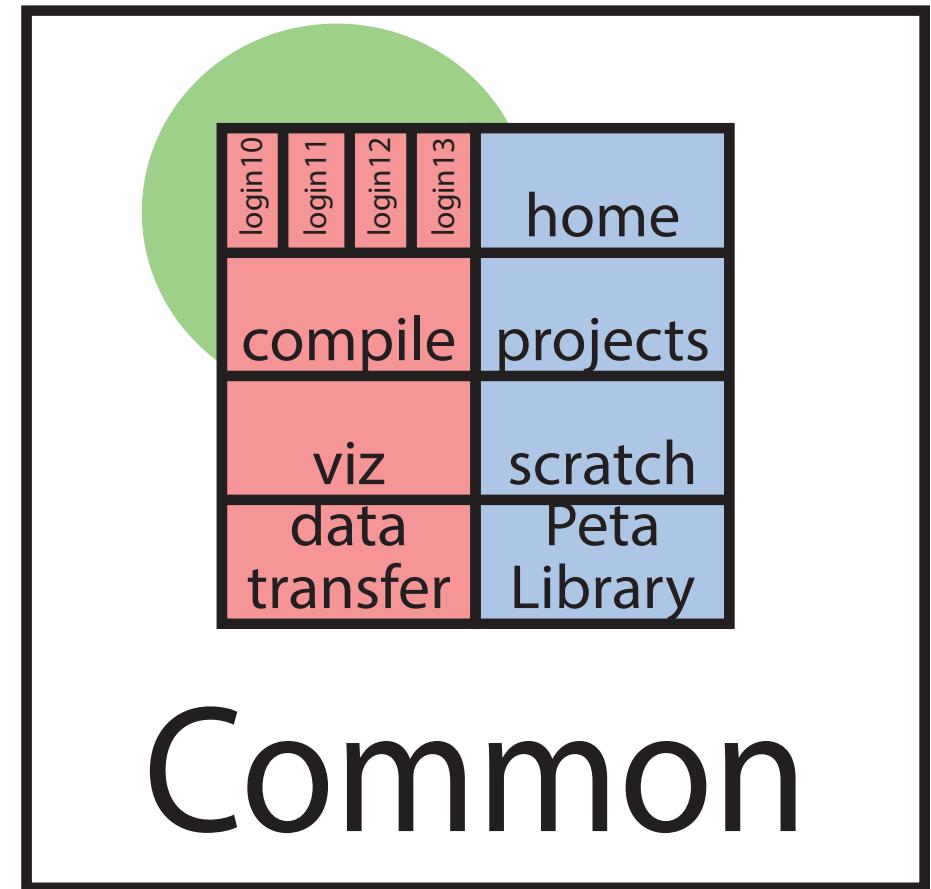
# Cluster architecture - overview.

1. Login
2. Viz
3. Compute
4. Compile



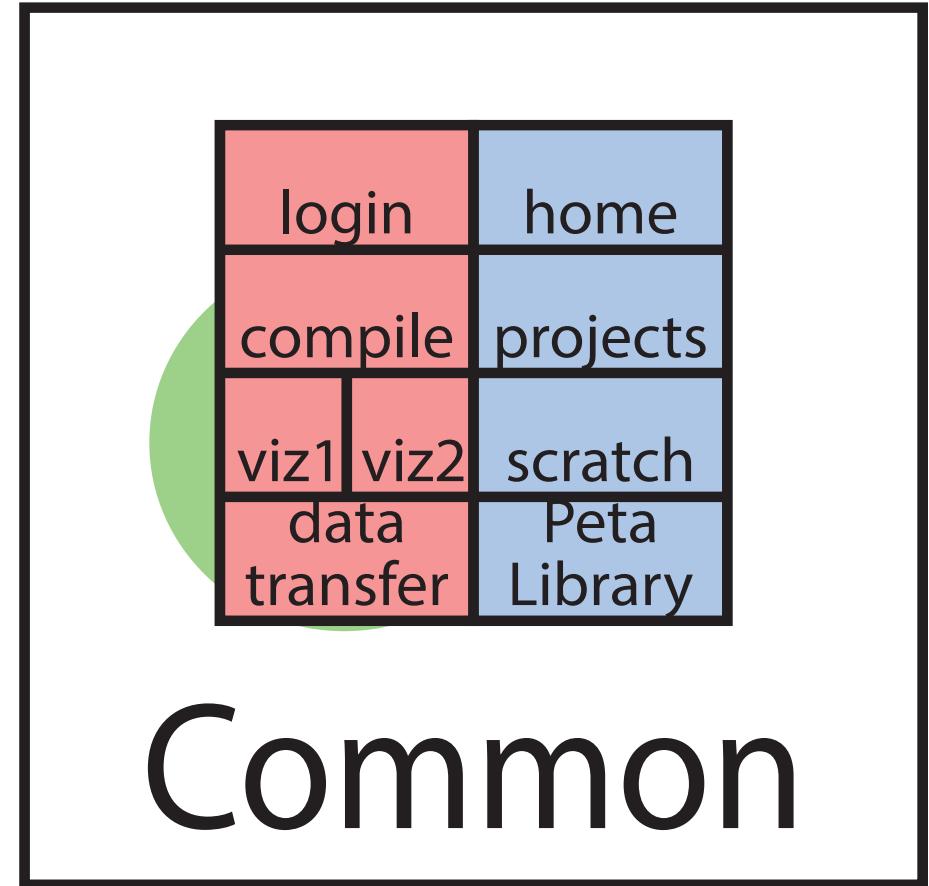
# Cluster architecture – login nodes.

- Used only for :
  - Submitting jobs
  - Editing scripts
  - Transferring data
- Not used for computing  
(compute software not accessible)
- Common to all CURC clusters
- 4 different nodes
  - Assigned to lowest traffic node
  - Can specify (if necessary)



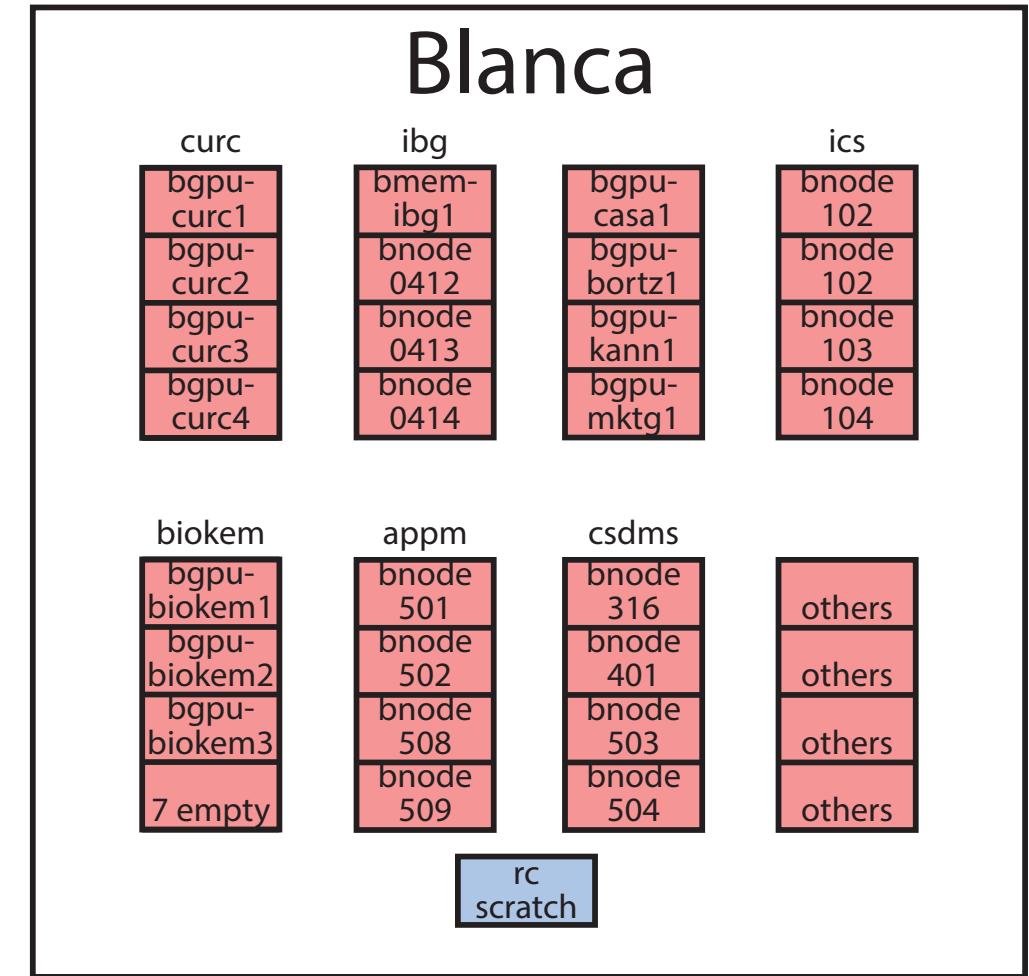
# Cluster architecture – viz nodes.

- 2 x RTX8000 GPU accelerated nodes meant for rendering graphics
- Can be used for computation, although won't work with SBGrid
- Can't be used to submit jobs
- Common to CURC clusters



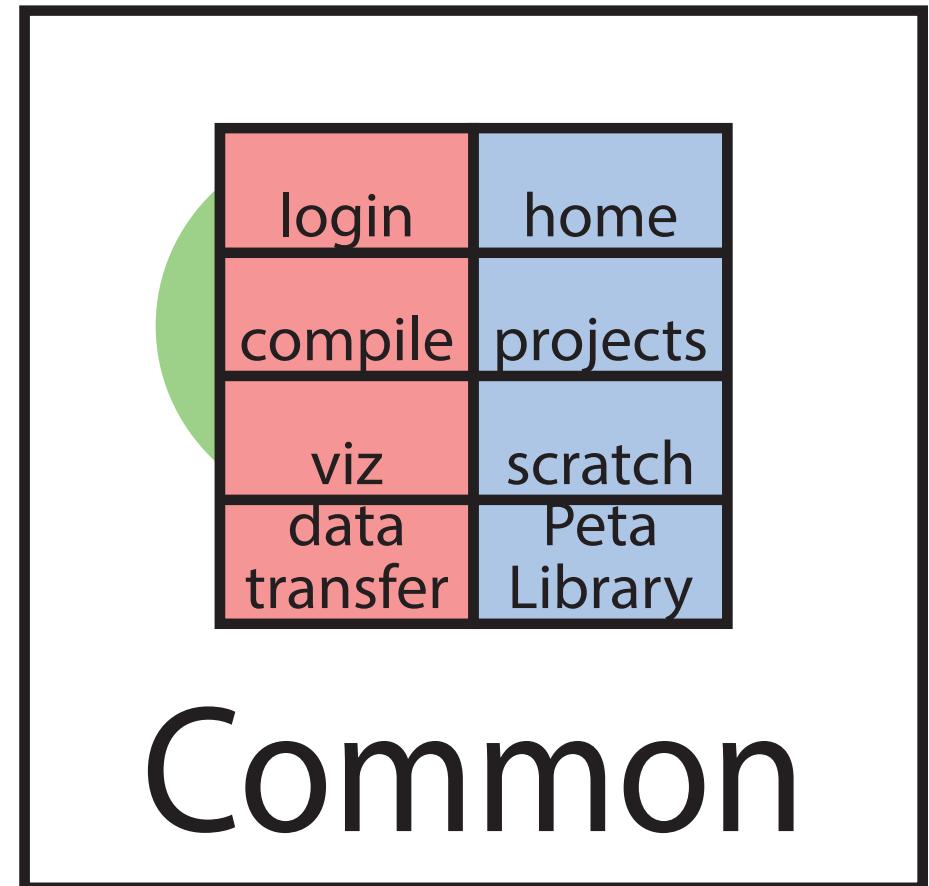
# Cluster architecture – compute nodes.

- 'Condo' computing cluster
- Buy nodes, get access to cluster
- Biokem nodes
  - 3 nodes (7xRTX6000, 2xA100)
  - Priority
  - 7d time limit
  - Consider contributing
- Blanca nodes
  - Lots of nodes (A100, RTX6000, P100, T4, V100)
  - Preemptable
  - 24hr time limit



# Cluster architecture – compile nodes.

- Used exclusively for compiling software
- Can see CURC SW stack to compile correct paths
- Could also compile software on compute nodes



# Logging on

# Logging on – OpenOnDemand.

- <https://ondemand.rc.colorado.edu>
- Web based remote desktop into CURC clusters
- Best way to access Blanca
  - Graphics rendered on cluster
  - Can rejoin/share session
  - Doesn't die if you lose connection
  - Can access from any computer with a web browser

The screenshot shows a login form for the University of Colorado Boulder's Federated Identity Service. The URL is "Log in to ondemand.rc.colorado.edu". There are two input fields: "Identikit Username" and "Identikit Password". Below the fields are "Log In" and "Advanced Settings..." buttons. A note at the bottom says: "Need help logging in? Quit or close all browser windows to exit completely. DO NOT bookmark this page! Bookmark the service homepage after logging in."

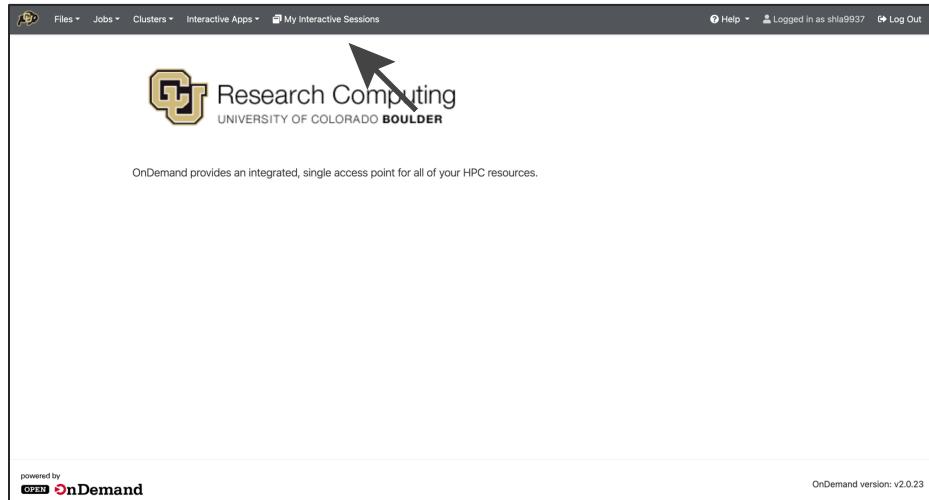
Enter credentials

The screenshot shows the Duo authentication selection screen. It features the CU logo and the text "Choose an authentication method". It lists four options: "Duo Push RECOMMENDED" (selected with a checked checkbox), "Call Me", "Passcode", and "Enter a Passcode". Below the options are links for "What is this?", "Add a new device", "My Settings & Devices", and "Need help?". At the bottom, it says "Secured by Duo". An arrow points to the "Send Me a Push" button.

Accept Duo push on phone

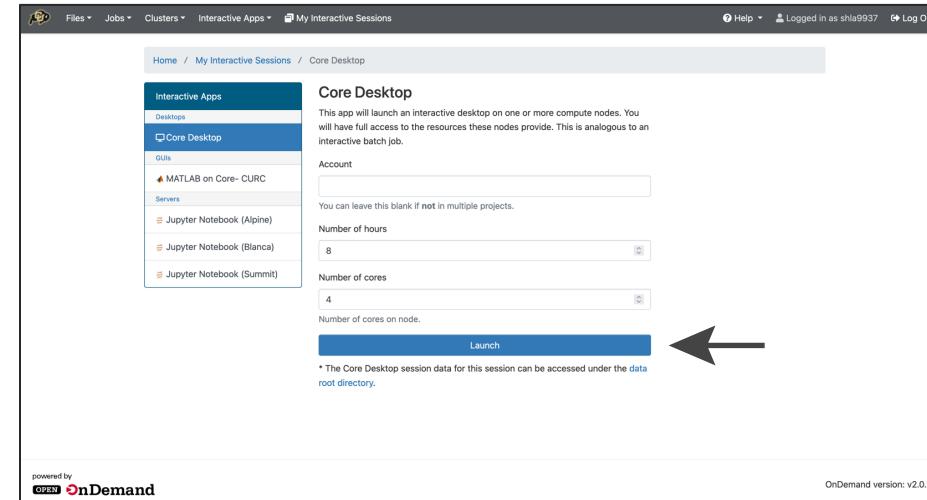
# Logging on – OpenOnDemand.

1



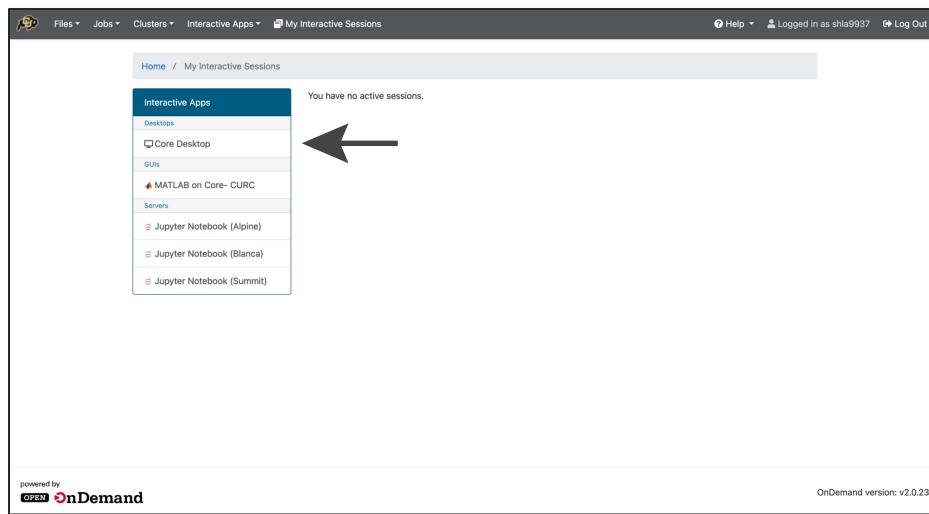
The screenshot shows the OpenOnDemand homepage. At the top, there is a navigation bar with links for 'Files', 'Jobs', 'Clusters', 'Interactive Apps', and 'My Interactive Sessions'. Below the navigation bar, the 'Research Computing' logo of the University of Colorado Boulder is displayed. A large arrow points to the logo. Below the logo, the text 'OnDemand provides an integrated, single access point for all of your HPC resources.' is visible. At the bottom of the page, it says 'powered by OPEN OnDemand' and 'OnDemand version: v2.0.23'.

3



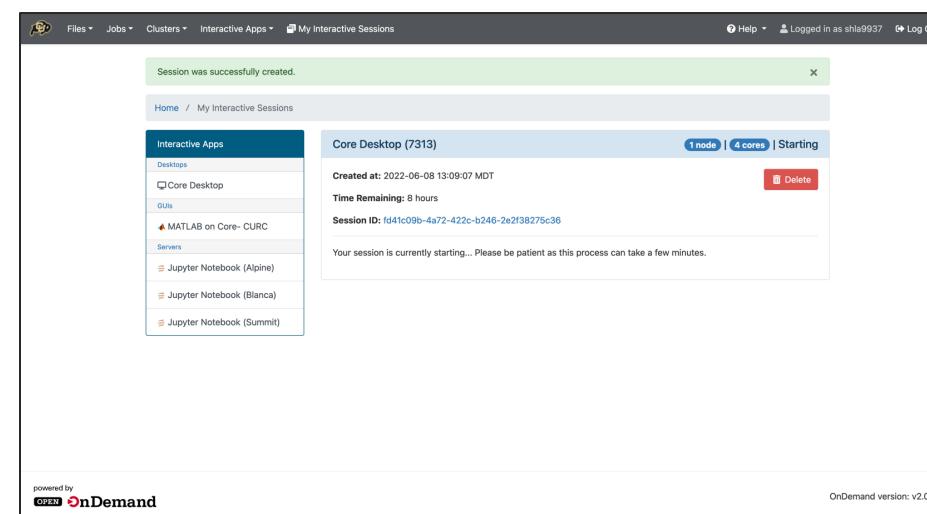
The screenshot shows the 'Core Desktop' session configuration page. The left sidebar lists 'Interactive Apps' such as 'Desktops', 'Core Desktop', 'GLUs', 'MATLAB on Core- CURC', 'Servers', 'Jupyter Notebook (Alpine)', 'Jupyter Notebook (Blanca)', and 'Jupyter Notebook (Summit)'. The main content area is titled 'Core Desktop' and contains a brief description: 'This app will launch an interactive desktop on one or more compute nodes. You will have full access to the resources these nodes provide. This is analogous to an interactive batch job.' It includes fields for 'Account' (a dropdown menu), 'Number of hours' (set to 8), and 'Number of cores' (set to 4). A large blue button labeled 'Launch' is at the bottom. A note below the button states: '\* The Core Desktop session data for this session can be accessed under the [data](#) root directory.' A large arrow points to the 'Launch' button.

2



The screenshot shows the 'My Interactive Sessions' page. The left sidebar is identical to the previous screenshot. The main content area displays a message: 'You have no active sessions.' A large arrow points to this message.

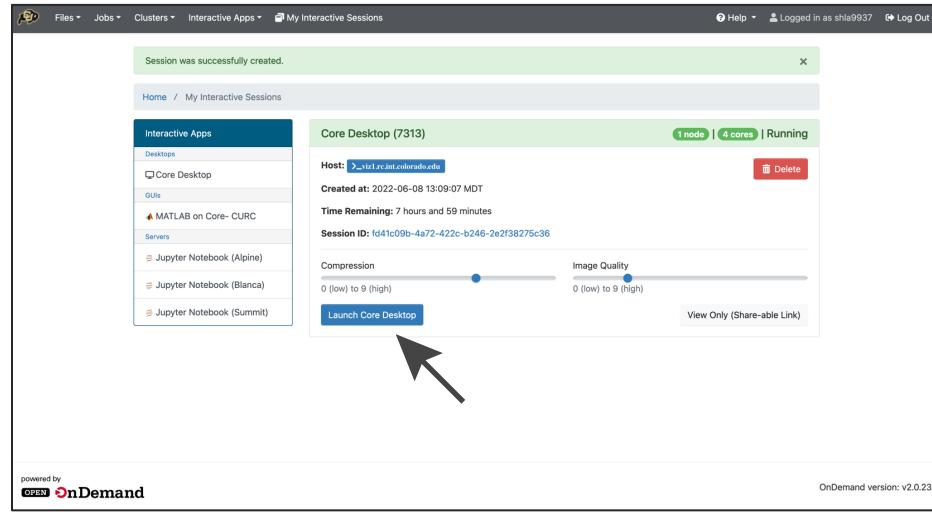
4



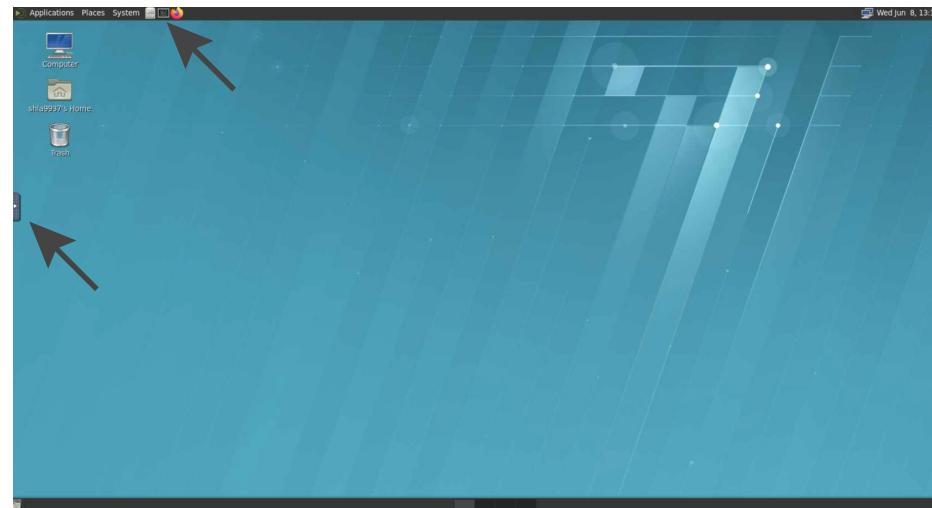
The screenshot shows the 'Core Desktop' session status page. The left sidebar is identical. The main content area shows a green success message: 'Session was successfully created.' Below it, the session details are listed: 'Core Desktop (7313)', '1 node | 4 cores | Starting', 'Created at: 2022-06-08 13:09:07 MDT', 'Time Remaining: 8 hours', and 'Session ID: f64fc09b-4a72-422c-b246-2e2f38275c36'. A note below states: 'Your session is currently starting... Please be patient as this process can take a few minutes.' A large arrow points to the session ID.

# Logging on – OpenOnDemand.

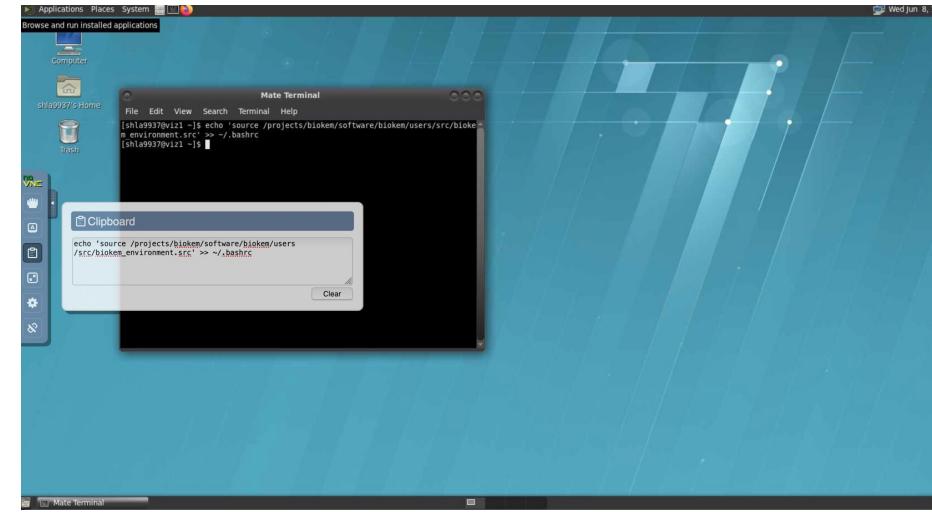
1



2



3



Only the first time you log on:  
- Paste into clipboard:

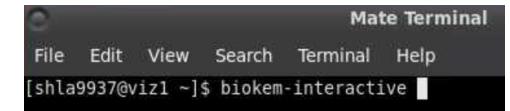
```
echo 'source  
/projects/biokem/software/biokem/users/  
src/biokem_environment.src' >> ~/.bashrc
```

- Use `ctrl+v` to paste into terminal
- Close and reopen terminal

# Logging on – interactive jobs.

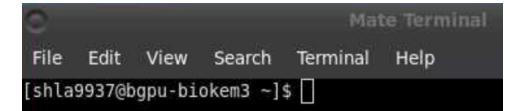
- `sinteractive`
  - SLURM command to start interactive job
  - Spawns job on free compute node
  - Default is 1CPU
- `biokem-interactive` - use this one most often
  - Spawns interactive jobs on biokem node
  - 1CPU
  - Use for starting GUIs that submit jobs
- `biokem-custom <arg1> <arg2>`
  - Spawns interactive job with <arg1> CPUs and <arg2> GPUs
  - Use for jobs that can't be submitted to SLURM scheduler
  - Use for graphics rendering off of Viz node
- `blanca-interactive` and `blanca-custom`
  - Useful for additional resources
  - Preemptable
- Need to start one of these to see software

Start interactive job



```
Mate Terminal
File Edit View Search Terminal Help
[shla9937@viz1 ~]$ biokem-interactive
```

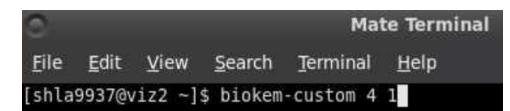
Now on compute  
node with 1CPU



```
Mate Terminal
File Edit View Search Terminal Help
[shla9937@bgpu-biokem3 ~]$
```

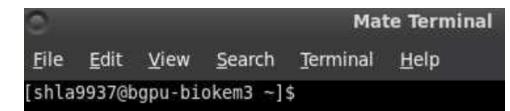
---

Start interactive job  
with 4 CPUs and 1  
GPU



```
Mate Terminal
File Edit View Search Terminal Help
[shla9937@viz2 ~]$ biokem-custom 4 1
```

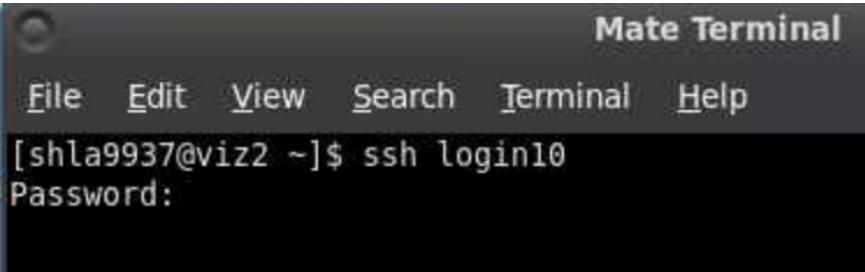
Now on compute  
node with 4 CPUs  
and 1 GPU



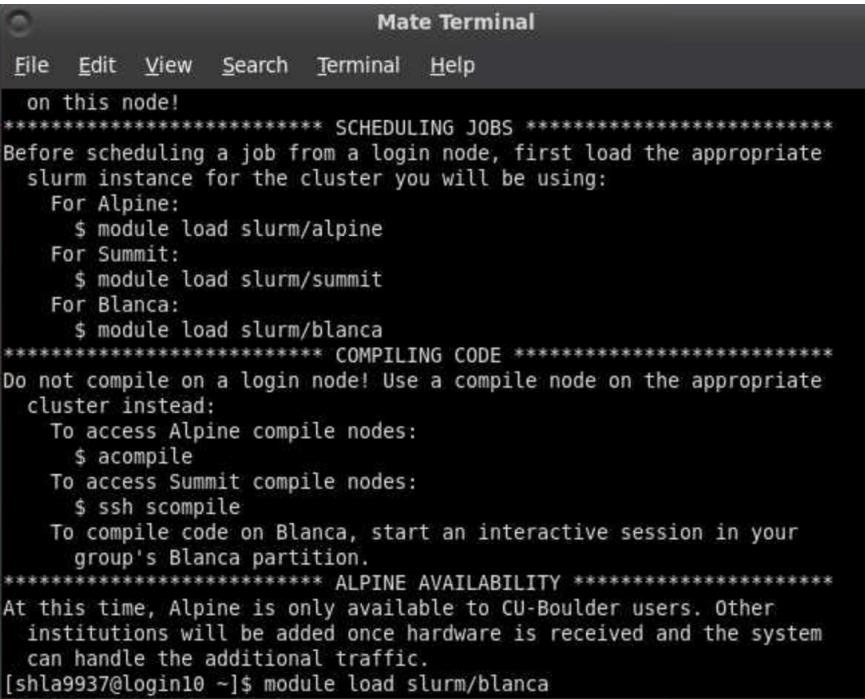
```
Mate Terminal
File Edit View Search Terminal Help
[shla9937@bgpu-biokem3 ~]$
```

# Logging on – login node.

- Submitting jobs
- Moving data
- Checking job status
- `ssh -Y login10`
  - Also login11, login12, login13
  - Enter password
  - Accept Duo push
- `module load slurm/blanca`
  - Starts blanca environment
  - Aliased as `blanca`



```
[shla9937@viz2 ~]$ ssh login10
Password:
```



```
on this node!
***** SCHEDULING JOBS *****
Before scheduling a job from a login node, first load the appropriate
slurm instance for the cluster you will be using:
  For Alpine:
    $ module load slurm/alpine
  For Summit:
    $ module load slurm/summit
  For Blanca:
    $ module load slurm/blanca
***** COMPILING CODE *****
Do not compile on a login node! Use a compile node on the appropriate
cluster instead:
  To access Alpine compile nodes:
    $ acompile
  To access Summit compile nodes:
    $ ssh scompile
  To compile code on Blanca, start an interactive session in your
  group's Blanca partition.
***** ALPINE AVAILABILITY *****
At this time, Alpine is only available to CU-Boulder users. Other
institutions will be added once hardware is received and the system
can handle the additional traffic.
[shla9937@login10 ~]$ module load slurm/blanca
```

SLURM

# SLURM – the queue.

- List of jobs currently running or waiting for resources to run
- The SLURM manager (a program) allocates resources based on priority and availability
- Once your job reaches the top of the queue and resources become available, your job will start
- If your job is preempted
  - 2mins to exit
  - Will requeue when resources open up
- `squeue` to see all jobs
  - `squeue -u <user>` to see only yours
  - `squeue | grep biokem` see biokem jobs
  - `scancel <job\_number>` to cancel job

```
File Edit View Search Terminal Help Mate Terminal
lsbla9937@login10:~$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
13876797 blanca test_ren misa5952 R 10:30:31 20 bhpc-c5-u7-[20-21],bhpc-c5-u19-[9-24],bhpc-c5-u31-[1-2]
13877783 blanca LOSR5LP maute R 3:25:22 1 bhpc-c5-u7-7
13877782 blanca LOSR5ZG maute R 3:27:00 1 bhpc-c7-u19-5
13877781 blanca LOSR5NHM maute R 3:29:52 1 bhpc-c5-u7-19
13877776 blanca sinterac maute R 3:46:15 1 bhpc-c7-u19-1
13877636 blanca 127NU7DD wach6115 R 9:47:42 12 bhpc-c7-u19-[2-4],bhpc-c7-u31-[8,10,13,17,20-24]
13877635 blanca 127NU6DD wach6115 R 11:27:55 12 bhpc-c7-u19-[8-19]
13877542 blanca 125NU6DD wach6115 R 14:23:16 12 bhpc-c7-u19-[20-24],bhpc-c7-u31-[1-7]
13877501_1121 blanca twis_pva luev6784 R 10:44 1 bhpc-c7-u31-14
13877501_1120 blanca twis_pva luev6784 R 12:16 1 bhpc-c7-u31-18
13877501_1119 blanca twis_pva luev6784 R 15:19 1 bhpc-c7-u31-19
13877501_1118 blanca twis_pva luev6784 R 15:49 1 bnode0413
13877501_1117 blanca twis_pva luev6784 R 22:57 1 bhpc-c7-u31-11
13877501_1116 blanca twis_pva luev6784 R 24:22 1 bhpc-c7-u31-12
13877501_1115 blanca twis_pva luev6784 R 25:21 1 bhpc-c7-u31-15
13877501_1114 blanca twis_pva luev6784 R 47:05 1 bhpc-c7-u7-21
13877501_1113 blanca twis_pva luev6784 R 55:57 1 bhpc-c7-u7-22
13877501_1112 blanca twis_pva luev6784 R 56:09 1 bhpc-c7-u7-20
13877501_1111 blanca twis_pva luev6784 R 56:40 1 bhpc-c7-u7-23
13877501_1110 blanca twis_pva luev6784 R 59:21 1 bhpc-c7-u7-24
13877501_1109 blanca twis_pva luev6784 R 1:13:14 1 bgpu-mktg1
13877501_1108 blanca twis_pva luev6784 R 1:16:57 1 bgpu-mktg1
13877501_1107 blanca twis_pva luev6784 R 2:04:03 1 bhpc-c7-u31-16
13877501_1106 blanca twis_pva luev6784 R 2:13:34 1 bhpc-c7-u31-9
13877501_1105 blanca twis_pva luev6784 R 2:17:06 1 bgpu-bortzl
13877501_1104 blanca twis_pva luev6784 R 2:21:34 1 bgpu-biokem1
13877501_1100 blanca twis_pva luev6784 R 2:36:18 1 bgpu-casal
13877501_1096 blanca twis_pva luev6784 R 2:52:07 1 bhpc-c7-u7-19
13877501_1095 blanca twis_pva luev6784 R 3:13:08 1 bgpu-kann1
13877984 blanca bunchofs emsa1620 R 1:37 1 bnode0408
13877981 blanca bunchofs emsa1620 R 2:38 1 bhpc-c5-u7-8
13877982 blanca bunchofs emsa1620 R 2:38 1 bnode0303
13877983 blanca bunchofs emsa1620 R 2:38 1 bnode0308
13877980 blanca bunchofs emsa1620 R 3:08 1 bhpc-c5-u7-23
13877979 blanca bunchofs emsa1620 R 4:39 1 bnode0309
13877978 blanca bunchofs emsa1620 R 5:40 1 bhpc-c5-u7-9
```

# SLURM – sbatch scripts/allocating resources.

- The way to run jobs
- Tells the SLURM manager which resources to allocate
- Can write your own if you know commands
- Submit with `sbatch <script>`
- Gulls like RELION or CryoSPARC will submit these for you
- Examples:  
`/projects/biokem/software/biokem/users/example\_sbatch\_scripts`

```
#!/bin/bash
#SBATCH --partition=blanca-biokem
#SBATCH --qos=blanca-biokem
#SBATCH --account=blanca-biokem
#SBATCH --job-name=my_biokem_job
#SBATCH --nodes=1
#SBATCH --gres=gpu:0
#SBATCH --ntasks=1
#SBATCH --mem=8gb
#SBATCH --time=24:00:00
#SBATCH --output=/home/%u/slurmfiles_out/slurm_%j.out    #output file of job - useful for checking progress of job
#SBATCH --error=/home/%u/slurmfiles_err/slurm_%j.err    #error file of job - useful for debugging
```

```
#!/bin/bash
#SBATCH --partition=blanca-biokem
#SBATCH --qos=blanca-biokem
#SBATCH --account=blanca-biokem
#SBATCH --job-name=md_heat_and_eq
#SBATCH --nodes=1
#SBATCH --gres=gpu:1
#SBATCH --ntasks=1
#SBATCH --mem=128gb
#SBATCH --time=24:00:00
#SBATCH --output=/home/%u/slurmfiles_out/slurm_%j.out
#SBATCH --error=/home/%u/slurmfiles_err/slurm_%j.err

module load amber/v18
NAME='bd0055_DNA_0M_KCL'

pmemd.cuda -O -i heat.in \
-o ./outputs/${NAME}_heat.out \
-p ./${NAME}_buffer.prmtop \
-c ./restarts/${NAME}_min2.rst \
-r ./restarts/${NAME}_heat.rst \
-x ./trajectories/${NAME}_heat.nc \
-inf ./mdinfo/${NAME}_heat.mdinfo \
-ref ./restarts/${NAME}_min2.rst

pmemd.cuda -O -i release1.in \
-o ./outputs/${NAME}_release1.out \
-p ./${NAME}_buffer.prmtop \
-c ./restarts/${NAME}_heat.rst \
-r ./restarts/${NAME}_release1.rst \
-x ./trajectories/${NAME}_release1.nc \
-inf ./mdinfo/${NAME}_release1.mdinfo \
-ref ./restarts/${NAME}_heat.rst

pmemd.cuda -O -i release2.in \
-o ./outputs/${NAME}_release2.out \
-p ./${NAME}_buffer.prmtop \
-c ./restarts/${NAME}_release1.rst \
```

```
#!/bin/bash
#SBATCH --partition=blanca-biokem
#SBATCH --qos=blanca-biokem
#SBATCH --account=blanca-biokem
#SBATCH --job-name=make_diffraction_movie
#SBATCH --nodes=1
#SBATCH --ntasks=4
#SBATCH --mem=32gb
#SBATCH --time=24:00:00
#SBATCH --output=/home/%u/slurmfiles_out/slurm_%j.out
#SBATCH --error=/home/%u/slurmfiles_err/slurm_%j.err

#make sure to launch this from an SBGrid environment

#Where .img's are located. Don't add slash on end of path.
DIR_NAME='/pl/active/LugerLab-EM/people/shawn/crystallography/synchrotron/bdello_dna/20220505/xtali'

#Where you want to put jpeg's. Don't add slash on end of path.
NEW_DIR='/pl/active/LugerLab-EM/people/shawn/crystallography/movies/20220505_xtali'

for IMAGE in ${DIR_NAME}/*.*img
do
    JPEG_NAME=$(basename $IMAGE .img)
    echo "adxv -sa '$IMAGE' '$NEW_DIR'/${JPEG_NAME}" >> adx_commands.txt
done

while read -r
do
    eval "$REPLY"
done < adx_commands.txt

#stitch into movie
ffmpeg -i xtali_5_%05d.jpeg -r 60 -vf scale=-1:1080 20220505_xtali.mp4
```

# SLURM – requesting specific hardware.

- `sinfo -o "%N %G"``
  - shows what GPUs are in the cluster
  - `sinfo` can be used to check status of nodes
- `#SBATCH --gres=gpu:<type>:<number>`
  - starts your job on a node with specific hardware
  - Ex: `#SBATCH --gres=gpu:a100:2`
  - Will not start if request more than possible

```
[shla9937@bgpu-biokem3 ~]$ sinfo -o "%N %G"
NODELIST GRES
bgpu-biokem1 gpu:rtx6000:4
bgpu-biokem2 gpu:a100:2
bgpu-biokem3 gpu:rtx6000:3
bgpu-bortz1 gpu:t4:1
bgpu-casa1,bgpu-papp1 gpu:v100:1
bgpu-curc[1-4] gpu:a100:3
bgpu-dhl1 gpu:p100:2
bgpu-ivc,bgpu-kann1 gpu:v100:4
bgpu-mktg1 gpu:p100:1
```

```
#SBATCH --nodes=1
#SBATCH --gres=gpu:0
#SBATCH --ntasks=1
```

# SLURM – when to use blanca vs biokem.

## Blanca

- Don't need immediate output
- If there is a long queue time on the biokem nodes
- Ex: Molecular dynamics, motion correction, tomogram reconstruction, etc.

## BioKEM

- Job can't be split into <24hr chunks
- Job can't be interrupted (lots of jobs are fine)
- Ex: OTF processing, long jobs

Data management

# Data management – storage.

- Home (/home/\$USER or ~)
  - 2GB
  - For personal configuration files
  - Backed up frequently (/home/\$USER/.snapshot)
- Projects (/projects/\$USER)
  - 250GB
  - For personal software or smaller datasets
  - Backed up frequently (/projects/\$USER/.snapshot)
- PetaLibrary (/pl/active/<group>)
  - ~\$45/TB/year
  - Allocated per lab
  - Store large datasets or compute from here
  - Backed up frequently (/pl/active/<group>/.snapshot)
- RC Scratch (/rc\_scratch/\$USER or `scratch`)
  - ~10TB/person
  - Not backed up, purged after 90d
  - Run jobs that will create large intermediate files, move results to PL
- SLURM scratch (\$SLURM\_SCRATCH)
  - Variable size
  - Local scratch storage on nodes
  - Created with job allocation
  - Deleted upon job exit

# Data management – internal transfer.

- `cp <from> <to>` - copy file
  - With CURC environment
  - Can overwrite
  - `cp -r <from> <to>` for directories
- `mv <from> <to>` - moves file/folder
  - Renames if within same device
  - Copies and deletes if moving between devices
- `rsync -auP <from> <to>` - syncs folder or file with second location
  - Only transfers new files/edits
  - Quick
- `rm <file>` - deletes file
  - `rm -r <directory>` for directory
  - Permanent, can recover if a snapshot exists

# Data management – external transfer.

- `rsync -auP <from> remote\_host:<to>`
- `scp <from> remote\_host:<to>`
  - Same as copy, but secure for interhost transfer
  - Need `-r` for folders
- Globus
  - Create endpoint on your computer
  - Use `CU Boulder Research Computing` as transfer endpoint
- AWS bucket
- Other options

Software

# Software – CURC software stack.

- Any node other than login: `module avail`
- Maintained by CURC
  - Submit ticket to get something installed
  - Some applications aren't visible until right compilers are loaded
- Module commands put the correct dependencies and paths in your environment:
  - `module load <app>`
  - `module unload <app>`
  - `module list` shows loaded apps
  - `module purge` unloads all apps
  - `module spider` shows all possible apps
- Can add own, but not trivial
- BioKEM has a few, but SBGrid obviates

```
[shla9937@viz2 ~]$ module avail
-----
-- /projects/biokem/modules --
imod/4.12.20 (D)    peet/1.16.0 (D)    relion/v3.1.3 (D)
imod/4.12.20 (D)    peet/1.16.0 (D)    relion/v3.1.3 (D)

-- /projects/shla9937/modules --
adxv/v1.9.14 (D)   amber/v18     amber/v20 (D)
adxv/v1.9.14 (D)   amber/v18     amber/v20 (D)

-- Compilers --
gcc/6.1.0  (D)    intel/16.0.3 (m)  intel/20.2 (m)
gcc/8.2.0          intel/17.0.0 (m)  pgi/16.5
gcc/10.2.0         intel/17.4   (m,D) pgi/18.4  (D)

-- Independent Applications --
R/3.3.0           mathematica/11.1.0 (D)
R/3.4.3           matlab/R2016b
R/3.5.0           (D)      matlab/R2017b
R/4.0.4           matlab/R2018b (D)
allinea/6.0.4      (m)      matlab/R2019b
anaconda/2020.11   matlab/R2020b
arm-forge/19.1.3   (m)      matlab/R2021b
autotools/2.69     maven/3.8.1
```

# Software – the BioKEM environment.

- Add environment to startup file
  - Biokem environment
  - Non-SBGrid users
- Enables SBGrid access
- BioKEM installed modules
- Executables (biokem-interactive, etc.)
- Aliases (sbgrid, scratch, blanca, etc.)
- Scripts (RELION submission, etc.)

# Software – starting SBGrid.

- Start interactive job (biokem-interactive)
- `sbgrid`
- Run application as normal

```
[shla9937@bgpu-biokem3 ~]$ sbgrid
*****
Software Support by SBGrid (www.sgrid.org)
*****
Your use of the applications contained in the /programs directory constitutes
acceptance of the terms of the SBGrid License Agreement included in the file
/programs/share/LICENSE. The applications distributed by SBGrid are licensed
exclusively to member laboratories of the SBGrid Consortium.
Run sbgrid-accept-license to remove the above message.
*****
SBGrid was developed with support from its members, Harvard Medical School,
HHMI, and NSF. If use of SBGrid compiled software was an important element
in your publication, please include the following reference in your work:

Software used in the project was installed and configured by SBGrid.
cite: eLife 2013;2:e01456, Collaboration gets the most out of software.
*****
SBGrid installation last updated: 2022-05-11
Please submit bug reports and help requests to: <bugs@sgrid.org> or
                                                <https://sgrid.org/bugs>
For additional information visit https://sgrid.org/wiki
*****
Configuration options in effect for this shell:
Variable          Setting          Configuration
RELION_X          4.0-beta2_cull   personal
EMAN2_X           20211129         personal
*****
```

# Software – specifying SBGrid versions.

- `sbgrid-info -l <app>` check versions of application
- Version variable
  - Resets with new terminal
  - `RELION\_X=4.0-beta2\_cu11`
- Config file
  - Sets versions on startup
  - `touch ~/.sbgrid.conf`
  - New entry for each application:  
`RELION\_X=4.0-beta2\_cu11  
EMAN2\_X=20211119  
IMOD=4.10.51`
- Modules
  - Added with environment script
  - Can use inside SBGrid shell to specify versions

```
[shla9937@bgpu-biokem2 ~]$ sbgrid-info -l relion
Version information for: /programs/x86_64-linux/relion

Default version:          3.1.3_cu10.2
In-use version:           4.0-beta2_cu11

Installed versions:        3.1.3_cu10.2 4.0-beta2_9b23e50_cu11 4.0-beta2_9
b23e50_cu10.2_amd 4.0-beta2_9b23e50_cu10.2 4.0-beta2_9b23e50_cu9.2 4.0-beta2_9b23e5
0_cu11.5 4.0-beta2_cu11.5 4.0-beta2_cu11 4.0-beta2_cu11.4.1 4.0-beta2_cu10.2_amd 4.
0-beta2_cu10.2 4.0-beta2_cu9.2 4.0-beta_cu11.4.1 4.0-beta_cu10.2_amd 4.0-beta_cu10.
2 4.0-beta_cu9.2 3.1.3_cu11.4.1 3.1.3_cu8.0 3.1.2_cu9.2 3.1.2_cu11.1 3.1.2_cu9.2_no
SAGD 3.1.2_cu8.0 3.1.2 3.1.1_cu9.2 3.1.1 3.1.0_cu8.0_compat 3.1.0_cu8.0 3.1.0 3.0.8_
_cu9.2 3.0.8_cu8.0 3.0.8 3.0.4_cu9.0 2.1_cu8.0 1.4-randomphase3d 1.4b 1.4 1.3
Other available versions:   4.0-beta2_9b23e50_cu11 4.0-beta2_9b23e50_cu10.2
_amd 4.0-beta2_9b23e50_cu10.2 4.0-beta2_9b23e50_cu9.2 4.0-beta2_9b23e50_cu11.5 4.0-
beta2_cu11.5 4.0-beta2_cu11 4.0-beta2_cu11.4.1 4.0-beta2_cu10.2_amd 4.0-beta2_cu10.
2 4.0-beta2_cu9.2 4.0-beta_cu11.4.1 4.0-beta_cu10.2_amd 4.0-beta_cu10.2 4.0-beta_cu
9.2 3.1.3_cu11.4.1 3.1.3_cu8.0 3.1.2_cu9.2 3.1.2_cu11.1 3.1.2_cu9.2_noSAGD 3.1.2_cu
8.0 3.1.2 3.1.1_cu9.2 3.1.1 3.1.0_cu8.0_compat 3.1.0_cu8.0 3.1.0 3.0.8_cu9.2 3.0.8_
cu8.0 3.0.8 3.0.4_cu9.0 2.1_cu8.0 1.4-randomphase3d 1.4b 1.4 1.3
Overrides use this shell variable: RELION_X
```

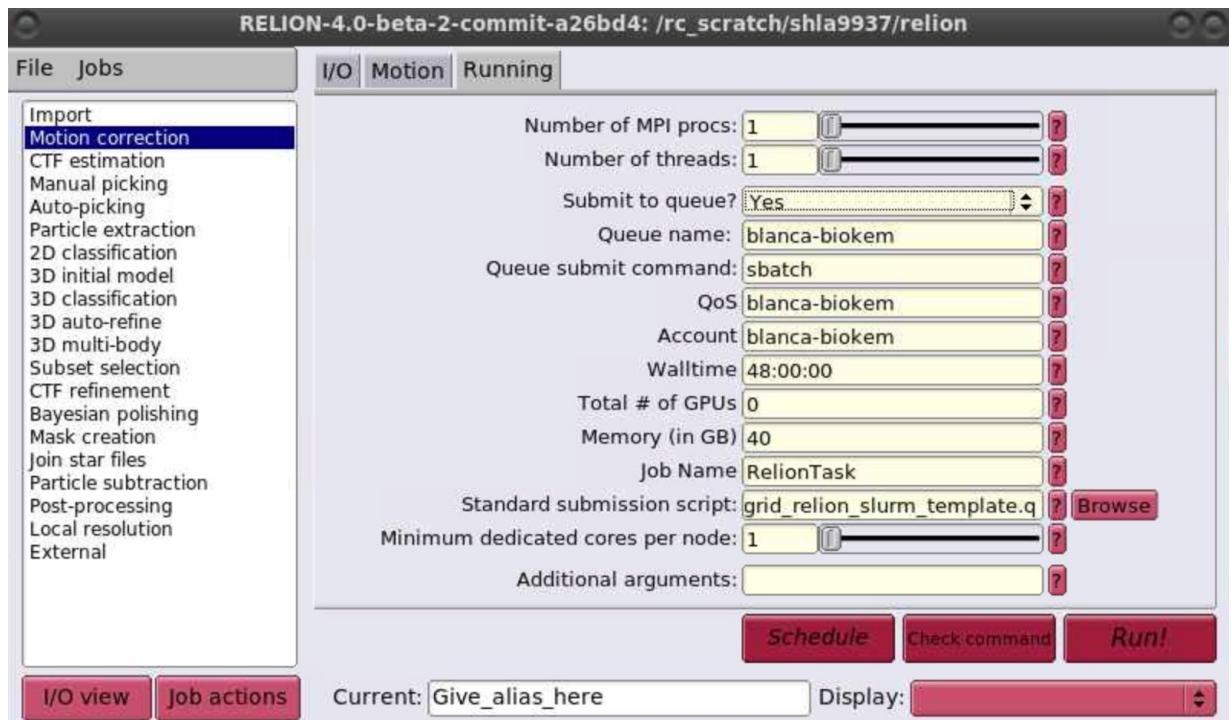
# Software – requesting new software.

- SBGrid
  - Updated frequently
  - Current versions, plus a few older ones
  - If you need specific version not installed, email me
- Non-SBGrid
  - Fill out help ticket with CURC
  - Should be able to install most apps

Practical examples.

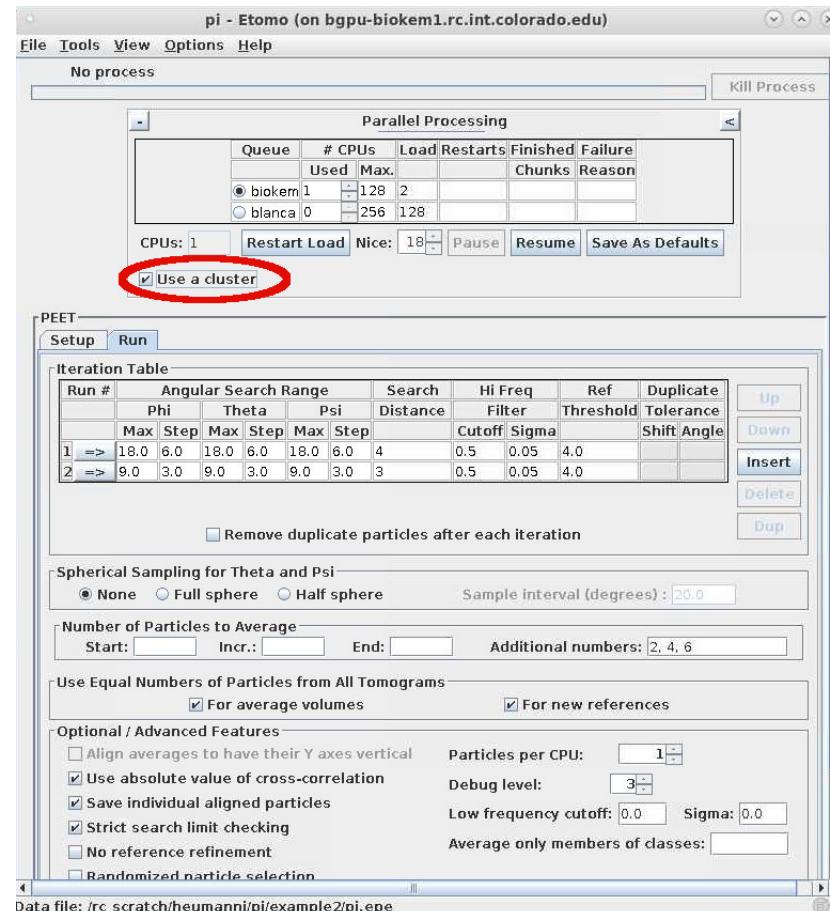
# Practical examples – RELION.

- `biokem-interactive`
- `sbgrid`
- `relion`
- Submit to queue
- Can also run:
  - Queue-name: blanca
  - QoS: preemptable
  - Walltime: 24:00:00



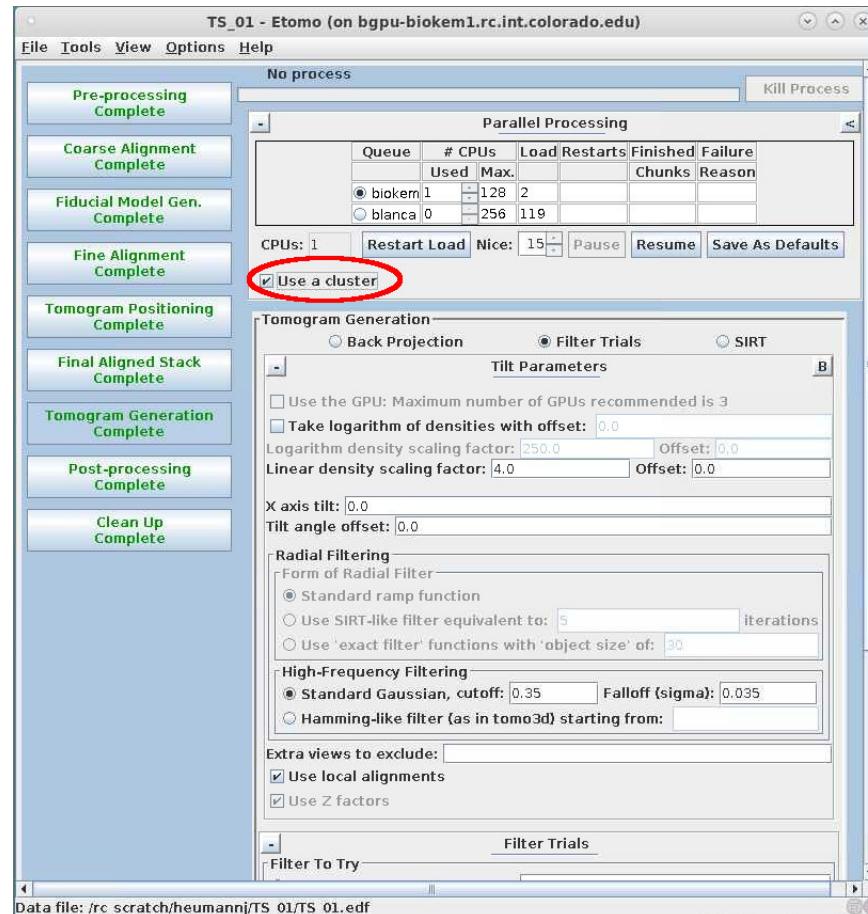
# Practical examples – PEET.

- `biokem-interactive`
- `sbgrid`
- `peet`
- Use a cluster



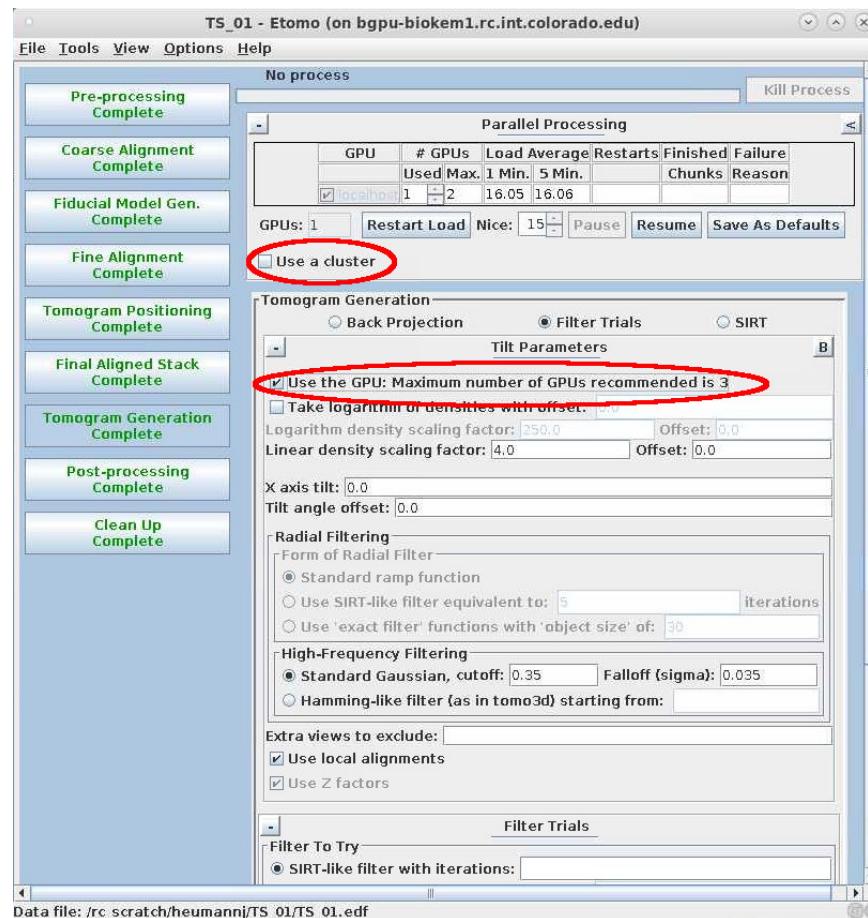
# Practical examples – IMOD-CPU.

- `biokem-interactive`
- `sbgrid`
- `etomo`
- `Use a cluster`



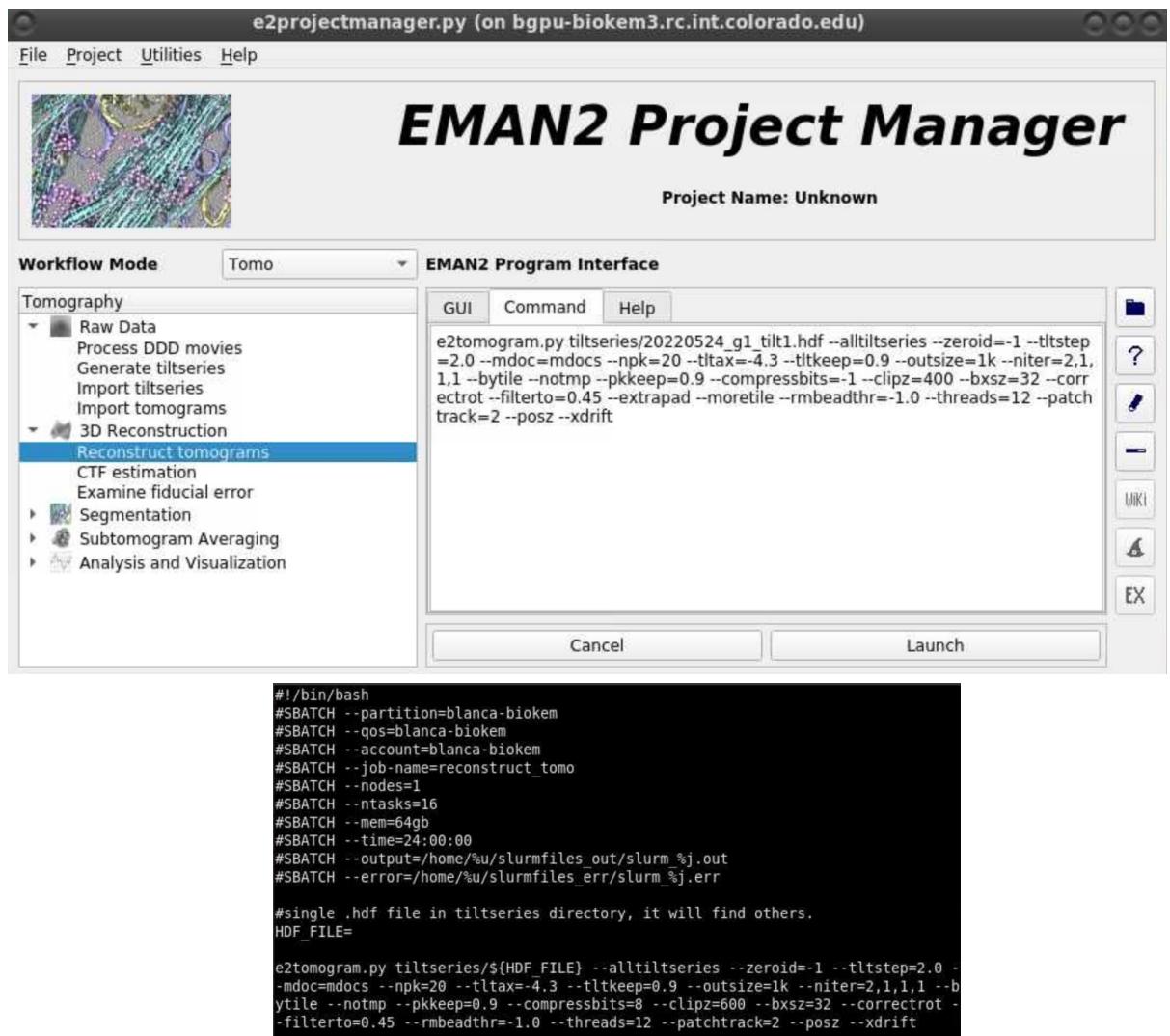
# Practical examples – IMOD-GPU.

- `biokem-custom 4 1`
- `sbgrid`
- `etomo`
- Don't `Use a cluster`
- `Use a GPU`
- Need to keep terminal alive while running job
- Exit job when done



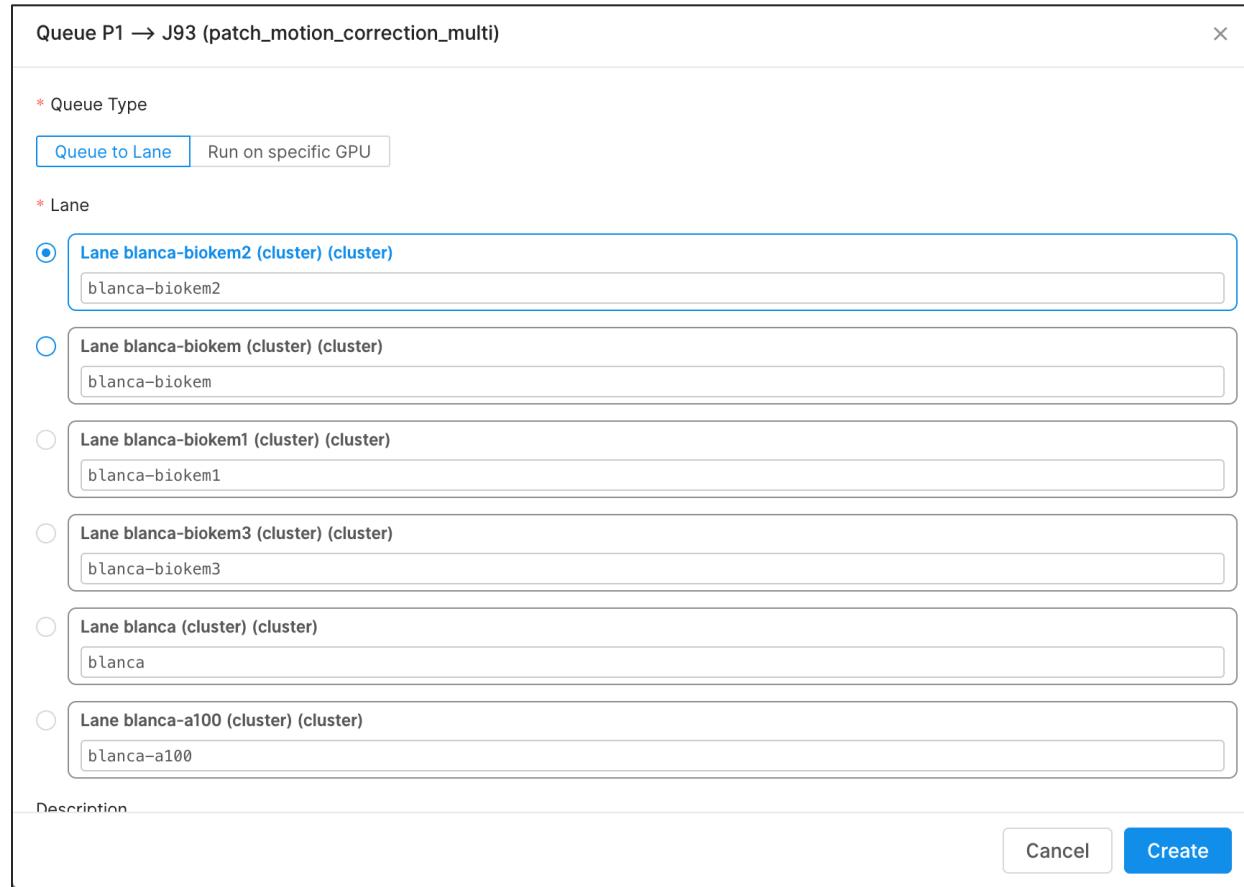
# Practical examples – EMAN2.

- `biokem-interactive`:
  - `e2projectmanager.py`
  - Example sbatch scripts in  
/projects/biokem/software/biokem/users/example\_sbatch\_scripts/eman2
- `biokem-custom 12 1`:
  - `e2projectmanager.py`
  - Run jobs interactively, keeping terminal open
  - remember to exit job when done



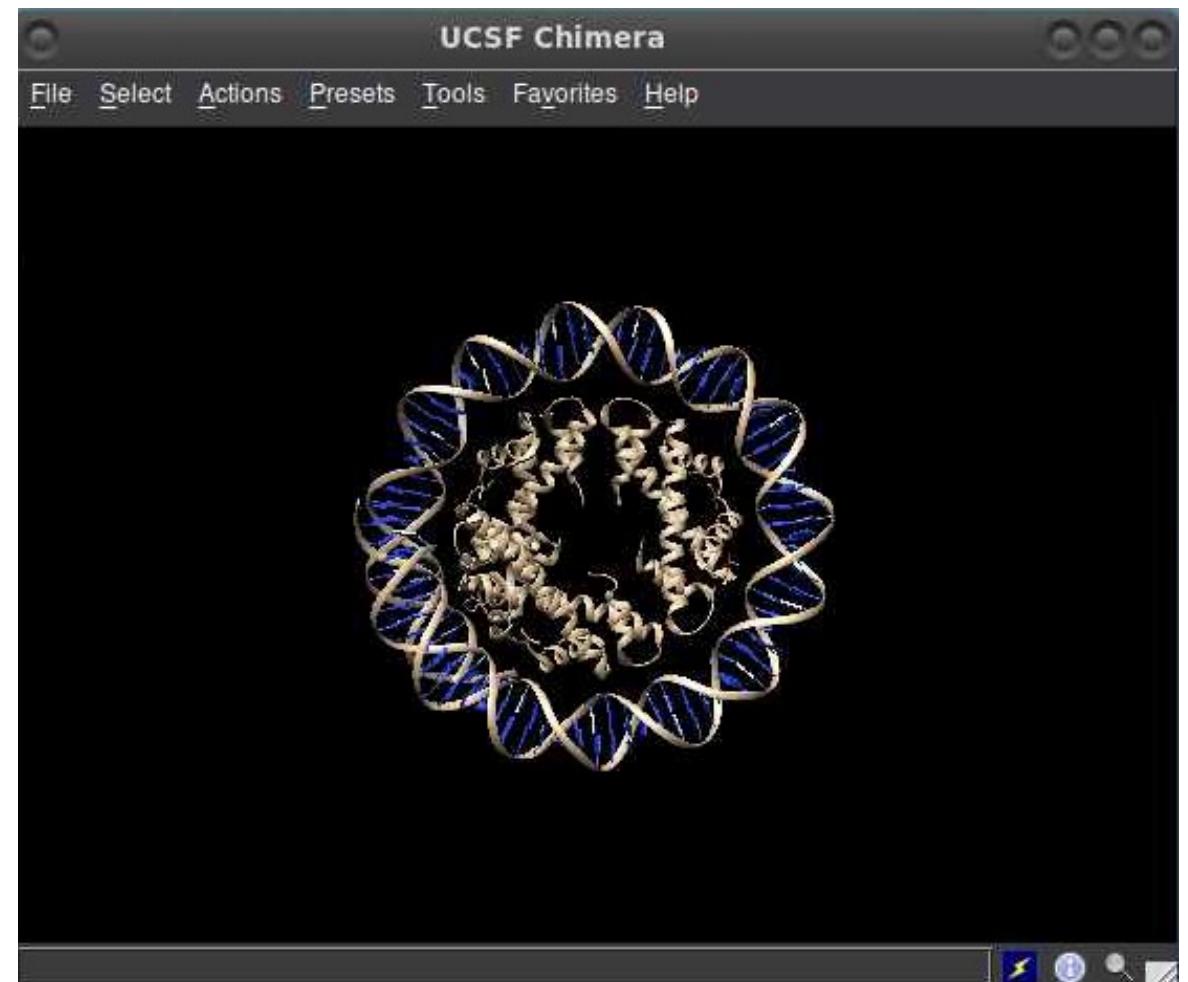
# Practical examples – CryoSPARC.

- Need to build VM for each lab
- `cryosparc` command will be put in path
  - Set up all behind the scenes stuff
  - Open browser in OpenOnDemand to correct page
- `Queue to lane`
  - blanca
  - biokem
- Prototype works, but need to iron out kinks



# Practical examples – Chimera/X, VMD.

- On compute nodes
  - Use SBGrid's version of Chimera/X
  - Or CURC's version of ChimeraX
  - VMD: allocate a GPU
- On Viz node use CURC's
  - `module load chimeraX`
  - Good place to use



Summary workflow.

# Summary workflow.

1. OpenOnDemand
2. `biokem-interactive`
3. `sbgrid`/other software
4. SLURM(`sbatch`)
5. Help
  - Biokem rtds (<https://biokem-user-guide.readthedocs.io>)
  - CURC rtds (<https://curc.readthedocs.io>)
  - CURC help (<https://www.colorado.edu/rc/userservices/contact>)
  - Program documentation (<https://sbgrid.org/software/>)
  - YouTube
  - Google forums

