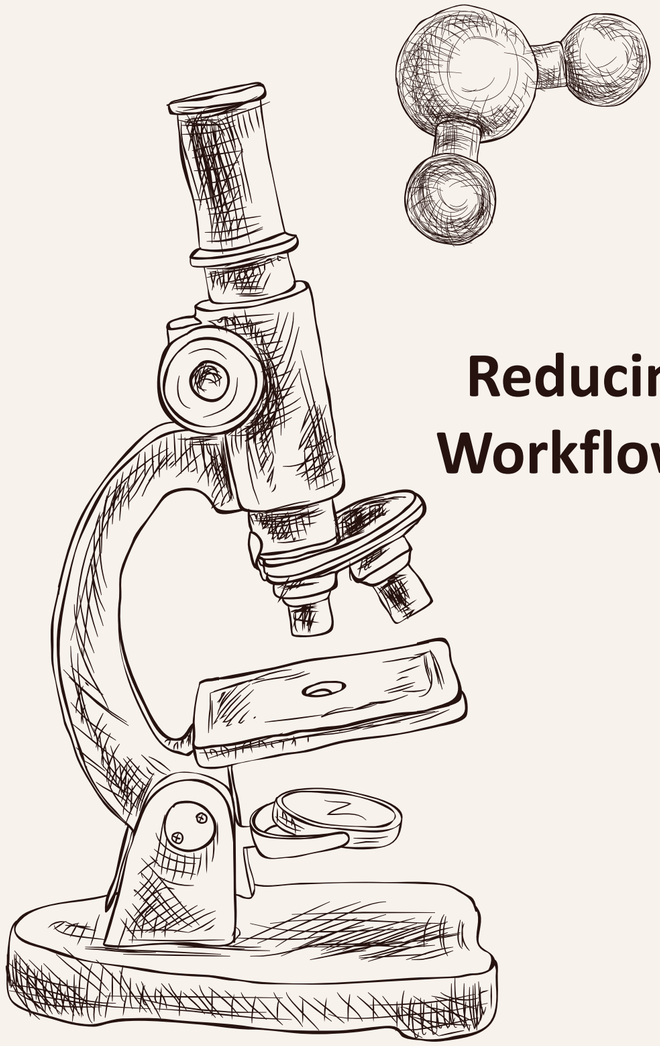
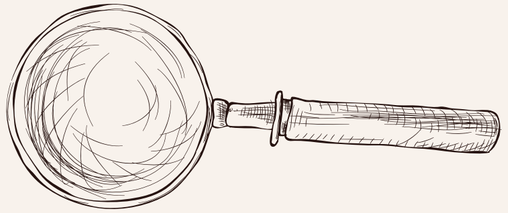
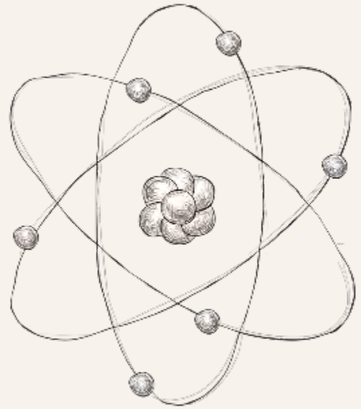
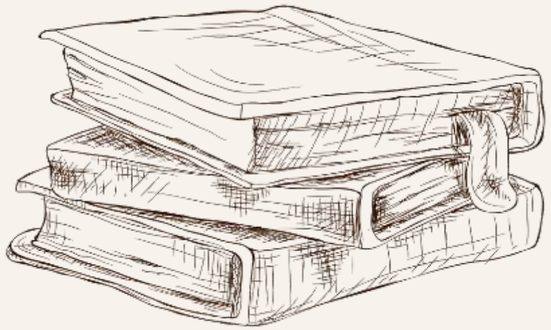






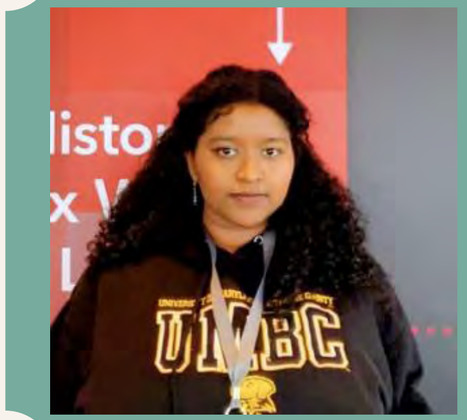
Reducing the Barrier of Entry for GPU Accelerated Workflows using Open OnDemand and Charliecloud Containers in HPC

Presented by GPU Unicorns



The Team

Abhinav Kotta
Cornell University
abhinavkotta.io@gmail.com
(689)-209-8517



Beamlak Bekele
University of Maryland Baltimore County
bekbeamlak@gmail.com
(240)-423-9786

Kelsey Tirado
Rochester Institute of Technology
kelseytirado3@gmail.com
(631)-807-9916

Introducing

Dr. Shirley!

She has PhD in Theoretical Elementary Physics and just started working at LANL. She needs the resources on our cluster to run her simulation

….but she's never used an HPC cluster before

What she N[illegible]ds to do

In order to run her simu[illegible]

- **Slurm workload m[illegible]**
  - *Creating jobs*
  - *Allocating the p[illegible]*
  - *Creating a work[illegible]*
- **The file system**
  - *She needs to e[illegible] readily available*
- **Installation and C[illegible]**
  - *She needs to b[illegible] her simulation*

slurm workload manager

Onboarding Made Easy

- Doesn't have to learn ***terminal commands***
- Doesn't have to learn ***Slurm***
- Makes use of ***existing*** scientific workflows

Open OnDemand

Open OnDemand (OOD) browser-based interface to access a cluster through the web

Now all Shirley needs to do is navigate to the custom OOD webpage where she has full access to the LANL cluster

Home Directory

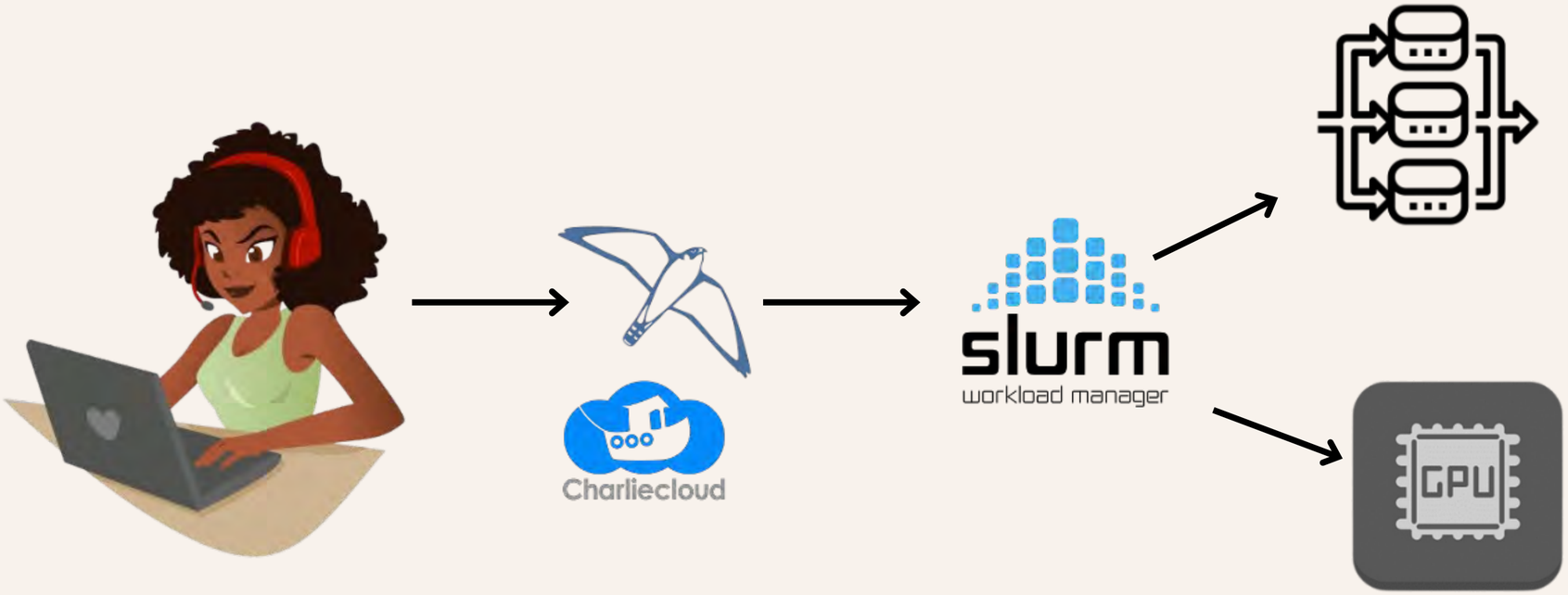
gpu-unicorn Shell Access

Active Jobs

System Status

Theoretical Physics to HPC

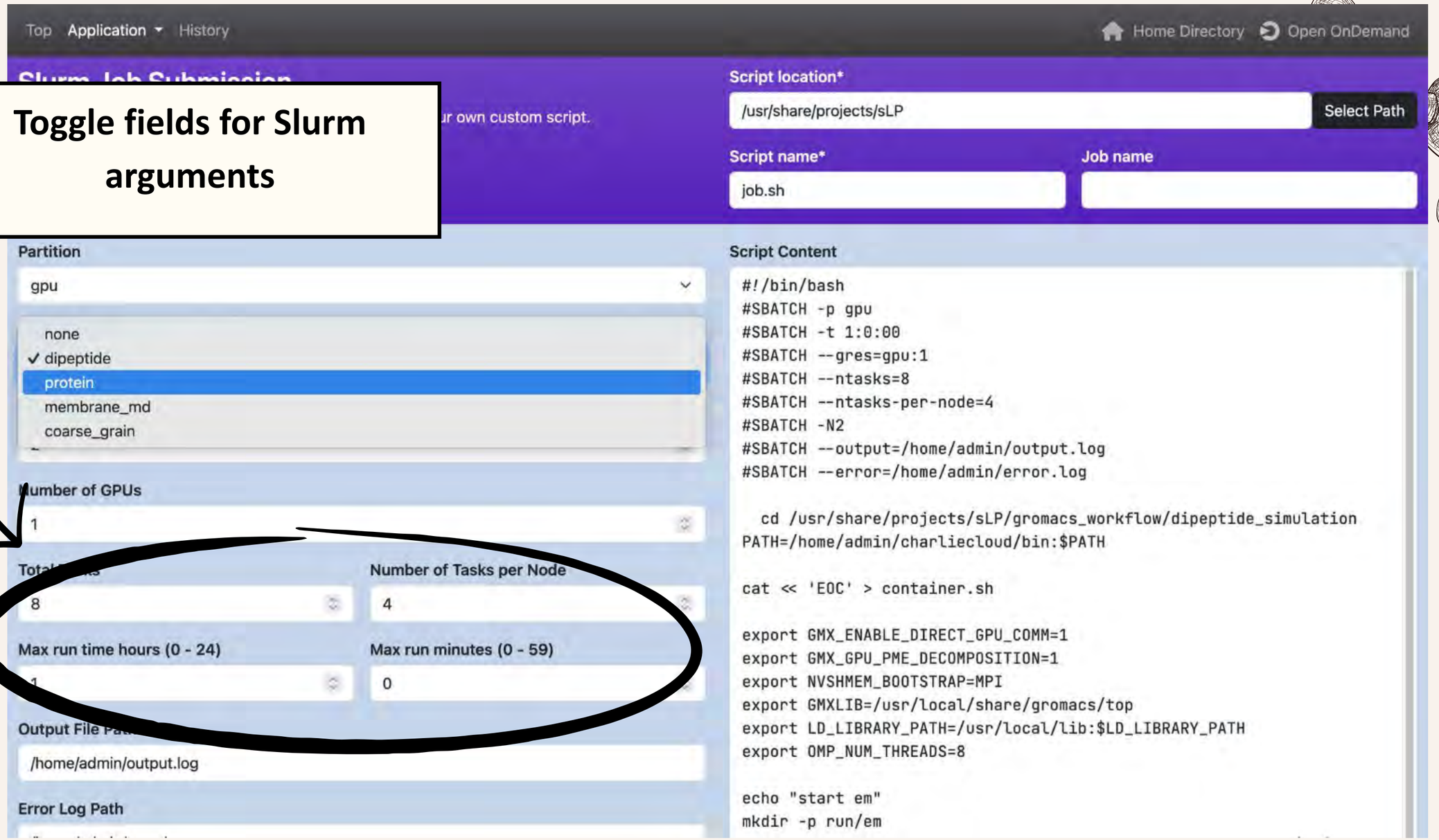
With her expertise in theoretical elementary physics, Shirley is leveraging GROMACS to model atomic-level interactions.



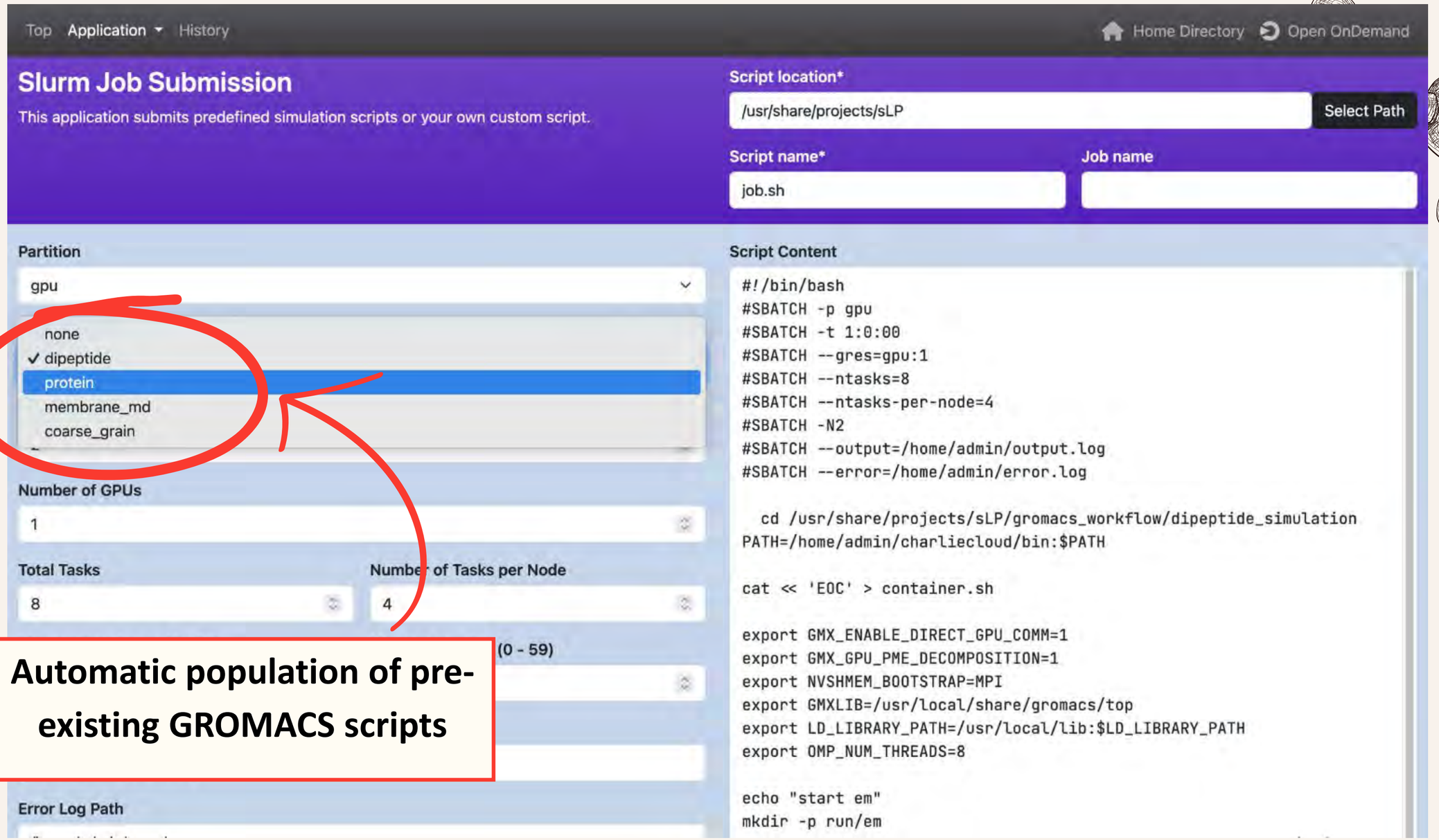
GROMACS in HPC

1. Runs across multiple nodes and GPUs
2. Supports GPU decomposition
3. Compatible GPUs can use CUDA aware MPI

Toggle fields for Slurm arguments

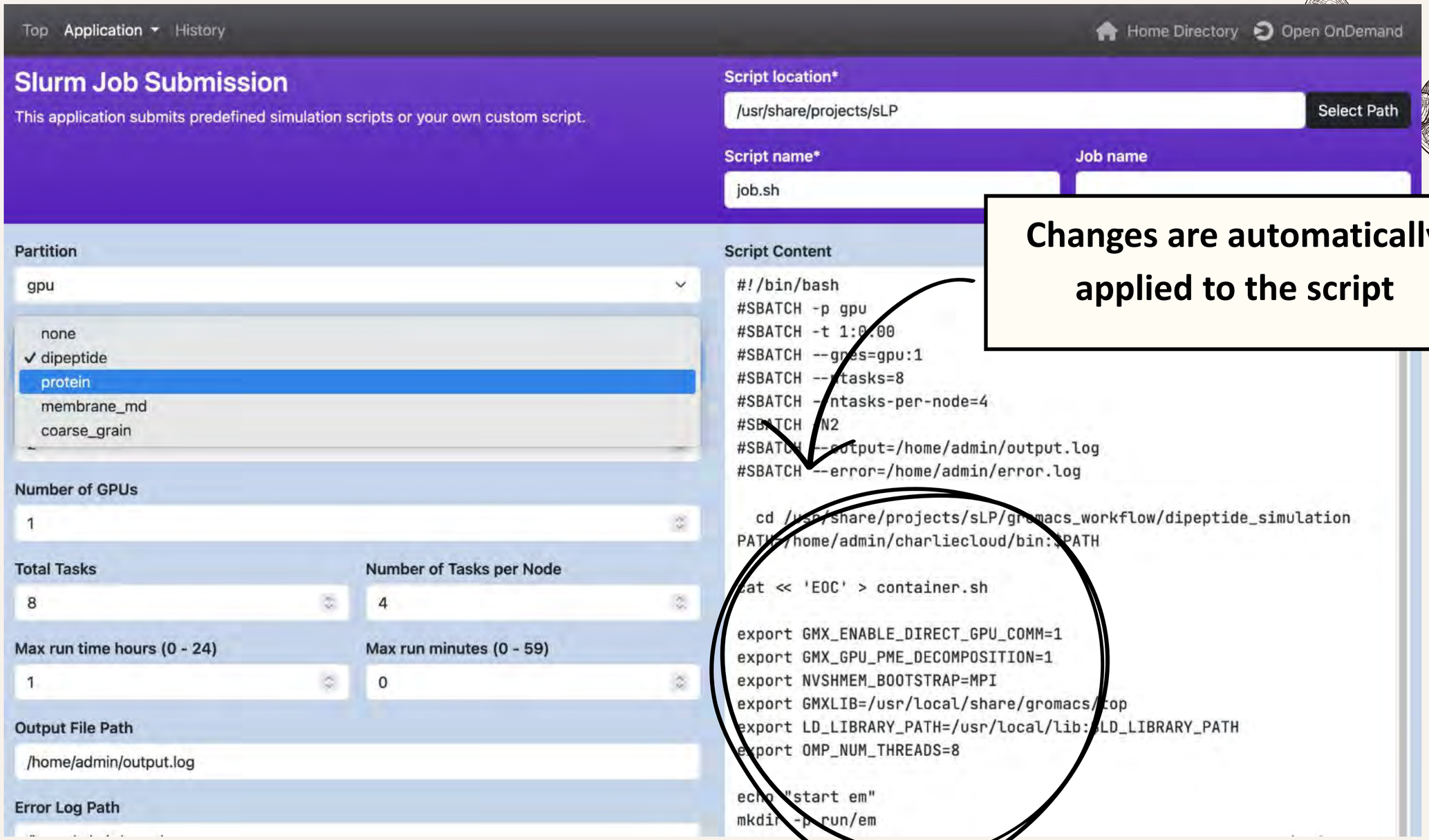


https://github.com/RIKEN-RCCS/OpenComposer



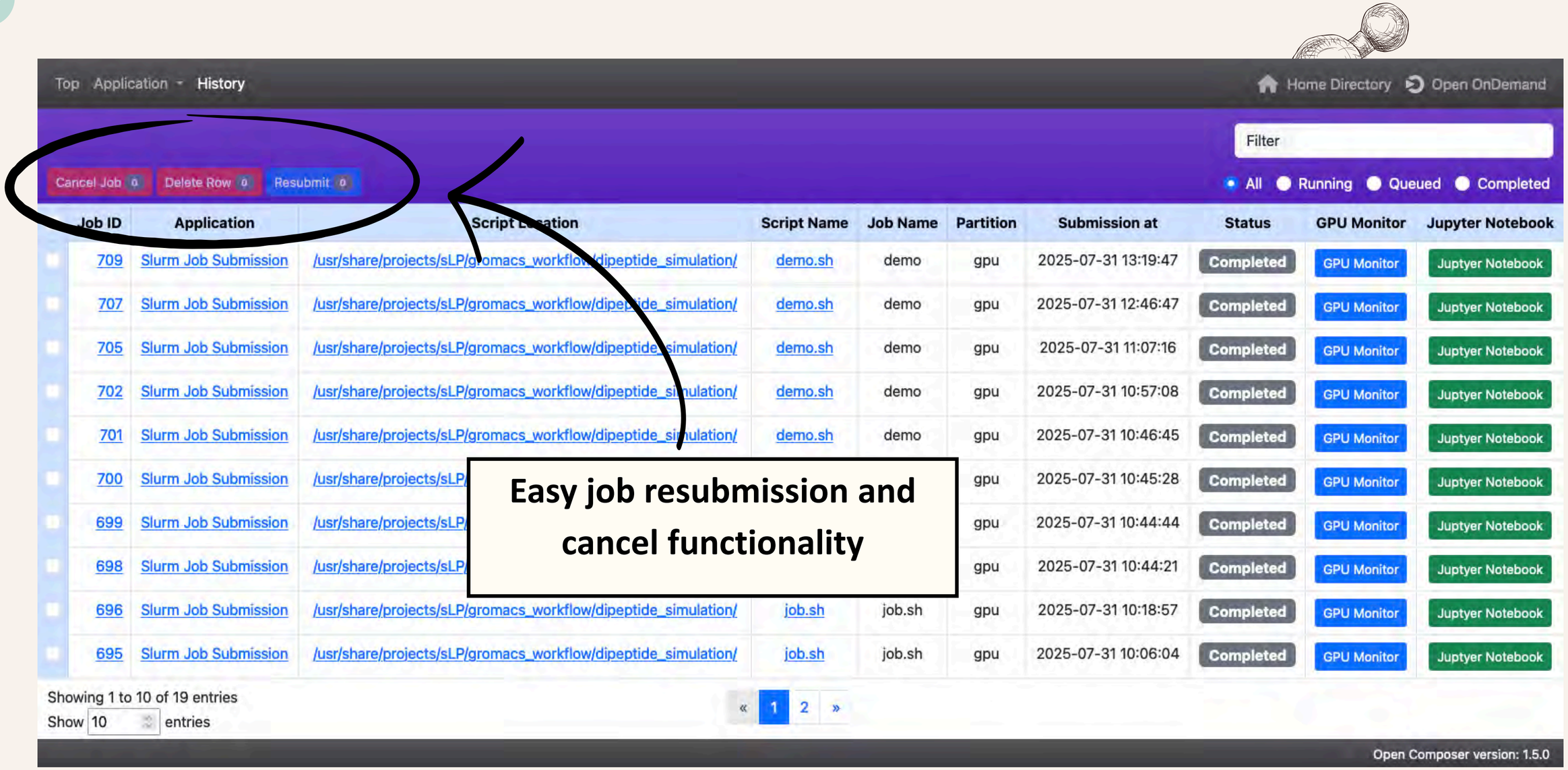
Automatic population of pre-existing GROMACS scripts

https://github.com/RIKEN-RCCS/OpenComposer



Changes are automatically applied to the script

https://github.com/RIKEN-RCCS/OpenComposer



Easy job resubmission and cancel functionality

https://github.com/RIKEN-RCCS/OpenComposer

Real Time **Monitoring**

"Real time monitoring is something we desperately want"

~ Everyone on the support team

Grafana

Shirley can monitor the resources her job is using

Custom GPU Monitoring

- Custom Sinatra app using Ruby on Rails
- Pulls real time logs from the compute nodes with GPU and displays them raw and through a table that updates every five seconds

Node	GPU	GPU Utilization	GPU Memory Used	Total GPU Memory
gu01	NVIDIA A2	58%	2084 MiB	15356 MiB
gu02	NVIDIA A2	56%	2084 MiB	15356 MiB

Custom GPU Monitoring

- Custom Sinatra app using Ruby on Rails
- Pulls real time logs from the compute nodes with GPU and displays them raw and through a table that updates every five seconds

Displays real time GPU usage

Node	GPU	GPU Utilization	GPU Memory Used	Total GPU Memory
gu01	NVIDIA A2	58%	2084 MiB	15356 MiB
gu02	NVIDIA A2	56%	2084 MiB	15356 MiB

Custom GPU Monitoring

- Custom Sinatra app using Ruby on Rails
- Pulls real time logs from the compute nodes with GPU and displays them raw and through a table that updates every five seconds

The total memory on each GPU

Node	GPU	GPU Utilization	GPU Memory Used	Total GPU Memory
gu01	NVIDIA A2	58%	2084 MiB	15356 MiB
gu02	NVIDIA A2	56%	2084 MiB	15356 MiB

Custom GPU Monitoring

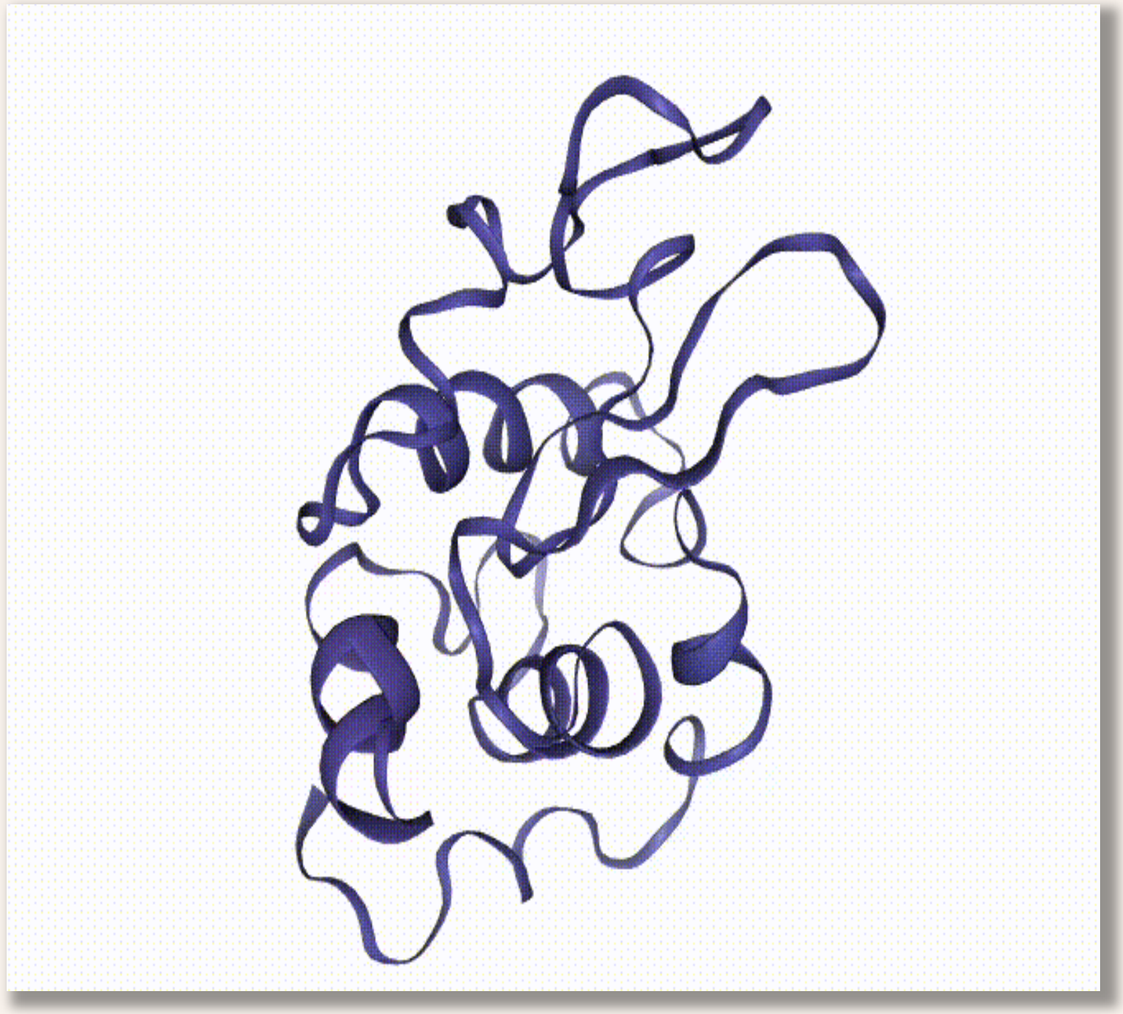
- Custom Sinatra app using Ruby on Rails
- Pulls real time logs from the compute nodes with GPU and displays them raw and through a table that updates every five seconds

The total memory used on each GPU

Node	GPU	GPU Utilization	GPU Memory Used	Total GPU Memory
gu01	NVIDIA A2	58%	2084 MiB	15356 MiB
gu02	NVIDIA A2	56%	2084 MiB	15356 MiB

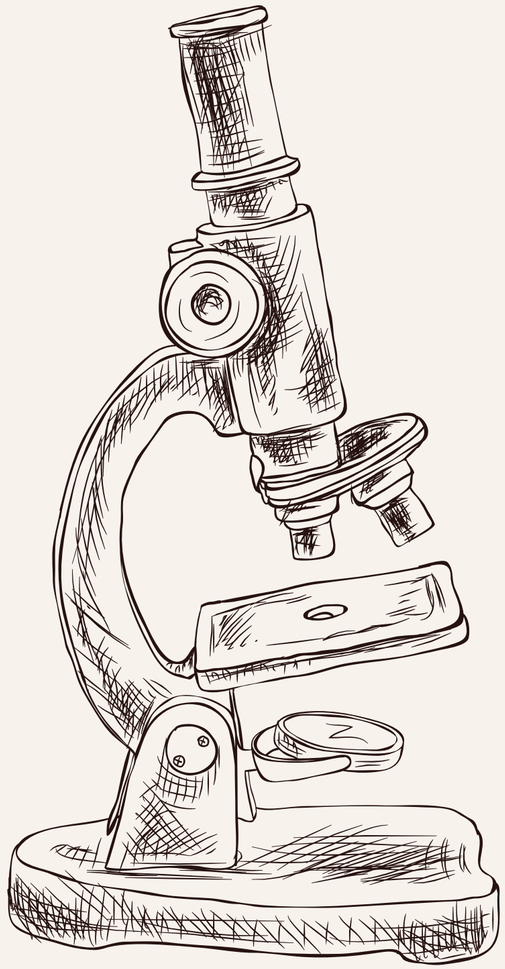
Post Processing

- Launch a **JupyterLab** through Open OnDemand
- **Visualize** results of your simulation.
- Our example uses **mdtraj** and **NGLView** for post processing



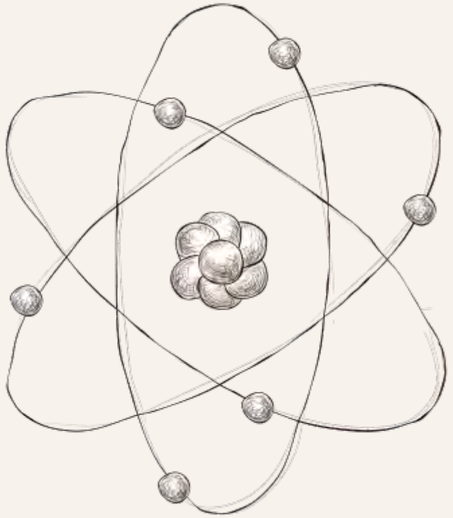
Charliecloud

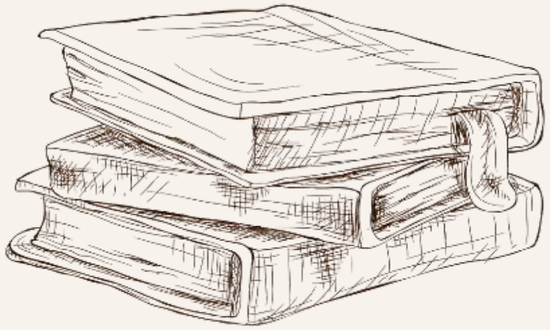
All Additional Applications are built in a



FAST. FLEXIBLE. FREE.
GROMACS

jupyter





Charliecloud Container

Charliecloud

DEMO Time!

Conclusions

01

OOD allows for users to share pre-made workflows with other people on their team

02

Users are given easy access to running and evaluating scientific simulations

03

Eliminates the need for users to SSH into compute nodes, navigate file structures manually, and write complex job scripts - replacing 30+ commands with a convenient interface.

Like and Subscribe for More Diva Features!

Questions?

Abhinav Kotta
Cornell University
abhinavkotta.io@gmail.com
(689)-209-8517

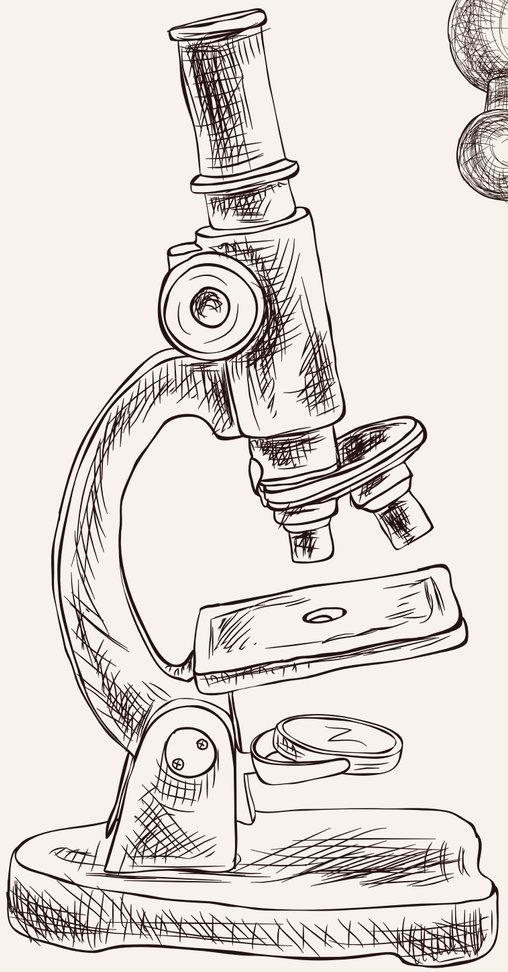
Beamlak Bekele
University of Maryland Baltimore County
bekbeamlak@gmail.com
(240)-423-9786

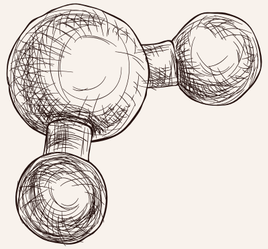
Kelsey Tirado
Rochester Institute of Technology
kelseytirado3@gmail.com
(631)-807-9916

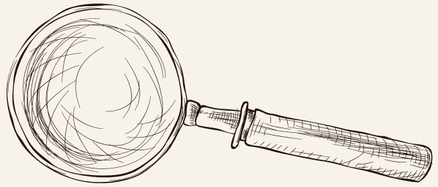
Mentors:

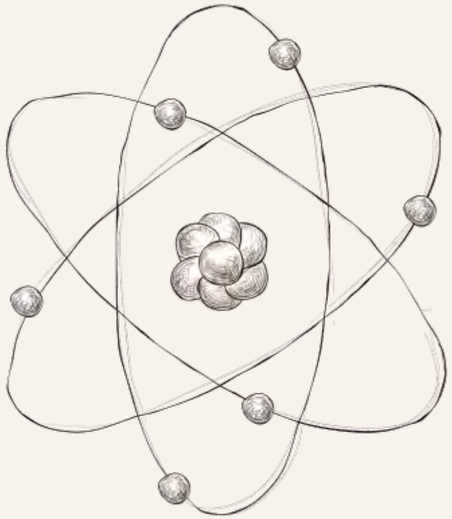
Jordan Ogas, Hunter Easterday, Angelica Loshak, and Andres Quan

https://tinyurl.com/Gpunicornsplaylist





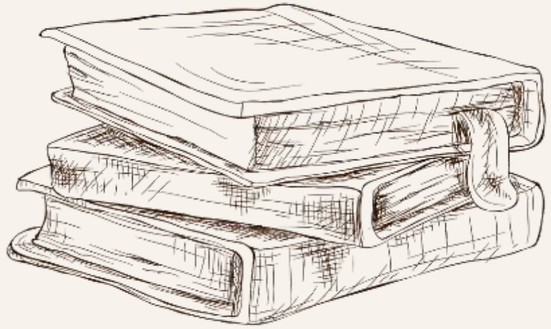


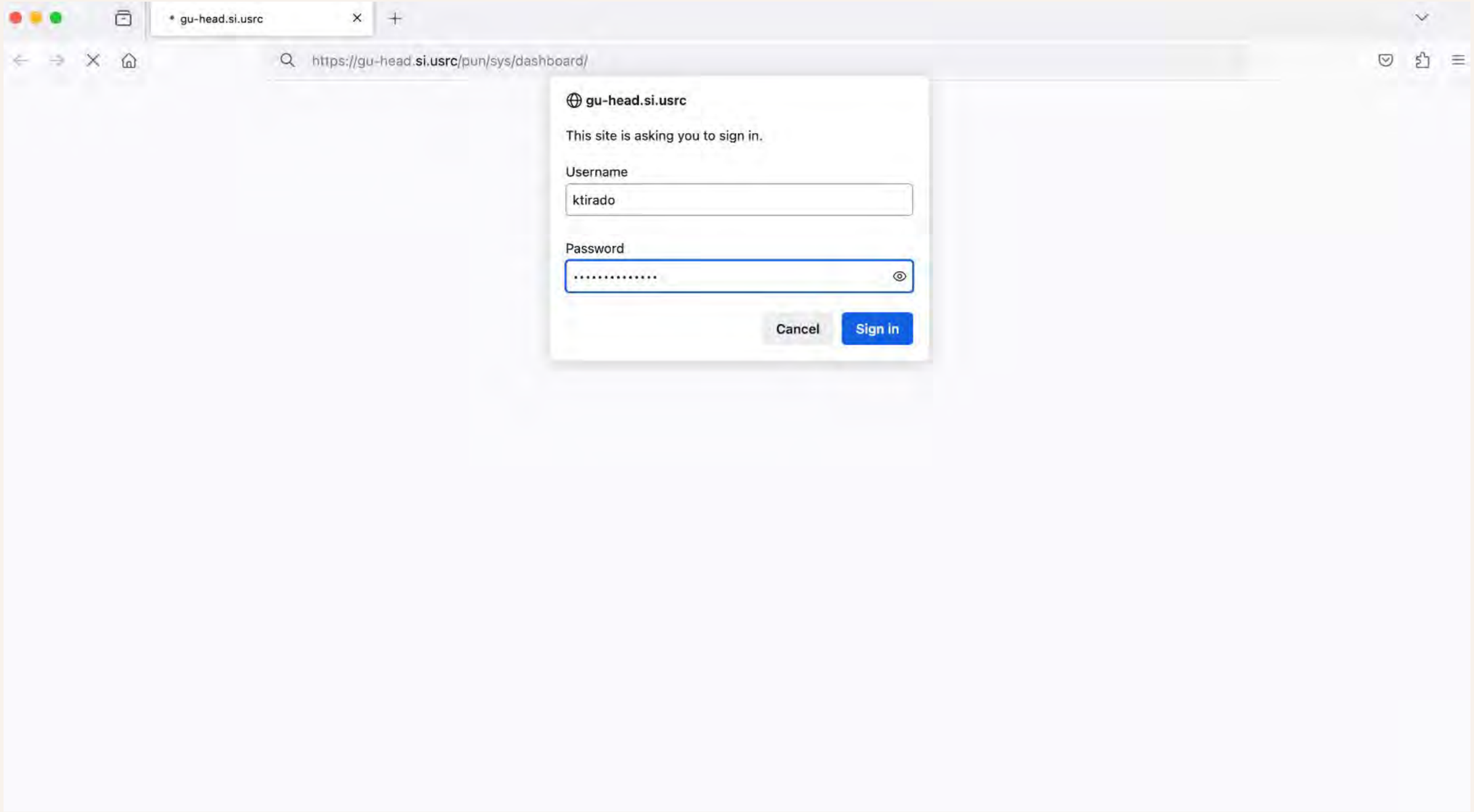


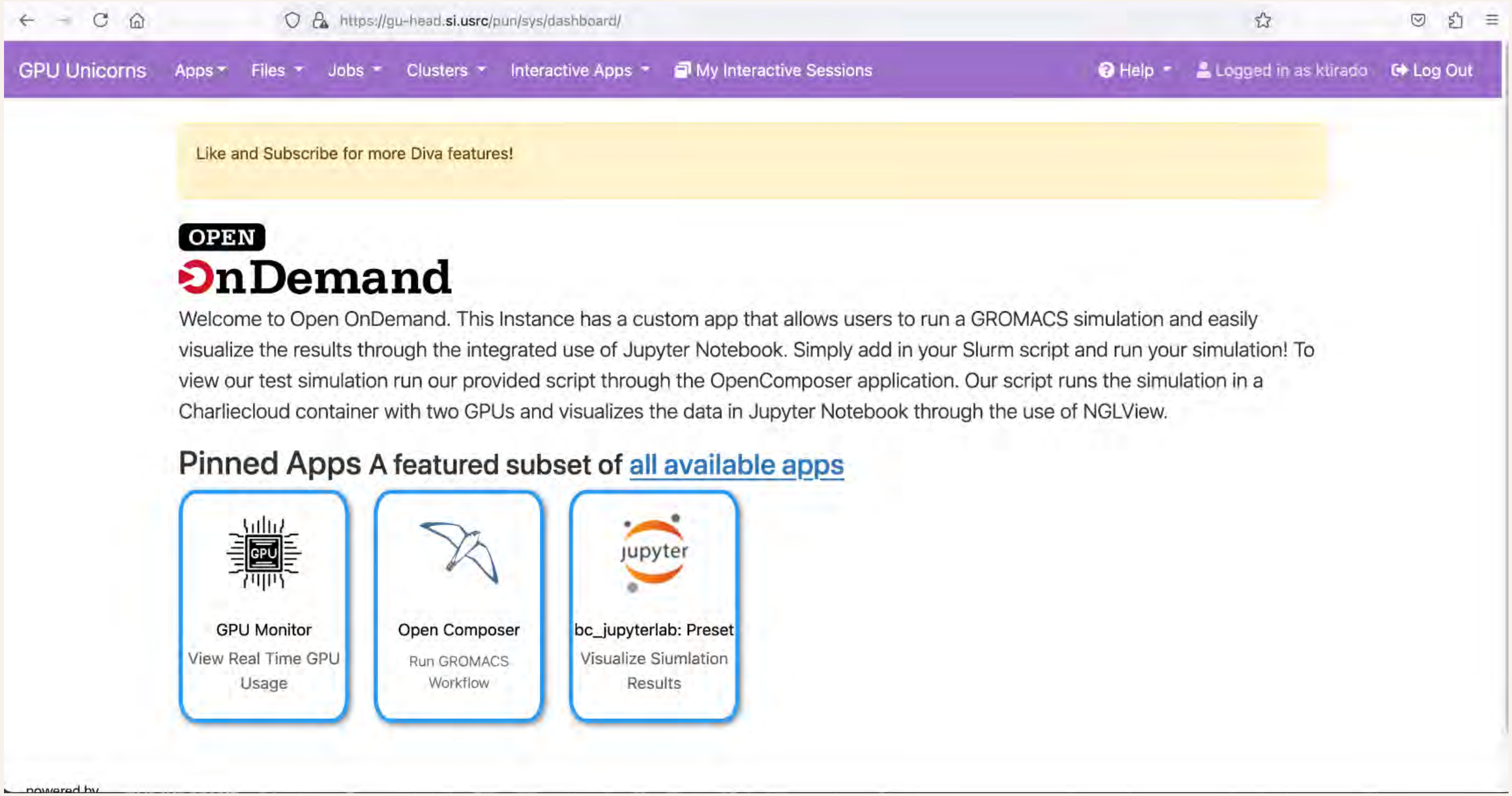
Thank you!

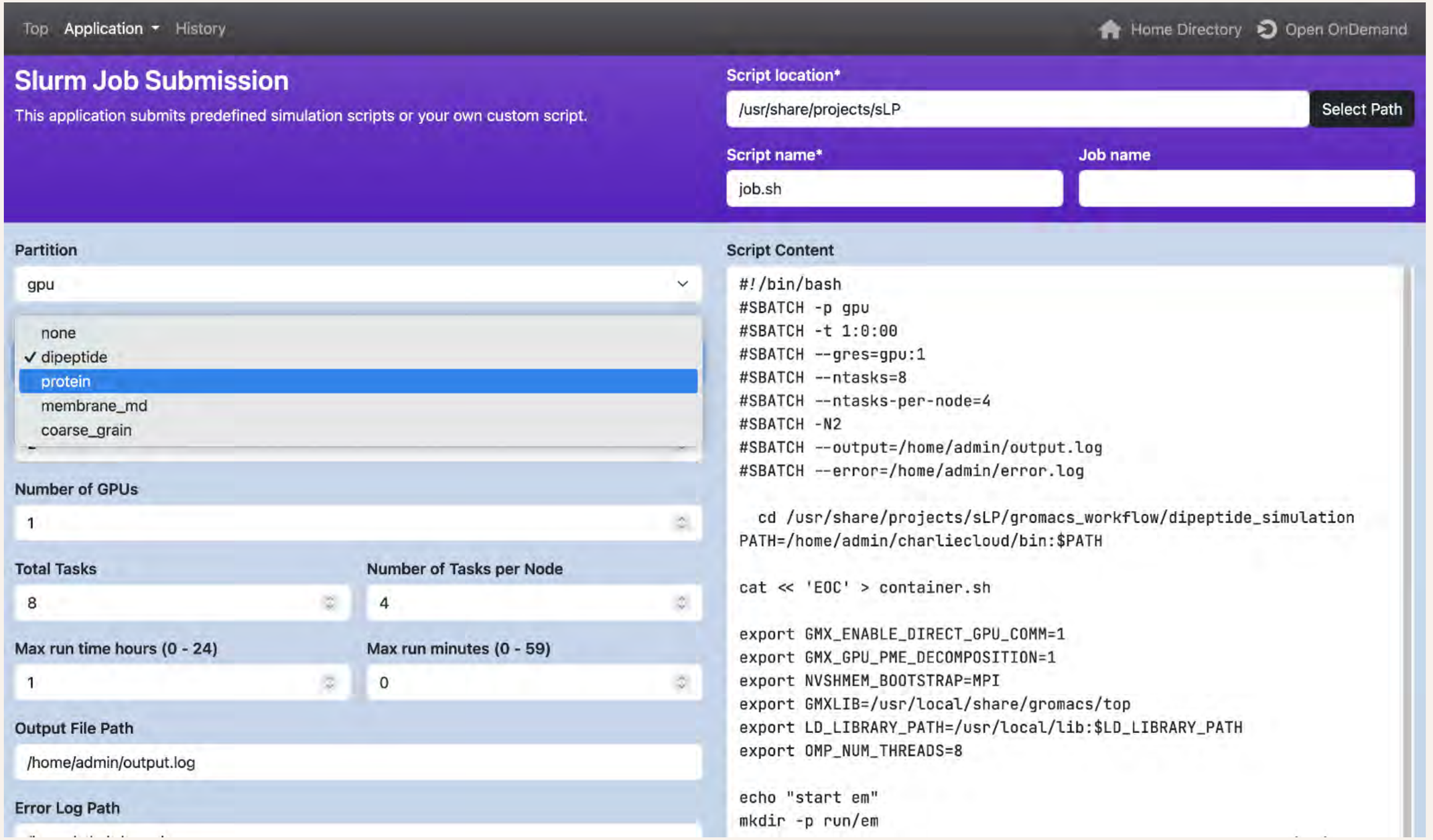
Mentors:

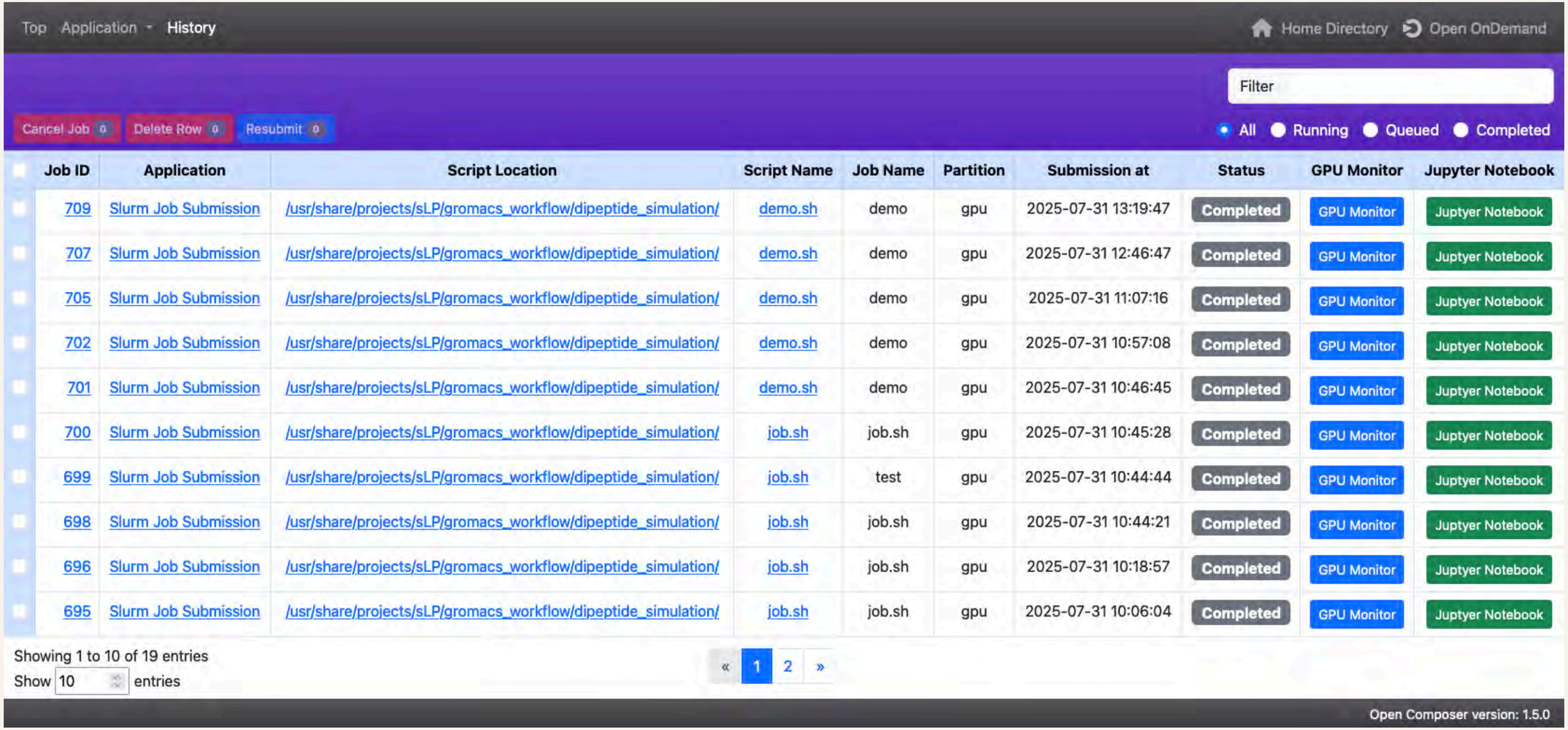
Jordan Ogas, Hunter Easterday, Angelica Loshak, and Andres Quan



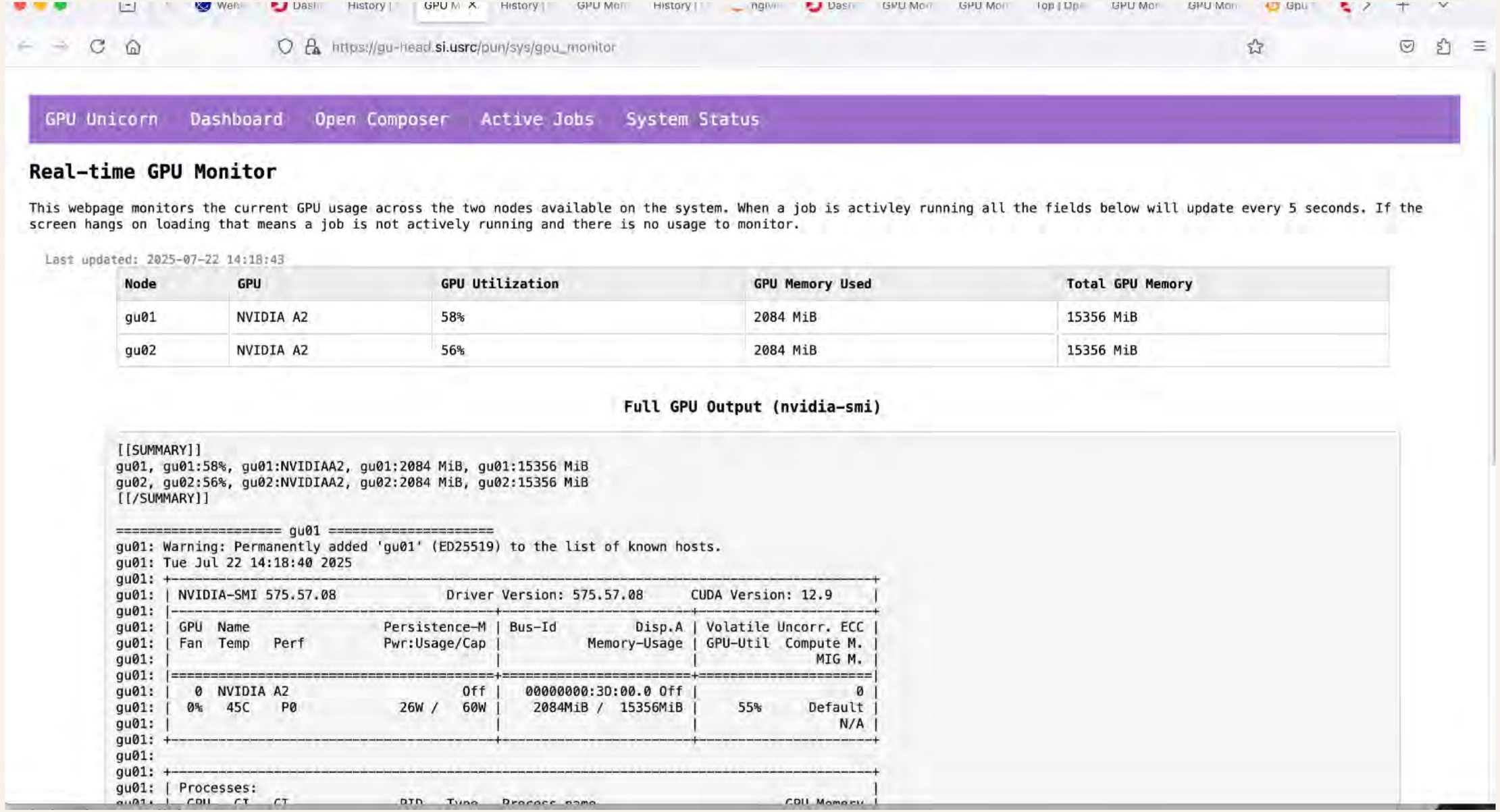

gu-head.si.usrc
https://gu-head.si.usrc/pun/sys/dashboard/
gu-head.si.usrc
This site is asking you to sign in.
Username
ktirado
Password
Cancel
Sign in


https://gu-head.si.usrc/pun/sys/dashboard/
GPU Unicorns
Apps
Files
Jobs
Clusters
Interactive Apps
My Interactive Sessions
Help
Logged in as ktirado
Log Out
Like and Subscribe for more Diva features!
OPEN
OnDemand
Welcome to Open OnDemand. This Instance has a custom app that allows users to run a GROMACS simulation and easily visualize the results through the integrated use of Jupyter Notebook. Simply add in your Slurm script and run your simulation! To view our test simulation run our provided script through the OpenComposer application. Our script runs the simulation in a Charliecloud container with two GPUs and visualizes the data in Jupyter Notebook through the use of NGLView.
Pinned Apps A featured subset of all available apps
GPU
GPU Monitor
View Real Time GPU Usage
Open Composer
Run GROMACS Workflow
jupyter
bc_jupyterlab: Preset
Visualize Siumlation Results
powered by


Top
Application
History
Home Directory
Open OnDemand
Slurm Job Submission
This application submits predefined simulation scripts or your own custom script.
Script location*
/usr/share/projects/sLP
Select Path
Script name*
job.sh
Job name
Partition
gpu
none
dipeptide
protein
membrane_md
coarse_grain
Number of GPUs
1
Total Tasks
8
Number of Tasks per Node
4
Max run time hours (0 - 24)
1
Max run minutes (0 - 59)
0
Output File Path
/home/admin/output.log
Error Log Path
Script Content
#!/bin/bash
#SBATCH -p gpu
#SBATCH -t 1:0:00
#SBATCH --gres=gpu:1
#SBATCH --ntasks=8
#SBATCH --ntasks-per-node=4
#SBATCH -N2
#SBATCH --output=/home/admin/output.log
#SBATCH --error=/home/admin/error.log
cd /usr/share/projects/sLP/gromacs_workflow/dipeptide_simulation
PATH=/home/admin/charliecloud/bin:$PATH
cat << 'EOC' > container.sh
export GMX_ENABLE_DIRECT_GPU_COMM=1
export GMX_GPU_PME_DECOMPOSITION=1
export NVSHMEM_BOOTSTRAP=MPI
export GMXLIB=/usr/local/share/gromacs/top
export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH
export OMP_NUM_THREADS=8
echo "start em"
mkdir -p run/em


Top Application History
Home Directory
Open OnDemand
Filter
Cancel Job 0
Delete Row 0
Resubmit 0
All Running Queued Completed
Job ID | Application | Script Location | Script Name | Job Name | Partition | Submission at | Status | GPU Monitor | Jupyter Notebook
709 | Slurm Job Submission | /usr/share/projects/sLP/gromacs_workflow/dipeptide_simulation/ | demo.sh | demo | gpu | 2025-07-31 13:19:47 | Completed | GPU Monitor | Juptyer Notebook
707 | Slurm Job Submission | /usr/share/projects/sLP/gromacs_workflow/dipeptide_simulation/ | demo.sh | demo | gpu | 2025-07-31 12:46:47 | Completed | GPU Monitor | Juptyer Notebook
705 | Slurm Job Submission | /usr/share/projects/sLP/gromacs_workflow/dipeptide_simulation/ | demo.sh | demo | gpu | 2025-07-31 11:07:16 | Completed | GPU Monitor | Juptyer Notebook
702 | Slurm Job Submission | /usr/share/projects/sLP/gromacs_workflow/dipeptide_simulation/ | demo.sh | demo | gpu | 2025-07-31 10:57:08 | Completed | GPU Monitor | Juptyer Notebook
701 | Slurm Job Submission | /usr/share/projects/sLP/gromacs_workflow/dipeptide_simulation/ | demo.sh | demo | gpu | 2025-07-31 10:46:45 | Completed | GPU Monitor | Juptyer Notebook
700 | Slurm Job Submission | /usr/share/projects/sLP/gromacs_workflow/dipeptide_simulation/ | job.sh | job.sh | gpu | 2025-07-31 10:45:28 | Completed | GPU Monitor | Juptyer Notebook
699 | Slurm Job Submission | /usr/share/projects/sLP/gromacs_workflow/dipeptide_simulation/ | job.sh | test | gpu | 2025-07-31 10:44:44 | Completed | GPU Monitor | Juptyer Notebook
698 | Slurm Job Submission | /usr/share/projects/sLP/gromacs_workflow/dipeptide_simulation/ | job.sh | job.sh | gpu | 2025-07-31 10:44:21 | Completed | GPU Monitor | Juptyer Notebook
696 | Slurm Job Submission | /usr/share/projects/sLP/gromacs_workflow/dipeptide_simulation/ | job.sh | job.sh | gpu | 2025-07-31 10:18:57 | Completed | GPU Monitor | Juptyer Notebook
695 | Slurm Job Submission | /usr/share/projects/sLP/gromacs_workflow/dipeptide_simulation/ | job.sh | job.sh | gpu | 2025-07-31 10:06:04 | Completed | GPU Monitor | Juptyer Notebook
Showing 1 to 10 of 19 entries
Show 10 entries
« 1 2 »
Open Composer version: 1.5.0


Temprature
43 °C
Gpu Memory utlization (nvidia)
2.50 GB
2 GB
1.50 GB
1 GB
500 MB
0 B
11:02:00
11:03:00
11:04:00
11:05:00
11:06:00
11:07:00
11:08:00
GPU memory used (Bytes)
Gpu Power Draw
25.9 W
Job load (Y- GB, x-Time)
24 GB
22 GB
20 GB
11:02:00
11:03:00
11:04:00
11:05:00
11:06:00
11:07:00
11:08:00
Used memory (GB)
Number of allocated vs free gpus
2
Allocated gpu
0
Free gpus
Total node loads reported by slurm
24
22
20
11:02:00
11:03:00
11:04:00
11:05:00
11:06:00
11:07:00
11:08:00
{__name__="slurm_node_load", instance="localhost:8090", job="slurm-exporter"}


https://gu-head.si.usrc/pun/sys/gpu_monitor
GPU Unicorn
Dashboard
Open Composer
Active Jobs
System Status
Real-time GPU Monitor
This webpage monitors the current GPU usage across the two nodes available on the system. When a job is activley running all the fields below will update every 5 seconds. If the screen hangs on loading that means a job is not actively running and there is no usage to monitor.
Last updated: 2025-07-22 14:18:43
Node
GPU
GPU Utilization
GPU Memory Used
Total GPU Memory
gu01
NVIDIA A2
58%
2084 MiB
15356 MiB
gu02
NVIDIA A2
56%
2084 MiB
15356 MiB
Full GPU Output (nvidia-smi)
[[SUMMARY]]
gu01, gu01:58%, gu01:NVIDIAA2, gu01:2084 MiB, gu01:15356 MiB
gu02, gu02:56%, gu02:NVIDIAA2, gu02:2084 MiB, gu02:15356 MiB
[[/SUMMARY]]
========== gu01 ==========
gu01: Warning: Permanently added 'gu01' (ED25519) to the list of known hosts.
gu01: Tue Jul 22 14:18:40 2025
gu01: +---+
gu01: | NVIDIA-SMI 575.57.08   Driver Version: 575.57.08   CUDA Version: 12.9 |
gu01: |---+------------------------+----------------------+
gu01: | GPU Name   Persistence-M | Bus-Id   Disp.A | Volatile Uncorr. ECC |
gu01: | Fan Temp   Perf   Pwr:Usage/Cap | Memory-Usage | GPU-Util Compute M. |
gu01: | | | MIG M. |
gu01: |===+========================+======================|
gu01: | 0 NVIDIA A2   Off | 00000000:3D:00.0 Off | 0 |
gu01: | 0%   45C   P0   26W /   60W | 2084MiB / 15356MiB | 55%   Default |
gu01: | | | N/A |
gu01: +---+------------------------+----------------------+
gu01:
gu01: +---+
gu01: | Processes: |


Session was successfully created.
Home / My Interactive Sessions
Interactive Apps
Desktops
Desktop
Servers
bc_jupyterlab: Preset
bc_jupyterlab: Preset (713)
1 node | 1 core | Running
Host: >_ gu01
Delete
Created at: 2025-08-04 10:27:10 MDT
Time Remaining: 57 minutes
Session ID: 663aedff-7bce-4870-9ae8-0e2020e2769a
Connect to Jupyter

https://gu-head.si.usrc/node/gu01/8252/lab/tree/gromacs_sim/lysozyme_simulation/nglview.ipynb

File Edit View Run Kernel Tabs Settings Help

/ gromacs_sim / lysozyme_simulation /

Name	Modified
run	5d ago
1AKI_clean.pdb	5d ago
1AKI_newbox.g...	5d ago
1AKI_processe...	5d ago
1AKI_solv_ions....	5d ago
1AKI_solv.gro	5d ago
1AKI.pdb	5d ago
container.sh	5d ago
error.log	5d ago
ions.mdp	5d ago
ions.tpr	5d ago
job.sh	5d ago
md.mdp	5d ago
mdout.mdp	5d ago
minim.mdp	5d ago
nglview.ipynb	2d ago
npt.mdp	5d ago
nvt.mdp	5d ago
output.log	5d ago
posre.itp	5d ago
README	5d ago
rungro.slurm	5d ago
rungro1.slurm	5d ago
save_nglview	5d ago

Untitled.ipynb | nglview.ipynb

Code | Notebook | Python 3 (ipykernel)

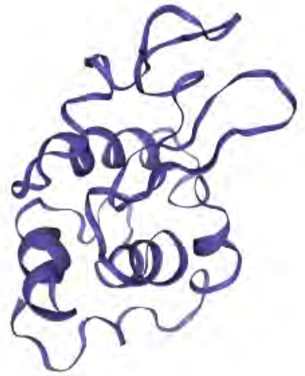
[1]:
```
import nglview as nv
import mdtraj as md
```

[2]:
```
#from simpletraj.trajectory import XtcTrajectory
import nglview as nv
import mdtraj as md  # or use parmed

# Load structure for topology
#top = md.load("1AKI_clean.pdb")  # can be PDB, PRMTOP+INPCRD, etc.

# Load trajectory directly with mdtraj
traj = md.load("run/md/md_0_1_ood.xtc", top="run/npt/npt.gro")  # or .gro

# Visualize with nglview
view = nv.show_mdtraj(traj)
view
```



Simple 1 Python 3 (ipykernel) | Idle

Mode: Command Ln 1, Col 1 nglview.ipynb 1