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

Abhinav Kotta
Los Alamos National Laboratory
Los Alamos, New Mexico, USA
ask329@cornell.edu

Beamlak Bekele
Los Alamos National Laboratory
Los Alamos, New Mexico, USA
bbekele1@umbc.edu

Kelsey Tirado
Los Alamos National Laboratory
Los Alamos, New Mexico, USA
kelseytirado3@gmail.com

Abstract

High performance computing (HPC) accelerates scientific research and discovery but requires proficiency with the Linux command line, the job scheduler, scripting, and software environment management. These requisite skills often prove to be a barrier of entry, preventing a broader range of researchers from leveraging HPC resources. We lower this barrier with established tools behind a single, browser-based interface: Open OnDemand (OOD) for cluster web access and job submission, Charliecloud for lightweight unprivileged containers, and Slurm for resource scheduling. Through OOD we deliver a ready-to-use workflow for GROMACS molecular dynamics simulation campaigns. Users launch pre-built Charliecloud containers that encapsulate the entire scientific software stack, establishing consistency across same-architecture systems and eliminating the need for environment modification. The combined use of these tools, reduces time to science and allows scientists to focus on core parts of their experiments.

Keywords

High Performance Computing (HPC), Slurm, Charliecloud, Grafana, Open Composer, JupyterLab, Open OnDemand (OOD)

ACM Reference Format:
Abhinav Kotta, Beamlak Bekele, and Kelsey Tirado. 2025. Reducing the Barrier of Entry for GPU Accelerated Workflows using Open OnDemand and Charliecloud Containers in HPC. In *The International Conference for High Performance Computing, Networking, Storage, and Analysis.* ACM, New York, NY, USA, 3 pages.

1 Introduction

High Performance Computing (HPC) plays a critical role in advancing scientific research, especially in domains like biology, physics, and chemistry [1]. While widely used, many scientists face a steep barrier when it comes to job submission, scripting, and resource management [3]. This often leads to inefficient resource usage: recent studies show that over 64% of both CPU and GPU jobs use less than half of available memory, and GPU memory is often underutilized [2]. To address this, we introduce an integrated workflow that simplifies job management using Open OnDemand (OOD) and a custom Open Composer interface. Researchers can generate, submit, and relaunch containerized simulations via Charliecloud with minimal manual scripting. Real-time monitoring via Grafana dashboards helps users understand and optimize resource use. This system lowers the barrier to HPC access, improves efficiency, and empowers domain scientists to focus more on discovery and less on infrastructure.

SC '25, St. Louis, MO

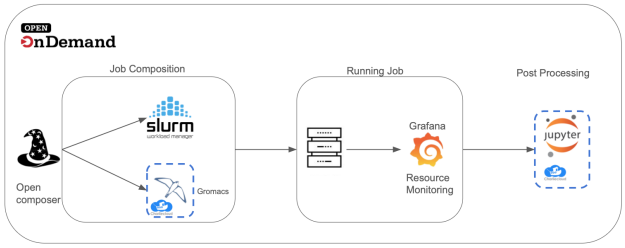


Figure 1: Workflow illustrating the technology stack used to run a scientific workflow with GROMACS

2 System Architecture

2.1 Toolchain Integration

Our system brings together Open OnDemand (OOD), Open Composer, Charliecloud, JupyterLab, and Grafana to deliver a seamless, browser-based HPC experience. OOD acts as the main interface, giving users access to file systems, interactive apps, and job submission from within a web browser.

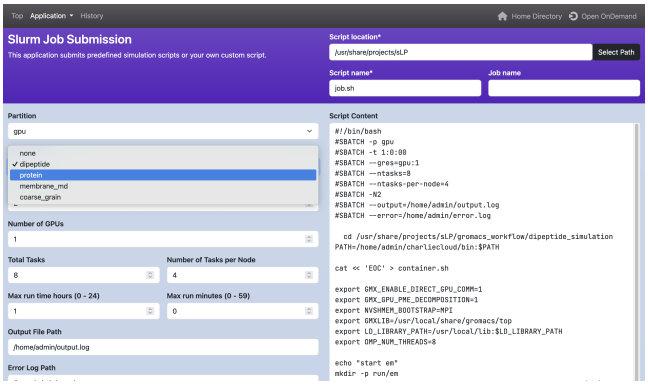
2.2 Containerized Scientific Stack

GROMACS (version 2024.5), a widely used molecular dynamics simulation package, is executed within pre-built Charliecloud containers that include all required runtime components: GROMACS binaries, CUDA libraries, and CUDA-aware MPI. By encapsulating the full software environment, containerization ensures consistency across jobs and eliminates the need for users to install, configure, or manage dependencies. Users can also run these containers with sudo access without requiring the container itself to run with elevated privileges [5].

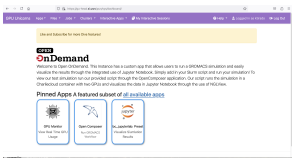
JupyterLab (version 4.4.4) is also containerized and can be built with any Python packages or scientific libraries required for analysis. This allows users to launch browser-based post-processing workflows without worrying about compatibility or setup.

2.3 Custom Workflow Composer App

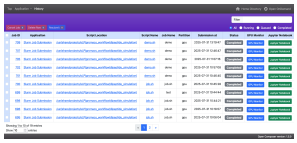
We developed a graphical job submission app using Open Composer (a tool built on top of Open OnDemand) to replace the traditional,



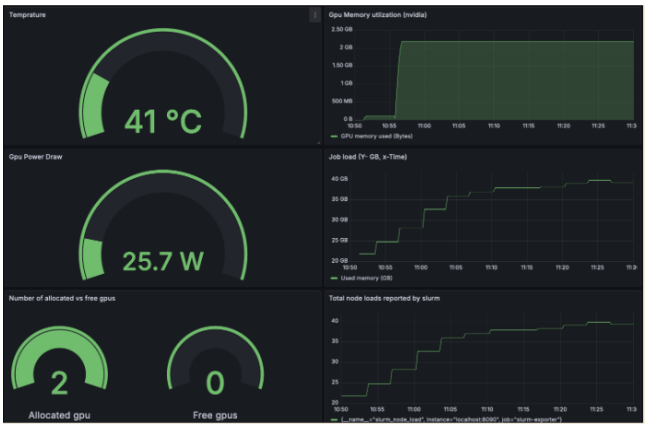
(a) Custom job launcher



(b) Customizable dashboard

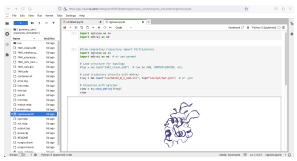


(c) Job history page



(d) Grafana resource monitoring

(e) Real-time GPU monitor



(f) Jupyter visualizations

Figure 2: Workflow interface and monitoring tools overview.

error-prone process of manually editing job scripts [4]. Users can construct and launch complete scientific workflows by selecting input files, setting parameters, and defining job stages through a point-and-click interface.

The app internally generates scripts based on pre-loaded default scripts, handles container execution, and supports pipeline saving and re-launching. This dramatically reduces the learning curve, promoting repeatable, shareable workflows across research groups.

3 User Experience and Scientific Workflow

To demonstrate the user workflow, we walk through a typical GROMACS molecular dynamics (MD) simulation, simulating dipeptide movement through a solvent, using our browser-based HPC platform.

3.1 Submitting a GROMACS Simulation

After logging into the cluster through Open OnDemand (OOD), the user opens the Open Composer application from the dashboard. From a dropdown, they select a pre-configured "dipeptide" workflow. The Slurm script auto-populates the relevant GROMACS script including GPU partition, wall time, and simulation parameters. The user sets the input directory containing any relevant input files and clicks submit. Behind the scenes, the system launches a Charliecloud container with a fully configured GROMACS + CUDA stack. No module loading, environment setup, or script editing is required.

3.2 Monitoring Simulation Progress

Once the job begins, the user can monitor performance from OOD's Active Jobs page. Embedded Grafana panels show real-time CPU, memory, and GPU utilization, while a custom HTML page provides detailed GPU stats via automatic refresh. This makes it easy to detect issues like underutilized GPUs or memory bottlenecks.

3.3 Visualization and Post-Processing

After completion, the user launches Jupyter Notebook directly from the job history page. Inside the notebook, preloaded code opens the trajectory file using mdtraj and NGLView, rendering a 3D animation of the dipeptide movement. Additional cells allow the user to calculate RMSD, radius of gyration, and energy profiles. Figure **??** shows each step in this streamlined workflow, enabling researchers to focus on their science rather than HPC infrastructure.

4 Conclusion

We introduce a streamlined workflow that empowers researchers to run, monitor, and analyze scientific simulations entirely through a browser, lowering technical barriers and accelerating scientific discovery.

References

[1] Simon Gathu. 2024. High-Performance Computing and Big Data: Emerging Trends in Advanced Computing Systems for Data-Intensive Applications. *Journal of Advanced Computing Systems* 4, 8 (Aug. 2024), 22–35. doi:10.69987/jacs.v4i8.42

[2] Jie Li, George Michelogiannakis, Brandon Cook, Dulanya Cooray, and Yong Chen. 2023. Analyzing Resource Utilization in an HPC System: A Case Study of NERSC's Perlmutter. In *High Performance Computing*, Abhinav Bhatele, Jeff Hammond, Marc Baboulin, and Carola Kruse (Eds.). Springer Nature Switzerland, Cham, 297–316.

[3] Julia Mullen, Chansup Byun, Vijay Gadepally, Siddharth Samsi, Albert Reuther, and Jeremy Kepner. 2017. Learning by doing, High Performance Computing education in the MOOC era. *J. Parallel and Distrib. Comput.* 105 (2017), 105–115. doi:10.1016/j.jpdc.2017.01.015 Keeping up with Technology: Teaching Parallel, Distributed and High-Performance Computing.

[4] Masahiro Nakao and Keiji Yamamoto. 2024. *Development of a Web Application 'Open Composer' for Batch Job Submission on HPC Clusters.* IPSJ SIG Technical Report. Information Processing Society of Japan (IPSJ), Kobe, Japan. https://github.com/RIKEN-RCCS/OpenComposer In Japanese.

[5] Reid Priedhorsky and Tim Randles. 2017. Charliecloud: unprivileged containers for user-defined software stacks in HPC. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (Denver, Colorado) *(SC '17)*. Association for Computing Machinery, New York, NY, USA, Article 36, 10 pages. doi:10.1145/3126908.3126925