

Research Computing at Mines Workshop Pre-HPC Preliminaries

January 29, 2024

Presented by:

Nicholas A. Danes, PhD

Computational Scientist

Cyber Infrastructure & Advanced Research Computing (ITS)

About Me

- Graduated from Mines in 2019
 - PhD in Computational & Applied Mathematics
 - Advisor: Dr. Karin Leiderman
 - Dissertation: *Computational modeling of extravascular platelet aggregation under flow*
 - Utilized the HPC system "Mio" for my research using Python (FEniCS)
- Computational Engineer at Ball Aerospace
 - September 2019 – August 2020
- Rejoined Mines in August 2020
 - Computational Scientist in the Cyberinfrastructure & Advanced Research Computing Group (ITS)



Who supports Cyberinfrastructure and Research Computing at Mines?



Matt Brookover
Solutions Architect



Nicholas Danes
Computational Scientist



Richard Gilmore
Visualization Engineer



Mike Robbert
System Administrator
& Network Engineer



Kira Wells
Manager, Research
Infrastructure



Matt Kettering
Sr. Director,
Infrastructure Services

What is Cyberinfrastructure and Research Computing?

- **Cyberinfrastructure (CI)** is the integrated system of computing resources, data storage, networking infrastructure, software and human technical support that enables modern scientific research and data analysis.
 - **Research computing** is one major part of CI.
- **Research computing (RC)** is a catch-all term using application of computational technology, to support research in science and engineering.
 - It involves the use **high performance computing (HPC)**, **data storage**, and other **cyberinfrastructure (CI)** to process, model, analyze, and/or visualize data.
 - It requires use of specialized software/libraries, advanced computational algorithms and methods, and/or **large-scale** hardware.

How does CIARC help researchers?

- Cyberinfrastructure
 - High Performance Computing
 - Cloud Computing
 - Data Management, Storage and Transfer
 - Advanced Research Computing
 - HPC
 - Job Management
 - Software Build
 - Troubleshooting & Support
- Consulting
 - Parallel Scaling and Optimization
 - Software Optimization
 - Advanced Workflows
 - Scientific Visualization

Help Center Tickets:

https://helpcenter.mines.edu/TDClient/1946/Portal/Requests/TicketRequests/NewForm?ID=4GCQlvW5OYk_&RequestorType=Service

What kind of compute do you need?

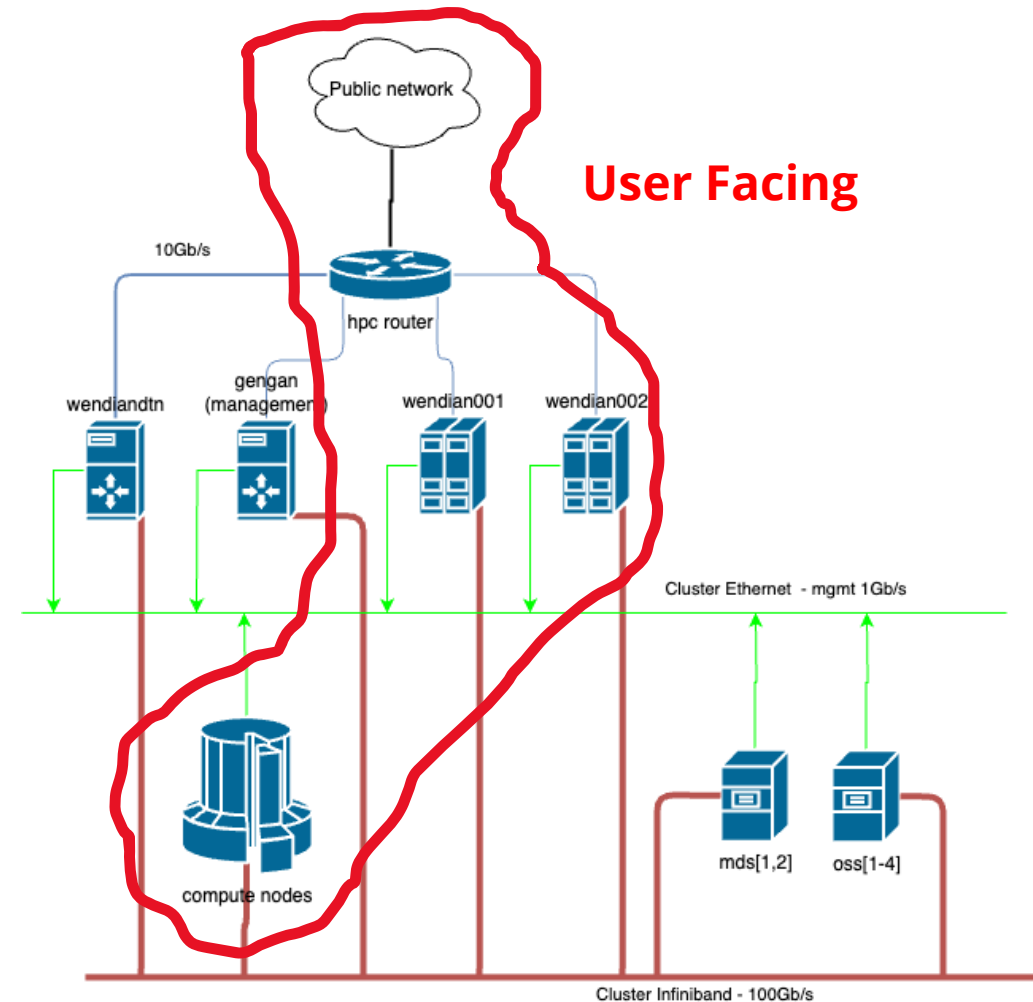
- **Exploratory or “interactive” research**
 - Workstation (local, virtual machine, and/or cloud)
 - Examples: ANSYS, Matlab, etc.
- **Large multi-node/multi-core jobs? HPC!**
 - More on this later!
- **Many single-core jobs? High Throughput Computing (HTC)!**
 - Workstation
 - Open Science Grid (<https://osg-htc.org/>)

What kind of storage do you need?

- Local
- Cloud
 - Microsoft OneDrive
 - Dropbox
 - Google Drive
- Shared/Project
 - Orebits Network-Attached Storage (NAS)
 - <https://ciarc.mines.edu/data-storage/orebits/>

The Anatomy of an HPC, Oversimplified

- Hardware
 - Head/Login Node
 - e.g. wendian001
 - Management node
 - e.g. gengan
 - Compute nodes
 - E.g. c001, g005, etc.
 - Nodes connected via
 - Ethernet: 1 Gb/s
 - TCP/IP
 - Inexpensive
 - High latency, but widely supported
 - InfiniBand: 100 Gb/s
 - Compute communication
 - Specific use case
 - High throughput
 - Low latency



Mines HPC Options

- On-campus
 - HPC
 - Wendian
 - <https://wpfiles.mines.edu/wp-content/uploads/ciarc/docs/pages/systems.html>
 - Launched 2018 on campus
 - Core-hour model
- Off-premise
 - NSF ACCESS
 - <https://access-ci.org/>
 - Different allocation tiers; requires proposal
 - NSF funding not required
 - CU Boulder “RMACC” Alpine
 - <https://www.colorado.edu/rc/alpine>
 - Mines Researchers have access through the RMACC program
 - “Gap” funding option for researchers
 - AWS and other Cloud Computing
 - CIARC is working on solutions to provide a seamless experience for Researchers
 - More details: https://wpfiles.mines.edu/wp-content/uploads/ciarc/docs/pages/computing_options.html

Wendian@Mines

- Available for new users
- Charging based on core-hour model (\$0.02 per core-hour)
- Typical CPU node configuration
 - Intel Xeon Gold (Skylake) Dual Socket
 - 12-18 cores, 24-36 threads per socket
 - 192 GB – 384 GB Memory per node
 - ~3000 CPU core total on Wendian
- GPU node available
 - NVIDIA Volta V100 x 4 cards, 24 Skylake cores
- More details: [Systems — HPC@Mines documentation](#)

What skills do I need to be a successful researcher on HPC?

- Linux
 - Log into a remote server
 - Navigate the filesystem in the command line
 - Learn the basics of how software is detected in an environment
 - PATH, LD_LIBRARY_PATH,CPATH, etc.
 - Set up automation via scripting (Bash)
- HPC Job Scheduler
 - Submit jobs to HPC compute node scheduler
 - Know how to request specific resources
 - Check status of jobs, computational efficiency, etc.

What skills do I need to be a successful researcher on HPC?

- Parallel Computing
 - Learning how multi-processing affects simulation/computation time
 - Differences between shared and distributed memory computing
 - GPU computing (if applicable)
- Data Management
 - How to archive and transfer data between systems
- Computational Lab Practices
 - Computational Notebook
 - Version Control (using git)

Lab #1

Intro to Linux & Command Line



What is a batch job? Job scheduler? Queue?

- How are HPC resources utilized?
 - A **batch job** is submitted to a **job scheduler** which sits in a **queue** until **resources** are available.
 - When resources are available, the **batch job** will be given to a compute node to process the job information
- **Batch jobs** are submitted using a script which contains the following information:
 - How much and how long resources are required for the program
 - Dependencies of the program from the OS environment
 - What program is run
 - Including options, input files, etc.
- **Batch jobs** let us create compute workloads that can be automatically submitted to be run on an HPC cluster.

What is a batch job? Job scheduler? Queue?

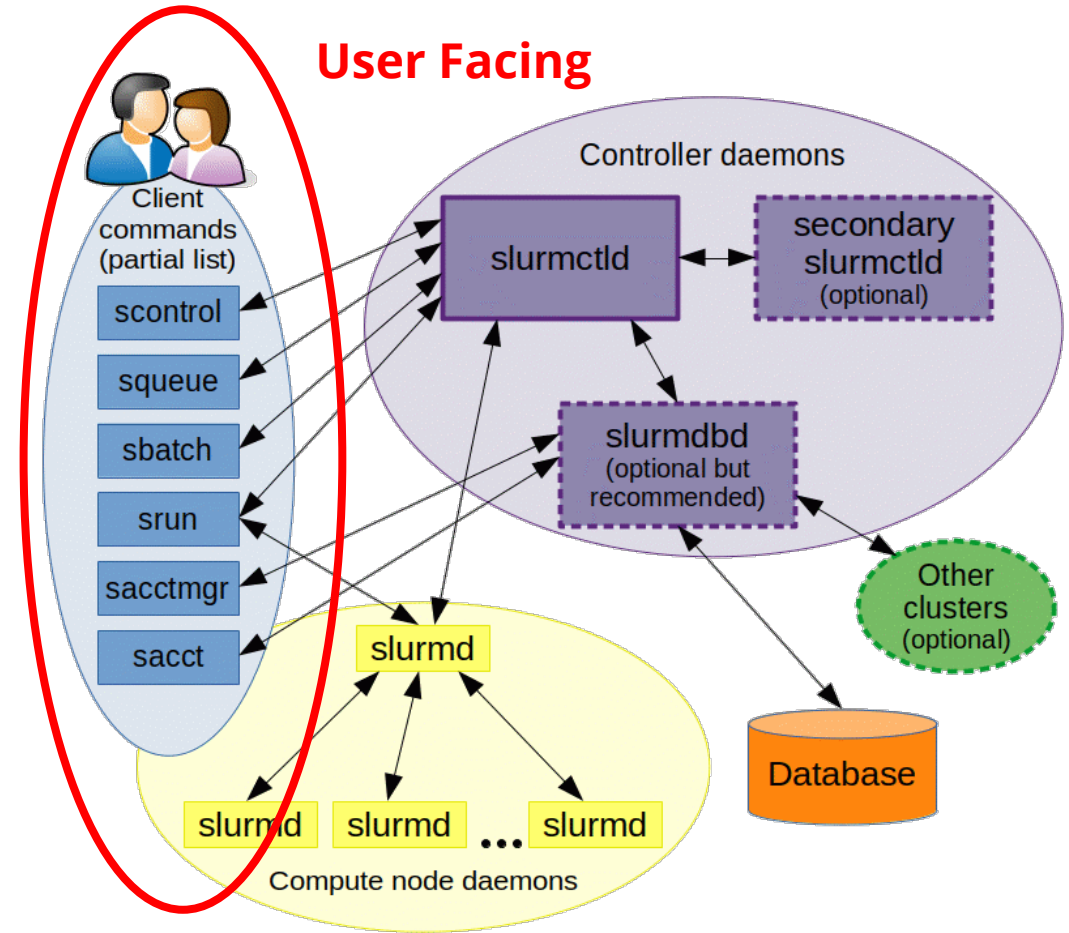
- A **job scheduler** is a software program that automates the scheduling and execution of **batch jobs**.
- The main user-facing component is the job **queue**, which is a list of all the jobs that the scheduler needs to run on the HPC cluster
- **Resources** are all the compute nodes available on the HPC cluster, which the job scheduler manages
- During an executed batch job, the job scheduler will allocate the requested resources and run the program within the script provided

Why do HPCs use a job scheduler?

- Goal of a Job Scheduler: *Maximize* utilization but *minimize* wait times!
- HPC is a shared compute resource
 - Researchers need them for:
 - Different periods of time (e.g. research paper, thesis deadline , etc.),
 - Different resources (single core, single node, GPUs, etc)
 - Different lengths of time (36 hours, 1 hour, etc) AND/OR
 - Different number of jobs
 - For example:
 - 5-node (36 cores per node) molecular dynamics simulation that takes 5 days
 - Researcher runs 5 of these jobs a week
 - 1-core parameter sweep of model, requiring thousands of 1 single-core jobs, each taking 1 hour
 - Researcher may run 100's or these jobs per week, for months

Slurm HPC Job Scheduler

- Slurm is an open source, cluster management and job scheduler for Linux clusters
- Several different daemons run on the nodes to manage Slurm's components



Slurm: Which jobs run first?

- Slurm uses a **priority** model to decide when jobs run
 - *Mostly* first come, first serve with caveats
- Priority depends on but is not limited to:
 - Job size
 - CPU tasks/cpus per task
 - Memory
 - Time requested
 - Job age
 - Priority of job grows while submitted, but not running
 - Fair share
 - Users who've consumed less resources recently get higher priority
 - Prevents overuse by a few heavy users.

Slurm User Features

Submit jobs with custom options, including but not limited to:

- Cores/Nodes
 - Number of Nodes
 - Tasks – Spawned processes
 - CPUs per task – number of cpu cores a given spawned process can utilize
- Memory
 - Memory per cpu
 - Memory per node

Slurm User Features

Submit jobs with custom options, including but not limited to:

- Time
 - On Wendian, defaults to 6 days
- Partitions
 - Access GPU nodes
 - Email Notification – job start, end, etc.
- Track jobs in the queue using **squeue**
- **You will learn more about this in the lab!**

Choosing software to run on HPC

- Your software application will be dependent on:
 - Research project
 - Your own personal preferences
 - **Advisor's personal preferences**
- For ease of use and to get things running, we will be using Python for most labs/tutorials in this workshop!

Python

- Widely Available
- Portable – Supported across MacOS, Windows, Linux
- Easy to read and learn
- Large community with scientific computing libraries & support
- Extensible: Supports bindings with
 - C/C++
 - Fortran
 - And more!

Using Python with a GUI/IDE

Popular Options:

- Spyder
- Atom (GitHub)
- Sublime Text 3
- Jupyter Notebooks – HPC compatible (*we will use these today*)

And many more!

Getting started Python on your local system

- Linux
 - Most up-to-date Linux distros ship Python 3 by default
 - Manage library installs using the python package mangager `pip`:
 - e.g. `$ pip install --user numpy`
- MacOS
 - Python 2.7 ships by default in MacOS Catalina ^&
 - Python 3.x available through Xcode
 - Homebrew or MacPorts can also provide Python 3 (Xcode required)
- Windows
 - Windows Subsystem for Linux can provide a Linux shell on your windows machine to use Python
 - Python can be installed by going to [Python.org](https://python.org)

Getting started Python on your local system

- Cross-platform option: Use Anaconda
 - <https://anaconda.org>
 - Binary distribution of package management
 - Available on Windows, Mac and Linux (+ our HPC systems)
 - Easy management of various environments
 - Supports `pip` and its own package manager `conda`
 - Community maintained packages available through conda-forge:
 - <https://anaconda.org/conda-forge>

We will be using this today!

Lab #2

Intro to Slurm and Python



Final Takeaways

- CIARC group provides support for Mines researchers' cyberinfrastructure and research computing needs
- Cyberinfrastructure primarily consists providing support of hardware related to researcher needs
- Research Computing is the application of computing technologies for research needs
- A successful researcher needs basic skills in Cyberinfrastructure and Research Computing, including:
 - Linux/Bash basics
 - HPC Job Schedulers
 - Knowledge of parallel computing
 - Data Management
- PATH, LD_LIBRARY_PATH, AND CPATH are important environment variables that setup your software to be used on a Linux system -> HPC

Final Takeaways

- Python provides a good baseline programming language to get new researchers up and running
- HPC Clusters use job schedulers to manage workloads for all researchers to maximize research usage, but minimize wait times
- PATH, LD_LIBRARY_PATH, AND CPATH are important environment variables that setup your software to be used on a Linux system, including HPC
 - Modules
 - Conda environments

Further Resources

- Mines CIARC HPC Website:
 - <https://ciarc.mines.edu/hpc>
 - Documentation: [Welcome to HPC at Mines documentation! — HPC@Mines documentation](#)
- For HPC-related questions:
 - Submit a ticket to the help desk!
 - https://helpcenter.mines.edu/TDClient/1946/Portal/Requests/TicketRequests/NewForm?ID=4GCQlvW5OYk_&RequestorType=Service
 - Schedule a meeting with one of us
 - <https://outlook.office365.com/owa/calendar/CIARCTeamServices@mines0.onmicrosoft.com/bookings/>