Python for HPC – Day 1

April 12, 2022

Presented by:
Nicholas A. Danes, PhD
Computational Scientist
Cyber Infrastructure & Advanced Research Computing (ITS)
Goals of Workshop Series

• Day 1: Intro to HPC@Mines
  • Introduce HPC@Mines as a computing resource for researchers at Mines
  • Introduce researchers to Python, an interpreted programming language, as an option for scientific computing
  • Show Python can be used in an HPC environment

• Day 2: Scientific Computing Fundamentals
  • Intro to core Scientific Python modules: NumPy, Scipy, Matplotlib
  • Optimizing a basic Python code

• Day 3: Parallel Computing & Other Advanced Topics
  • MPI, multiprocessing and other advanced parallel computing options
Questions for Audience (Zoom Chat)

• Have you used Python before?

• Have you used HPC before?

• Are you a current Mines @ HPC user?

• What software/libraries are you interested in from Python?
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)
What is HPC?

• Stands for **High Performance Computing**
  • “High Performance Computing most generally refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business.” – insideHPC

• HPC achieves this by:
  • Interconnecting many computers (”nodes”) through a high throughput networking interface which allows all computers to talk to one another
  • Using scientific computing software that can leverage the HPC environment

[What is high performance computing? - insideHPC](http://example.com)
HPC Systems @ Mines

- Two main HPC systems:
  - Mio
  - Wendian
- Mio
  - Legacy system – 150+ TFLOPS
  - “Condo” model
    - Advisors/PI’s purchased nodes and have priority access to those nodes
    - Nodes available to all users, but may be kicked off ("preempted") if a user with node owner access requests those resources
- Wendian
  - Newest HPC System@Mines – 350+ TFLOPS
  - Priority Access Model
    - Advisors/PI’s purchase priority access to nodes through a quality-of-service (QoS) queuing system
Wendian @ Mines

• Still available for new users, pending PI proposal submission and approval

• Typical CPU node configuration
  • Intel Xeon Gold (Sky Lake) Dual Socket
    • 12-18 cores, 24-36 threads per socket
  • 192 GB – 384 GB Memory per node
  • ~3000 CPU core total on Wendian

• GPU and Power Nodes also available
  • NVIDIA Volta V100 x 4 Nodes
  • NVIDIA Tesla A100 x 4 Nodes
  • OpenPower 8 Nodes
  • Open Power 9 Nodes

What is high performance computing? - insideHPC
Who supports HPC@Mines?

- Cyber Infrastructure and Advanced Research Computing (CIARC) Group supports HPC@Mines!
  - **Director:** Matt Ketterling
  - **AD of Advanced Research Computing:** Dr. Torey Battelle
  - **Systems Administrator:** Mike Robbert
  - **Visualization Engineer:** Dr. Richard Gilmore
  - **Computational Scientist:** Dr. Nicholas Danes
Why use 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!
Quick Note on Python 2.7 vs 3.x

• Python 2.7 was end-of-life (EOL) on January 1\textsuperscript{st}, 2020

• Many Linux distributions are no longer shipping Python 2.7
  • Make the transition to Python 3.x as soon as possible!

• Some packages have not made the transition yet

• As of now, Python 2.7 still available on Mines HPC systems
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 manager *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 installed by going to Python.org
Getting started Python on your local system

• Cross-platform option: Use Anaconda
  • [https://anaconda.org](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](https://anaconda.org/conda-forge)

We will be using this today!
Writing your first program in Python!

Make a new file called `hello_world.py`:

```
#!/usr/bin/python
print("Hello World!")
```

Run the script in your command line:

```
$ python hello_world.py
Hello World!
```
Demo: Setting up & using Jupyter locally and on HPC

1. Let’s open Anaconda Navigator do the following:
   - Install packages/manage environments
   - Show Spyder IDE
   - Jupyter notebook locally on our machines!

2. Intro to HPC environment (Wendian)

3. Show how Jupyter can be used on Wendian via Open OnDemand
Further Resources

• Mines CIARC HPC Website:
  • https://ciarc.mines.edu/hpc
  • Pages are under construction!

• For HPC-related questions:
  • Submit a ticket to the help desk!
  • https://helpcenter.mines.edu/TDClient/1946/Portal/Requests/ServiceCatalog?CategoryID=11036

• More References:
  • https://realpython.com/matlab-vs-python/
  • https://matplotlib.org/3.1.1/tutorials/index.html
Day 2 Plans

- Using Python for Scientific Computing
  - Comparisons to MATLAB
  - 1D Poisson Differential Equation Test Problem
Questions?