Python on Trillium and Open OnDemand

Ramses van Zon

October 27, 2025

In this workshop...

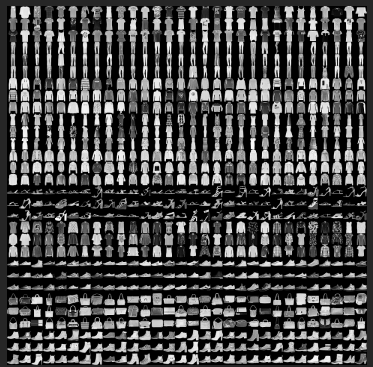
- Why Python?
- Why Supercomputers?
- Access
- Using Trillium
- Installing packages
- More about OnDemand

Why Python?

Python is great

- Python is a high-level, interpreted language.
- Python is fairly easy to learn, very expressive, and, not surprisingly, very popular.
- Its greatness is in large part due to the available packages.
- And in its interactive computing paradigm (=> Jupyter Lab)
- Development in Python can be substantially easier (and thus faster) than when using compiled languages.
- But the interpreted and dynamic nature of Python is often at odds with "high performance". **Yes, Python itself is slow!**
- This matters a lot less when Python is the 'driver' or 'glue language' for optimized packages or programs, such as for AI and ML.

Running example



- We have a data set of images of fashion items, (“T-shirt/top”, “Trouser”, “Pullover”, “Dress”, “Coat”, “Sandal”,“Shirt”, “Sneaker”, “Bag”, “Ankle boot”):
  See: https://github.com/zalandoresearch/fashion-mnist
- We want to train an artificial neural work on this data set so we could recognize items in other images.
- We’ll use PyTorch for this task.

This use case was taken from a PyTorch tutorial:
https://docs.pytorch.org/tutorials/beginner/basics/quickstart_tutorial.html

Although this example would be too small to warrant running on the Trillium supercomputer, it will demonstrate many aspects of running Python applications on such a system.

Why use a supercomputer?

Why use a supercomputer?

Your research project may need more resources than your laptop can provide.

This may be for several reasons:

1. Your research computations are too large to fit on your laptop.
2. The computations are too slow.
3. The computations are too plentiful.

So you go to one of the Alliance's 'advanced research computing' clusters: like Nibi, Fir, Narval, Rorqual and Trillium.

Digital Research Alliance of Canada

Congratulations, you are now doing Advanced Research Computing!

Advanced Research Computing

A supercomputer is just like your laptop

Haha! You didn't really think so, right?

We are going to need to make some adjustments.

Using a supercomputer is different

1. It is remote.
2. It's usually command-line driven.
3. It is a shared resource.
4. It is not your own machine.

But it's still got Python, right?

Well yes, but:

Many tutorials on Python, AI and ML assume that you are working on your own machine and have full privileges to reconfigure it (and mess it up).

We'll show you how to operate in this shared space, focusing in particular on Trillium (but touching upon the other national systems as well).

When do we get to running Jupyter notebooks?

Patience, we'll get there.

Getting started

Let's get onto Trillium!

What do you need to follow along this afternoon:

- An Alliance CCDB Account:
  https://ccdb.alliancecan.ca
- Setup MFA on CCDB
  https://ccdb.alliancecan.ca/multi_factor_authentications
- Access to Trillium (Resource -> Access Systems)
  https://ccdb.alliancecan.ca/me/access_systems
- Optional for today:
  - An ssh client;
  - Setup SSH keys. https://docs.alliancecan.ca/wiki/SSH_Keys

This will give you access to both Trillium terminal and SciNet's OnDemand service.

You can learn a lot more about using Trillium than we will cover today, in the self-guided course "Intro to Trillium", see https://scinet.courses/1389.

Logging in

Option 1: Through an ssh client

Connects directly to the Trillium command line.

The supercomputer runs the remote **ssh server**. You local computer run the **ssh client**.

- Open a (local) terminal
- Type (uses SSH keys):

```
ssh USERNAME@trillium.alliancecan.ca
```

- Use your Yubikey or Duo app as 2nd factor.
- You now get a command line prompt on a Trillium login node.

Option 2: Through Open OnDemand

This is SciNet's **web interface** to Trillium meant for interactive applications.

OnDemand can also be used to get to the Trillium command line in your browser.

- Go to https://ondemand.scinet.utoronto.ca
- Log in with your CCDB USERNAME and password. (note: don't use your email).
- Use your Yubikey or Duo app as 2nd factor.
- You can now go to "Clusters; Trillium Shell Access" to get a command line on one of the Trillium login nodes.

Hands-on 1

Hands-on 1 (5 min)

Get logged into Trillium by one of these two methods.

Then, type the command

```
$ which python
```

(and press Enter).

It should say:

```
/cvmfs/soft.computecanada.ca/gentoo/2023/x86-64-v3/usr/bin/python
```

Note: The dollar sign ("$") in the slides will be an abbreviation of the full prompt, which will look more like `[rzon@tri-login01 ~]$`.

Different organizations

A digression about all those different organizations

- **Digital Research Alliance of Canada**
- **CCDB**
- **Compute Canada**
- **SciNet**

Command line

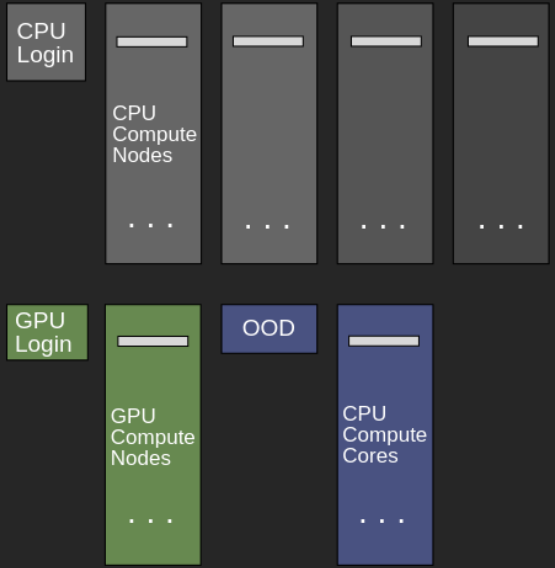
So we're always using this ~~black screen of death~~ command line?

Pretty much, yes, because

- In HPC and supercomputing, that's what people use.
- Any repetitive or large scale computational work requires working with the command line.
- Graphical User Interfaces (GUIs) would only offer existing functionality and GUI workflows are harder to automate or documents.
- Being familiar with the command line makes you more efficient, consistent, and productive in managing your data and your workflows.

Need to brush up on the Linux command line? SHARCNET has a self-guided course for that: https://training.sharcnet.ca/courses/enrol/index.php?id=182.

Understanding the Trillium system



Login nodes

- Ssh reaches the CPU or GPU login nodes.
- OnDemand reaches the OOD server.
- Shared among users
- Meant for preparing your work and software.

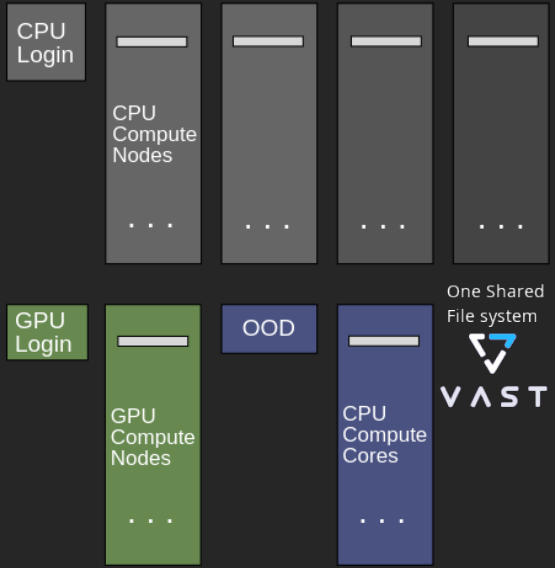
Compute nodes

- CPU: scheduled by 192-core node.
- GPU: scheduled by full NVIDIA H100 GPU.
- No internet access.
- Read-only home directory.

OOD compute cores

- Scheduled by core and memory
- Internet access.
- Writable home directory

Understanding the Trillium system



Login nodes

- Ssh reaches the CPU or GPU login nodes.
- OnDemand reaches the OOD server.
- Shared among users
- Meant for preparing your work and software.

Compute nodes

- CPU: scheduled by 192-core node.
- GPU: scheduled by full NVIDIA H100 GPU.
- No internet access.
- Read-only home directory.

OOD compute cores

- Scheduled by core and memory
- Internet access.
- Writable home directory

Software packages

It's a shared system

On Demand

Not everything needs 192 cores

But what if you have that one postprocessing step that you need less than 192 cores for? What if you need to do some visualization? For interactive work of that and other kinds in python, JupyterLab is typically used.

We installed the OnDemand to provide this JL and other features in the browser.

What is OnDemand?

Let's jump in (hands on)

In your browser, log into https://ondemand.scinet.utoronto.ca

Use your CCDB account Use your CCDB password Use your MFA

You'll see the ondemand interface.

OnDemand is this web interface. It was developed at OSC, and is getting widely adopted for many supercomputing systems. In Canada, Trillium, Nibi, and Vulcan, as well as on Grex

More

Introduction to JupyterLab

Best practices

Notebooks (Python, R, Julia)

Virtual Desktop, VS Code, OpenRefine, LibreQDA

SciNet

Resource Monitoring