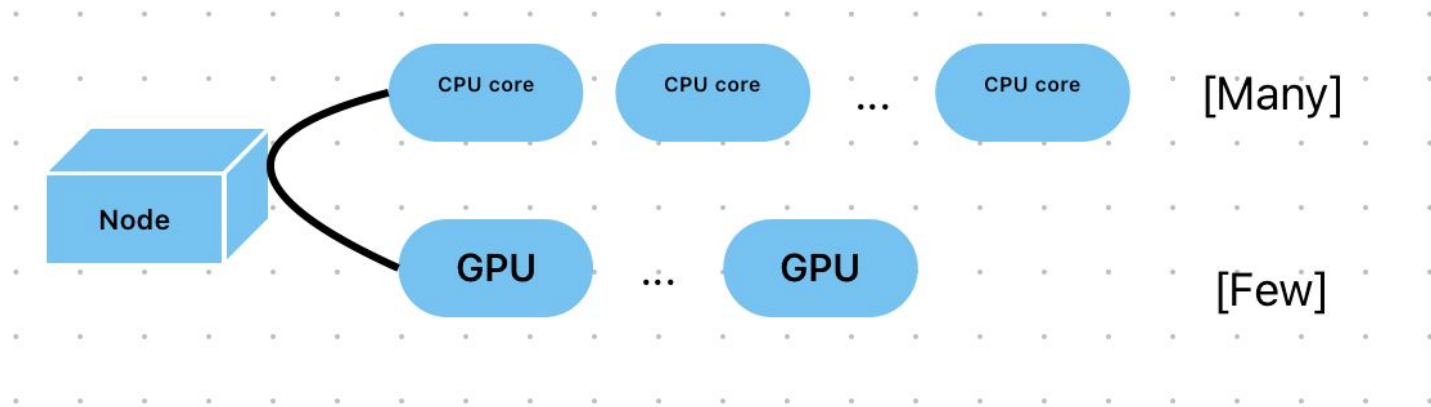# Slurm tutorial

Bohan Jiang

What is cluster?
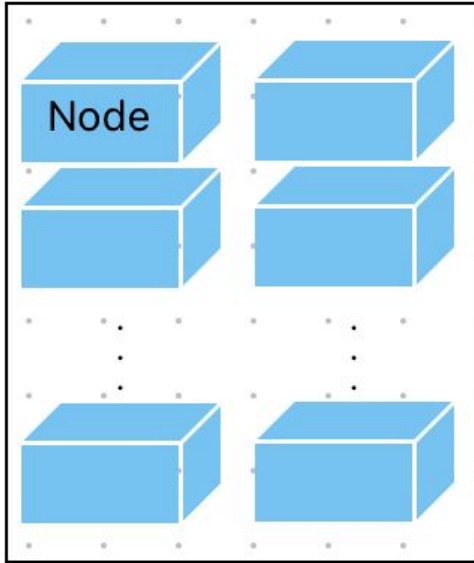
- A cluster is a group (or groups) of computers interconnected to each other, and perform computations for requested tasks.
- Hierarchy
  a. Cluster
  b. Partitions
  c. Nodes

Node

Partition
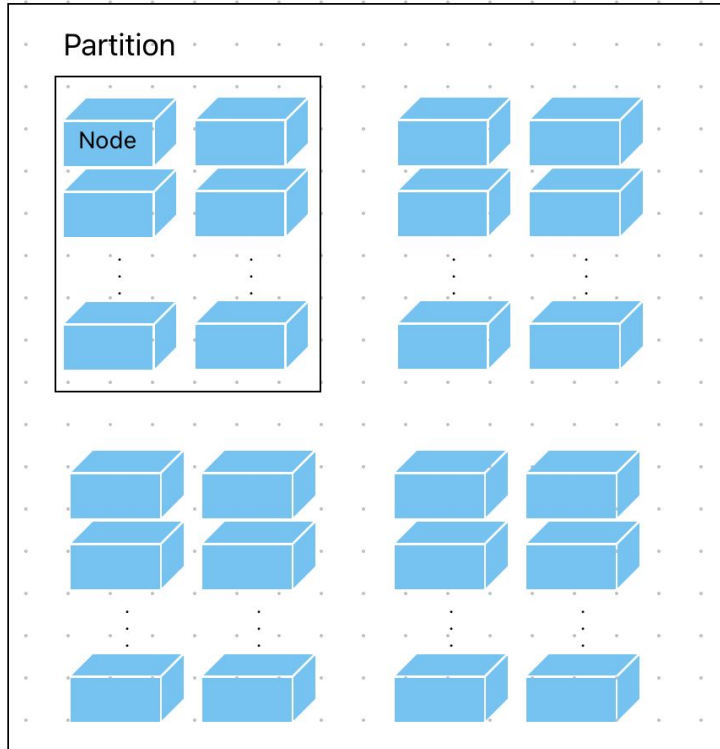

- Partition is a group of connected nodes that serves a specific purpose
  - Interactive partition for debugging
  - GPU partition training deep neural models

Cluster

Cluster

Partition

Node

- So, a cluster is a group of partitions
- Partitions may or may not communicate with each other
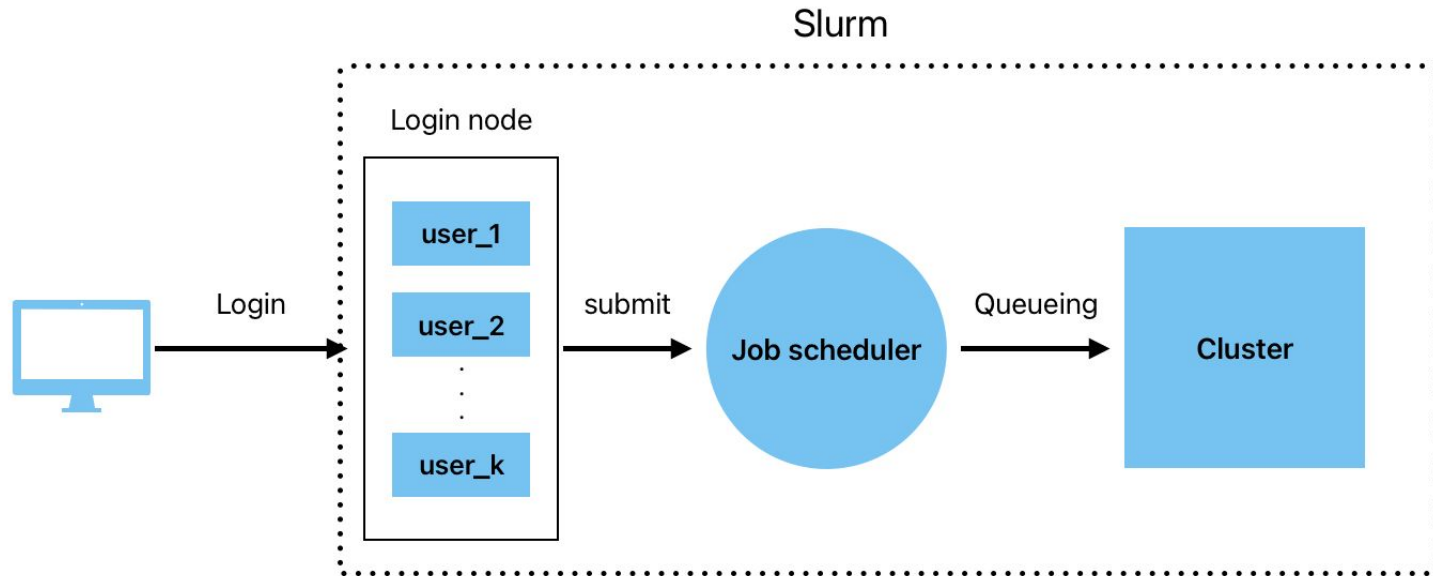
Brandeis CS student gpu cluster (SGC)

- 2 partitions
  - `regular`
    - `student-gpu-[001-004]`
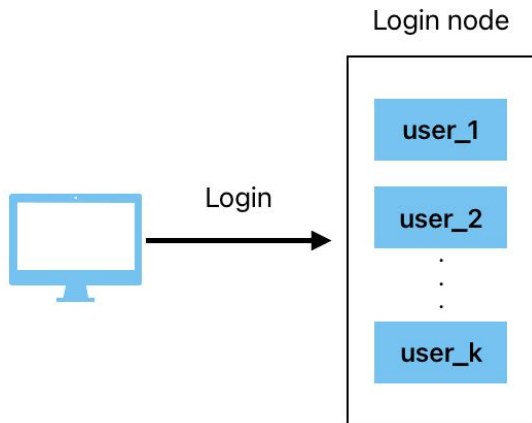 - `gpu48g`
    - `student-gpu-[003-004]`

Slurm

- Slurm is a queuing system that effectively manages the cluster to process submitted job tasks
- That means,
  - You don't have to monitor system usage before running your job
  - On the other hand, you're not the primary user in this world, so your request may be delayed or rejected if it's too resource demanding
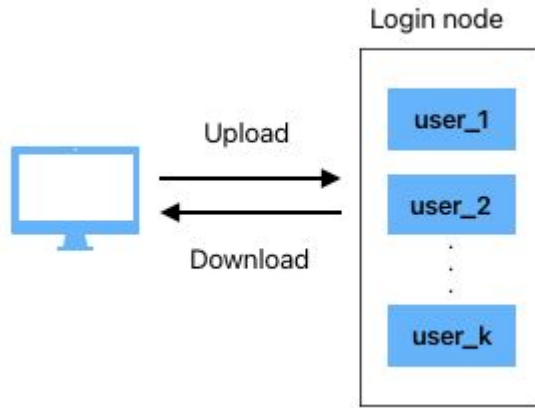
Overview

Login node

Login node

- Slurm provides a shared node for all users to log in from their local machines via ssh
  - On campus: direct ssh
  - Off campus: you need VPN
- This shared node is called "login node", it's used for submitting job scripts
- Login node has limited resource, so each user has even more limited resource (~2G memory), so DO NOT run any computation in this node
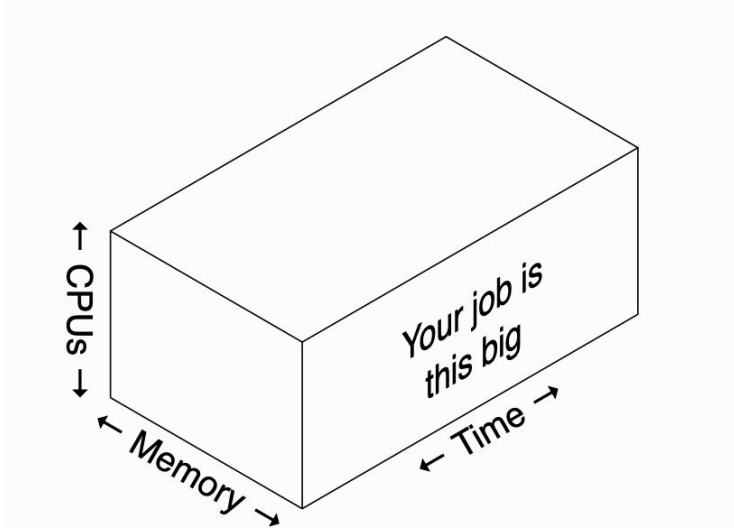
(For VScode users, the "remote-ssh" extension is not working here, as it would install softwares in login node which takes more memory than assigned)

How to edit and run your code?

Login node

Upload

Download

user_1

user_2

⋮

user_k

- ● Solutions
  1. Edit in your local, and run in the cluster

  OR

  2. Edit and run in your local in a small scale, and scale it up in the cluster
- ● Tools
  - ○ `scp` command
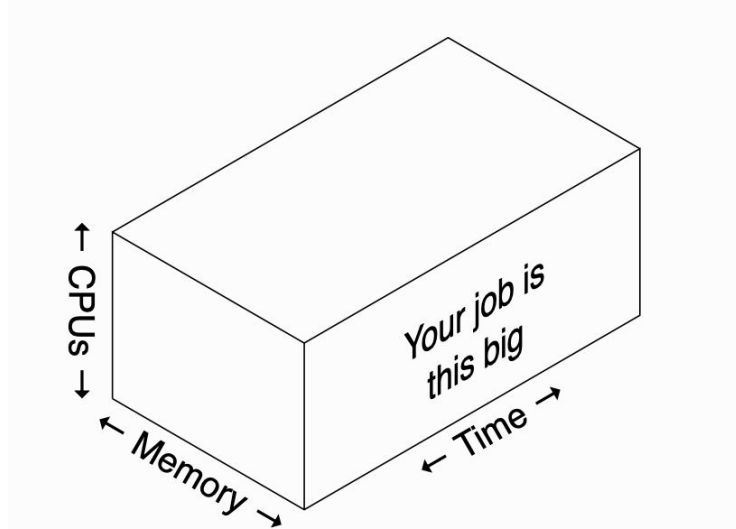 - ○ SFTP in IDE (e.g. pycharm)

Submit a job

- 2 scripts:
  - Your root/main script to automate your tasks
  - A "special" bash script
- 1 command:
  - `sbatch <bash_script>`

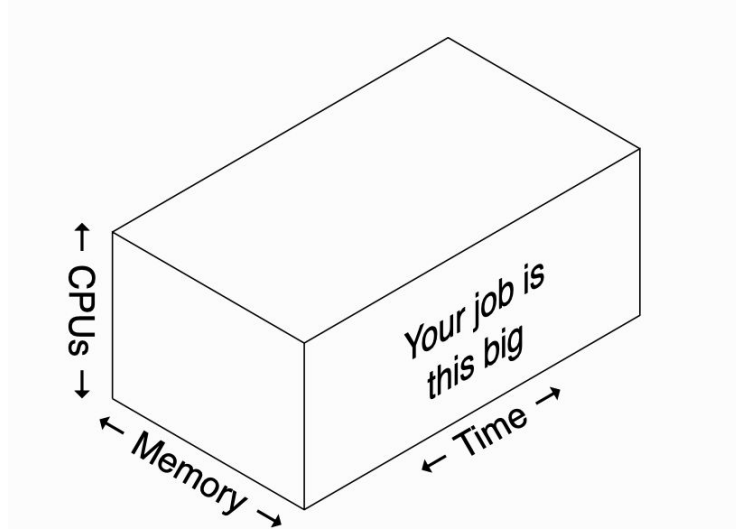*Source: https://scicomp.aalto.fi/triton/tut/slurm/*

Submit a job

- Before submission, consider sources you need:
  - Memory
  - Number of CPU cores
  - Number of GPUs
  - How long your job would take

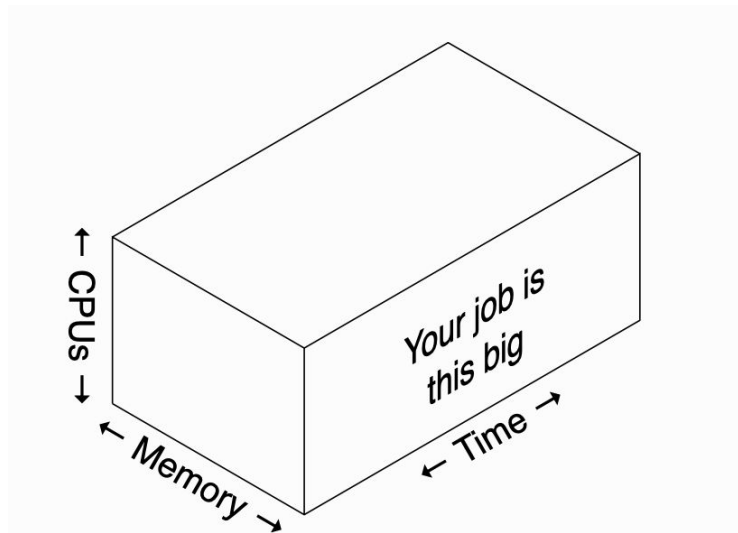*Source: https://scicomp.aalto.fi/triton/tut/slurm/*

Submit a job

- Configure the "special" bash script
- Assume you request
  - 4 CPU cores
  - 8G memory
  - 1 GPU device
  - Expect to run 20 mins

Source: https://scicomp.aalto.fi/triton/tut/slurm/

Submit a job


```
#!/bin/bash

#SBATCH --job-name demo
#SBATCH --ntasks 1
#SBATCH --cpus-per-task 4
#SBATCH --mem 8G
#SBATCH --gres 1
#SBATCH --time 00:20:00

cd <WORK_DIR>        # (optional)
export <ENV_VAR>   # (optional)
python main.py
```

Source: https://scicomp.aalto.fi/triton/tut/slurm/

More commands

- `squeue`: show information about job(s) in the queue
  - `squeue -u`: only show info about your job(s) in the queue
- `scancel`: cancel a job
  - `scancel <job_id>`: cancel a job by specifying its id
- `sacct`: show information about your current and previous jobs
- `sinfo`: show availability for nodes in the cluster

Debug results

- Pop quiz: how do you quickly know if your submitted job is done?
- Your job is done because of 2 reasons:
  - It's executed successfully
  - It's terminated due to errors
- No interactive session once you've submitted your job, where do you find stderr/stdout message?

  ```
  #SBATCH -o <stdout_path>
  #SBATCH -e <stderr_path>
  ```

- These files are often called log files, python has `logging` module which is more powerful than print statement
- Open them using a text editor or `less/cat` command

Demo: useful commands

Demo: CPU job

Demo: GPU job