## High Performance Cluster (HPC) Guide:

The Econ-Server guide contains commands that can be useful in the HPC, so please see that first. This guide is for using GPUs via batch jobs.

* Don't run jobs on the login node (figs-cluster)
* Don’t try to work interactively, but use Slurm batch jobs

1. Installing and Configuring HPC:

* Raise access: <https://docs.google.com/spreadsheets/d/1LEcPF_tNwX6tu-ElTDboUpMsO3C_o0krMLnikJwXTDw/edit?usp=sharing>
* Download the Google Cloud SDK installer for Mac from Google Cloud SDK Downloads and follow the installation instructions.
  + gcloud init
  + After running "gcloud init", sign in with your GCP account.
  + Using a Web Browser:
  + In the GCP Console with your web browser, please login with your NetID at the top right corner and select/type the GCP Project ID (gcp-gu-hpc-medusa).
  + To login to the HPC cluster, please select Compute Engine from the top left menu bar in the GCP Console.
  + Then click the SSH button on the <login\_node> instance.
* Using gcloud command-line tool:
* gcloud init
* Select the following values for each category for the project gcp-gu-hpc-medusa:

```

account = <Your\_NetID>@georgetown.edu

disable\_usage\_reporting = True or False

project = gcp-gu-hpc-medusa

zone = us-east1-c

```

* For example, the following shows commands to initialize the default configuration and to display it:

```

$ gcloud config list

[compute]

region = us-east1

zone = us-east1-c

[core]

account = <Your\_NetID>@georgetown.edu

disable\_usage\_reporting = True or False

project = gcp-gu-hpc-medusa

Your active configuration is: [default]

```

* After the initial setup, you should be able to login to HPC clusters as follows:

```

$ gcloud compute ssh figs-controller

```

* When you login for the first time, it may ask for a passphrase to generate a key. You can just type the Enter key twice without typing any passphrase.

2. **Moving Files:**

- Box: first create external password in box, then use:

- lftp -u pp712@georgetown.edu

- !ls (list files in server)

- ls (list files in box)

- put /home/<netid>/folder/example.py folder/example.py

- get folder/example.py /home/<netid>/folder/dist.py

- Moving files within GCP:

- gcloud auth login (and authenticate via browser)

- Moving files to Econ Department cluster:

- gsutil -m cp /home/pp712/folder/example.py gs://gu-econ-pi-pinetid/netid

- You can also use FileZilla, see Econ-Server guide.

3. HPC Cluster commands (Slurm):

* sinfo (get information)
* PARTITION: This column shows the name of each partition available on the cluster. Each partition may have different configurations and resource allocations.
* AVAIL: Indicates the availability status of the partition. Common values include "up" (available) and "down" (unavailable).
* TIMELIMIT: Shows the default time limit for jobs in each partition. This is the maximum wall clock time that jobs in the partition are allowed to run by default.
* NODES: Displays the number of nodes in each partition. This tells you how many compute nodes are available for scheduling jobs in each partition.
* STATE: Describes the current state of the nodes in the partition. Common states include:

- idle: Nodes are available and not running any jobs.

- alloc: Nodes have been allocated to a job and are actively running tasks.

- mix: A mix of idle and allocated nodes in the partition.

* NODELIST: Lists the specific nodes within each partition. Node names often follow a pattern, such as figs-base-ghpc-[0-11], which indicates a range of nodes.
* squeue -u <net-id> (look at your processes)
* srun ( to log into compute nodes)
* scontrol show job job\_id (look at your jobs)
* Use virtual environments to install packages and running Python CLI / scripts:

- pip install torch

- python3

- python3 example.py

* Get the config script (for different softwares)

$ ls $JB\_SCRIPTS

$ cat $JB\_SCRIPTS/README

$ cp $JB\_SCRIPTS/jb\_python.sh ./

5. Using GPUs:

- GPU does not exist on the login-node. So you need to use compute-nodes

- srun --partition=gpuspot --gres=gpu --pty bash

- python3 example.py

- sinfo

- nvidia-smi

- nvcc --version

- module avail cuda

- module load cuda

- python3 example.py

- batch jobs:

- #SBATCH --partition=gpuspot --gres=gpu

- #SBATCH --mail-user=<Your\_NetID>@georgetown.edu

- #SBATCH --mail-type=END,FAIL

- Running jobs:

- write example.sh

```bash

#!/bin/bash

#SBATCH --partition=gpuspot --gres=gpu

python3 example.py

```

- sbatch example.sh

- squeue -u <netid>

- scontrol show job <JobId>

- kill slurm-<jobid>.log

- tail -n 10 slurm-<jobid>.log

- scancel <jobid>

- scancel -u <Your\_NetID>

- Python CLI:

```python3

import torch

torch.cuda.is\_available()

```

6. Different versions on Python:

- module load anaconda3

- which python

7. Integrating with github

git config --global user.email "pranjal.rawat@outlook.com"

git config --global user.name "rawatpranjal"

git clone https://github.com/rawatpranjal/algorithmic-auctions.git

git add .

git commit -m 'dfs'

git push https://rawatpranjal:<TOKEN>@github.com/rawatpranjal/algorithmic-auctions.git --all

[chung@figs-controller ~]$ module avail git

---------------------- /home/share/apps/modules/modulefiles ----------------------

git/2.33.1

[chung@figs-controller ~]$ module load git

[chung@figs-controller ~]$ module list

Currently Loaded Modulefiles:

1) git/2.33.1

[chung@figs-controller ~]$ git --version

git version 2.33.0