

HPC Instruction

Chaojie Zhang

1. Download NYU VPN
<https://www.nyu.edu/life/information-technology/infrastructure/network-services/vpn.html>
2. Access HPC
<https://sites.google.com/nyu.edu/nyu-hpc/accessing-hpc?authuser=0>
`ssh <NYU_NetID>@greene.hpc.nyu.edu`
3. Greene Introduction
<https://sites.google.com/nyu.edu/nyu-hpc/home?authuser=0>
4. Slurm Tutorial
<https://sites.google.com/nyu.edu/nyu-hpc/training-support/general-hpc-topics/slurm-main-commands>
<https://sites.google.com/nyu.edu/nyu-hpc/training-support/tutorials/slurm-tutorial>
Check job status
`squeue -u $USER`
Cancel a job
`scancel JOBID`
5. Graphical User Interface (GUI) Tools
<https://ood.hpc.nyu.edu/>
6. Available GPUs
There are 200 GPU hours for everyone. The job will be canceled if there is low GPU usage for 2 hours.
`partitions = {"n1s8-v100-1", "n1s8-t4-1", "n1s8-p100-1" }`
7. Data Transfer (from your local PC or Google Drive)
PC: `scp -rp images.gz <NetID>@greene-dtn.hpc.nyu.edu:/scratch/<NetID>`
<https://sites.google.com/nyu.edu/nyu-hpc/hpc-systems/hpc-storage/data-management/data-transfers/transferring-cloud-storage-data-with-rclone>
8. Singularity with Miniconda
<https://sites.google.com/nyu.edu/nyu-hpc/hpc-systems/greene/software/singularity-with-miniconda>
9. Squash File System for image data
<https://sites.google.com/nyu.edu/nyu-hpc/hpc-systems/hpc-storage/data-management/squash-file-system-and-singularity>

Example:

```
ssh burst
```

```
# Total GPU hours
```

```
sacctmgr list assoc format=user,qos,defaultqos,account%20,GrpTRESMins%30 \
where account=bmsc_ga_4493 user=$USER
```

```
# GPU hours used
```

```
sshare --format=user,account,GrpTRESRaw%120 --account=bmsc_ga_4493 \
--user=$USER
```

```
# Run an interactive job
```

```
sbatch --account=bmsc_ga_4493 --partition=interactive --time=04:00:00 \
--wrap "sleep infinity"
```

```
squeue -u $USER
```

```
# Google Cloud Platform (different directory from Greene)
```

```
ssh NODE # use the node in your NODELIST, e.g. ssh b-9-1
```

```
# Create a pytorch environment using Singularity and Anaconda
```

```
mkdir /scratch/$USER/DL4Med
```

```
cd /scratch/$USER/DL4Med
```

```
scp -rp greene-dtn:/scratch/work/public/overlay-fs-ext3/overlay-7.5GB-300K.ext3.gz .
```

```
gunzip overlay-7.5GB-300K.ext3.gz
```

```
singularity exec --overlay overlay-7.5GB-300K.ext3 \
/share/apps/images/cuda11.3.0-cudnn8-devel-ubuntu20.04.sif /bin/bash
```

```
wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

```
sh Miniconda3-latest-Linux-x86_64.sh -b -p /ext3/miniconda3
```

```
vim /ext3/env.sh
```

```
# Create a wrapper script /ext3/env.sh using vim (basic vim commands), paste the
following lines, then [‘esc’] and ‘:wq’
```

```
#!/bin/bash
```

```
source /ext3/miniconda3/etc/profile.d/conda.sh
```

```
export PATH=/ext3/miniconda3/bin:$PATH
```

```
export PYTHONPATH=/ext3/miniconda3/bin:$PATH
```

```
# Activate conda environment and install packages
```

```
source /ext3/env.sh
conda update -n base conda -y
conda clean --all --yes
conda install pip
conda install ipykernel
pip3 install torch==1.10.2+cu113 torchvision==0.11.3+cu113 \
torchaudio==0.10.2+cu113 -f https://download.pytorch.org/whl/cu113/torch_stable.html
pip3 install jupyter jupyterhub pandas matplotlib scipy scikit-learn scikit-image Pillow
```

Exit the Singularity container and then rename the overlay image

```
exit
mv overlay-7.5GB-300K.ext3 DL4Med_pytorch.ext3
```

Copy the Jupyter Notebook script from Greene (upload the file first)

```
scp -rp greene-dtn:/scratch/$USER/run-jupyter-gpu-HW2.sbatch .
```

Run Jupyter Notebook, use the HW2 script for HW2, it includes the image data

Use cpu scripts if you don't need gpu to run the models

```
sbatch run-jupyter-gpu-HW2.sbatch
```

```
squeue -u $USER
```

```
cat slurm-xxxxx.out
```

Run in a new terminal, then open the link

```
ssh -L xxxxx:localhost:xxxxx NETID@greene.hpc.nyu.edu
```

Sometimes the v100 GPUs are not available, modify the run-jupyter-gpu.sbatch file

```
#SBATCH --gres=gpu:t4:1
```

```
#SBATCH --partition=n1s8-t4-1
```

Or

```
#SBATCH --gres=gpu:p100:1
```

```
#SBATCH --partition=n1s8-p100-1
```

Working with Data for Projects

1. Data transfer from your local PC

```
scp -rp images.gz <NetID>@greene-dtn.hpc.nyu.edu:/scratch/<NetID>
```

2. Data transfer from your Google Drive

You can also follow the instructions [here](#).

DON'T run 'ssh burst' after login to Greene

```
module load rclone/1.53.3
```

```
rclone config
```

```
n
```

```
remote1
```

```
13
```

```
<Enter>
```

```
<Enter>
```

```
1
```

```
<Enter>
```

```
<Enter>
```

```
n
```

```
n
```

Go to the link and copy the authorization code

```
<authorization code>
```

```
n
```

```
y
```

```
q
```

Copy data from your Google Drive

```
rclone copy remote1:<GoogleDriveDataPath> ./Data
```

3. Optional: Squash File System and Singularity

If you have many tiny files as fixed datasets, please make squashFS files to work with Singularity.

You can also follow the instructions [here](#).

Set access permissions for <images> folder

```
find images -type d -exec chmod 755 {} \;
```

```
find images -type f -exec chmod 644 {} \;
```

Convert to a single squashFS file

```
mksquashfs images images.sqf -keep-as-directory
```

4. Data transfer from Greene to GCP

Login to GCP then transfer the data

```
scp -rp greene-dtn:/scratch/$USER/images.sqf .
```

5. Sbatch file to run the model

```
#!/bin/bash
```

```
#SBATCH --job-name=Model
```

```
#SBATCH --nodes=1
```

```
#SBATCH --gres=gpu:v100:1
```

```
#SBATCH --time=24:00:00
```

```
#SBATCH --partition=n1s8-v100-1
```

```
#SBATCH --account=bmsc_ga_4493
```

```
module purge
```

```
singularity exec --nv \
```

```
--bind /home,/scratch,/share/apps,/mnt \
```

```
--overlay /scratch/$USER/DL4Med/DL4Med_pytorch.ext3:ro \
```

```
--overlay /scratch/$USER/DL4Med/Project/images.sqf:ro \
```

```
/share/apps/images/cuda11.3.0-cudnn8-devel-ubuntu20.04.sif \
```

```
/bin/bash -c 'source /ext3/env.sh; python /scratch/$USER/DL4Med/Project/model.py'
```