

Cluster

09 April 2025 16:45

```
prabal@Prabal:~$ ssh prghosh@access.grid5000.fr
prghosh@access-south:~$ ssh Nancy
prghosh@fnancy:~$ oarsub -I -I host=1,walltime=1:00
prghosh@grvingt-2:~$ ls
prghosh@grvingt-2:~$ cd p
prabal_1/ public/
prghosh@grvingt-2:~$ cd prabal_1/
prghosh@grvingt-2:~/prabal_1$ ls
prghosh@fnancy:~/prabal_1$ vim script_fff.sh
```

Use **ECHAP** key and press **:wq** and **ENTER** to save the script file in

```
(.venv) prghosh@fsophia:~/prabal_ghosh/lnria_Medical_Imageing_Internship_prabal$ cd ..
(.venv) prghosh@fsophia:~/prabal_ghosh$ ls
cluster_1_test.sh lnria_Medical_Imageing_Internship_prabal OAR.1249823.stderr
OAR.1249823.stdout run_job.sh
(.venv) prghosh@fsophia:~/prabal_ghosh$ cd lnria_Medical_Imageing_Internship_prabal/
(.venv) prghosh@fsophia:~/prabal_ghosh/lnria_Medical_Imageing_Internship_prabal$ ls
'cluster notes' 'New folder' s0002 s0005 s0008 s0124 test_s0005.ipynb
TotalsegmentatorMRI dataset_v200
cluster_test_0005.py papers s0003 s0006 s0009 s0560 test_s0006.ipynb
'my papers to read' s0001 s0004 s0007 s0010 software TotalSegmentator
(.venv) prghosh@fsophia:~/prabal_ghosh/lnria_Medical_Imageing_Internship_prabal$ cd ..
(.venv) prghosh@fsophia:~/prabal_ghosh$ ls
cluster_1_test.sh lnria_Medical_Imageing_Internship_prabal OAR.1249823.stderr
OAR.1249823.stdout run_job.sh
(.venv) prghosh@fsophia:~/prabal_ghosh$ vim cluster_1_test.sh
(.venv) prghosh@fsophia:~/prabal_ghosh$ vim cluster_1_test.sh
(.venv) prghosh@fsophia:~/prabal_ghosh$ oarsub -S ./cluster_1_test.sh
# Redirecting to the 'abaca' queue (new name for the 'production' queue)
# Setting queue to: p2
# Tie job resource request for GPU to resources with GPU
# Set walltime to default (3600 s).
OAR_JOB_ID=1250130
(.venv) prghosh@fsophia:~/prabal_ghosh$ oarstat -u
Job id Name User Submission Date S Queue
-----
1250130 prghosh 2025-04-10 17:25:37 W p2
(.venv) prghosh@fsophia:~/prabal_ghosh$
```

```
oarstat -u # Check status
cat OAR_${OAR_JOB_ID}.out # View output
cat OAR_${OAR_JOB_ID}.err # View errors
```

```
oarsub -S ./cluster_1_test.sh
```

1. **For interactive jobs (if you really need one):**

```
oarsub -I -I host=1/gpu=1,walltime=1:00:00 -p "gpu='16GB' AND gpu_compute_capability_major>=5" -q production
```

Then manually run:

```
./cluster_1_test.sh
```

```
oarstat -u # Check status
cat OAR_${OAR_JOB_ID}.out # View output
cat OAR_${OAR_JOB_ID}.err # View errors
```

```
oarsub -I -q production -I gpu=1,walltime=0:05:00
```

```
module load conda
module load cuda/11.8
conda activate mri_2025_4
cd ~/prabal_ghosh/lnria_Medical_Imageing_Internship_prabal
python cluster_test_0005.py
```

```
/usr/bin/oarsub -I -q production -I gpu=1,walltime=0:05:00
```

```
chmod +x cluster_1_test.sh
```

```
/path/to/cluster_1_test.sh
```

```
./cluster_1_test.sh
```

```
oarsub -S ./cluster_1_test.sh
```

UPLOAD DATA

```
prabal@Prabal:~$ scp -r lnria_Medical_Imageing_Internship_prabal
```

```
prghosh@access.grid5000.fr:sophia/prabal_ghosh/
```

```
prabal@Prabal:~$ scp test_s0005.ipynb  
prghosh@access.grid5000.fr:sophia/Inria_Medical_Imageing_Internship_prabal_1/
```

```
DOWNLOAD DATA
```

```
mri_2025_4
```

```
this virtual environment is working for cpu in cluster
```

Inside Ubuntu terminal

```
prabal@Prabal:~$ ssh prghosh@access.grid5000.fr
```

```
prghosh@access-north:~$ ssh sophia
```

Creation of virtual environnement

```
Module load cuda/11.8  
Module load conda
```

```
Conda create -n virtual_prabal python=3.10.0
```

```
Conda activate virtual_prabal
```

Installation of pytorch with gpu cuda 11.8- check these 2 methods inside your virtual environment

```
pip install torch==2.3.1 torchvision==0.18.1 torchaudio==2.3.1 --index-url  
https://download.pytorch.org/whl/cu118
```

```
From <https://pytorch.org/get-started/previous-versions/>
```

```
conda install pytorch==2.3.1 torchvision==0.18.1 torchaudio==2.3.1 pytorch-cuda=12.1 -c pytorch -c  
nvidia
```

```
From <https://pytorch.org/get-started/previous-versions/>
```

```
pip install TotalSegmentator
```

Now the setup is done. There might be some errors, which you will have to fix by looking at the error messages during code execution..

This is the python script

Vim cluster_test_0005.py

```
import nibabel as nib  
from pathlib import Path  
from totalsegmentator.python_api import totalsegmentator  
# import numpy as np  
# import os  
# import matplotlib.pyplot as plt  
  
input_path_1 =  
Path("/home/prghosh/prabal_ghosh/Inria_Medical_Imageing_Internship_prabal/s0005/m  
ri.nii.gz")  
output_path_1 =  
Path("/home/prghosh/prabal_ghosh/Inria_Medical_Imageing_Internship_prabal/s0005/s  
egmentations_test_2")  
  
img = nib.load(input_path_1).get_fdata()  
print(img.shape)
```

```
print(f"*****The .nii files are stored in memory as numpy's: {type(img)}.*****")

if __name__ == "__main__":
    # Segment the first MRI image
    print(f"*****Segmenting {input_path_1}*****")

    # totalsegmentator(input=input_path_1, output=output_path_1,device='gpu',
    task="total_mr", roi_subset= ["lung_left", "lung_right"])
    totalsegmentator(input=input_path_1, output=output_path_1,device='gpu',
    task="total_mr")

    print(f"*****Segmentation completed for {input_path_1}. Results saved
    to {output_path_1}.*****")
    print("thanks for using TotalSegmentator! prabal_ghosh")

=====
```

1. Method1:

This is the shell script

vim prabal_test_shell.sh

Inside this .sh file write the following code

```
#!/bin/bash
#OAR -q production
#OAR -l host=1/gpu=1
#OAR -l walltime=00:30:00
#OAR -p gpu_count > 0
#OAR -O OAR_%jobid%.out
#OAR -E OAR_%jobid%.err
# display some information about attributed resources
echo "=== Host and GPU Info ==="
hostname
nvidia-smi
nvcc --version

echo "=== Loading environment ==="
module load conda
module load cuda/11.8
conda activate mri_2025_4

echo "=== Checking PyTorch GPU Availability ==="
python3 -c "import torch; print('CUDA available:', torch.cuda.is_available());
print('Device:', torch.cuda.get_device_name(0))"
echo "=== Starting TotalSegmentator Job ==="
cd /home/prghosh/prabal_ghosh/Inria_Medical_Imageing_Internship_prabal
# Run your script
python cluster_test_0005.py
echo "=== Done ==="
conda deactivate
```

Now to run the shell script use the Following commands

```
chmod +x prabal_test_shell.sh      # is to check that its executable or not

oarsub -S ./prabal_test_shell.sh    # to run the script

oarstat -u # Check status
cat OAR_S(OAR_JOB_ID).out # View output
cat OAR_S(OAR_JOB_ID).err # View errors
```

2. Method2:

If you want to run it interactively, use the following commands — you don't need to write any shell script file.

```
oarsub -I -q production -l gpu=1,walltime=0:05:00
```

```
module load conda  
module load cuda/11.8  
conda activate mri_2025_4  
cd ~/prabal_ghosh/Inria_Medical_Imageing_Internship_prabal  
python cluster_test_0005.py
```