
HPC

Access HPC

You can access the HPC in two different ways:

1. You can access to the command-line/terminal:
    - Open the terminal
    - ssh -XY [username]@login.hpc.imperial.ac.uk
    - Insert password
    - module load anaconda3/personal
    - Source activate [name_conda_environment]

2. You can access the Jupyter session:
    1. https://jupyter.rcs.imperial.ac.uk/
    2. Choose amount of resources needed and press “Spawn”

Create a conda environment: first access to HPC

```
# LECTURE HPC - DATA SCIENCE HELPER TEAM

module load anaconda3/personal

# To run the first time you access the HPC
anaconda - setup

# Create a python conda environemnt
conda create -n test1 python=2.7 ipykernel # You can choose which python version to have
# Create a R conda enviornment
conda create -n test2 r=3.4.3 r-irkernel

#Activate the environment:
source activate test1

#Install desired packages:
conda install package_name[=version]

#Install python kernel for Jupyter:
python -m ipykernel install --user --name python2_test1 --display-name "Python2.7 (test1)"

# If you want an environment based on python 3
python3 -m ipykernel install --user --name python3_test1 --display-name "Python2.7 (test1)"

# If yoou want to install the R in your jupyter
R

IRkernel::installspec(user = TRUE)
IRkernel::installspec(name = 'ir343', displayname = 'R 3.4.3 (test2)')
```

Jobs submission

1. Write a python/R **script**
2. Write a **bash script**:
    1. Select how much **time** you think your job would take
    2. Select the **type and amount** of resources
    3. Specify the **path to your script** and the programming language required

3. **Submit** your job

Python script

The python script needs to be saved with **.py**
The R script needs to be save with .R

```
"""

Script to get the first N fibonacci numbers (defaults to 1.000)

"""

import sys



print("="*80)
print("-"*30 + " FIBONACCI " + "-"*30)

def fibo(n):
    L = [1,2]
    for j in range(n-2):
        L.append(L[-1] + L[-2])
    return L



def main():
    if len(sys.argv) > 1:
        N = int(sys.argv[1])
    else:
        N = 1000
    print("Getting the first {} fibonacci numbers".format(N))
    fibolist = fibo(N)
    print("Writing output to fibo_numbers.txt...")

    with open('fibo_numbers.txt', 'w') as file:
        for i,num in enumerate(fibolist):
            file.write("[{}] {}\n".format(i+1,num))

    print("-"*35 + " DONE " + "-"*35)
    print("="*80)


if __name__ == "__main__":
    main()
```

Bash script

How do you choose the amount of resources needed?
https://www.imperial.ac.uk/admin-services/ict/self-service/research-support/rcs/computing/job-sizing-guidance/

CPUs

```
#PBS -lwalltime=24:00:00
#PBS -lselect=1:ncpus=32:mem=124gb

module load anaconda3/personal
source activate test1

python3 $HOME/test_script.py

mkdir $WORK/$PBS_JOBID
cp * $WORK/$PBS_JOBID
```

The bash script needs to be saved with **.sh**

GPUs

```
#PBS -lwalltime=24:00:00
#PBS -lselect=1:ncpus=16:mem=96gb:ngpus=4:gpu_type=RTX6000

module load anaconda3/personal
source activate test1

python3 $HOME/test_script.py

mkdir $WORK/$PBS_JOBID
cp * $WORK/$PBS_JOBID
```

- Mainly required to train deep learning models
- Long waiting time
- If possible run with throughput

Submit the job

qsub bash_script.sh # To submit the job

qstat # To check whether your job is running or in the queue

OUTPUT:

1) Output file
2) Error file
3) Directory with output of the job script submitted

The output will be saved in your working directory unless specified differently