Python on NUS HPC

September 2019
Research Computing, NUS IT

Contents

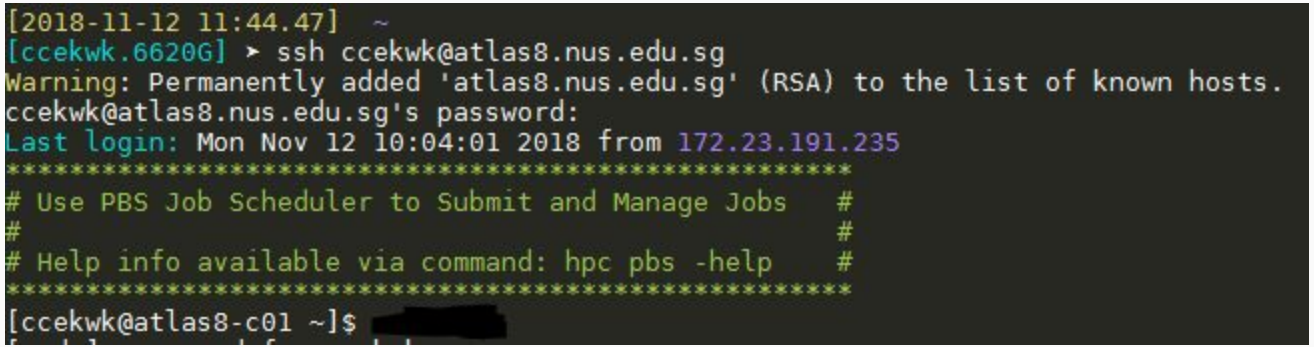
Access

Access

- Login via ssh to NUS HPC login nodes
  - atlas9
  - atlas6-c01.nus.edu.sg
  - atlas7-c10.nus.edu.sg
  - atlas8-c01.nus.edu.sg
- If you are connecting from outside NUS network, please connect to Web VPN first
  - http://webvpn.nus.edu.sg

Access

OS	Access Method	Command
Linux	ssh from terminal	`ssh nusnet_id@atlas8-c01.nus.edu.sg`
MacOS	ssh from terminal	`ssh username@hostname`
Windows	ssh using mobaxterm or putty	`ssh username@hostname`


[2018-11-12 11:44.47] ~
[ccekwk.6620G] ➤ ssh ccekwk@atlas8.nus.edu.sg
Warning: Permanently added 'atlas8.nus.edu.sg' (RSA) to the list of known hosts.
ccekwk@atlas8.nus.edu.sg's password:
Last login: Mon Nov 12 10:04:01 2018 from 172.23.191.235
Use PBS Job Scheduler to Submit and Manage Jobs
# #
Help info available via command: hpc pbs -help
[ccekwk@atlas8-c01 ~]$

Logging in

Resources

Resources: Hardware

- Standard CPU HPC Clusters
  - Gold
  - Atlas 5, 6, 7, 8, 9
  - Tiger
- GPU Clusters
  - 5 nodes x 4 Nvidia Tesla V100-32GB

Resources: Hardware/Storage

Directories	Feature	Disk Quota	Backup	Description
`/home/svu/$USERID`	Global	20 GB	Snapshot	Home Directory. U:drive on your PC.
`/hpctmp/$USERID`	Local on all clusters	500 GB	No	Working Directory. Files older than 60 days are purged automatically

Note: Type “hpc s” to check your disk quota for your home directory

Queues

Queue Name	Min. No. of CPU Cores	Max No. of CPU Cores	Max Nodes	Memory (Max) per node
parallel24	24 (2 Socket)	96 (8 Sockets)	4 (2 Socket/node)	197 GB
parallel20	20 (1 Sockets)	80 (4 Sockets)	4	196 GB
parallel12	12 (2 Sockets)	48 (8 Sockets)	4 (2 Socket/node)	49 GB
parallel8	8 (2 Sockets)	32 (8 Sockets)	4 (2 Socket/node)	49 GB
openmp	1	40 (8 Sockets)	4 (2 Socket/node)	49 GB
short	1	36 (3 Sockets)		49 GB
serial	1	1		49 GB
gpu	1	32 (8 Sockets)	4 (2 Socket/node)	49 GB

Anaconda Python

module load miniconda

Using Anaconda

To use conda, run the following commands **once** in the login node:

```
echo ". /app1/bioinfo/miniconda/3.6/etc/profile.d/conda.sh" >> ~/.bashrc
mkdir ~/conda_envs
echo “export CONDA_ENVS_PATH=~/conda_envs/” >> ~/.bashrc
```

Creating Conda Environments

- You can create conda environments in the login nodes with the following commands

```
bash
. ~/.bashrc
module load miniconda
conda create -n conda_env_name python=3.6
```

- You can replace 3.6 with the version you want.
  - e.g.: 2.7, 3.5, 3.7

Creating Conda Environment

It is suggested to install the conda environment on a login node that corresponds to the queue you intend to use.

There will be dependency issues otherwise.

Queue Name	Login Node to Use
parallel24	Atlas8
parallel20	Atlas8
parallel12	Atlas6, Atlas7
parallel8	Atlas6, Atlas7
openmp	Atlas6, Atlas7
short	Atlas6, Atlas7
serial	Atlas6, Atlas7

Activate Conda Environment

- To use the conda environment you've create, use the following command

```
conda activate conda_env_name
```

- The name of the conda env will then appear next to your shell prompt

```
(conda_env_name) [ccekwk@atlas8-c01 ~]$
```

- You can now install packages and use your python environment
- **Use Atlas 6/7/8 to install packages**
- **Atlas 9 has NO INTERNET ACCESS**

Installing Packages

```
conda install [-c channel_name] package_name -n env_name
```

Example:

```
conda install -c conda-forge numpy -n myenv
```

More details here:
https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-pkgs.html

PBS Job Script Template: Atlas8 Queue (CPU)

```
#!/bin/bash
#PBS -P Project_Name_of_Job
#PBS -j oe
#PBS -N Job_Name_1
#PBS -q parallel24
#PBS -l select=1:ncpus=24:mem=48gb
#PBS -l walltime=00:24:00

cd $PBS_O_WORKDIR;
np=$(cat ${PBS_NODEFILE} | wc -l);

source /etc/profile.d/rec_modules.sh

module load miniconda
bash
. ~/.bashrc

conda activate conda_env_name
python my_python_script.py
```

Green is user configurable
Black is fixed

Job Scripts for Other Queues

To utilise other queues (e.g.: parallel8, parallel12, short, serial)

- Change `#PBS -q` to target queue
- Change `#PBS -l select=1:ncpus=20:mem=48gb` to respective queue resource values. (Or do not include mem=XXgb to use default value)

Installing Python Packages

Python Packages

- **For Anaconda:** If you are using miniconda module,
  - please use conda to install packages (see next slide)
- To search Anaconda's package repository,
  - https://anaconda.org/anaconda/repo
- If the package you want to install is not available on Anaconda repositories, you can use pip install.
- If you need help, drop us a request
  - https://ntouch.nus.edu.sg/ux/myitapp/#/catalog/home
- **Use Atlas 6/7/8 to install packages**
- **Atlas 9 has NO INTERNET ACCESS. DO NOT USE**

Installing Packages using Conda

- Ensure that you have activated your conda environment on the login node
- When you are in your conda environment, use the following command to install packages

```
conda install package_name
```

- If you're using conda-forge

```
conda install -c conda-forge package_name
```

Installing tar.gz Packages using Conda

- If for some reason you're installing a .tar.gz downloaded from conda-forge or Anaconda's repository, use the following command to install

```
conda install package.tar.gz
```

1. For example if you download this:

https://anaconda.org/conda-forge/xgboost/0.81/download/linux-64/xgboost-0.81-py37hf484d3e_1000.tar.bz2

2. The command would be

```
conda install xgboost-0.81-py37hf484d3e_1000.tar.bz2
```

PBS Job Scheduler

Steps

You have to run:

1. Prepare your python script in your working directory
2. Create a PBS job script and save it in your working directory
    a. Example job scripts are in the following 2 slides
3. Submit PBS job script to PBS Job Scheduler

Server will run:

1. Job is in PBS Job Scheduler queue
2. Job Scheduler waits for server resources to be available
3. If available, Job Scheduler runs your script on remote server

Submitting a Job

Save your job script (previous slides for examples) in a text file (e.g. train.pbs) then run the following commands

```
shell$ qsub train.pbs
675674.venus01


shell$ qstat -xfn

venus01:
                                                                  Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
669468.venus01  ccekwk   azgpu    cifar_noco    --    1   1   20gb 24:00 F   --
    --
674404.venus01  ccekwk   azgpu    cifar_noco    --    1   1   20gb 24:00 F   --
   TestVM/0
675674.venus01  ccekwk   azgpu    cifar_noco    --    1   1   20gb 24:00 Q   --
    --
```

Statuses: Q(ueue), F(inish), R(unning), E(rror), H(old)

```
[ccekwk@atlas8-c01 classification]$ qsub train.pbs
697978.venus01
[ccekwk@atlas8-c01 classification]$ qstat -xfn

venus01:
                                                                  Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
695126.venus01  ccekwk   azgpu    cifar_noco    --    1   4   40gb 24:00 F   --
    --
697978.venus01  ccekwk   azgpu    cifar_noco    --    1   4   40gb 24:00 R   --
   TestVM/0*4
[ccekwk@atlas8-c01 classification]$
```

Statuses: Q(ueue), F(inish), R(unning), E(rror), H(old)

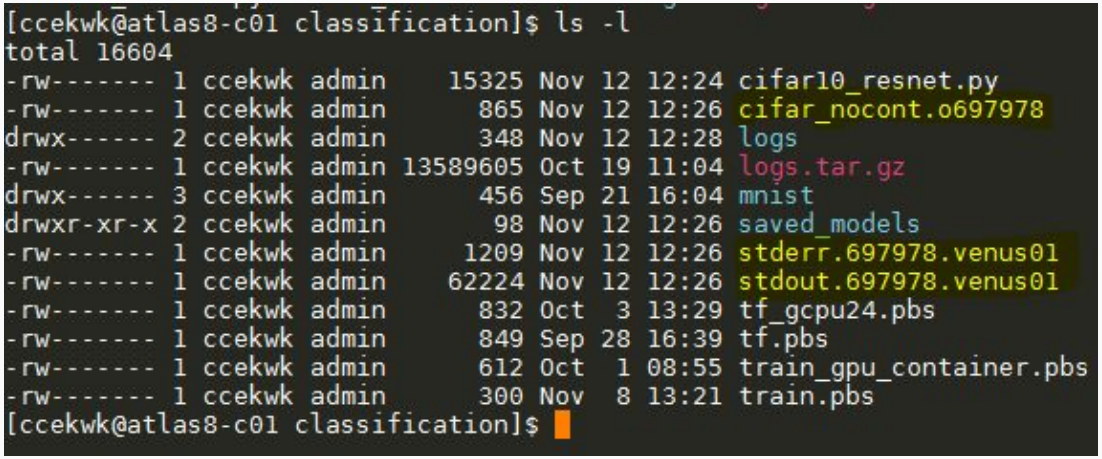
Submitting a Job

Useful PBS Commands

Action	Command
Job submission	`qsub my_job_script.txt`
Job deletion	`qdel my_job_id`
Job listing (Simple)	`qstat`
Job listing (Detailed)	`qstat -ans1`
Queue listing	`qstat -q`
Completed Job listing	`qstat -H`
Completed and Current Job listing	`qstat -x`
Full info of a job	`qstat -f job_id`

Log Files

- Output (stdout)
  - `stdout.$PBS_JOBID`
- Error (stderr)
  - `stderr.$PBS_JOBID`
- Job Summary
  - `job_name.o$PBS_JOBID`



FAQ

Permissions Error

If you encounter a permission error when creating environments or installing packages, execute the following commands:

```
mkdir ~/conda_envs
echo “export CONDA_ENVS_PATH=~/conda_envs/” >> ~/.bashrc
```

FAQ

Q: I submitted a job and it failed. The error is

```
#PBS: bad interpreter: No such file or directory
```

A: There are hidden characters (^M, etc) in your job script.

Check: cat -v my_text_file.txt

Fix: Use dos2unix tool to remove them, or manually remove them in vim.

This happens when you create or copy text files from Windows systems to Linux.

FAQ

Q: I encounter some tkinter error when using matplotlib, I need to install python-tk

A: tkinter is one of many backends for matplotlib. Tkinter is a GUI framework but our environment is headless.

Tkinter will not work. You'd have to use an alternative backend.

```
import matplotlib
matplotlib.use('agg')
import matplotlib.pyplot as plt
```

Help is available:

https://ntouch.nus.edu.sg/ux/myitapp/#/catalog/home