TACC System User Guide

Joohyun Lee (joohyun.lee@austin.utexas.edu, https://joohyun-lee.github.io/)
Updated 7/30/24

This tutorial guide is primarily written for Stampede3, but other TACC supercomputers, such as Frontera or Lonestar6, have similar system architecture and job submission systems, so this guide can be helpful for users of those supercomputers.
https://docs.tacc.utexas.edu/hpc/stampede3/ Referring to the official documentation provided by TACC is also a good starting point.
Before we start, if you have any questions, please ask ChatGPT first! It is really, really good at explaining things you are not familiar with, but widely used and well-known to other people!

Please feel free to suggest edits and reach out to me if you have any suggestions, comments, or questions about this guide.

1. How to access the system

Here, myusername is your username that you use to login to the TACC user portal.

```
ssh myusername@stampede3.tacc.utexas.edu
ssh myusername@login1.stampede3.tacc.utexas.edu
```

There are four login nodes and you can fix the login node by specifying the node number. This is useful if you use tmux (Section 2).

2. Useful bash/text editor setup & tmux

Setup .bashrc file, where you can store the default login setups. Refer to this article for more information: https://www.marquette.edu/high-performance-computing/bashrc.php.
Example .bashrc file: downloadable, google doc
Example .vimrc file: downloadable, google doc

In the example .bashrc file, you will notice lines like $PATH and $LD_LIBRARY_PATH. These are the paths that the system looks up executables or libraries including C/C++ libraries. Understanding the paths in the Linux system is very critical in installing, compiling, running, and debugging software.

The paths can also be added easily by "modules" on TACC computers. Refer to this part of the documentation https://docs.tacc.utexas.edu/hpc/stampede3/#admin-modules for details.

Tmux is a very useful tool when you work in a Linux terminal. Tmux allows you to retain sessions even if there's an interruption in network connectivity or you just want to go to sleep. Also, you don't have to open multiple windows locally and login to the cluster multiple times to have multiple panels. Read this blog post to learn how to use tmux: https://www.redhat.com/sysadmin/introduction-tmux-linux.

3. Set up the Python environment & use the TACC visualization portal

There are two ways to use Python: pre-installed Python module and conda environment.

To use the Python module, load the module by `module load python` (or `module load python/3.x.x`) and make your own virtual environment like this. Then from there, you should be able to install packages with pip. However, to use Jupyter notebook/lab on the TACC visualization portal, you need to install Jupyter locally and it has to be loaded automatically when you login. Follow the instructions here:
https://docs.tacc.utexas.edu/hpc/stampede3/#python

Another way to install Python is to use a conda environment through Anaconda or Miniconda. Anaconda is heavier but comes with commonly used packages and Miniconda is a minimal version of Anaconda. To install Miniconda, download the Miniconda distribution from here and execute the bash script to complete the installation. Then, in the same way, make a conda environment to `conda` or `pip install` packages. Installing Anaconda works in the same way as installing Miniconda. To use Jupyter notebook/lab on the TACC visualization portal, again you have to automatically load the conda environment with Jupyter installed when you login. This can be done in the .bashrc file.

The TACC visualization portal (https://vis.tacc.utexas.edu/) is a perfect platform for running a Jupyter notebook/lab to analyze data. You run a job on the system and access the computing nodes you requested to run Ipython notebooks (.ipynb). If you need to add kernels, refer to this documentation: https://ipython.readthedocs.io/en/latest/install/kernel_install.html.

See Section 6 for useful Python packages.

4. Submit and manage jobs

On supercomputer clusters, to run codes, you will have to submit jobs to request compute nodes and run codes on those nodes. This process is explained in the documentation: https://docs.tacc.utexas.edu/hpc/stampede3/#running. You can monitor jobs with this command: `squeue -u myusername` or `showq -u` and see the queue information with this command: `sinfo -S+P -o "%18P %8a %20F"`.

Example job.sh:

```
#!/bin/bash

#SBATCH -J reduce_C4      # Job name
#SBATCH -o stdout1         # Name of stdout output file
#SBATCH -e stderr1         # Name of stderr error file
#SBATCH -p skx            # Queue (partition) name
#SBATCH -N 1               # Total # of nodes (must be 1 for serial)
#SBATCH -n 32              # Total # of mpi tasks
#SBATCH -t 2:00:00        # Run time (hh:mm:ss)
#SBATCH -A TG-PHY24005    # Project/Alloc. name (req'd if you have more than 1)
#SBATCH --mail-type=all   # Send email at begin and end of job
#SBATCH --mail-user=your_email@utexas.edu

# Any other commands must follow all #SBATCH directives...

module list
which python
which mpirun
date

# export OMP_NUM_THREADS=68 # 1 thread/core
# export OMP_NUM_THREADS=48 # 1 thread/core

ibrun -np 32 python reduce.py
```

5. Globus

Globus is a useful tool for transferring data from/to/between clusters. It also supports a personal laptop with a downloadable app, which you can use to download/upload data from/to TACC computer systems. Go to https://www.globus.org/ and learn more about it.

6. Useful Python packages & other package managers

Basic Python packages:
Numpy
Scipy
Pandas
h5py
Jupyterlab

Visualization packages:
Matplotlib
Seaborn
OpenCV-Python

Generic astronomy packages:
Astropy
yt

High-performance computing packages:
Mpi4py
Multiprocessing
Dask

Machine learning packages:
Scikit-learn
JAX
PyTorch
Tensorflow

Debugging/linting:
Pytest
Ruff
Flake8

Package manager:
Pixi

Acknowledgment

Melanie Rowland’s Python guide was really helpful in writing this.