

# Notes on CUDA practicals on TACC's Frontera system

Prof. Mike Giles  
mike.giles@maths.ox.ac.uk

## 1 The Frontera system

The layout of the GPU nodes on the Frontera system is shown in Figure 1. Each node has 4 NVIDIA Quadro RTX 5000 GPUs but you will be using only one of them in the practicals.

The login node (which is actually a set of nodes) sits on the external network and can be accessed from machines on the university network through an SSH command:

```
ssh -X <username>@frontera.tacc.utexas.edu
```

Note that your username on TACC may be different to your username on your laptop or other home system which is why you will probably need to specify it.

The `-X` option provides X-window forwarding which is convenient when using windows-based editors such as `emacs`, or the Nsight IDE. Alternatively, TACC [recommends](#) the use of either DCV or VNC for remote viewing.

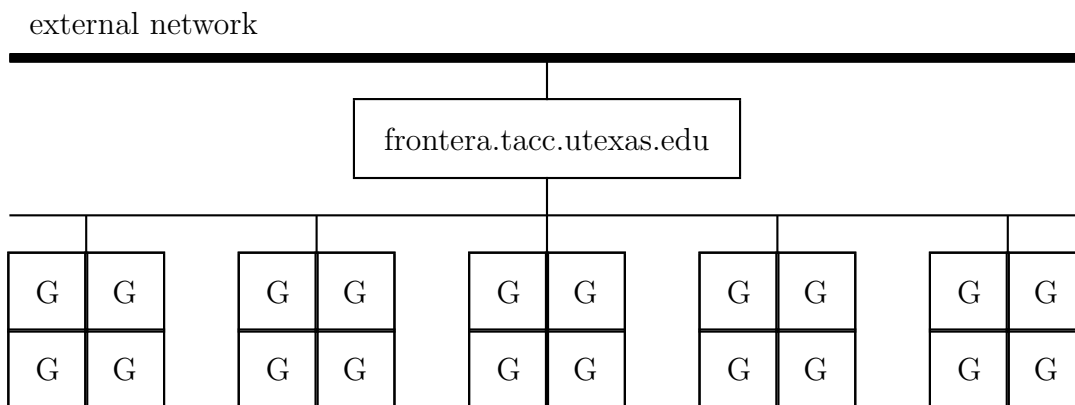


Figure 1: TACC's Frontera system with login node(s) and compute nodes, each with 4 NVIDIA Quadro RTX 5000 GPUs

## 2 Practical files

The tar file `practicals.tar.gz` can be downloaded from <https://people.maths.ox.ac.uk/gilesm/cuda/practicals.tar.gz> and put in your working directory using the commands:

```
cd $WORK
wget https://people.maths.ox.ac.uk/gilesm/cuda/practicals.tar.gz
```

It can then be unpacked using the command:

```
tar zxvf practicals.tar.gz
```

which will give you a separate sub-directory for each practical plus a sub-directory with a few header files.

## 3 CUDA module

To use the CUDA compiler and other CUDA tools you need to add the CUDA module using the command:

```
module add cuda
```

which will load the default version of the CUDA module which is currently 12.2. The command:

```
module save
```

will ensure that the module is automatically reloaded the next time you login.

Some of the later practicals may require you to set the following environment variable so that Make can correctly find certain CUDA library header files:

```
export CUDA_HOME=/home1/apps/cuda/12.2/
```

**WARNING:** doing a cut-and-paste of this instruction may not get the underscore `_` and may instead set the value of your `HOME` environment variable which will cause you various problems.

## 4 Interactive sessions on the GPU nodes

Editing, and perhaps also compiling, is best done on the login node, but to run your CUDA codes you need to create an interactive session on one of the GPU nodes by issuing the following command:

```
idev -N 1 -n 1 -m 120 -p rtx-dev
```

(one node, one job, 120 minutes, `rtx-dev` queue for GPUs)

You can then compile your code as usual, and run it on one of the GPUs attached to the compute node. Using an interactive session like this can be appropriate when developing codes using test runs which last no more than a few seconds, but for anyone doing “production” runs you should probably use the [Slurm batch scheduler](#).