Spring 2023

ECE 361E: Machine Learning and Data Analytics for Edge AI
Appendix A1

A1.1. Setting up the TACC Environment

We use TACC machines to train the models on powerful GPUs. Each account can have 4 running jobs simultaneously on Maverick2. You will access your TACC account remotely from your computer. This can be done via SSH. Open a terminal (or Windows bash/PowerShell for Windows users) and use the command:

```
ssh <your_TACC_username>@maverick2.tacc.utexas.edu
```

You need to first introduce your password and then access the DuoMobile app for a second security code required to access UT TACC instances. While in TACC, **always use the $WORK directory**! To change directory to $WORK use the `cdw`[1] command.

IMPORTANT! Every single time you SSH into TACC, it is compulsory to run the `cdw` command as the first command. If you type **`pwd`** and you do not see a directory path starting with **`/work`** (but you see for example **`/home`**), then you are working on the wrong path and you will encounter errors. Make sure you are in the **`/work`** directory and every file or virtual environment is also there. Then everything will run just fine on TACC as long as you don't have errors in your code.

1) Load the Python3 module in the system with the following command:

```
module load python3
```

2) To validate, you can check if you have Python3 by typing:

```
which python3
```

3) We use *virtualenv*, which is installed by default. You can check it using:

```
which virtualenv
```

4) To avoid package conflicts with other Python packages and versions on the machine, we create a virtual environment using *virtualenv*:

```
virtualenv -p '<output_of_which_python3>' $WORK/HW1_virtualenv
```

5) We activate the virtual environment:

```
source $WORK/HW1_virtualenv/bin/activate
```

6) Check your current working directory (you will need it next) using `pwd`; it should give you something like:

```
/work/<number>/<your_TACC_username>/maverick2
```

7) Download ***HW1_files.zip*** from Canvas and unzip it.
8) Open a new terminal on your computer and go to the location where you downloaded HW1_files:

```
cd Downloads/
```

9) Move the files from your computer to Maverick2 using `scp`[2]

```
scp -r HW1_files <your_TACC_username>@maverick2.tacc.utexas.edu:<pwd_output>
```

10) After moving the files from your computer to Maverick2, on the terminal for TACC install the required packages from ***requirements.txt***:

```
pip install -r $WORK/HW1_files/requirements.txt
```

---

[1] **This should be the first command when you login on TACC using SSH.**

[2] `scp -r` is used to move recursively to all subdirectories and files.

A1.2. Running Jobs on TACC machines

1) ***After you write and test the code for a problem on your computer***, copy the Python file(s) from your computer to Maverick2. Open a new terminal window on your computer and execute:

```
scp <your_file_location>
<your_TACC_username>@maverick2.tacc.utexas.edu:<pwd_output>/HW1_files
```

2) After the code has run and plots are produced, the CSV files and other result files have been created, you will move them from Maverick2 to your computer. Open a new terminal window on your computer and execute:

```
scp -r <your_username>@maverick2.tacc.utexas.edu:<pwd_output>/HW1_files
                                <your_destination>
```

3) You will then edit the code in the job file ***config.slurm***.

[Required] Make sure to change <username>@utexas.edu to your UT email/ email associated with your TACC account

[Optional] Change the desired name of the job

Click here to read more on Maverick2.

4) From the ***config.slurm*** you can see that the output can be accessed in: `$WORK/HW1_files/out`. Everything you need is already provided there, but feel free to personalize the script according to your needs, if needed.

5) Start the job by executing:

```
sbatch config.slurm
```

WARNING: Be sure to *test your code on your local machine* and only when you see it working with no errors (and training the model), move it to Maverick2. For a good testing experience, we suggest to use 1 epoch for training and the ***'cpu'*** as the device to run your code such that it will not take too much time and it will verify the full functionality of your code (i.e., CSV files generation, plot creation, etc.).

A1.3. Interacting with TACC Instances and Managing Jobs

1) After you start the job, check the running jobs using `squeue`. You will see:
   - JOBID - the unique ID associated to your submitted job
   - PARTITION - the type of machine running your job (for you should be gtx)
   - NAME - the name you gave your job in the ***config.slurm*** file
   - USER - <your_TACC_username>
   - TIME - time spent running this job
   - NODELIST - the address of the machine running your job

NOTE: If your job is not listed, then it might have been killed due to an error. You will get an email confirming your job has stopped mentioning if it was Completed, Canceled, or Failed due to an error.

2) With the information obtained from `squeue` you can access the machine your job will be running on. Use the NODELIST from above and SSH into that machine:

```
ssh <NODELIST>
```

3) Once you are in that machine you can view the CPU statistics by running `htop` or you can monitor the GPU continuously, updating the readings at every 0.1 seconds using:

```
watch -n 0.1 nvidia-smi
```

4) To cancel a job simply run:

```
scancel <JOBID>
```