

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

### Appendix A3

#### A3.1. Running Tasks in Parallel on TACC

Running a single Python program on a computing node with 4 GPUs is a waste of resources. As this class focuses on efficiency from different perspectives, it comes naturally to teach you how to run commands in parallel, each of them on a different GPU.

- 1) To run in parallel, in the *config.slurm* file provided in *HW3\_files* place the two commands you want to run in parallel one after the other.
- 2) Add at the beginning of a command **CUDA\_VISIBLE\_DEVICES=0** to make that specific command to run on the first GPU (i.e., GPU0). Do the same with the rest of the commands, assigning them to run on the other GPUs 1, 2 or 3.
- 3) Add “&” at the end of each command; this will make the processes run in the background.
- 4) After inserting all commands, write “**wait**”. If the processes are running in the background, you want to wait until all of them finish running, and not kill the entire script (with all its running processes) when the first command from the background finishes running.
- 5) Example:

```
CUDA_VISIBLE_DEVICES=0 python main.py &  
CUDA_VISIBLE_DEVICES=1 python main.py &  
wait
```

#### A3.2. Installing ONNX for Linux on your personal computer

- 1) The following steps are for starting using [Conda](#). Python 3.9 is highly recommended.
  - `$ sudo apt-get update`
  - `$ sudo apt-get install -y python3 python3-dev python3-setuptools gcc libtinfo-dev zlib1g-dev build-essential cmake libedit-dev libxml2-dev`
  - `$ conda create --name ml python=3.9`
  - `$ conda activate ml`
- 2) Make sure you have vim installed or run `$ sudo apt install vim`
- 3) While inside the virtual environment created in step 1), execute the following
  - `$ pip install onnx onnxruntime`

#### A3.3. Connecting to the Devices

- 1) For RaspberryPi: `ssh student@sld-rpi-<your RPi number>.ece.utexas.edu`
- 2) For Odroid MC1: `ssh student@sld-mc1-<your MC1 number>.ece.utexas.edu`

#### A3.4. Monitoring Power and Temperature on RaspberryPi 3B+ and MC1

- 1) For Raspberry Pi 3B+ you need to measure temperature as follows:

```
import gpiozero  
cpu_temp = gpiozero.CPUTemperature().temperature
```
- 2) We provide an example code at *get\_power\_temp\_mc1.py* for MC1. We also provide the example code for *get\_power\_temp\_rpi.py* for RaspberryPi 3B+.

### **A3.5. Monitoring Memory Consumption on Edge Devices**

- 1) For both RaspberryPi 3B+ and Odroid MC1, to get the memory consumption you need to run the following command:

```
free -m
```

which gives you the total, used and free RAM memory in MB. The memory consumption stabilizes during the inference process so you can see exactly how much memory the inference consumes compared to the idle state of the device.