Task 01

Login to GPU cluster and submit an interactive job to get into development mode.

Issue "`lspci | grep V100`" command w/o quotes. Exit from the development mode.

A. What is the hostname of the GPU node you were assigned to?

B. Which command did you use to submit an interactive job?

C. What is the name of Nvidia GPU returned when you issued `lspci` command?

Task 02

Start development mode on the system by submitting an interactive job. Use "`module avail`" command to identify available CUDA versions on the system. DO NOT exit from the job yet.

A. Which specific CUDA versions are available on the system?

B. What is the latest CUDA version available on the system?

C. Which command did you use to load latest version of CUDA?

D. Did you see any MATLAB version available when you issued "`module avail`"? If yes, what is the version of MATLAB available on the system?

Task 03

While you are still in development mode, load the latest version of the CUDA on GPU node and issue "`nvidia-smi`" command. Exit from the interactive mode.

From the output answer following;

A. CUDA version?

B. Driver Version?

C. GPU memory?

Before Task 04, copy example files from /lustre/share/examples/gpu-tutorial.tar to your home directory.

```
cp /lustre/share/examples/gpu-tutorial.tar .
tar -xvf gpu-tutorial.tar
cd gpu-tutorial
```

Task 04

Start development mode by submitting interactive job to the cluster. Compile and run a sample CUDA code. Navigate to "01_cuda" directory. Review the readme file and compile/run the CUDA code.

A. Issue "`nvcc --version`". What is the version of nvcc you see?

Task 05

Start development mode by submitting an interactive job and create Anaconda Python virtual environment and install few packages.

1. Issue command "`which python`" and "`python -V`"

   A. What is the python executable path you see?

   B. What is the Python Version?

2. To be able to use Anaconda, we need to make conda command available in our environment. For this add below line in your .bashrc by issuing below

   ```
   echo "source /cm/shared/apps/anaconda3/etc/profile.d/conda.sh" >> .bashrc

   source ~/.bashrc
   ```

3. Create virtual environment named "myproject" with python 3.6

   ```
   conda create -n dlproject python=3.6
   ```

4. Activate the newly created Python Virtual Environment

   ```
   conda activate dlproject
   ```

5. Repeat Step 1 and note following;

A. What is the python executable path you see?

B. What is the Python Version?

6. Deactivate Conda environment

```
conda deactivate
```

7. Issue "`conda env list`" command.

A. Do you see your newly created environment? If yes, what is the name?

8. Reactivate the conda environment you just created and install following Python packages.
   a. tensorflow-gpu
   b. keras

A. Which command did you use to install the python package in conda virtual environment?

9. Navigate to "02_python" directory and run sample code "tfkeras.py" in development mode.

```
python tfkeras.py
```

A. How many epochs were completed?

B. What is the final accuracy?

C. Increase the epochs to 10 and run the code again. What is the final accuracy now?

10. Run sample code in production mode by submitting a batch job file.
    Hint: Modify "gpu.job" file.

```
sbatch gpu.job
```

A. How many epochs were completed?

B. What is the final accuracy?

Task 06

Launch MATLAB in development mode and run example codes placed under "03_matlab"

1. Start development mode by submitting an interactive job to the system.
2. Load Matlab module by issuing "`module load matlab`"
3. Setup Display variable in your environment. This will allow MATLAB graphical session to forward the stream to your local desktop system. Refer to "https://rc-docs.qatar.tamu.edu/index.php/Raad2_faqs" X11 Forwarding.
4. Launch MATLAB by issuing "`matlab`" command.
5. Issue command "gpuDevice" and observe the output.
6. Run mandlebortcpu.m code from Matlab GUI session.
7. Run mandlebortgpu.m code from Matlab GUI session.

    A. What is the compute capability of the GPU present on the GPU node?
    Hint: Look for output in gpuDevice.

    B. How much time it took to run mandlebort GPU version?

    C. How much time it took to run mandlebort CPU version?

8. Submit batch job to run mandlebortgpu.m in production mode.
    Hint: Modify gpu.job and issue below command;

    ```
    sbatch gpu.job
    ```

Task 07

Launch Jupyter Lab on GPU cluster

Step 01: On Local machine, login to raad2-gfx with port forwarding

```
ssh -L PORT:localhost:PORT username@raad2-gfx.qatar.tamu.edu
```

Step 02: Start development session on raad2-gfx by submitting interactive job

```
srun --ntasks=18 --gres=gpu:v100:1 --tunnel LPORT:PORT --pty /bin/bash
```

Step 3: Setup conda environment

```
source /cm/shared/apps/anaconda3/etc/profile.d/conda.sh
```

Step 4: Create and activate Conda Environment

```
conda create -n myproject python=3.6

conda activate myproject

conda install jupyter

conda install jupyterlab
```

Step 5: Unset XDG_RUNTIME_DIR (Fix for known bug)

```
unset XDG_RUNTIME_DIR
```

Step 6: Launch Jupyter Notebook

```
jupyter lab --port PORT
```