CMPE 297 Lab#1

Due: Friday, Aug 26, 5:15pm
Total Score: /100

Instructor: Hyeran Jeon

Computer Engineering Department, San Jose State University

Group ID				
Member 1	Name		Student ID	
Member 2	Name		Student ID	

In this Lab, you will compile a code that runs CUDA APIs to check the properties of the GPU in your system. Following the steps below and answer the questions. Submit this form when you leave the class.

Step 1: Board setup and login

- Connect the provided Jetson board to the monitor in front of you via HDMI cable.
- Connect keyboard and mouse to the USB hub attached to the board.
- Turn on the provided Jetson board
- Login with the following credentials
  - ID: ubuntu
  - PASSWORD: ubuntu

Step 2: CUDA toolkit installation

- Open web browser (*epiphany*), go to Canvas→Labs→Lab1 and download following files
  - cuda-l4t.sh
  - cuda-repo-l4t-7-0-local_7.0-76_arm64.deb
- Open *terminal* and type following commands
  - cd "*folder that you downloaded the two files"*
  - sudo ./cuda-l4t.sh cuda-repo-l4t-7-0-local_7.0-76_arm64.deb 7.0 7-0
- Check if cuda toolkit is successfully installed by typing the following command

- nvcc –version
- You will see the following message if the CUDA toolkit is successfully installed

  nvcc: NVIDIA (R) Cuda compiler driver

  Copyright (c) 2005-2015 NVIDIA Corporation

  Built on Sun_Nov_15_11:52:02_CST_2015

  Cuda compilation tools, release 7.0, V7.0.72

Step 3: Compile a CUDA code

- Open web browser, go to Canvas→Labs→Lab1 and download following file
  - lab1.tar.bz2
- Type the following commands to uncompress the file
  - cd "*folder that you downloaded the two files"*
  - mkdir ~/CUDACourse
  - cp lab1.tar.bz2 ~/CUDACourse
  - cd ~/CUDACourse
  - tar –xvjf lab1.tar.bz2
- Check the code "deviceQuery.cpp"
  - This code prints GPU device properties. Followings are CUDA related code you can find in the file.
    - #include <cuda_runtime.h>
      - Should include when you use CUDA APIs
    - cudaGetDeviceCount()
      - return # GPUs in the system
    - cudaGetDeviceProperties(&deviceProp, dev)
      - return various properties of the selected GPU
      - deviceProp
        - totalGlobalMem // GPU device memory size
        - multiProcessorCount // SM count
        - warpSize // # threads in a warp
        - sharedMemPerBlock // maximum shared memory size that can be allocated by each thread block
        - maxThreadsPerMultiProcessor // maximum # threads that can be used in each SM

- maxThreadsPerBlock // maximum # threads that can be assigned per thread block
- Many more..

- Check Makefile
  - This file helps to compile CUDA code
    - CUDA_HOME := /usr/local/cuda
      - CUDA home directory path
    - CC := nvcc
      - CUDA compiler
    - LIB := -lcudart
      - CUDA runtime library
- Type following command to compile deviceQuery.cpp
  - make
- You can also compile the code by simply running the following command:
  - nvcc –o deviceQuery deviceQuery.cpp

Step 4: Run deviceQuery

- Type following command to run deviceQuery
  - ./deviceQuery
- Check the GPU properties and fill the following table

CUDA Driver Version	
CUDA Capability version	
# Multiprocessors	
# CUDA Cores per Multiprocessor	
Size of constant memory	
Max size of shared memory per thread block	
Max # registers per thread block	