CS5225 Parallel and Concurrent Programming

Take Home Lab 2

Due – June 21 before 11:55 PM

Learning Outcomes

In this lab we will learn the programming model for Graphics Processing Units (GPUs) and how to develop parallel programs for GPUs. At the end of the labs you will be able to:

- understand the fundamental concepts of GPUs, CUDA programming model, parallel programming, and well-known processing patterns
- develop simple algorithms to solve embarrassingly parallel programs using GPUs

Step 1: Each lab group consists of up to 2 students. Find your lab buddy.

Step 2: Register to follow the Udacity course entitled "Introduction to Parallel Programming" conducted by John Owens, David Luebke, Cheng-Han Lee, and Mike Roberts. Course page can be accessed from https://www.udacity.com/course/cs344

Step 3: Follow the first 2 lessons. Make sure to attempt all the exercises.

Step 4: While following the lessons, try to answer the following questions.

1. Certain ALU operations to be carried on an array with 200 elements. Array elements can be processed independently from each other. It takes 50 ns to perform those ALU operations on a single element using a uniprocessor. What will be the latency and throughput, if the operations to be performed on all array elements on a:
   a. quad-core processor
   b. a GPU with 480 cores

   Clearly state any assumptions. Due to typically slower clock speed on GPUs and the overhead involved in transferring data between the host and the device, suppose GPU is 1.6 times slower than the CPU. [2 marks]
2. What are the use of following functions/keywords? [2 marks]
   a. __global__
   b. cudaMemcpy( )
   c. cudaMalloc( )
   d. blockDim( )
3. What is the meaning of the following kernel launch parameters? How many threads will be created? [2 marks]

   ```
   kernel<<< dim3(4, 8), dim3(16, 32)>>>(...);
   ```
4. Compare and contrast the following operations while providing suitable diagrams. [4 marks]
   a. Map
   b. Gather
   c. Scatter
   d. Reduce
5. CUDA decides how to map blocks to streaming processors. What are the pros and cons of such automated mapping? Briefly describe. [2 marks]

Step 5: Watch the instructions for Problem Set #1 at the end of Lesson 1 in Udacity.

Step 6: Modify the given source code to perform the task given in Problem Set #1. Properly comment your code. [3 marks]

Step 7: Run the modified program and make sure it is working correctly. Save a copy of the screenshot that confirms your code is working correctly (including the picture). [1 marks]

Step 8: There are several other techniques to convert a color image to grayscale. The *lightness* technique is another which averages the most prominent and least prominent colors using the following equation:

$$\frac{\max(R,G,B)+\min(R,G,B)}{2}$$

where R, G, B refers to three fundamental colors of a pixel. A grayscale image is obtained by applying this equation to each pixel separately.

Modify your CUDA program by adding another kernel function to convert a given color image to grayscale using the lightness technique. [3 marks]

Step 9: Run the modified program and make sure it is working correctly. Save a copy of the screenshot that confirms your code is working correctly (including all pictures). [1 marks]

What to Submit

- Submit following files as a single .zip file
    - Answers to Questions 1-6 in Step 4 as a PDF
    - student_func.cu with both kernel functions
    - reference_calc.cpp
    - reference_main.cpp
    - reference_hw1.cpp
- Name the .zip file as **lab2_*<index no 1>*_*<index no 2>*.zip**. Replace *<index no x>* with your index number.