School of Computer Science and Applied Mathematics
UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

COMS4040A & COMS7045A Assignment 2

Hand-out date: 10:00 am, April 28, 2020
Due date: 24:00 pm, May 14, 2020

Instructions

- This is an individual assignment, adhere to the academic integrity. Plagiarism will result in 0%.
- The due date is strictly applied.
- Hand in the electronic files and source code on Sakai. See more instructions in Section **Hand-in**.
- The total marks available for this assignment is 25.

Outcome

1. Write programs in CUDA for CPU + GPU systems;
2. Optimize the performance of a CUDA program using CUDA memory hierarchies efficiently.

Problems [25 marks]

Note The following problems are common exercises for CUDA programming. They are simple, so do not use any source program files from the Internet unless otherwise stated!

1. Implement matrix transpose using global memory and shared memory, respectively. Your program needs to work with square matrices. In your testing, use square matrices of size $2^9, 2^{10}, 2^{11}, 2^{12}$, and report the performance on these sizes. Verify and compare your results against a serial implementation, and compare the results among your various implementations. Your performance should be measured with timing, speedup, throughput, and ratio of throughputs. [8]
2. Implement two kernels which compute the sum of all the elements in a vector (a reduction operation). You should implement the kernels in two different ways:
   (a) Using shared memory.
   (b) Using global memory.

   In this implementation, you should also include a CPU reduction in order to verify the results of your GPU version. Test with different large data sizes and compare the performances. Note that, for reduction, when the array size is large, summation of the values may cause overflow, hence wrong results. To verify your implementation, start with very simple array entries, such as all 1's. [9]
3. Based on the matrix multiplication example in the lecture slide, implement the tiled matrix multiplication. You may start from the base code `matrix_ multiplication.cu` given in cuda_lab_2. Run the program using different data size, execution configuration and tile width to measure the performance. [8]

1 Hand-in

1. Your submission should be a single compressed file named as `yourStudentNo_hw2.tar.gz` that when extracts, gives me a folder named `yourStudentNo_hw2`. In this folder, you may include subfolders of source files for different questions, `Makefiles` to build your code, a `readme` file on building and running your program, as well as a write-up named `write-up.pdf`.
2. In your write-up document, include running outcome of your program, where you can simply copy and paste the results from running your code.
3. Such outcome of your program should indicate what kind of GPU devices you are using, and clearly indicate timing, speedup from implementation 1 over implementation 2, throughput, and ratios of the throughputs from implementation 1 over implementation 2 etc.
4. Attach the kernels for each question as appendixes and clearly indicate for which question the kernel is written.
5. References, if any, should be cited properly.