Parallel/Distributed Computing (CSEG414/CSE5414) Assignment #3

2020. 12. 14

Due Date: Dec. 29th (Tuesday)

1. **(120 Points: OpenMP Programming)** In this problem, you are to develop an OpenMP program to find all *palindrome* words in an input file (an example file will be given) and write the results to an output file. A palindrome is a word or phrase that reads the same in either direction or the reversed word is also in the dictionary file. For example, "noon" is palindromic, because it is a palindrome and it's reverse is also in the dictionary. A word like "draw" is palindromic because "ward" is also in the dictionary. Your parallel program should use **for-loop** parallelism. The number of threads used and the input/output files for your program should be command-line arguments. (e.g., palindrome 4 input-file output-file). Run the program on different number of processors and report the analysis of the results as well as the speedup (up to at least 4). To measure the execution time, use the *omp_get_wtime* function.

2. **(80 Points: CUDA Programming 1)** In this problem, you are to write a *block-wise matrix multiplication* program (refer to the provided attachment at slide #14-#15) (assume 4096 x 4096 matrix, where each element is randomly generated floating number):

(1) Compare the performance of CUDA implementations with different block/thread sizes and discuss your results.

(2) Modify your program so that you use shared memory instead of global memory. Discuss your results in terms of memory type.

(3) Try to further optimize your program using any techniques learned in the class and discuss your results.

3. **(100 Points: CUDA Programming 2)** Assume a *reduction* algorithm that finds the maximum of an array of 10,000 integers (randomly generated).

(a) Write a sequential version in C.

(b) Write a CUDA version that does not take path divergence into account.

(c) Update the version in (b) to take path divergence into account.

(d) Optimize the version in (c) by using different block/thread sizes, shared memory instead of global memory, and other memory optimizations techniques.

(1) Compare the performance of each version and discuss your results and findings.

(2) Also repeat the same comparison with different array sizes. Did you get the same results as (1) ?

< How to submit >

Create a **tar** file with the name of "pdc3_학번.tar" that consists of your source file(s) and a makefile, a readme file, and a hwp file containing the answers for each question (<u>Do not zip your files before making a tar file</u>). Then, upload the **tar** file to Sogang Cyber Campus.