GPU Programming
Academic Year: 2021 - 22

Dr. Praveen Kumar Alapati
praveenkumar.alapati@mechyd.ac.in
Department of Computer Science and Engineering
Ecole Centrale School of Engineering

GPU Programming Assignment Submission Guide Lines

- Mail-ID: cs481.gpu.mec@gmail.com
- Sub:TEAM_NUM
- Attach.Name and Type: (TEAM_NUM).zip
- Late Submission<=3-Days:50%.
- Write a readme file to understand your solutions.
- Submit source files only.

Programming Assignment1 (Weightage 15%)
Due Date: February 20, 2022

Develop a parallel code for the following problems using OpenMP. Report the speedup of your implementations by varying the number of threads from 1 to 16 (i.e., 1, 2, 4, 6, 8, 10, 12, 14, and 16).

1. Develop a program for the Magic-square problem when $n = 1000, 1001, 1002, 1004, \textit{and}\ 1006$.
2. Develop a program to multiply two square-matrices of order 2048 X 2048 using Block Matrix Multiplications by considering the block sizes: 4, 8, 16, 32, and 64.
3. Write a program to find a median of 100000000 elements using median-of-medians.
4. Write a program to sort 10000000 (1 crore integers) elements using Merge sort technique.
5. Write a program to generate all possible permutations for a given set of elements. Assume that cardinality of the set is 10.

Programming Assignment2 (Weightage 10%)
Due Date: March 23, 2022

Develop a parallel code for the following problems using OpenMP. Report the speedup of your implementations by varying the number of threads from 1 to 16 (i.e., 1, 2, 4, 6, 8, 10, 12, 14, and 16).

1. Generate all 24-digit binary numbers without any consecutive 1's.

2. We are given an unlimited supply of coins or notes with denominations: $\{1, 2, 5, 10, 20, 50, 100, 500, 2000\}$. Generate the distinct ways to get the desired change for 1 Million Rupees.

3. Find the shortest path between every pair of cities in INDIA and store the paths. Consider at least 1000 cities.

Programming Assignment2 (Weightage 10%)
Due Date: March 23, 2022

4. Convolution Problem: Consider that you have a 1D or 2D input data and a kernel (which is simply a small matrix of weights). This kernel "slides" over the input data, performing an element-wise multiplication with the part of the input it is currently on, and then summing up the results into a single output cell. The kernel repeats this process for every location it slides over, converting a 1D or 2D matrix of features into yet another 1D or 2D matrix of features. Solve the convolution problem by considering the following problem instances:
   a. 1D array of size 4 Million and all elements are pre-filled with the numbers: 1 to 4 Million, in ascending order. Assume that your friend supplied eight 1D-kernels (with lengths 1, 2, 4, 8, 16, 32, 64, 128) and the kernels are initialized with binary values of 1, 3, 5, 7, -7, -5, -3, -1, respectively.
   b. 2D array of size 4000 X 4000 and every row of the array is pre-filled with the numbers: 1 to 4000, in ascending order. Assume that your friend supplied 8 kernels with dimensions: 1X2, 1X4, 1X16, 1X64, 2X4, 2X16, 2X64, 2X 256. All elements of the kernels are initialized with 1.

Programming Assignment3 (Weightage 10%)
Due Date: April 24, 2022

Develop a parallel code for the following problems using CUDA C. Report the speedup of your implementations (ref. to single threaded CPU implementation) by varying the number of threads from 32 to 1024 (i.e., 32, 64, 128, 256, 512, 1024). Launch appropriate number of blocks (i.e.,depends upon the size of the problem and number of threads).

1. Find all prime numbers between $1 \ldots 10^{20}$.

2. Multiply two matrices of order 8192 X 8192 using block matrix multiplication by varying block sizes (i.e., from 16, 32, 64, 128, 256, 512, 1024).

3. Find dot product of two vectors. Assume that length of each vector is 10^{20} and the vectors are initialized randomly from the set $\{ 1.0 \ldots 2.0 \}$.

Programming Assignment3 (Weightage 10%)
Due Date: April 24, 2022

4. Convolution Problem: Consider that you have a 1D or 2D input data and a kernel (which is simply a small matrix of weights). This kernel "slides" over the input data, performing an element-wise multiplication with the part of the input it is currently on, and then summing up the results into a single output cell. The kernel repeats this process for every location it slides over, converting a 1D or 2D matrix of features into yet another 1D or 2D matrix of features. Solve the convolution problem by considering the following problem instances:
   a. 1D array of size 2^{28} and all elements are pre-filled with the numbers: 1 to 2^{28}, in ascending order. Assume that your friend supplied six 1D-kernels (with lengths 32, 64, 128, 256, 512, 1024) and the kernels are initialized with binary values of -31, -15, -7, -5, -3, -1, respectively.
   b. 2D array of size 8192 X 8192 and every array element is pre-filled with 1. Assume that your friend supplied 6 kernels with dimensions: 1X32, 1X64, 1X128, 16X16, 16X32, 32X32. All elements of the kernels are initialized with 1.

Programming Assignment4 (Weightage 15%)
Due Date: May 24, 2022

1. Write an efficient CUDA program to multiply two large positive integers that are represented using Hexa-Decimal number system.
   (Assume that number of digits in each number is greater than $(FFFF)_{16}$ and read the digits from files, say file1 and file2. Write the output to file3.)
2. Write an efficient CUDA program to compute the following:
   $X^n + X^{n-1} + X^{n-2} + ... + X^2 + X + I$,
   where X is a matrix of order 4096 X 4096 and
   I is the Unit Matrix of order 4096 X 4096 and
   n is positive integer greater than 15.
3. Write an efficient program to find the factorial of $(100000000)_4$ and write the output to a file.
4. Write an efficient program to find the prime factors of all the composite numbers between $[2^{30}$ to $2^{31} - 1]$.

Thank You ☺

Any ? Please