CS6023: GPU Programming

Assignment 1 (7 marks)

Submission deadline: Feb 9, 2020, 23:55 on Moodle

1 Problem specification

Write **three separate CUDA C++ kernels** for converting a given lower triangular matrix of integer values to the corresponding upper triangular matrix, which is the mirror image of the input matrix on the main diagonal. In the first kernel *per_row_kernel*, each thread should process a complete row of the input matrix. In the second kernel *per_element_kernel*, each thread should process exactly one element of the input matrix. In the third kernel *per_element_kernel_2D* also, each thread should process exactly one element of the input matrix. For the evaluation purpose, *per_row_kernel* will be invoked with **1D grid and 2D blocks**, *per_element_kernel* will be invoked with **3D grid and 1D blocks** and *per_element_kernel_2D* will be invoked with **2D grid and 2D blocks**.

Triangular matrices: A square matrix is called lower triangular if all the entries above the main diagonal are zero. Similarly, a square matrix is called upper triangular if all the entries below the main diagonal are zero.

Points to be noted:

- The file **kernels.h** provided by us contains the prototypes of the two kernels.
- **Do NOT change the names and the signatures of the kernels provided.**
- Sample input and sample output matrices are shown below. Pay attention to the position of each element in the input and the output matrices.
- The size, N, of the square matrices used for evaluation will be in the range: $5 \leq N \leq 2^{13}$
- The updates should be performed on the input matrix (input to the kernel) itself. Do not use any intermediate matrices.
- **Do not write any print statements inside the kernel.**
- You can use your own main.cu to test your code. We will be using main.cu written by us for evaluating your code.
- Test your code on large matrices.

$$\underset{\text{Input}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 5 & 2 & 0 & 0 \\ 8 & 6 & 3 & 0 \\ 10 & 9 & 7 & 4 \end{bmatrix}} \quad \overset{Transformation}{\Longrightarrow} \quad \underset{\text{Output}}{\begin{bmatrix} 1 & 5 & 8 & 10 \\ 0 & 2 & 6 & 9 \\ 0 & 0 & 3 & 7 \\ 0 & 0 & 0 & 4 \end{bmatrix}}$$

Figure 1: Sample input and output matrices

2 Submission guidelines

- Submit only one file that contains the implementations of both the kernels on moodle: https://courses.iitm.ac.in/mod/assign/view.php?id=39257
- The name of the file submitted should strictly be of the format ROLL_NUMBER.cu

  For example, if your roll number is CS16D019, the name of the file you submit should be CS16D019.cu
- Make sure that the ROLL_NUMBER part of the filename is in upper case.
- Do not upload anything other than the ROLL_NUMBER.cu file.
- After submission, download the file and make sure it was the one you intended to submit.

Learning suggestions:

- Write a CPU-version of code achieving the same functionality. Time the CPU code and GPU code separately for large matrices and compare the performances.