Programming Assignment 3

Due date: 11:59:59PM 4/3/2018

Sparse matrix vector multiplication (SpMV) is the core of many scientific applications. Optimizing its performance is an important topic in high-performance computing research. In this homework, you will need to implement sparse matrix vector multiplication in CUDA.

Optional: Read the following paper for the background and optimization techniques for SpMV:

http://inside.mines.edu/~bwu/CSCI_440_18SPRING/sparse_mat_vec.pdf

The code in sparse_matvec.c is a sequential version of a sparse matrix-vector multiply. The matrix is sparse in that many of its elements are zero. Rather than representing all of these zeros which wastes storage, the code uses a representation called Compressed Row Storage (CRS), which only represents the nonzeros with auxiliary data structures to keep track of their location in the full matrix.

I provide:

Sparse input matrices which were generated from the MatrixMarket (see http://math.nist.gov/MatrixMarket/). The format is a sorted coordinate representation (row, col, value) and will need to be converted to CRS. Two example matrices can be found at:

http://inside.mines.edu/~bwu/CSCI_440_18SPRING/code/sm1.txt
http://inside.mines.edu/~bwu/CSCI_440_18SPRING/code/sm2.txt

A sequential implementation of SpMV in C, which can be found at:

http://inside.mines.edu/~bwu/CSCI_440_18SPRING/code/sparse_matvec.c

You write:

1. A CUDA implementation of SpMV which optimizes for memory coalescing or load balancing. You should use the CPU implementation (sparse_matvec.c) to check whether the results produced by your GPU code are correct.

2. A one-page report (no format requirement) in PDF to describe the optimization technique(s) you apply to improve memory coalescing or load balancing.

Submit your CUDA file named FirstName_LastName_homework3.cu and the report in PDF in Canvas.

Grading criteria:

30%: compilation success

40%: output correctness

30%: optimization for improving memory coalescing or load balancing