Programming Assignment 2

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 background and optimization techniques for SpMV:
http://crd-legacy.lbl.gov/~oliker/papers/SC07_SPMV.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 for these is a sorted coordinate representation (row, col, value) and will need to be converted to CRS.

An implementation of dense matrix matrix multiply in CUDA. To compile, do:

```
nvcc -I /usr/local/cuda-samples/common/inc -o matrixMul matrixMul.cu
```

You write:

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

2. A report to describe the memory access inefficiency problem of SpMV, and the optimization

technique you apply to improve memory coalescing.

Send your CUDA file named FirstName_LastName_SpMV.cu and the report in PDF to our TA.

Grading criteria:

30%: compilation success

40%: output correctness

30%: optimization for improving memory coalescing

NOTE: Your GCC version must be the default 4.9 to successfully compile!