Jeremy Bonnell
CS4230
Programming Assignment #5 – CUDA (CRS – Sequential & CUDA Sparse Matrix/Vector Multiply)

README

I was able to parallelize the code with acceptable speedup for the size of matrix. I calculated the blocksize and gridsize by the number of rows in the matrix. For the 1024x128 matrix the result was:
blksz == 48, grdsz == 22,
for the 5x5 matrix the result was:
blksz == 3, grdsz == 2.
This was calculated by the number of rows within the matrix.
Blocksize was:
sqrt(NumberOfRows) + sqrt(NumberOfRows)/2
Gridsize was:
NumberOfRows/Blocksize
If the gridsize was not a whole number, I rounded up. This allowed me to solve for blocksize number of thread ids for rows with gridsize number of block ids. Each thread in each block looked at the CRS index for starting its row and read the vector in a 1-dimensional fashion until reaching the next row. After __syncthreads() was called, the entire problem was solved.

I compiled, ran, and tested the code on the CADE lab-1 machines. I have provided the makefile for my code and also generated matrixCPU and matrixGPU files to check the results.

TO use my compile, run, and Validate my code, I used the following commands:

- setenv LD_LIBRARY_PATH /usr/local/apps/cuda/3.2/cuda/lib64
- make
- ./sparse sm1.txt
- ./sparse sm2.txt

RESULTS:

```
[bonnell@lab1-7 proj5]$ ./sparse sm1.txt
VALID!
  Sequential Time: 1.34 msec
  Parallel Time: 0.28 msec
 Speedup = 4.76
```

```
[bonnell@lab1-7 proj5]$ ./sparse sm2.txt
VALID!
  Sequential Time: 1.67 msec
  Parallel Time: 0.69 msec
 Speedup = 2.43
```