

Homework 6- Even More CUDA

Compile and running instructions:

The code was compiled using the flags `--resource-usage`.

The code was executed on the peanut cluster (Titan) in interactive mode

The repo includes `CMakeLists.txt`.

Run `make cpu` for the CPU version and `make gpu` for the GPU version.

Run with 2 arguments for the CPU version: `<size> <sparseness value>`

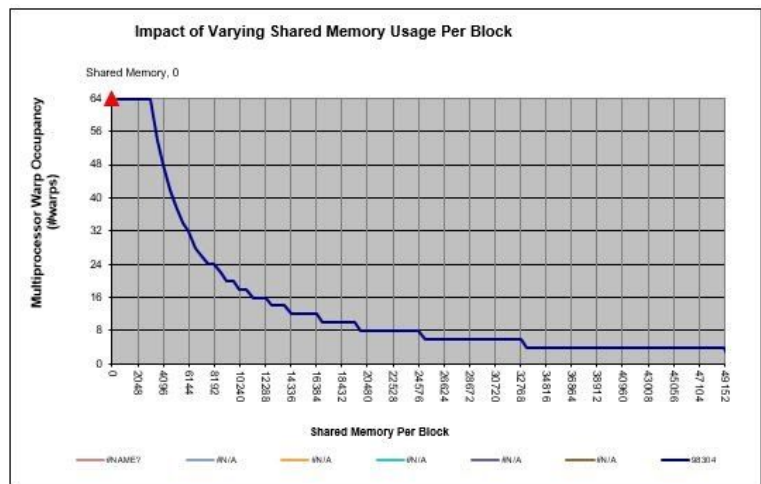
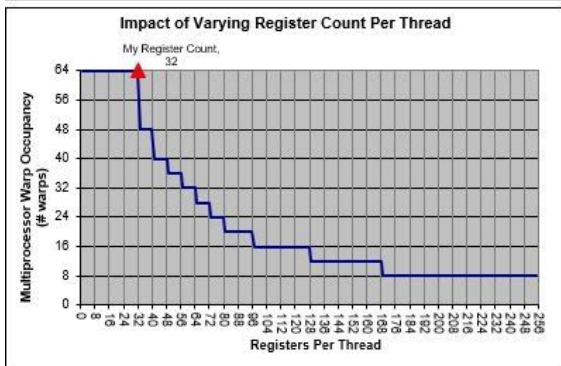
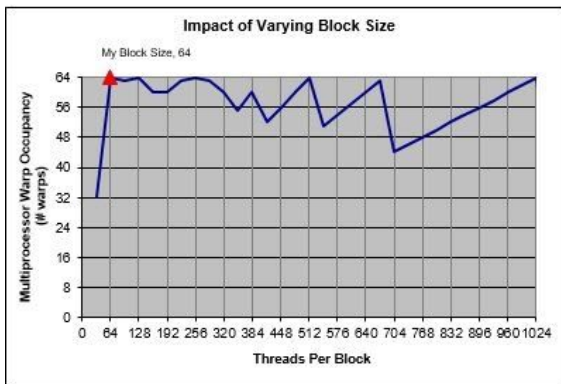
Run with 4 arguments for the GPU version: `<size> <sparseness value> <gridDim> <blockDim>`

Functionality:

The repo contains two files, `spmv_cpu.c` and `spmv_gpu.cu`. They are both programs that perform Sparse Vector Multiplication on a given input, tests the resultant values with the given file and reports the time it took to execute the function. One runs on serial CPU operations, while another runs on CUDA optimizations.

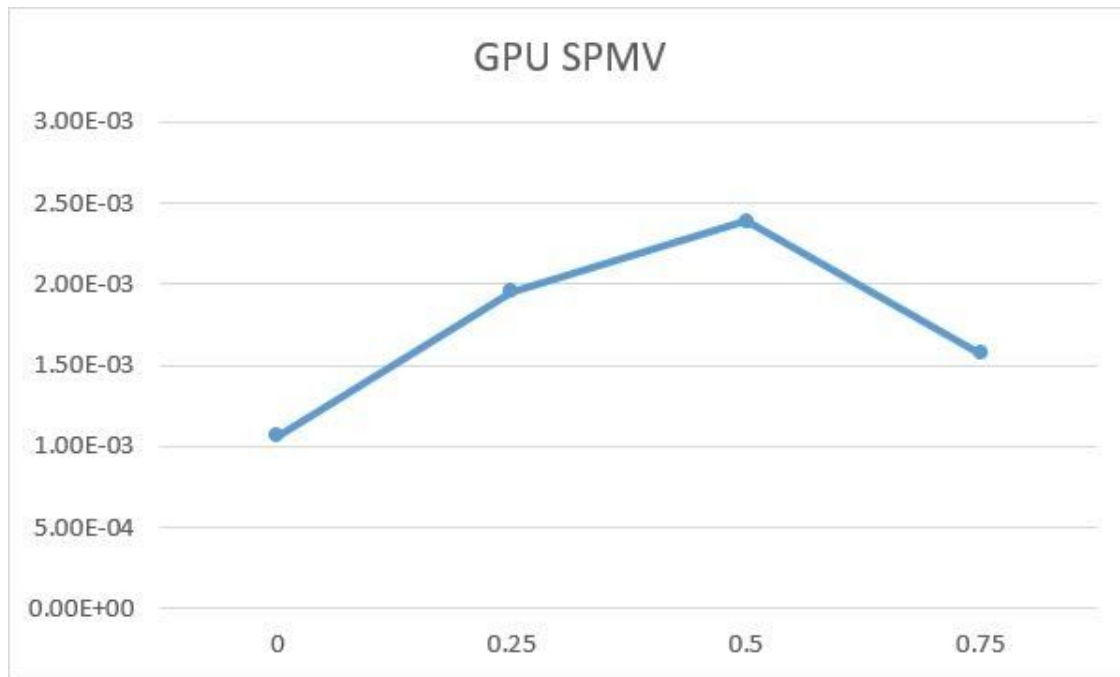
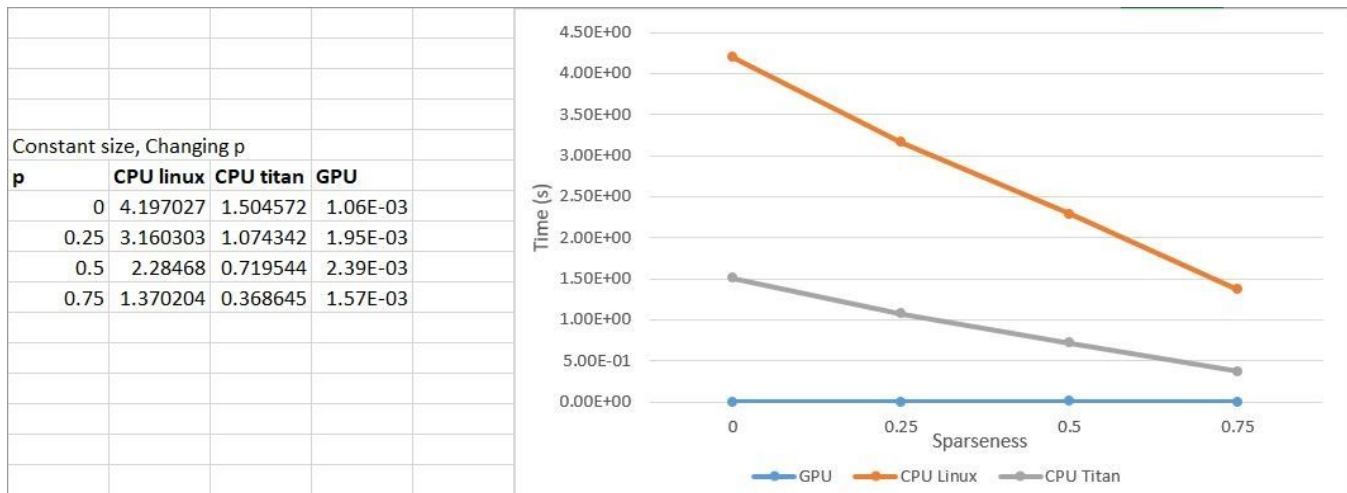
Optimum block size calculation:

Using the excel sheet from the Nvidia website, I calculated the optimum block size for my program, which used 32 registers per thread, as reported by `--resource-usage`. Max occupancy was 100% at 64 threads per block. Therefore the dimensions used were (1024, 64).



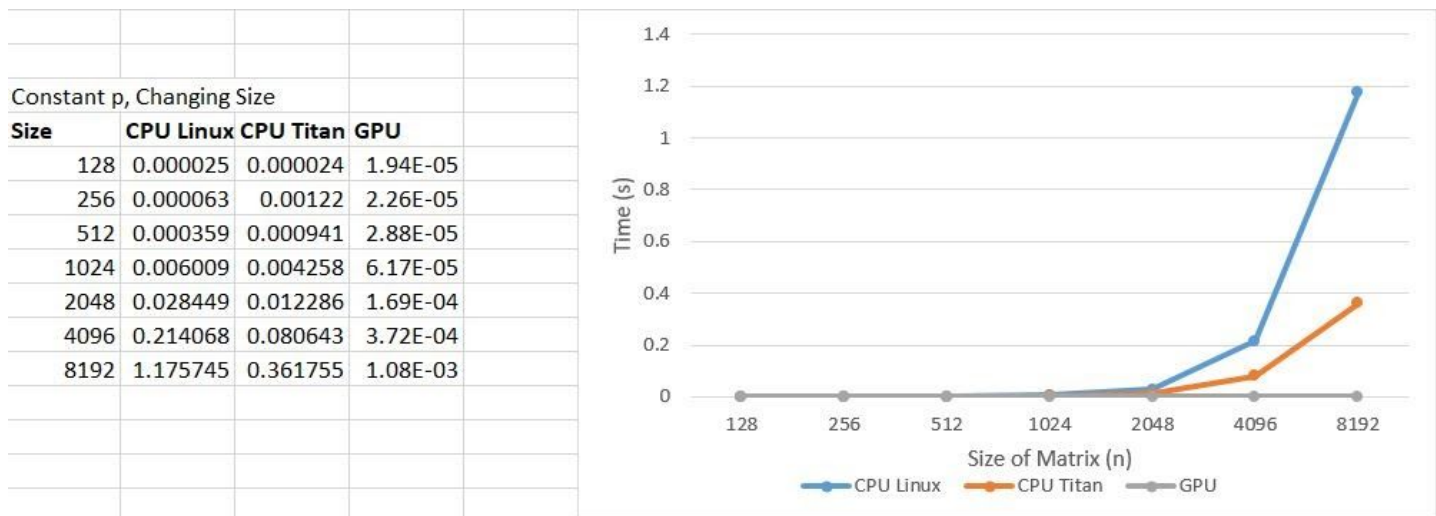
Timing tests:

The following is the graph for the programs run with a constant matrix size of 8192 x 8192, with p values ranging from 0.0 to 0.75

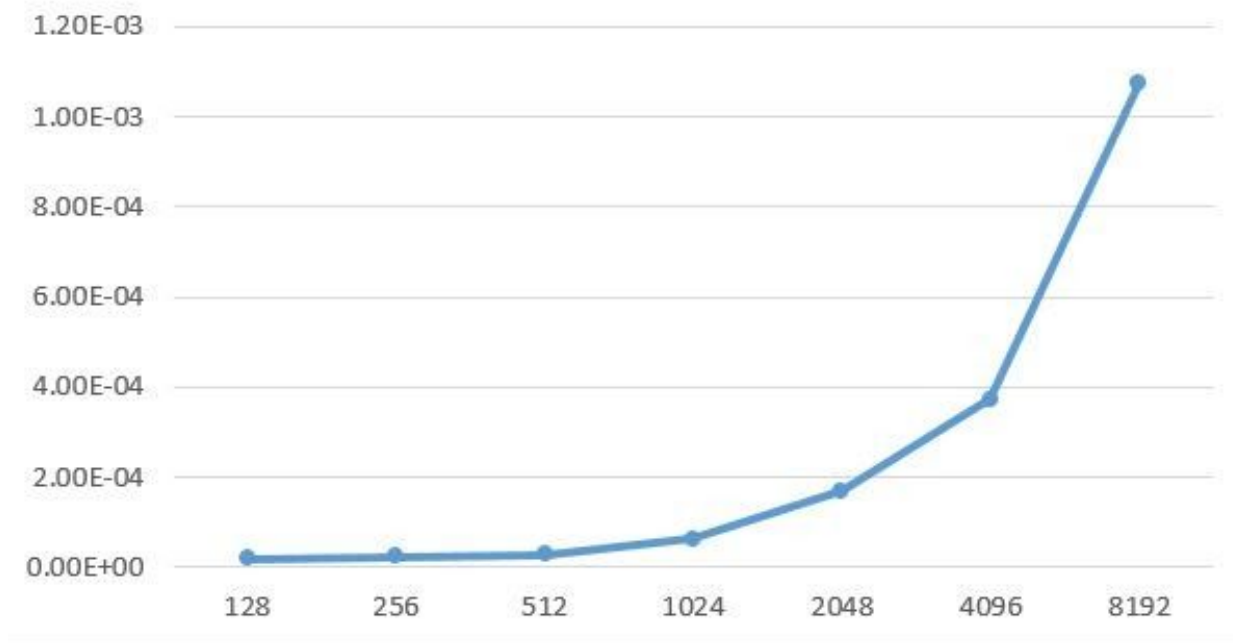


Observations: The GPU version is almost a 1000 times faster. So much so that it wasn't really visible in the first graph, so there is a second graph with just the GPU timings. The CUDA SIMD reads and coalesced reads really made a huge difference. The CPU trends are predictable, with timing increasing with increasing density of the matrix. However, the CPU code run on the Titan partition performs better than the linux cluster as opposed to the previous homework where the opposite happened.

The following graph shows the execution times of the programs with p value constant at 0.75, and sized ranging from 128 x 128 to 8192 x 8192:



GPU SPMV



Observations: The results are as expected. The GPU version is leagues ahead of the CPU version. The CPU titan version outperforms the CPU linux version by a margin as well.