CSE 746 - Assignment #1
Due 5 pm, Tuesday, February 7, 2017. Submit by email.

Accelerating matrix multiplication

Develop a matrix multiplication code that is accelerated on the GPU. A starting program with matrix multiplication implemented on the CPU has been provided on the course website (here). Use that as your starting point, with the CPU calculation used as a check on the correctness of the GPU calculation. For simplicity, consider only the case of A x B = C where matrix A is Nx32 and matrix B is 32xN, and N is a multiple of 32. (We have chosen 32 because then the matrix lends itself very well for decomposition into optimal 32 by 32 blocks)
A CPU only matrix multiplication benchmark implemented with MKL library is also provided on the course website (here and here). It can be run in serial and threaded fashion.

Part 1 - Accelerating with CUDA

Implement a GPU-accelerated matrix multiplication via CUDA. Use the CPU code provided as a starting point. Develop your GPU kernel by altering the CPU matrix multiplication function.

Part 2 - Accelerating with CUBLAS

Take your CUDA program and replace the kernel you wrote with CUBLAS library function cublasSgemm. Use the calling scheme in MKL benchmark.

Part 3 - Accelerating further with shared memory

Once both your CUDA code and CUBLAS code are working, attempt to accelerate performance of your CUDA kernel to get a result closer to the highly optimized library. You can attempt to do this by using shared memory. In your matrix multiplication kernel, define a shared block of size 32x32, copy data from global memory to it (don't forget to synchronize threads after copying), and then use it in the kernel loop doing row times column multiplication.

Part 4 - Measuring performance

The goal is to acquire a quantitative feel for the acceleration capability of the GPU so timing your code is important. For code in part 1,2 and 3, try different problem sizes to observe how speedup varies with matrix size N. Provide a plot to illustrate this. Use two different GPU cards of different Compute Capability for these tests. Compare the

GPU results with those of the CPU benchmark provided. Systems you could test on include monk.sharcnet.ca and mosaic.sharcnet.ca.