BLG562E – Parallel Computing for GPUs using CUDA
Homework #3
Due on: 01/05/2023

1. *Dense Matrix-Matrix Multiplication using GPUs*

In the first part of this homework, you are expected to implement the dense matrix-matrix multiplication kernel using CUDA using with a simple naïve approach without any optimization (do not implement tiling, do not use scratchpad memory, etc.). Do not use any matrix-multiplication libraries. Implement the host code to benchmark your dense matrix-matrix multiplication kernel.

a) Measure the running time of your kernel with a matrix size of N=1024x1024, M=1024x1024.
b) Do a performance profiling using NVVP/nvprof/NSight profiling tool and discuss the performance bottlenecks and identify opportunities for performance optimizations. If the occupancy is a problem work on the Block size to make sure you have enough occupancy per SM, then re-do the profiling.
c) Repeat the steps in (a) and (b) with matrix sizes of N=2048x2048, M=2048x2048 and n=4096x4096, M=4096

2. *Sparse Matrix-Matrix Multiplication using GPUs*

In the second part of the homework, you are expected to implement the sparse matrix-matrix multiplication kernel using CUDA with a simple naïve approach without any optimizations. Do not use any matrix-multiplication libraries. Make sure you use a condensed format for storage of sparse matrix (CSR, CSC, COO etc.). Implement the host code to benchmark your sparse matrix-matrix multiplication kernel.

For benchmarking and testing your code, you can use sparse matrixes from https://www.cise.ufl.edu/research/sparse/matrices/list_by_id.html

Choose three different matrix combinations to study performance of your sparse matrix-matrix multiplication kernel. For each matrix set:

a) Measure the running time of your kernel.
b) Do a performance profiling using NVVP/nvprof/NSight profiling tool and discuss the performance bottlenecks and identify opportunities for performance optimizations. If the occupancy is a problem work on the Block size to make sure you have enough occupancy per SM, then re-do the profiling.

In your report, provide details on the GPU model, specs, compiler etc. For both parts, provide the pseudo-code of your kernel implementation. For sparse matrix-matrix multiplication, explain with format that you use for sparse matrix storage and why, explain which matrixes that you use for benchmarking your kernel and why that you have chosen them. Give the final block-size that you use for each part.

Submit your homework as a single zip file. Include your source code and an your report (as a pdf file) in your submission.