CSCI 3160 Matrix Multiplication Project Writeup

Team 3: Brendan Dalhover, Deep Desai, Jacob Hebert, James Letterman, Russell Payne

# C:

In the C program, we utilized NVIDIA’s CUDA cores to make matrix multiplication as efficient as possible. This involved the use of an RTX 3090, and a lot of electricity.

The CUDA toolkit, which includes the nvcc to compile .cu files can be downloaded from NVIDIA’s website: <https://developer.nvidia.com/cuda-downloads>. The test desktop that used the toolkit used version 11.7, as that was what was available on the AUR. The matrix multiplication file is from the Matmult sample in NVIDIA’s cud-samples githup repo: <https://github.com/NVIDIA/cuda-samples/tree/master>

Initial tests of CUDA showed that 128, 256, and 512 sized matrices would not be enough to even show up on a results page, so we went with 1024, 2048, and 4096 sized matrices. The complexity of each operation is O(N^3), so each increase in size results in a cubic increase in the number of operations performed. In the default code, an 8196 sized matrix takes half a second to complete, but about 2 minutes to output to the console.

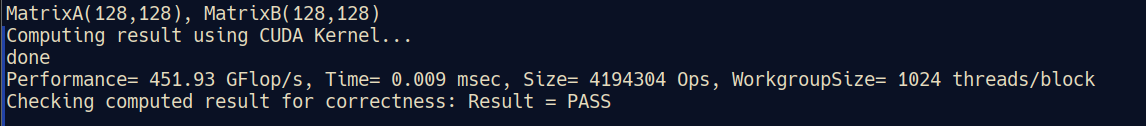


Figure 1- 0.009 ms for 128 size square matrix

We optimized the code:

# Python:

# Java: