Assignment 5 – CS398S21-A.SG

Ang Hiap Lee, a.hiaplee, 390000318

**1.** Testing environment

The experiment was conduct with the machine of the following specifications.

|  |  |
| --- | --- |
| CPU | Intel Xeon E6-1620v4 |
| GPU | Nvidia GeForce GTX 1060 6 GB |
| RAM | 16GB |

# 2. Performance Analysis: CPU vs GPU Single Stream vs GPU Multi-stream

The graph below shows the result of running matrix multiplication of various single precision floating point matrix using sequential CPU, GPU single stream and GPU Multi-stream. (Block size = 32 & Tile size = 256)

As we can observe, both GPU versions offered a performance improvement over the CPU version and scales relatively well as the matrix size grows. However, we can also see that the multi-stream version was slower than the single stream version.

# 3. Performance Analysis: Various block sizes and tile sizes

To ensure that we were not bottlenecked by the given block size and tile size (32 & 256). The experiment was then conducted again with various combinations of block size and tile sizes.

First, we monitor how the timing scales if we increase only the block size:

(Recording timings with various block sizes, presented in logarithmic scale for better view)

We observed that if the block size grows too huge, we start observing a drop in timings this effect is more apparent in the smaller 1000 x 1000 matrix.

Next, we repeat the experiment again with constant block sizes but increasing tile sizes:

(Recording timings with various tile sizes, presented in logarithmic scale for better view)

We observed a small improvement for each step up of the tile size while keeping the block size constant without performance lost. With that being said, the performance increase with using various tile sizes while using constant block size were not as good as the just increasing the block sizes alone.

Finally, with all the obtained information, we benchmarked the application with increasing block and tile sizes and keeping it capped at 1024, since performance gains began to drop when sizes go beyond 1024.

(Recording timings with various block & tile sizes, presented in logarithmic scale for better view)

With a combination of increasing block & tile sizes capped at 1024, slight performance drops were observed when the tile size reaches 1024 but we managed achieved the best timings using multi-stream out of the experiments so far. The timings however were still slower than the single stream version, this is probably caused by the overhead incurred by each stream’s allocations and operations needing to wait for pinned memory access between threads.