Nathan Dunn

Professor Liu

CS-4370-90 Par. Prog. Many-Core GPUs

10-24-19

Project 2: Tiled Matrix Multiplication

**Introduction**

For this project, I was required to implement tiled matrix multiplication on both the CPU and GPU device. I was also required to measure the CPU computation time, along with the GPU computation and memory transfer time.

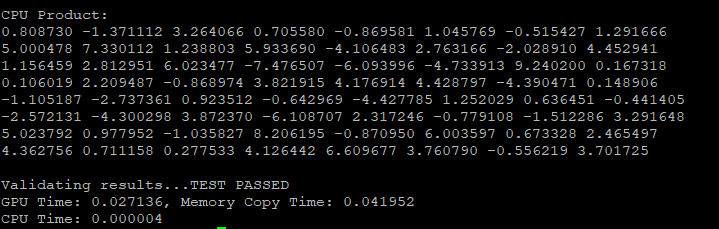
**Progress**

I was able to complete every part of the report. For measuring the computation time of the CPU, I excluded the matrix initialization. I strictly measured the host multiplication function. For measuring the computation time of the GPU, I measured both the memory copy to and from device and combined the times into one memory transfer time. I then strictly measured the kernel function call to obtain the GPU computation time.

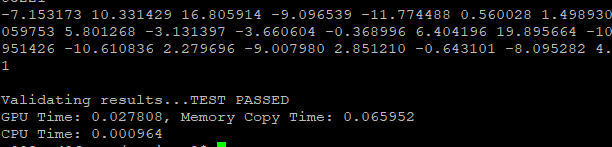
**Screenshots**

The output displays both matrix A and B’s contents, The CPU and GPU product, and the computation times. The following are screenshots of the program computing matrix multiplication at different matrix sizes:

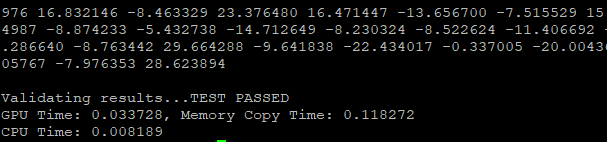
*8x8 matrix with 4x4 tile size*.

****

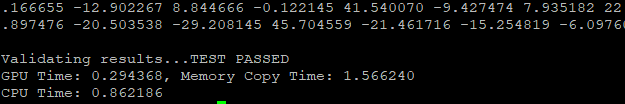
*64x64 matrix with 16x16 tile size*.



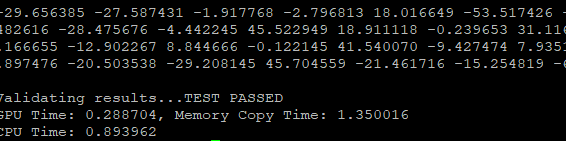
*128x128 matrix with 16x16 tile size*.



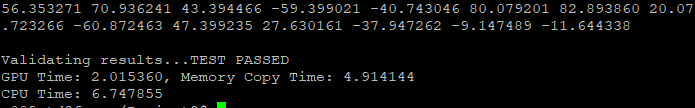
*512x512 matrix with 8x8 tile size.*



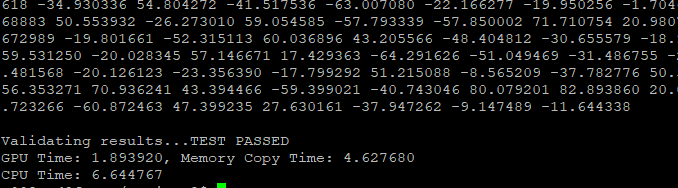
*512x512 matrix with 16x16 tile size*.



*1024x1024 matrix with 8x8 tile size*

**

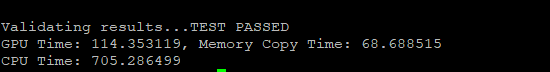
*1024x1024 matrix with 16x16 tile size*



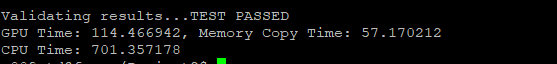
*4096x4096 matrix with 8x8 tile size*



*4096x4096 matrix with 16x16 tile size*



*4096x4096 matrix with 32x32 tile size*

******

**Computation Time Table**

*Note: Each measurement is in* ***seconds***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Time | 512\*512 | | 1024\*1024 | | 4096\*4096 | | |
| CPU computation time | *(Averaged from two samples)*  **0.878074** | | *(Averaged from two samples)*  **6.696311** | | *(Averaged from three samples)*  **714.77063** | | |
| GPU computation time | Tile width: 8  **0.294368**  **2.98 Speedup** | Tile width: 16  **0.288704**  **3.04 Speedup** | Tile width: 8  **2.015360**  **3.32 Speedup** | Tile width: 16  **1.893920**  **3.54 Speedup** | Tile width: 8  **129.077347**  **5.54 Speedup** | Tile width: 16  **114.353119**  **6.25 Speedup** | Tile width: 32  **114.466942**  **6.24 Speedup** |
| GPU memory transfer time | *(Averaged from two samples)*  **1.458128** | | *(Averaged from two samples)*  **4.770912** | | *(Averaged from three samples)*  **61.70108** | | |

The larger the matrix, the larger the speedup of the GPU compared to the CPU. For example, when the matrix size is less than 512 (observe the screenshots above), the CPU tends to compute the matrix computations faster. But as the matrix size grows to 512, the GPU and its parallel processing power is able to bridge the gap on the CPU to process the matrix computations faster.