ELEC 374

Machine Problem 4

Luka Gobovic

20215231

# Results

Text

Description automatically generated

The first test with 125x125 fails because the matrix size is not divisible by the tile width of 2, and this, without proper checks in place, causes the matrix multiplication to fail. I will however attempt to remedy this in the bonus section.

For the purposes of speeding up the testing, I removed the host multiplication portion for the remaining tests since it would take on average around 2-3 minutes. The tests should pass except for when the size is not divisible evenly by the tile width, so 125 and 10, 125 and 20, and 250 and 20. That is why the above screenshot will look different than the rest.

**SIDE NOTE, ALL OF THE RESULTS SHOULD SAY TILED MATRIX MUL NOT NORMAL**

**This was a mistake on my part.**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

# Code

Tile width parameter at the top was changed every run.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated Text

Description automatically generated

# BONUS CODE

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated Text

Description automatically generated

# Analysis

The 2000x2000 result for the regular matrix multiplication was removed from the chart above to allow the rest of the results to scale better. The result for that test was 23,907.50 ms

It can be seen that when comparing tiled multiplication to single thread single block multiplication, it is significantly faster, and tends to decrease as tile width is increased, especially for the larger matrices. However, when comparing the results to the second portion of MP3, where we experimented with different block widths, the results tend to be much more closely matched. For example, when using a block width of 20, the results for the 2000x2000 matrix was 497.89 ms, marginally faster than the tiled multiplication using a tile width of 25.

## Questions

1. When using the Tesla C2075, which has 14 streaming multiprocessors, the maximum number of threads that can be simultaneously executing on the device is 14 X 2048 = 28,672 threads.

Maximum number of threads per block: 1024

Maximum number of blocks per SM: 8 (for compute capability 2.x)

Total number of blocks: 14 \* 8 = 112

Total number of threads: 112 \* 1024 = 114688

114688 threads can be simultaneously scheduled on the GPU.

1. All of these values are using a tile width of 25: This was found using the nvcc-- ptxas - options = -v options when compiling.  
     
   Number of registers used by matrixMultiplication kernel: 11

Number of registers used by tiled\_matrix\_multiply kernel: 17

Size of shared memory used by tiled\_matrix\_multiply kernel: 5000 bytes

From above, the number of blocks per streaming multiprocessor is 8, and the maximum total threads simultaneously scheduled is 114,688, with 28,672 executing.

If however, I use a tile width of 2, the size of shared memory decreases to 32 bytes. From this, it can be seen that the size of both of the shared matrices in the kernel is equal to (tile\_width)^2 \* 2(size of float). So in this case, one of the share matrices is 4 \* 4 bytes = 16 bytes, equating to 32 total bytes for both.

# BONUS

Tiled Multiplication for 1900x1600 and 1600x1300, tile width of 8 by 15

Text

Description automatically generated

Tiled Multiplication for 350x400 and 400x500, tile width of 8 by 15

Text

Description automatically generated

Analysis  
The tests above passed because I ran the host multiplication on non-square matrices as well, simply by changing around some of the parameters being passed in along with the for loops.

For a 400x400 matrix, using a tile size of 8, it took roughly 6.5ms for the computation. Seeing as the output matrix of the boundary check matrix has 15,000 more elements, that could explain the discrepancy. However, it is difficult to exactly compare the results of the two different methods. While it may take longer to execute the calculations due to additional calculations and checks that are necessary to avoid out of bounds issues, the revised kernel allows us to perform operations on non-square matrices and where the dimensions are not a multiple of the dimensions of the tiles.