# Second assignment

## *Advanced Computer Architecture*

*CUDA Programming*

| | |
|---|---|
| Diogo Ferreira | *76504* |
| Luís Leira | *76514* |

***T3G02***

# Optimization of the thread launch grid

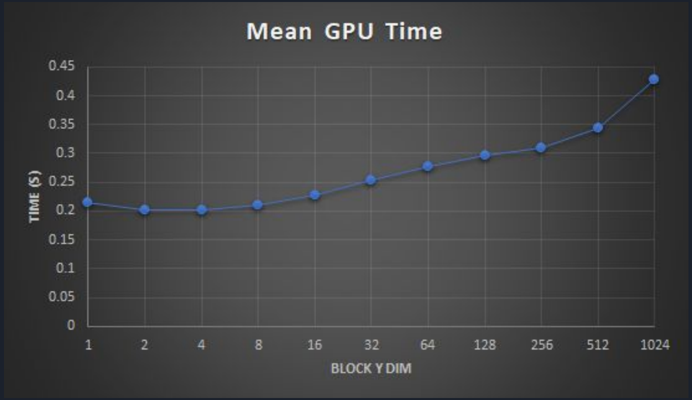
- Subsequent elements are read in row order.
- Less GPU time with the launch grid <<(32, 2048), (1, 32)>> : 0.1072 seconds
- More than 3 times faster than with the CPU (3.17 times faster)
- Given the data transfer time between the host and the device, the GPU operation overall takes longer than the CPU (1.36 times faster overall with the CPU, counting with the transference times)

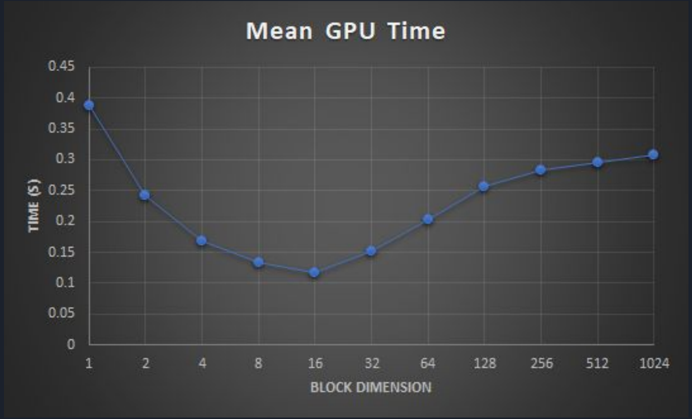
| Grid X | Grid Y | Block X | Block Y | GPU Time (s) | CPU Time (s) |
|---|---|---|---|---|---|
| **32** | **2048** | **1** | **32** | **0.1072** | **0.3399** |
| 16 | 4096 | 2 | 16 | 0.1091 | 0.3522 |
| 16 | 4096 | 1 | 32 | 0.1114 | 0.3403 |
| 4 | 32768 | 8 | 2 | 0.1117 | 0.3402 |
| 8 | 16384 | 4 | 4 | 0.1117 | 0.3523 |

# Optimization of the thread launch grid

- If the block Y dimensions is higher than sixteen, the performance starts to deteriorate. It has to do with the fact that the GPU GeForce 480 has 15 Streaming Multiprocessors. On this case, due to cache misses, the performance starts to deteriorate when the block Y dimension is higher than 4.

- The GPU mean time is lower when the block dimension (x*y) is 16, which means that the grid dimension (x*y) is 131072. The performance lowers because of cache misses in the blocks.





# Optimization of the memory accesses

- Optimization was made by changing the reading order for each thread. Subsequent elements are read in column order instead of row order. With that, the cache misses are reduced, because each thread simultaneously accesses the same lines of memory, but in different columns.

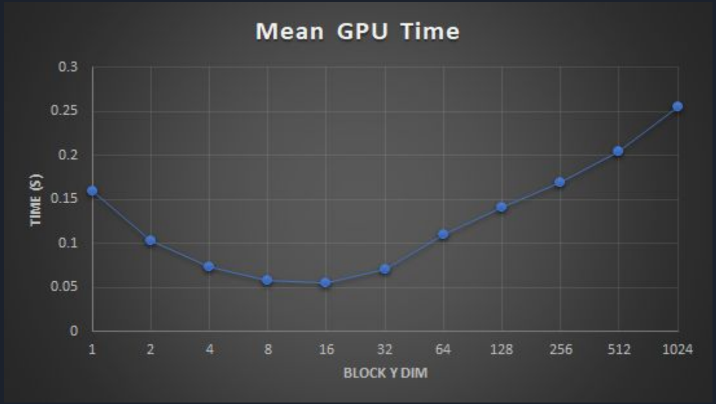
- The best time with the memory optimization is more than 7 times faster than the best time without the optimization (7.30 times faster).

- With the GPU transfer times into account, the overall GPU time (optimized) is still higher than the best CPU time (without optimization).

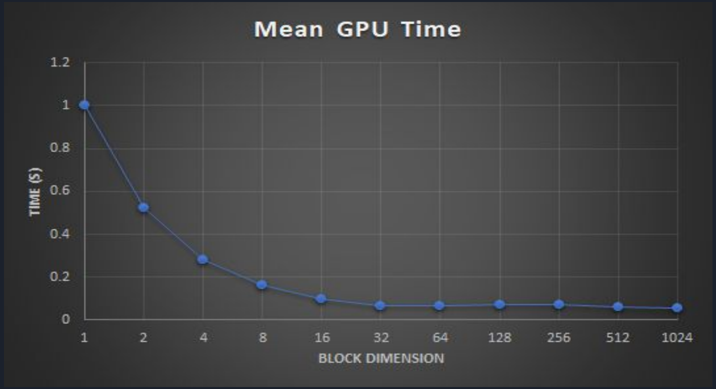
| Grid X | Grid Y | Block X | Block Y | GPU Time (s) | CPU Time (s) |
|---|---|---|---|---|---|
| **32** | **128** | **128** | **4** | **0.01468** | **24.55** |
| 64 | 128 | 64 | 4 | 0.01474 | 24.56 |
| 64 | 128 | 128 | 2 | 0.01478 | 24.47 |
| 64 | 256 | 64 | 2 | 0.01478 | 24.52 |
| 16 | 256 | 256 | 2 | 0.01488 | 24.53 |

# Optimization of the memory accesses

- On this case, because there are less cache misses, the performance starts to deteriorate only when the block Y dim is higher than 16, instead of 4.



- The GPU mean time is lower when the block dimension is higher. This happens because the cache misses are lower than in the previous case, so the most blocks in parallel the faster.


Mean GPU Time
TIME (S)
BLOCK DIMENSION