**Applied Parallel Programming With GPU** **Assignment 1**

**Matrix x Matrix Product ( O = M x N)**

**Semih Kekül**

**1347657**

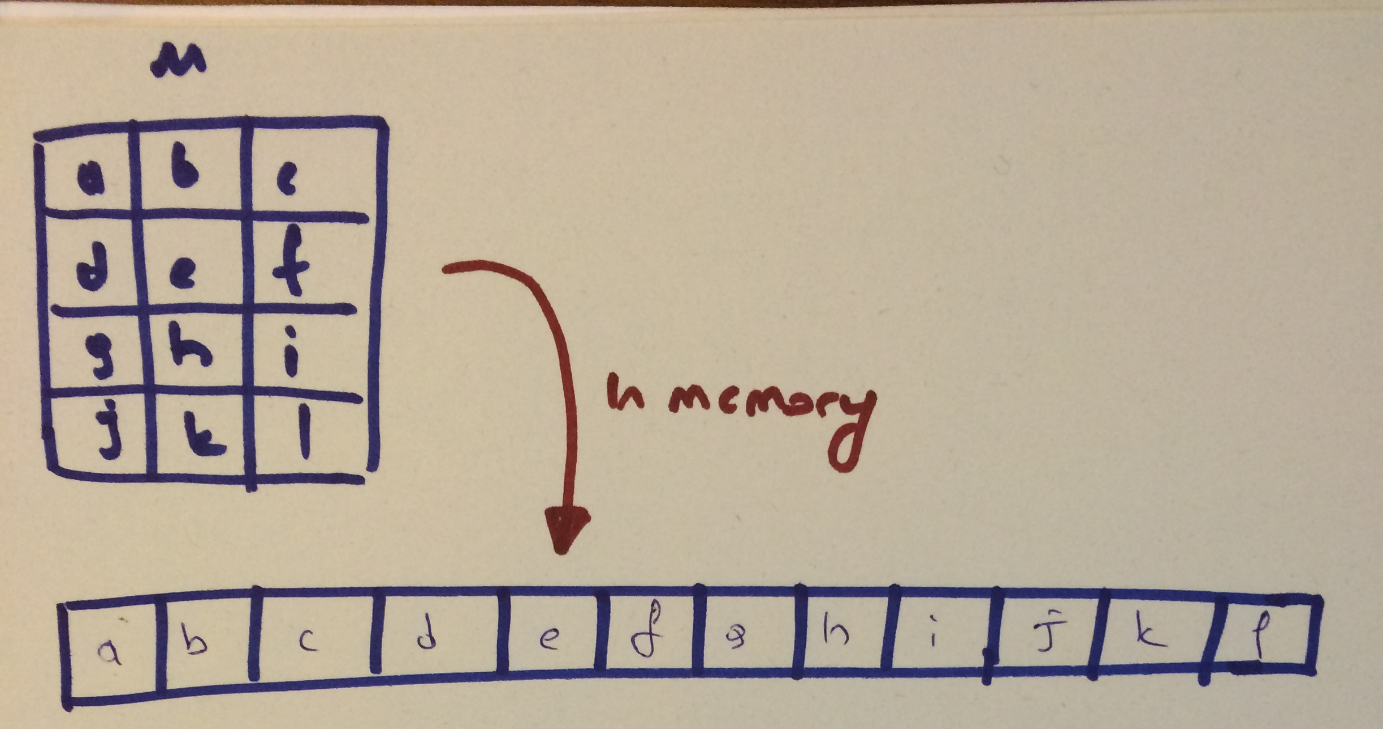
**Matrix - Vector Product**

1. **Problem definition:**

Implementation of prodcution of two matrices with CUDA. Parallelization would fasten up the element by element production, however, it would not effect the summation speed.

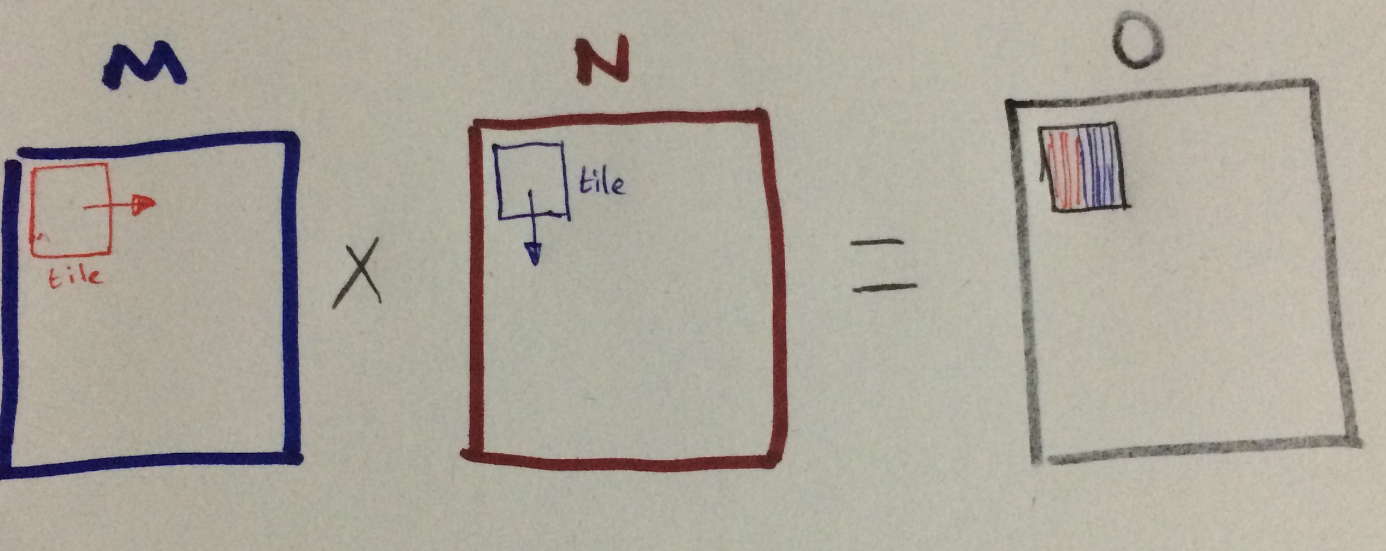
1. **Algorithm:**

The matrix data are 1D arrays :



Grid is 2D and block is also 2D. The dimension of the grid is determined by block dimension and matrix dimension (matrix size / block size).

The algorithm uses sliding tiles for allowing coalased data access. Tiles slides on both M and N:



The tiles are shared memories:

\_\_shared\_\_ int tM\_s[BLOCK\_SIZE][BLOCK\_SIZE]; // tile from M

\_\_shared\_\_ int tN\_s[BLOCK\_SIZE][BLOCK\_SIZE]; // tile from N

And the tiles are filled by each thread block, after filling its own part each thread waits for tiles are to be filled by

\_\_syncthreads(); // wait all threads of the block fill its part of the tile

Afterwards, a regular inner product is done

for (int j = 0; j < BLOCK\_SIZE; ++j)

{

sum += tM\_s[tiY][j] \* tN\_s[j][tiX]; // regular inner product

}

1. **Benchmarking:**

**Memory**

Memory usage is the global memory for the an input matrices and output matrix and two shared memory double arrays.

**Time**

|  |  |  |  |
| --- | --- | --- | --- |
| Matrix sizes | Block size | Time Without I/O | Time With I/O |
| {128 x 128} x {128 x 128} | 16x 16 | ~0.28 ms | ~ 0.5 ms(higher as expected) |
| {1024 x 1024 } x {1024 x 1024 } | 16x 16 | ~115 ms | ~ 121 ms(higher as expected, however not much higher shows the algorithm is GPU bound) |
| {2048 x 1024 } x {1024 x 2048 } (not square) | 16x 16 | ~220 ms | ~ 235 ms(higher as expected) |

1. **Pros and** **Cons**

The first idea come to my mind is using CPU implementation; however it was not for a parallel device and would be very slow on a GPU. So I used tiles for parallelism and shared memory for coalased access. However, there may be bank conflicts .

1. **Environment**:

|  |  |
| --- | --- |
| GPU processor | GeForce GTX 760M |
| CUDA Cores: | 768 |
| Core clock: | 627 MHz |
| Memory data rate: | 4008 MHz |
| Total available graphics memory: | 4096 MB |
| Memory interface: | 128-bit |
| Memory bandwidth: | 64.13 GB/s |
| Processor: | Intel Core i7 2.20 GHz |
| Memory | 32 GB |
| OS | Windows 10 64-bit |