Cuda Transpose

Lorenzo Fresco

June 2019

1 Introduction

The last exercise asked us to implement two different versions of a function that transpose a matrix inside the GPU. The difference resides in the fact that in the optimized version we make use of the shared memory logic in order to fasten the execution. In the graph below we can see the results obtained with a Matrix of fixed size of 8192.

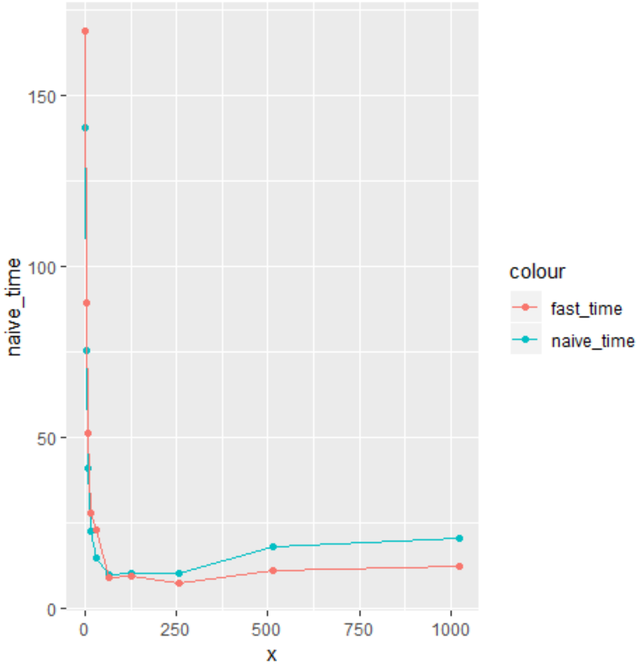

Figure 1: Number of Threads vs Execution Time

We can see that in order to obtain sensible results we must use a minimum of 128 threads. And that after that break.point we see that the optimized function time required as an average of 10ms which is almost one third of the one used by the naive one(which is around 30 ms).