Report 4

Tommaso Rodani

August 2020

1 Introduction

This exercise aim to be an example of CUDA programming by implementing matrix transposition on GPU. Moreover, a comparison of running time between two different implementation, one naive and another optimized, is shown.

2 Assignment 6

This algorithm transpose a matrix of 8192x8192 using block of threads where each block transposes a 32x32 tile. Different number of threads have been tested to seek maximum bandwidth. Also the difference between a naive and an efficient implementations was tested, where the efficient one avoids global memory inefficiencies by making more use of the GPU shared memory. In fact transpose_gpu performs the just swap the indices of the input and output vectors that can lead to strides in the global memory. The transpose_gpu_opt function avoid this problem using shared memory to optimise memory access.

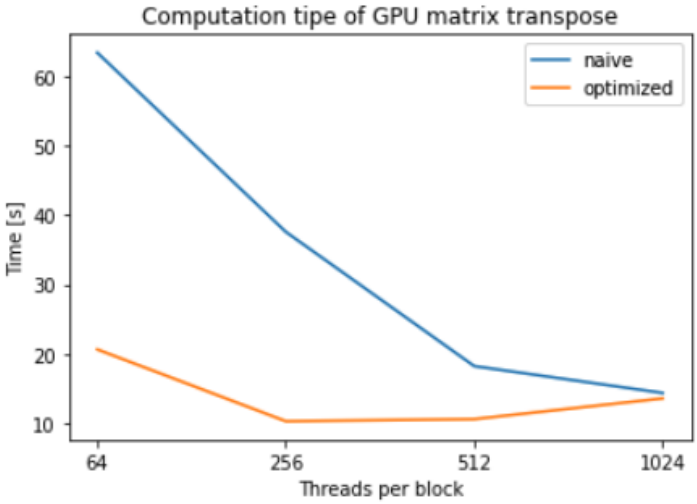

Figure 1: Computational time: naive and optimized

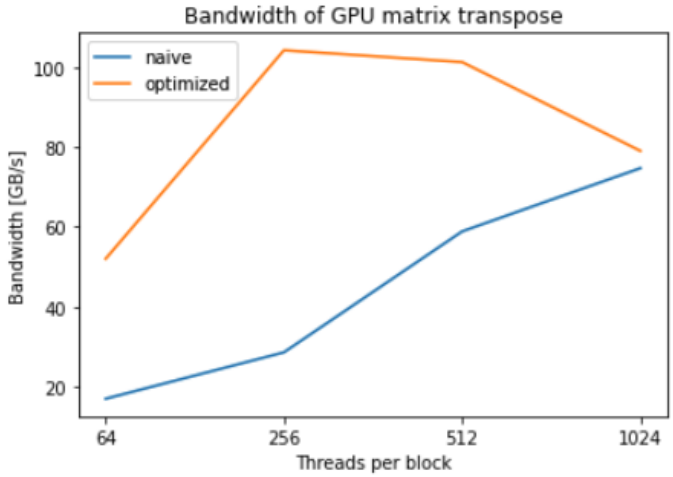

Figure 2: GPU memory bandwidth: naive and optimized