



# CLE - Assignment 3

Universidade de Aveiro - DETI

31/May/2024

Pompeu Costa, 103294  
Guilherme Craveiro, 103574

## Idea

Use the GPU threads to sort a sequence of integers using bitonic sort. Divide the sequence into  $K$  subsequences and have each thread sort their subsequence. On the next iteration, halve the number of threads and double the size of their subsequence. Continue this process until only one thread remains and have that thread sort its subsequence.

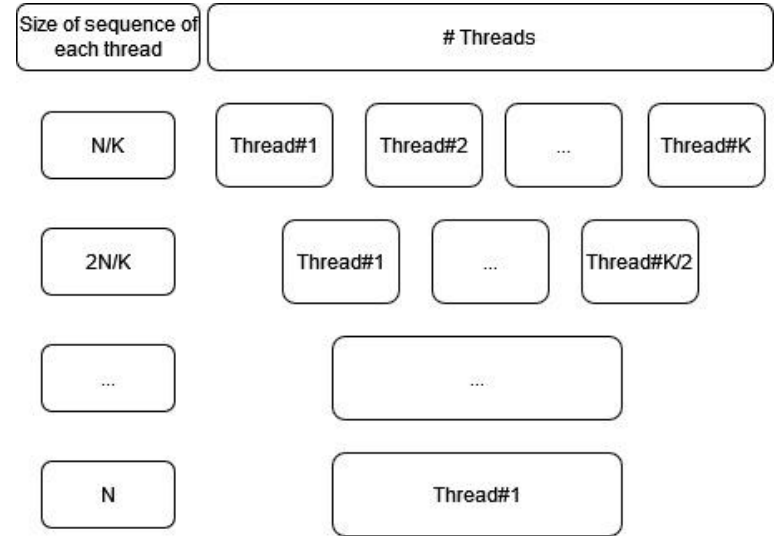


Figure 1: GPU threads and their sequence at each iteration



## Memory Access

Two different approaches: row sort and column sort.

Row sort benefits from coalesced memory access which is good for caching, leading to better performance.

Column sort suffers from non-coalesced memory access which may result in multiple cache misses, resulting in poor performance.

# Timings

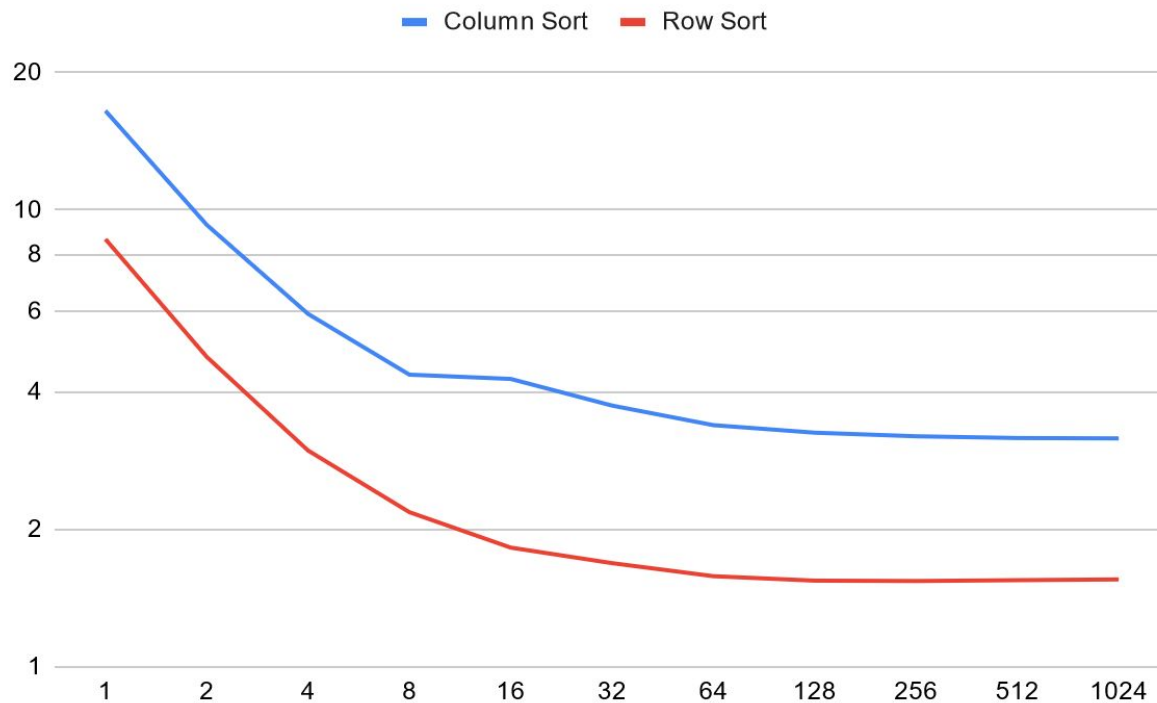


Diagram 1: Timings of both sorting approaches



# Is it worthwhile to use the GPU to solve this kind of problem?

## Key Considerations

1. Parallel Processing Power
  - a. GPU is designed for massive parallelism, while CPU has fewer cores, but each core is more powerful
2. Problem Size
  - a. GPUs excel in handling large arrays due to parallel processing, however in smaller data sets the overhead of data transfer to/from the GPU might outweigh the benefits

## Conclusion

Use the GPU for large data sets due to its performance benefits from parallel processing; for small data sets, CPU multithreading/multiprocessing has less overhead and is often sufficient.