GPU Computing

Homework 1: Matrix Transposition

Murtas Cristian
248025
cristian.murtas@studenti.unitn.it
GitHub Repository

April 2, 2024

1 Problem Description

The goal of this homework is to implement a matrix transposition algorithm, where the transpose of a matrix is an operator which flips the matrix over its diagonal. That is, if A is a matrix of size $m \times n$, then the transpose of A is a matrix of size $n \times m$ where the element at row i and column j is the element, also denoted as A^{T}.
Additionally, we are asked to measure the effective bandwidth of our implementation, considering also the usage of different optimization flags such as: `-O0`, `-O1`, `-O2`, `-O3`.
Furthermore an analysis of the cache behavior of the algorithm is required, for this purpose we are going to use valgrind.

1.1 Algorithm

In order to dimostrare qualcosa, two algorithms have been implemented: the first one is a naïve approach, which consists in iterating over the matrix and swapping the elements, while the second one is a more optimized version which takes advantage of a block mechanism to reduce the number of cache misses.

Algorithm 1 Naïve Matrix Transposition

```
src ← create_matrix(size)
for i = 0 to size do
    for j = 0 to size do
        dest[j * size + i] = src[i * size + j]
    end for
end for
```

2 Bo da decidere

2.1 Hardware

1. **Desktop PC**
   - **CPU**: AMD Ryzen 5 5600X
     - **Cores**: 6
     - **Threads**: 12
     - **Base Clock**: 3.7 GHz
     - **L1 Cache**: 64 KB (per core)
     - **L2 Cache**: 512 KB (per core)
     - **L3 Cache**: 32 MB
   - **RAM**: 16 GB DDR4
   - **OS**: Ubuntu 22.04 (on WSL2)

2. **MacBook Air M1**

- **CPU**: Apple M1
  - **Cores**: 8 (4 Firestorm + 4 Icestorm)
  - **Threads**: 8
  - **Base Clock**: 3.2 GHz
  - **L1 Cache Firestorm**: 192 + 128 KB (instructions + data, per core)
  - **L2 Cache Firestorm**: 12 MB (shared)
  - **L1 Cache Icestorm**: 128 + 64 KB (instructions + data, per core)
  - **L2 Cache Icestorm**: 4 MB (shared)
- **RAM**: 8 GB LPDDR4X
- **OS**: macOS Ventura 13.2.1

2.2 Results

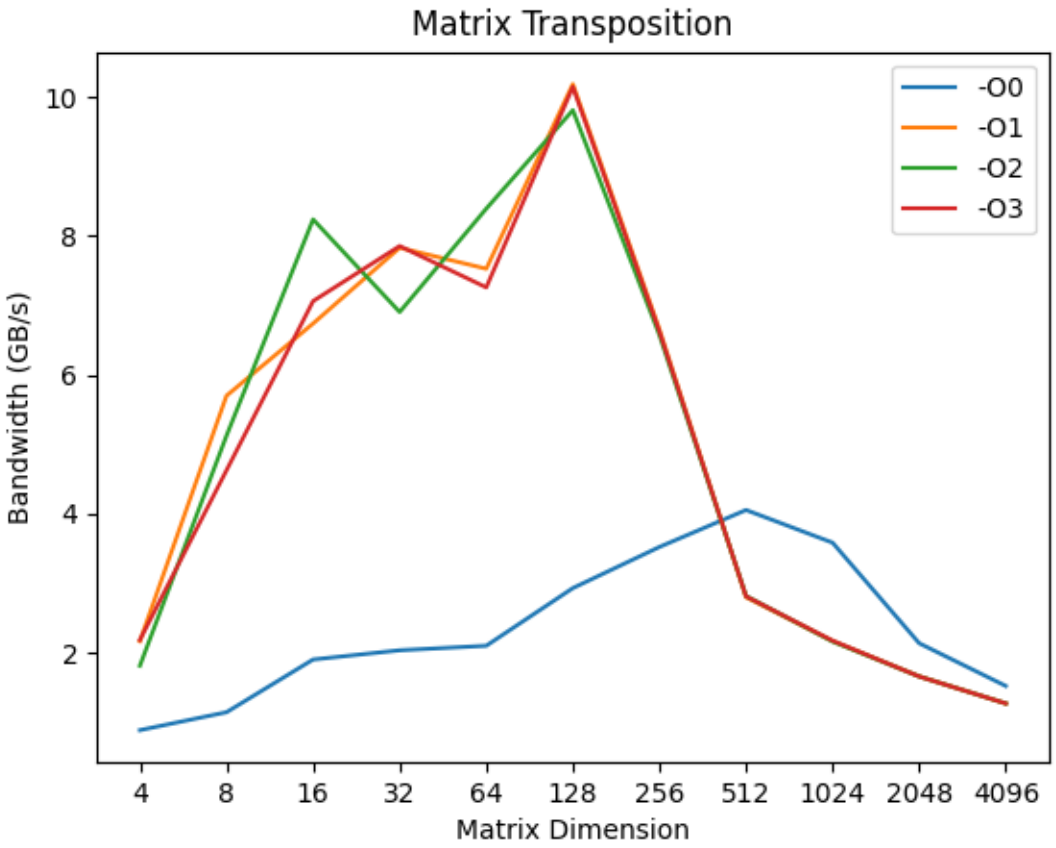

Figure 1: