GPU Computing

First Assignment

Report

Christian Dalvit

April 22, 2024

1 Introduction

The goal of this homework is to implement an algorithm that transposes a non-symmetric matrix. Furthermore, different metrics of the algorithm should be measured and analyzed. In this report I describe the problem setting, algorithms and experimental results of my implementation.
The code used for this homework is made available through a public Github repository. Details on how to run the code and reproduce the results can be found in the `README.md` file of the Github repository.

2 Problem Description

For a given matrix $A \in \mathbb{R}^{n \times m}$, the transpose of the matrix $A^T \in \mathbb{R}^{m \times n}$ is defined as

$$A^T_{ij} = A_{ji}$$

In this homework, matrices have dimensions of 2^N for $N \in \mathbb{N}$, so only square matrices are considered. As a result, the implemented algorithms don't need to accommodate changes in the output matrix's shape. For the purpose of this homework we assume row-major memory layout of the matrix.

While implementing an algorithm that computes the transpose of a matrix is straightforward, comming up with an efficient implementation is quite tricky. In general, leveraging spatial and temporal locality can improve efficiency. Because each element of the matrix is accessed only once, temporal locality cannot be exploited for computing the transposed matrix [1], so spatial locality becomes the only source for improvement. The issue with leveraging spatial locality in matrix transposition is that data is accessed along rows but written along columns, potentially leading to poor cache performance. Algorithms that respect spatial locality in their memory access pattern can benefit from quicker access to cached data.

2.1 Algorithms

All human things are subject to decay. And when fate summons, Monarchs must obey.

3 Experiments

3.1 Setup

3.2 Results

All human things are subject to decay. And when fate summons, Monarchs must obey.

References

[1] Siddhartha Chatterjee and Sandeep Sen. "Cache-efficient matrix transposition". In: *Proceedings Sixth International Symposium on High-Performance Computer Architecture. HPCA-6 (Cat. No. PR00550)*. IEEE. 2000, pp. 195–205.