School of Computer Science and Applied Mathematics
UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

COMS4040A & COMS7045A Assignment 2 – Report

Abdulkadir Dere - 752817 - Computer Science Hons

14 May 2020

1 Introduction

This report will focus on parallel implementation of different computation problems. Parallel computing is the method of executing multiple processes simultaneously. CUDA, which stands for Compute Unified Device Architecture, will be used to create the parallel processes. CUDA is a parallel computing platform developed by Nvidia to enable developers to utilise the processing power of Nvidia GPUs. CUDA is a model extension to the C programming language.

We will implement matrix transpose, vector reduction and matrix multiplication using CUDA. The sequential methods have also been created for evaluation of parallelised versions of the problems.

2 Problems

2.1 Matrix Transpose

The matrix transpose is the process of switching the rows and columns of a matrix. We have only considered square matrices for simplicity of the implementations. We have implemented sequential method, parallel method using the global memory and parallel method using the shared memory methods. Square matrices of size 64, 256, 512, 1024, 2048 and 4096 has been used for testing the performances of different implementation methods. Shared memory version makes use of TILE_WIDTH to compute the transpose. The method is to create tiles within the matrix and transpose each tile using a different thread to speed up the transpose process. Each thread block will be responsible of a single tile.

2.2 Vector Reduction

The vector reduction is the process of summing each element of a vector. The reduction process is associative so order of summing elements does not matter. The vector is initialised with random integers between the values of 0 and 9. The sequential reduction process is defined as adding each element until you reach the end of the vector so we will compute $(N-1)$ computations to sum all the elements. The parallel reduction method involves two implementations. Global memory implementation computes the reduction process by dividing the block size into two until only one element is left. So each thread in each block can add the values in its block. Hence, summation of elements is shared amongst blocks. Shared memory implementation is similar to global memory however data is copied from shared memory to global memory. Each thread in shared memory computes the numbers in its block and all the values are copied to global memory when all the threads are done.

2.3 Matrix Multiplication

The matrix multiplication is the process of multiplying two matrices and recording the results in an output matrix. Each value in output matrix is calculated by computing values from row of the first matrix and column of the second matrix. So each value is accessed as the size of width (number of rows or number of columns). This sequential method slows down the process as same element is accessed redundantly. Tiled matrix multiplication is a substantial method which improves the memory access by making use of the shared memory. The base code from *cuda_lab_*2 is used to implement the tiled matrix multiplication.

3 Experiment

3.1 Experiment Data

4 Conclusion