6CS005 Learning Journal - Semester 1 2019/20

Put your name and student number here

# Table of Contents

[Table of Contents 1](#_Toc527650969)

[1 POSIX Threads 2](#_Toc527650970)

[1.1 Password Cracking 2](#_Toc527650971)

[1.2 Image Processing 2](#_Toc527650972)

[1.3 Linear Regression 2](#_Toc527650973)

[2 CUDA 3](#_Toc527650974)

[2.1 Password Cracking 3](#_Toc527650975)

[2.2 Image Processing 3](#_Toc527650976)

[2.3 Linear Regression 3](#_Toc527650977)

[3 MPI 4](#_Toc527650978)

[3.1 Password Cracking 4](#_Toc527650979)

[3.2 Image Processing 4](#_Toc527650980)

[3.3 Linear Regression 4](#_Toc527650981)

[4 Verbose Repository Log 5](#_Toc527650982)

# Posix Thread

# CUDA

## Password Cracking

Insert a table that shows running times for the original and CUDA versions.

|  |  |
| --- | --- |
| Running Time for Original | Running Time for CUDA version |
|  |  |

Write a short analysis of the results

Here we can see, Run time for cuda version took considerably less time than the original version.

## Image Processing

Insert a table that shows running times for the original and CUDA versions.

|  |  |
| --- | --- |
| Running Time for Original | Running Time for CUDA version |
|  |  |

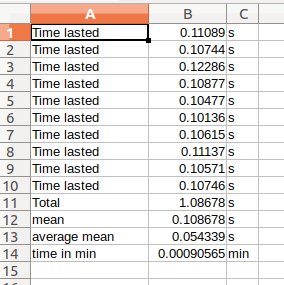
Write a short analysis of the results

Here, we can see that the cuda version took slightly more time.

GUPs generally have a good floating point computing power but not integer computing power, and modular (and division) operation is very slow. And for original CPU version, each integer can exit the loop right the given condition. However for GPUs, an integer must wait until all of its neighbour threads exit. This is the reason it took slightly more time.

## Linear Regression

Paste your source code for your CUDA based linear regression

Insert a table that shows running times for the original and CUDA versions.

|  |  |
| --- | --- |
| Running Time for Original | Running Time for CUDA version |
|  |  |

Write a short analysis of the results

Here also we can see that the cuda version took slightly more time. And it is due to the same reason.

GUPs generally have a good floating point computing power but not integer computing power, and modular (and division) operation is very slow. And for original CPU version, each integer can exit the loop right the given condition. However for GPUs, an integer must wait until all of its neighbour threads exit. This is the reason it took slightly more time.