## Declaration on Plagiarism

*This form must be filled in and completed by the student(s) submitting an assignment*

|  |  |
| --- | --- |
| **Name:** |  |
| **Student Number:** |  |
| **Programme:** | MCM |
| **Module Code:** | CA670 |
| **Assignment Title:** | Concurrent Programming |
| **Submission Date:** |  |
| **Module Coordinator:** | Dr. David Sinclair |

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. I have read and understood the Assignment Regulations. I have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

I have read and understood the referencing guidelines found at <http://www.dcu.ie/info/regulations/plagiarism.shtml>, <https://www4.dcu.ie/students/az/plagiarism> and/or recommended in the assignment guidelines

Name:\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_ April 2020\_\_\_\_\_\_\_\_\_\_\_\_

## Project Description

1. Project overview

使用openMP完成大型矩阵的乘法运算。对大型矩阵进行分块并使用多线程并行计算的方式提高运算效率。

1. The Purpose of the Project

应用openMP技术开发多线程并行计算，提高运算效率。

1. Design ideas

创建两个N\*M阶矩阵matrix\_a和matrix\_b相乘的程序，a和b中的所有值均为0~1的随机数，结果存放在矩阵result中. 在并行计算中，大型矩阵由4个task 来划分成4个小矩阵分别计算以此提高计算效率，最后输出运算时间。作为对照，串行计算的运算时间也被输出。具体函数如下：

* void init\_matrix()：初始化矩阵。
* void smallMatrixMult（）：矩阵乘法运算
* void matrixMulti（）：大型矩阵分隔
* void serial\_matrixMulti()：串行矩阵乘法运算

## Result Analysis

1. Running time

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 500\*500 | 1000\*1000 | 3000\*3000 | 6000\*6000 |
| 串行计算 | 0.3238 | 4.0445 | 163.4990 | 2021.4100 |
| 2线程 | 0.2457 | 1.9550 | 62.1604 | 936.9560 |
| 4线程 | 0.2036 | 1.3560 | 47.7519 | 619.1990 |
| 8线程 | 0.1338 | 1.1809 | 40.4989 | 545.890 |

## Conclusion

线程越多运算时间越少，计算效率越高。 并行计算小于远远高于串行计算。