

# TSN3151

## Parallel Processing

### Group Assignment

### Trimester 1, 2014/2015

**GROUP MEMBERS:**

| <i>Name</i>                | <i>Student ID</i> | <i>Contact Number</i> | <i>Tutorial Sections</i> | <i>Email</i>                     |
|----------------------------|-------------------|-----------------------|--------------------------|----------------------------------|
| <b>Moahmed Saleh</b>       | <b>1111113245</b> | <b>0183976570</b>     | <b>TC101</b>             | <b>Mohamed.saleh16@gmail.com</b> |
| <b>Tayyebah Azadzamani</b> | <b>1101109095</b> | <b>0122964099</b>     | <b>TC101</b>             | <b>Azadzamani@Gmail.com</b>      |

*Table of Contents*

|                          |   |
|--------------------------|---|
| Table of Contents.....   | 2 |
| Introduction.....        | 3 |
| Instructions.....        | 3 |
| Implementation:.....     | 4 |
| Sequential Program:..... | 4 |
| Parallel Program :.....  | 4 |
| Table Of Results:.....   | 6 |

## ***Introduction***

The Julia Set fractals are a group of mathematical functions on the complex number plane. In the file `julia.c` (provided with this assignment) is a C program to calculate the Julia fractal set where :  $c = -0.74543 + 0.11301*i$  and  $f(z) = z*z + c$

## ***Instructions***

1. Modify the code in `julia.c` so that the data is not written to a file immediately, but instead written to a two-dimensional array in memory, and only written to a file at the end, after all the calculations are finished.
2. Modify the usage of the Performance timer code in this program to measure the computational time of the calculation part, but not the file writing part.
3. Compile the modified code and run the program serially for 5 times on a dual core and above machine
  - a. You should have 5 sets of measures time results.
  - b. From the time results, obtain an average time taken to complete the program execution.

## **Message Passing Interface (MPI)**

- i. Note that all the data needs to be sent back to the master to be written to the file.
- ii. Instead of using the provided *Performance timer* code, use the **MPI\_Wtime()** function to measure the time taken to complete the program execution. The timing code must only be run on the master program. Be sure to place the timing code correctly so that the timings will validly calculate only the computational time and the communication time and not the file writing time.
- iii. Run the MPI program 5 times on a single quad core machine using 4 processes (i.e. each core represents one process).
- iv. Connect together at least 3 dual core and above machines using MPI and run the MPI program with 4 processes per machine for 5 times.
- v. Obtain an average time taken to complete the program execution for parts (iii) and (iv).
  - For each of the MPI code developed, plot the results on a graph. Evaluate your results for parts (iii) and (iv) to justify the selected partitioning scheme.

***Implementation:***

***Sequential Program:***

***1. Explanation of Serial Code:***

***2. Modify Code :***

***3. Result :***

***Parallel Program :***

***1. Explanation of Partition Scheme:***

***2. Modify Code :***

***3. Plot of Result :***

***— On Single Dual Core Machine :***

***— On Three Dual Core Machines :***

***Table Of Results:***

| Value of iXmax        |                | 10,000 (default)         |                      |
|-----------------------|----------------|--------------------------|----------------------|
| Value of iYmax        |                | 10,000 (default)         |                      |
| Value of IterationMax |                | 2,000 (default)          |                      |
| No                    | Serial program | Parallel Program (MPI)   |                      |
|                       |                | Single dual core machine | 3 dual core machines |
| Run_1                 | 86.514021      | 104.285692               | 74.139541            |
| Run_2                 | 86.110111      | 100.374868               | 74.371215            |
| Run_3                 | 86.157673      | 93.307541                | 74.566592            |
| Run_4                 | 86.829289      | 110.741109               | 74.892645            |
| Run_5                 | 87.116931      | 96.679407                | 74.389152            |
| Average Time          | 86.514021      | 103.9020693333           | 74.532926            |