EVALUATION REPORT
OPERATING SYSTEM
PROGRAMMING ASSIGNMENT 2: THREADS
UNIVERSITY OF TEXAS AT ARLINGTON

Date:10/06/2019

Name: Bijay Raj Raut
ID: 1001562222
Prof: Trevor Jay Bakker
CSE 3320-003

DESCRIPTION:

Exploring different patterns in the Mandelbrot the image, the pattern in figure 1 is the one that I chose to work with, and the image can be produced by using the following values of:
-x 0.29262
-y 0.01501
-s 0.000099
-m 1000
-W 1028
-H 1028

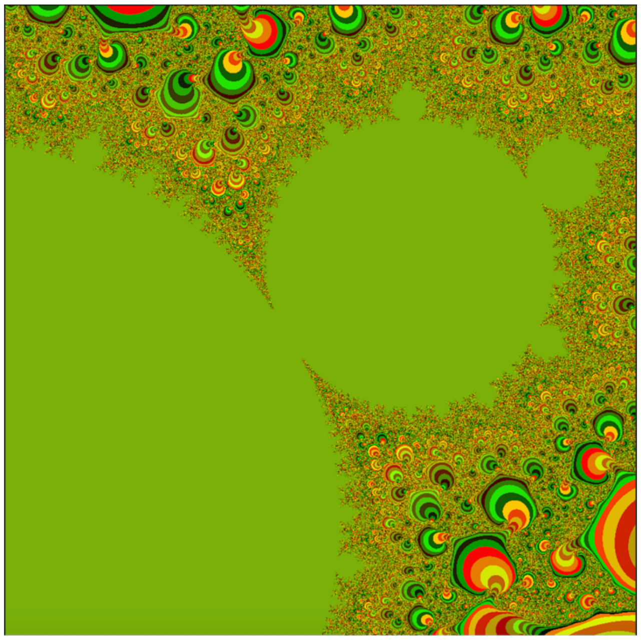
Figure 1 Generated Mandelbrot Image

The provided code was then modified to use the thread by adding the number of the thread in the parameter as -n followed by a number or the default of one thread. The compile time of the image gets reduced while using higher number of threads than the original program with no thread and one single process to compile and generate the image. After that the provided set of measurements were tested and the times were recorded to plot a graph which has been described below.

PURPOSE:

The purpose of this assignment is to learn and then utilize the usage of the threads. Mandelbrot use the escape time algorithm which can have very high number of computations depending on the value of the max iterations. The higher the max, height, and width of the picture is the lengthier would be the compile time. So, in order to improve the compile time. of the image we can either divide the work in multiple processes or have multiple threads working on the portion of the image. In the following assignment the work is being done using multi-threading in which the height of the image would be divided for number of threads and then each thread will work on the equal slice of the image to complete the task.

POSIX THREAD (pthread):

POSIX thread is the library that is going to be used to implement the threading concept. They are comparatively easier to implement and easier to manage. POSIX threads are cross-platform, so it is more portable to different environments compare to any other libraries.

MEASUREMENTS:

Attached table is measured value in omega:

A: mandel -x-.5-y.5-s1-m2000-W2048 -H2048
B: mandel -x 0.2869325 -y 0.0142905 -s .000001 -W 2048 -H 2048 -m 1000

Threads	A	B
1	24.65	14.5
2	22.38	7.51
3	15.75	5.55
4	11.22	4.12
5	10.59	4.14
10	6.91	3.77
50	6.44	3.78

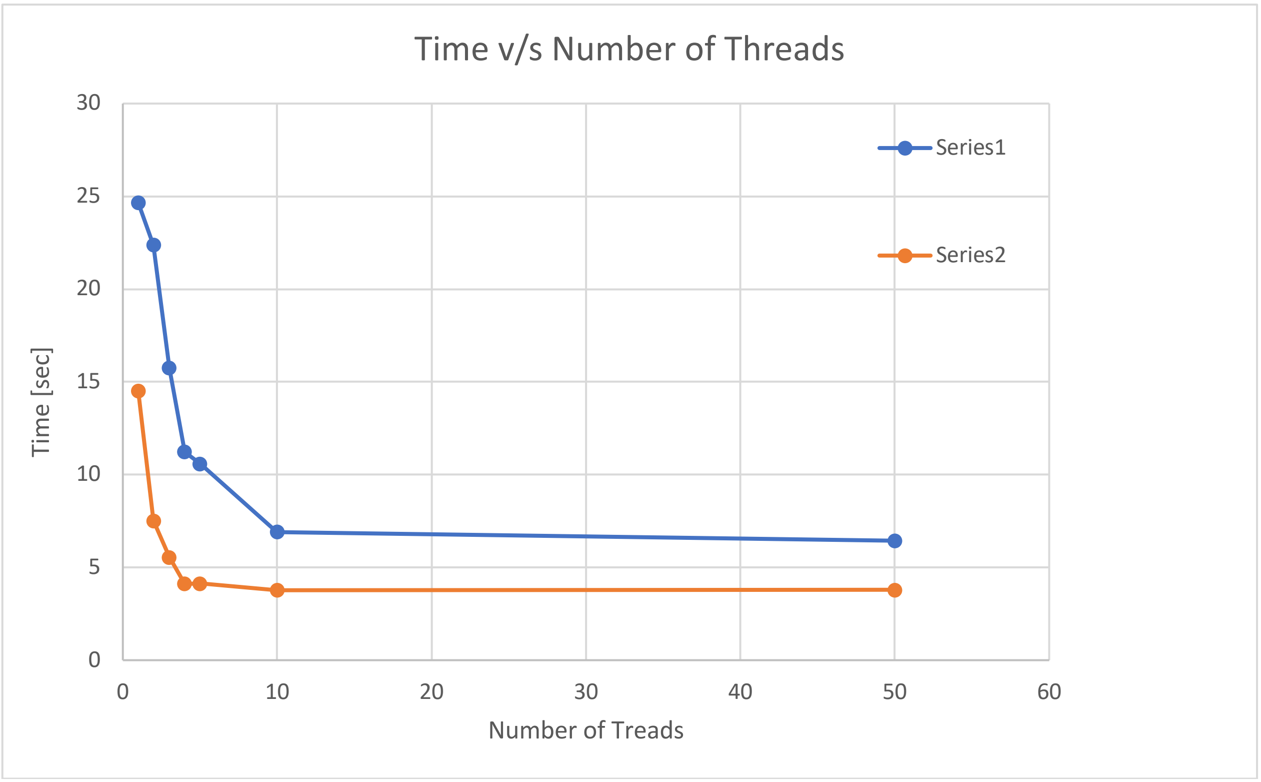

Figure 2 Time v/s Number of Threads: Series 1->A , Series 2->B

The graphical representation of the data's collected from the omega shows that after 10 threads the performance is kind of slim or non-existent. Graphs for A and B look quite similar but if we take a closer look at the graph and their corresponding run time in omega for some number of threads, we can see that they differ. Graph A has shown significant performance gain and after that there is a very small slope in the graph for increasing number of threads. While Graph B has performed in the somewhat similar manner, there are some ups and downs as we increase the number of threads. Which shows for B the performance gain is negative in some cases for the increasing number of threads.

To explain that we need to understand how to select the optimal number of threads. The performance boost cannot just simply be achieved by spawning off multiple threads. It takes time and memory to create a thread and for any given system it is finite. Omega is a quadcore system and there are not that many numbers of CPUs to be able to provide the resource required by each thread to operate. One thread can utilize one CPU at a time to do the computation and omega just have 4 CPU which all of the threads need to share including all the other users process that it is handling.

So, according to my research the optimal number of the threads for running in omega would be 10. As we can see from the Graphs. There is little to none performance gain after 10 threads.

CONCLUSION:

The run time of a high computation program can be cut short using multi-threading but there is a fixed number of threads up to which performance gain can be observed and that number varies from system to system depending on number of CPUs, their cores and whether it supports hyperthreading or not.