

OS2-PROGRAMMING ASSIGNMENT1

SADINENI ABHINAY
CS21BTECH11055

January 13, 2023

1 code design

The code constitutes total of three:

1. thread creation and data splitting
2. main thread computing pi value
3. create log of the events

2 Detailed explanation of code desgin

Total Four function:

1. **Main function:** Main function take input parameters through inp.txt file and create K threads which does the task "thread_work" and joins all threads. When all threads completes the given tasks the main thread invokes function "main_thread_work" after that it invokes log_creation
2. **task thread_work:** Using the index, the first block the code calculates size and starting index of the Generating_ set array ,now it generates random points of computed size and check whether it is inside the circle of radius "1" and centre (0,0) and stores it in the array. After the loop it exits
3. **main_thread_work:** main thread checks each random point stored in the global array for whether inside the circle and increments the no_ of_circle after the loop it returns

$$computed_value = \frac{4 * \text{No of points inside circle}}{\text{No of points inside square}} \quad (1)$$

4. **log_ creation:** Using the global array ,Random points generated by each thread are printind in the file "output.txt" No of points inside circle and square are determined by sizes and bool value stored in each element of the array

3 Analysis

1. No of threads v/s Time Taken

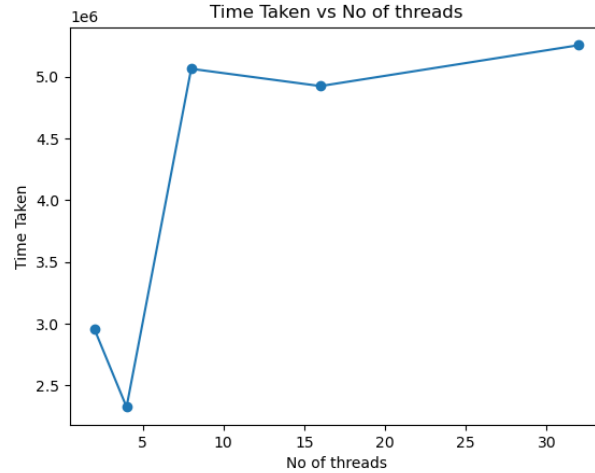


Figure 1: intel-i51035G1 chip(4 core)

Observations:

- (a) There is significant decrease in the execution time when no of threads is changed from 2 to 4, this is because 4 parallel threads are running and data is also split, which time taken by each thread to complete is decreased.
- (b) There is drastic increase in the execution time when no of threads is changed from 4 to 8, this is because the speed up due to the data splitting is not enough to compensate the overhead of the thread creation.
- (c) For $K=8, 16, 32$ the time taken is almost the same, slight variation due to the CPU scheduling and other factors.

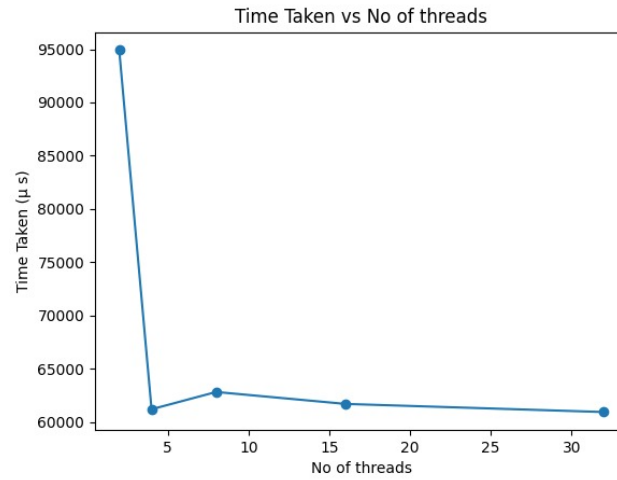


Figure 2: Apple M1 chip(8 core)

Observations:

- (a) In this case also the execution time decreases when no of threads is changed from 2 to 4, this is because 4 parallel threads are running and data is also split so time taken by each thread to complete is decreased.
- (b) there is slight increase in time when K changed from 4 to 8, because extra time by create 4 extra threads which is not compensated by the speedup.
- (c) From $k=8$ to 32 there is expected trend, As the data been split among threads time taken is expected to decrease.
- (d) if noticed properly the two graphs exhibit different trends mainly to due chip architecture and os in which this program is executed.

2. No of points v/s Time Taken

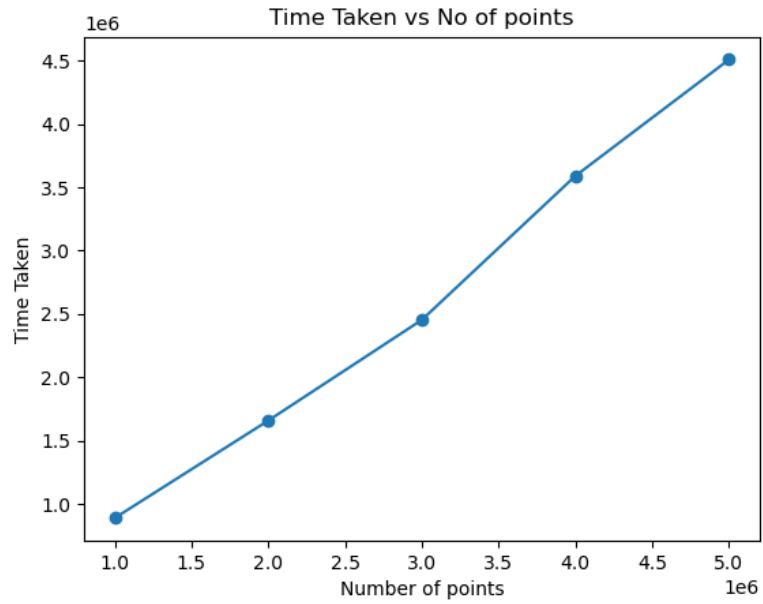


Figure 3:

When No of points are increased ,the work of each thread increases as the no of iterations done in the task also increases ,so the execution time also increases