Assignment 1

Ameet Deshpande (CS15B001)

August 14 2017

1 Using pthreads.h

- *pthreads.h* facilitates the creation of threads and is thus supposed to speed up computation.
- *pthreads_create* function creates a new thread and calls the function using that thread. It also indicates if the thread has been created or not.
- *pthreads_join* pauses the parent process until the threads have returned. Once the threads return, the function continues it's execution.
- The matrix multiplication is done using the standard "for loops" algorithm. The complexity of the code is $O(n^3)$.

2 Analysis of Running times

- A testcase was created which contained matrices in the order 500×500. This is expected to yield a running time of about 0.4 seconds. The link for the testcase can be found here.
- The running time is expected to decrease as the number of threads increases and at some point it is supposed to start increasing again.
- The analysis was conducted on thread values from 1 to 10.
- The trend that followed was that the running time decreased when the number of threads increased from 1 to 2 till 3 and then the time taken started remaining almost constant till few cores. It started increasing a little later on. (The graph and table can be found below)
- The trend can be explained as follows. As the number of threads starts increasing, the number of cores utilization also increases. The system which I worked with clearly has three cores. This is because, the parent thread itself is one thread. And after generation of three other threads, all cores are being utilized for the following program. More number of cores than this will just stagnate the values.

3 Data and Graph

Threads	Time
1	0.402
2	0.395
3	0.390
4	0.404
5	0.403
6	0.404
7	0.400
8	0.412
9	0.411
10	0.415

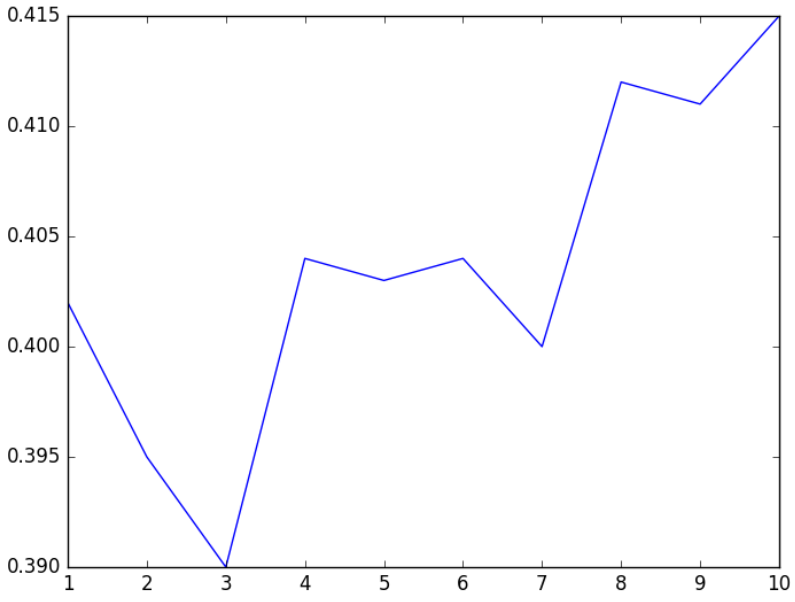

Figure 1: Graph

4 C and Java

- Java Threads are expected to take a longer time than C threads because of the abstraction that Java Virtual Machine provides.