CPSC 457 – Assignment3

Name: Farica Mago

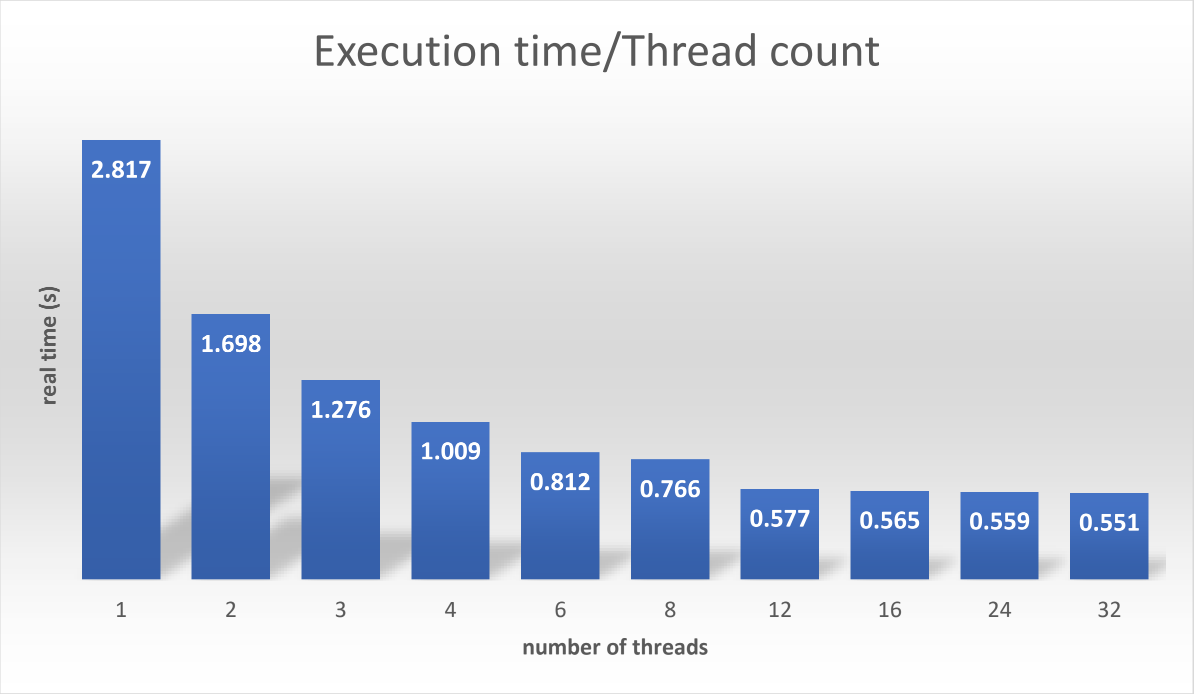
UCID: 30111924

Question 2: Written answer

A. Record your timings in a table and create a corresponding bar graph.

Threads	Timing(s)
1 (Original)	2.728
1	2.817
2	1.698
3	1.276
4	1.009
6	0.812
8	0.766
12	0.577
16	0.565
24	0.559
32	0.551

Graph:



B. When you run your implementation with N threads, you should see N-times speed up compared to the original single threaded program. Do you observe this in your timings for all values of N?

When I run my code, I do see that my program speeds up N times for N threads up to first two values (as compared to the original single threaded program).

I did not observe this for all the values of N. The speeding of the program becomes less and at some point, where the number of threads = 6 and it reaches a stable point when number of threads becomes 12.

C. Why do you stop seeing the speed up after some value of N?

We stop seeing the speed up after some value of N because, the number of threads becomes large and the thread function keeps getting called for each of the threads. After a specific number of threads, the efficiency of threading does not make much of an affect since the large number of thread itself has so much to do. After a specific number of threads, the number of threads becomes more than the number that is actually needed.

Question 4: Written answer

For medium.txt

# threads	Observed timing	Observed speedup compared to original	Expected speedup
Original program	17.592	1.0	1.0
1	17.686	1.00	1.0
2	9.369	1.88	2.0
3	6.569	2.68	3.0
4	5.349	3.29	4.0
8	3.649	4.82	8.0
16	2.907	6.05	16.0

For hard.txt

# threads	Observed timing	Observed speedup compared to original	Expected speedup
Original program	6.022	1.0	1.0
1	5.996	1.00	1.0
2	3.117	1.93	2.0
3	2.122	2.84	3.0
4	1.603	3.75	4.0
8	1.056	5.70	8.0
16	0.870	6.92	16.0

For hard2.txt

# threads	Observed timing	Observed speedup compared to original	Expected speedup
Original program	6.090	1.0	1.0
1	6.052	1.00	1.0
2	3.165	1.92	2.0
3	2.123	2.87	3.0
4	1.678	3.63	4.0
8	1.398	4.36	8.0
16	0.794	7.67	16.0

We can see from the results that the overall efficiency of the program increases as we increase the number of threads. The results for the first four observations seem to be close to what is expected. The observation for the 8 and 16 threads is way off than what was expected. I think is reason is the same as we discussed above, i.e. at a specific number of threads, the number of thread is way more than actually needed for the required efficiency of the program. The threads begin to clutter once the number become more than required. If the number of threads is too much, then each thread will get so little amount of work to do, and more work will be done to switch between the threads.