ASSIGNMENT GNUPLOT

Raghav Garg
160534

March 26, 2018

ONE POINT GRAPH FOR EACH THREAD WHERE X REPRSENTS NUMBER OF ELEMENTS AND Y AXIS REPRESENTS EXECUTION TIME FOR EACH SAMPLE

Figure 1: ....

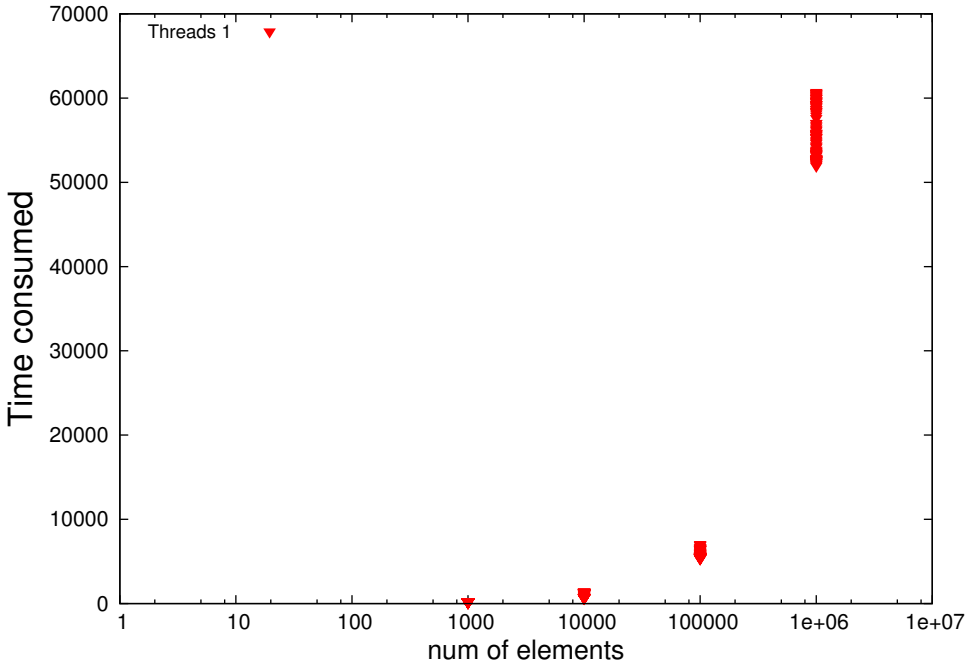


Figure 2: ....

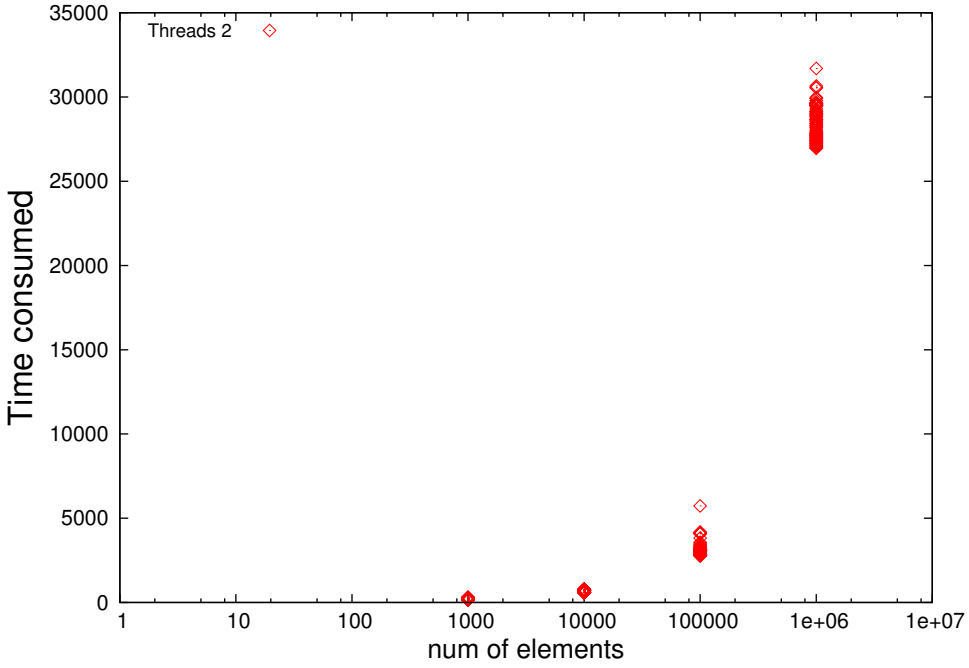

Threads 2
Time consumed
num of elements
35000
30000
25000
20000
15000
10000
5000
0
1
10
100
1000
10000
100000
1e+06
1e+07

Figure 3: ....

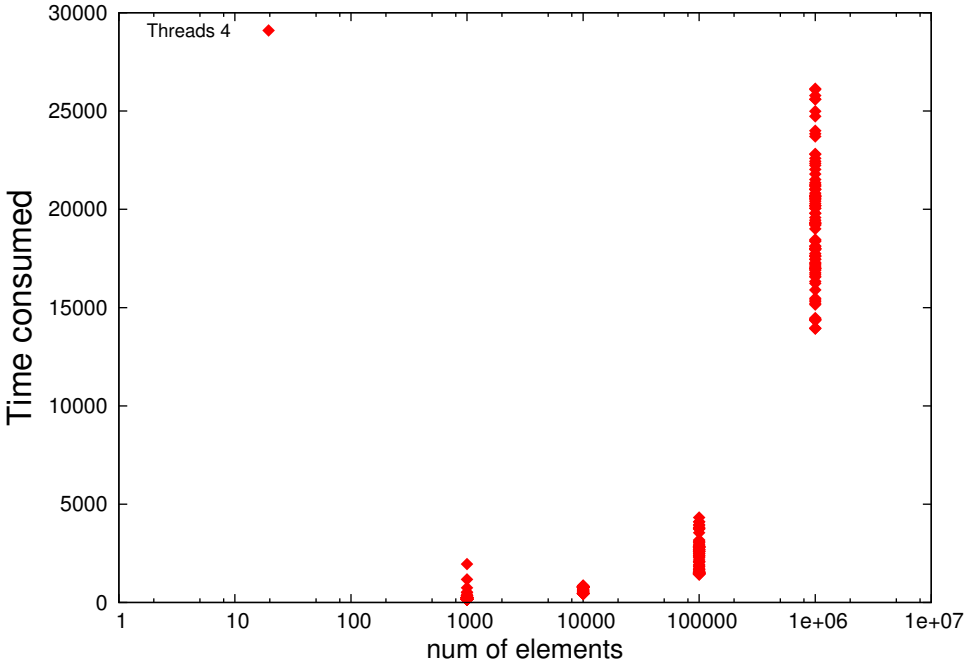

Threads 4
Time consumed
num of elements
30000
25000
20000
15000
10000
5000
0
1
10
100
1000
10000
100000
1e+06
1e+07

Figure 4: ....

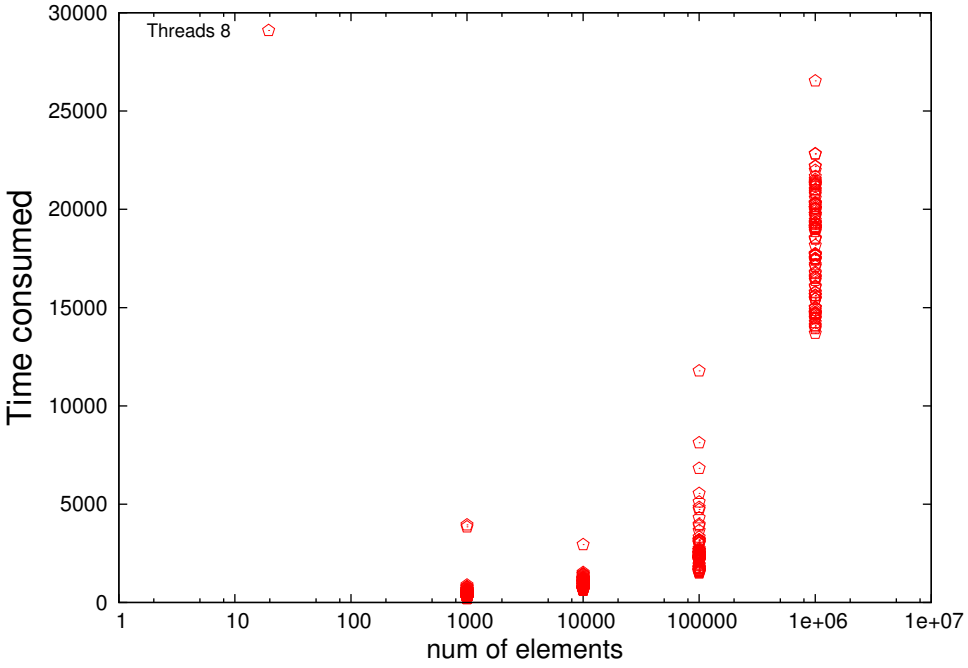
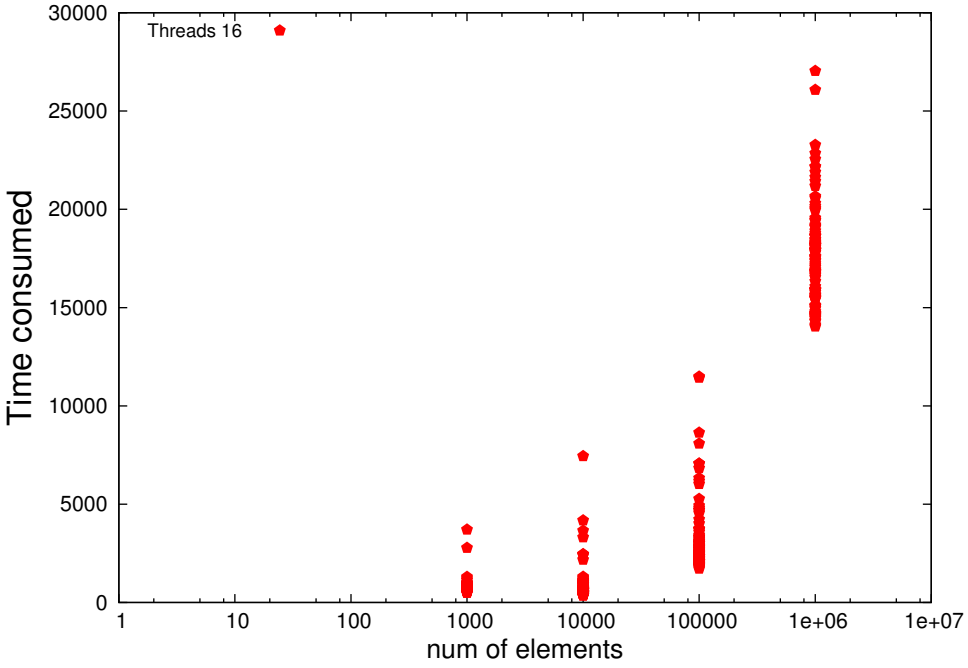

Threads 8
Time consumed
num of elements
30000
25000
20000
15000
10000
5000
0
1
10
100
1000
10000
100000
1e+06
1e+07

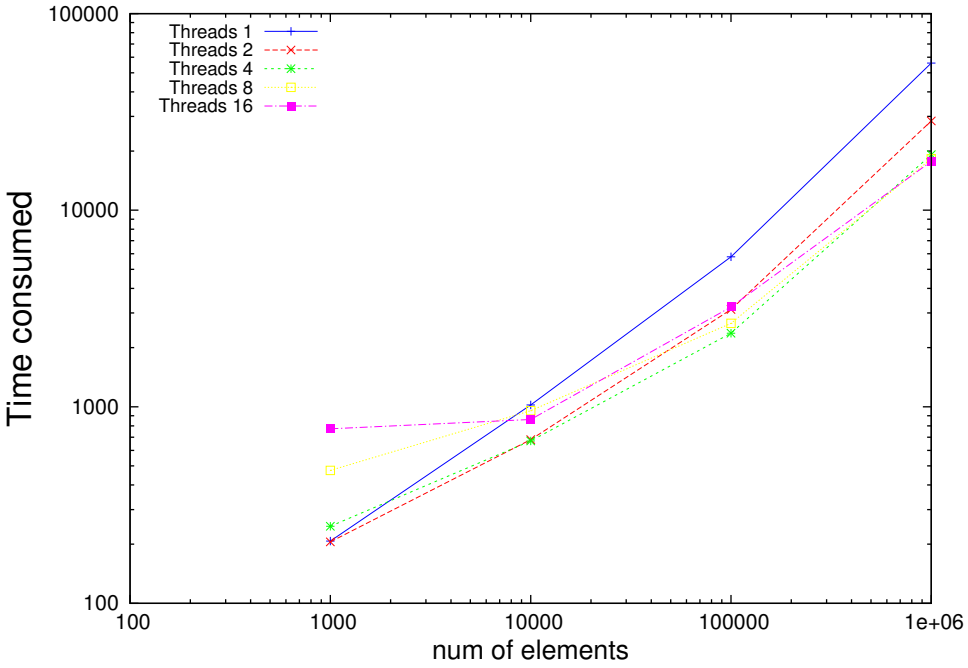
Figure 5: ....



The Given Figure represents the execution time for 100 samples for the given number of elements for different number of threads.With increase in threads, execution time decreases.However this thing is not very much evident with smaller number of elements as the latency induced in thread creating is not overcome by task time.

X is number of elemenets and Y is average Execution time over 100 samples

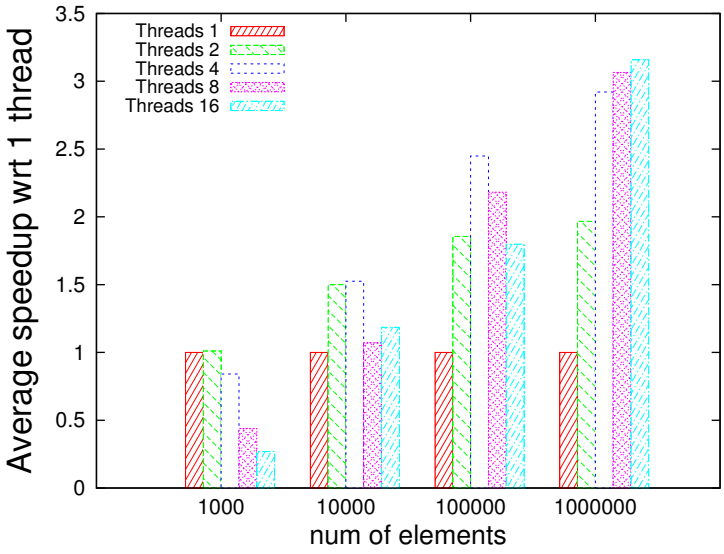
Figure 6: ....



The above graph is plotted against log scale of X.We can see that the average execution time decreases with increase in number of threads but since my compute had quad core processor increasing threads beyond 4 did not help much.

X is numner of elements and Y is average time for 1 thread

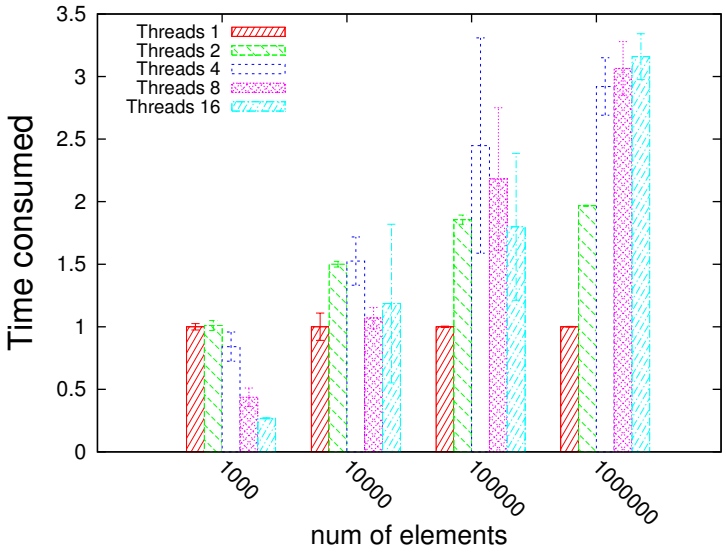
Figure 7: ....



This is just a bar graph reprenation of the average speedups where the average is calculated for each thread for different number of samples and then the ratio is taken against 1 thread for speedup calculations.Thread 1 has default average of 1.

X is numner of elements and Y is average time for 1 thread with error bars shown

Figure 8: ....




```
Variance is calculated in the avergae speedup of each threads treating average
 time of thread  1 as the golden time against which I calculated my means.More
 error is induced with increasingnumber of threads as the data gets more scattered.
```