HOMEWORK #5 - TRAVELLING SALESMAN PROBLEM

TABLE A : 1 COMPUTER

Communication Latency Optimization	Workers	Avg. time as seen by the client (Tc milliseconds)	Parallel Efficiency : T1/C.Tc
On	Multiple	13071 (T1)	1
On	Single	15224 (T1)	1
Off	Multiple	19430 (T1)	1
Off	Single	20200 (T1)	1

TABLE B : 16 COMPUTERS

Communication Latency Optimization	Workers	Avg. time as seen by the client (Tc milliseconds)	Parallel Efficiency : T1/C.Tc
On	Multiple	8102	0.1008
On	Single	11045	0.0861
Off	Multiple	60660	0.02
Off	Single	73330	0.0172

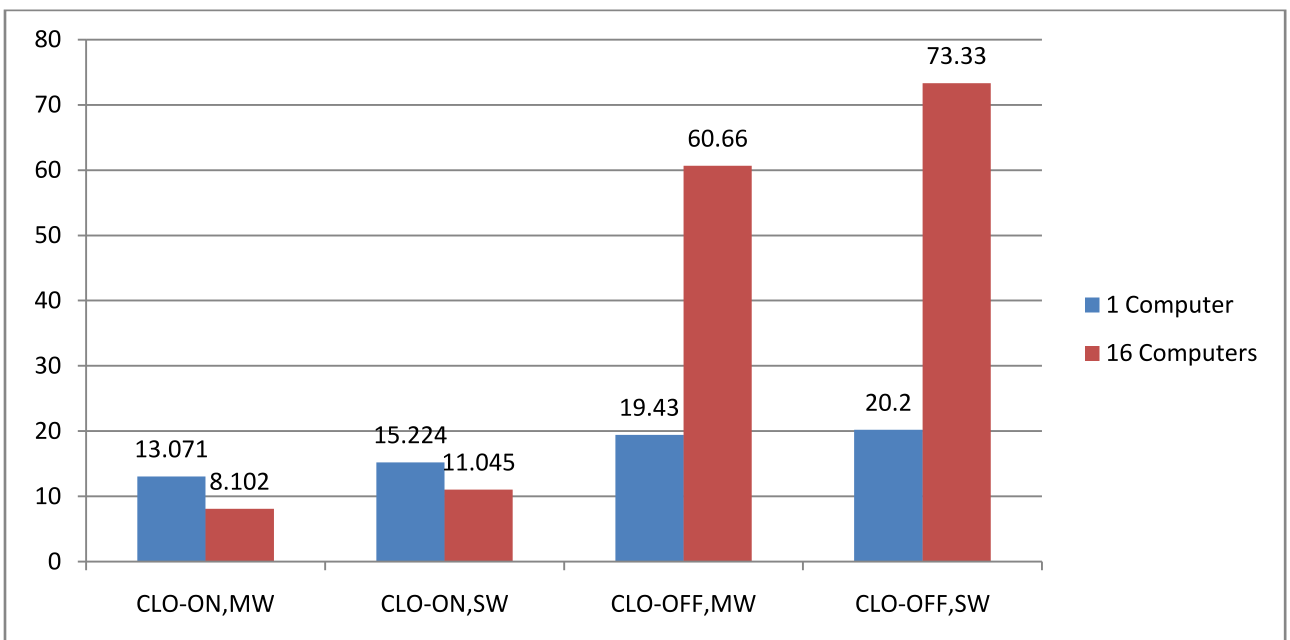
TABLE C : AVERAGE EXECUTION BASED ON LATENCY OPTIMIZATION

Communication Latency Optimization	Computers	Avg. execution time across different worker types viz. multiple/single (milliseconds)
On	1	14147.5
Off	1	19815
On	16	9573.5
Off	16	66995

Diff. = 5 .6 s

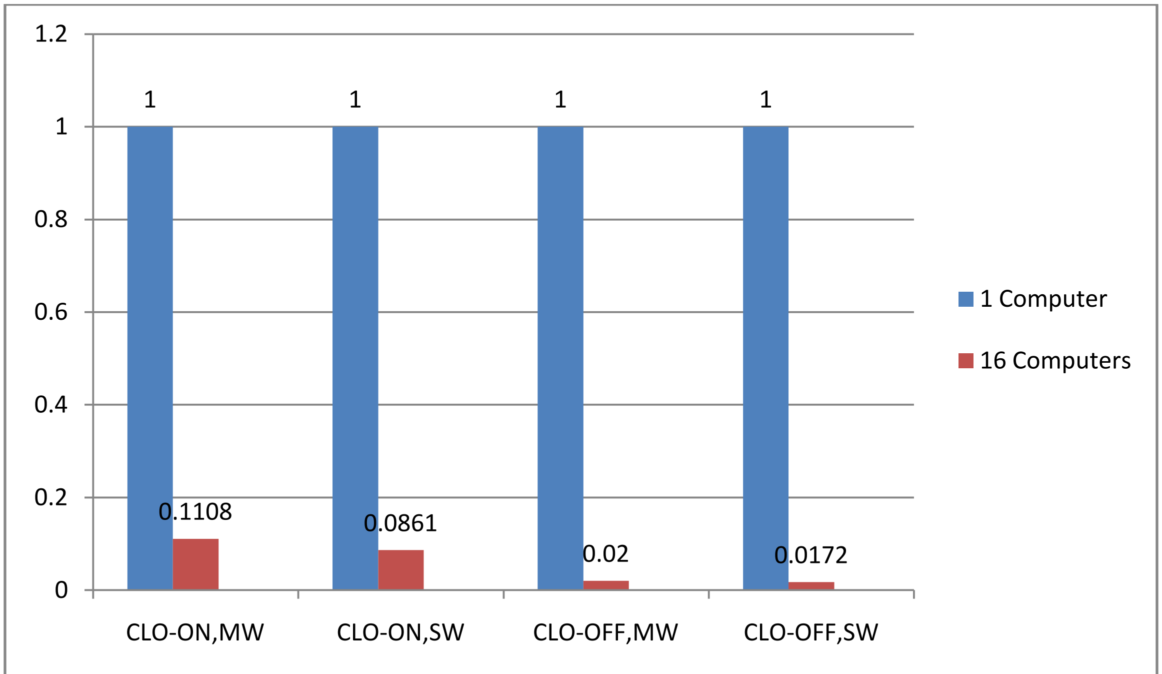
Diff. = 57.42 s

GRAPH A - EXECUTION TIMES



CLO – Communication Latency Optimization
MW – Multiple Workers
SW – Single Worker

GRAPH B - EFFICIENCY PLOT



CLO – Communication Latency Optimization
MW – Multiple Workers
SW – Single Worker

In the case : C=16, it can be seen that the parallel efficiency decreases with increase in number of parallel processors. This suggests that the degree of parallelization keeps decreasing and approaches zero as more parallel computational power is added to the infrastructure. This was an expected result. Not all tasks can achieve perfect parallelization, and this also holds good for the TSP task which we experimented with.

However, we see a clear difference in numbers when compared to homework #4. In the table below, we have considered the best case in homework #5 viz. (Multiple Workers + Communication Latency Optimization) verses the normal case in homework #4.

No. of computers (C)	Homework #4 Parallel Efficiency	Homework #5 Parallel Efficiency
1	1	1
16	0.05	0.1008

For both homeworks, we used the same input containing 12 cities and the same decomposition depth of 5. After the 5th level of decomposition, each computer used a depth-first search algorithm to systematically traverse all possible paths locally and prune nodes along the way using the branch-and-bound strategy.

As seen in the table above, the parallel efficiency has improved twofold.

(1) This can be attributed to the decrease in time lost in communication due to the presence of a local queue in each of the 16 computers. From Table C (see above), one can note the drastic difference in times between cases where communication latency optimization was switched on and off. In the absence of any optimization, a massive number of subtasks generated in the 5^{th} level of decomposition increase the time spent in communicating with the compute space and also the idle time of the computers.

(2) The algorithm implemented by us for homework #4 to compute a tighter lower bound for TSP was computationally intensive with a complexity of O(N^2). This has been replaced with a lighter algorithm discussed in class that has a linear time complexity for tightening the lower bound. This should have also contributed to the improvement in numbers.