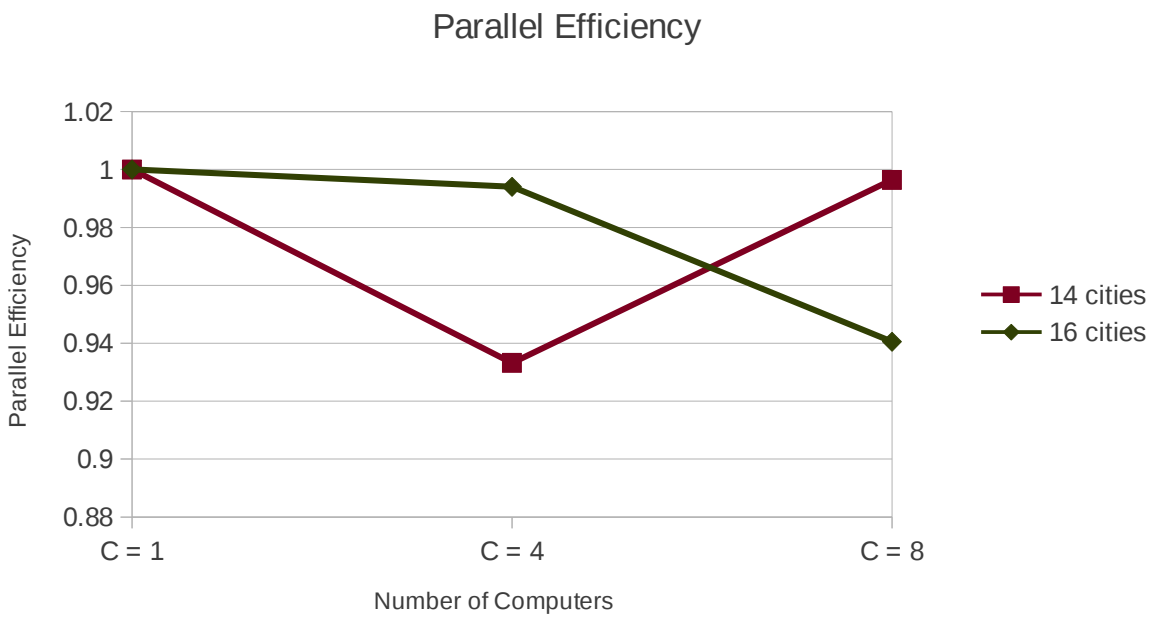
Parallel Efficiency



As we can see, for both the cases (14 cities and 16 cities), the parallel efficiency increases as the number of computers increases. Since the tasks in the ready queue are processed in a depth first manner, the lower bound is computed for a subtree rooted at at least one of the nodes in the current level is computed before all the subtrees having the root as the other nodes in that level are explored. This can facilitate faster pruning.

Also, when an atomic task is executed, instead of computing all the permutations for the remaining unexplored cities, a stack is used to prune the sub-trees. In this way, more sub-trees are pruned without actually exploring each unexplored node in the search tree.

Since the upper bound is stored in a shared object, when the tasks are executed in parallel across many computers, each task can prune its sub-tree more quickly by comparing its partial tour with the shared value. Otherwise, each atomic task would have to compute the minimal tour and then be composed to form the least tour among the minimum tours computed by all the tasks.