Experiment 06

Name : Kiran K. Patil Sem : 06
Reg ID : 211070904 Course : PC Lab.

Aim :- Parallel Travelling salesman Problem using openMP implementation.

Theory :

- The Traveling Salesman Problem (TSP) is a problem to find shortest possible route that visits a set of cities and returns the starting city.

- OpenMp is a popular ~~(api)~~ API for parallel programming is a shared-memory architecture

- we can parallelize the local search part of the TSP algorithm using openmp to speedup its execution

- To parallelize the algorithm, we can use the '#pragma omp parallel for' directive and the openMP 'reduction' clause to parallelize the loop that iterates over all pairs of cities and find the best swap for each thread.

To implement Dijkstra's algorithm using OpenMP, we can parallelize the loops that updates the distanc values of the neighbors of the selected vertex Each thread maintains a private copy of the distance array and set of visited vertices.

When a thredd selects a vertex to add to the set, it updates the distance array and the set of visited vertices, when a thread selects a vertex to add to a set it updates the distance array and marks the vertex as visited in its private copy

After all thredds have finished processing their private copies, the main thread merge all the results by selecting the smallest distance value for each vertex across all private copies.

- To achive correct and efficient parallelization we need to use synchronization primitives such as 'reduction', 'atomic' and 'critical' directives to manage shared and private data and ensure correct results. By parallelizing Dijkstra's algorithm using openM we can - achive significant speedup on multicore multicore processors, especially for large graphs with many vertices and edges.