Muneel Haider
I21-0640

PDC Assignment 2

Q1 - Report

Introduction

The Traveling Salesman Problem (TSP) poses a classic challenge in combinatorial optimization, seeking the shortest possible route through a set of cities, visiting each once before returning to the start. This report explores a parallel solution to the TSP using the Message Passing Interface (MPI), aiming to harness the computational power of multiple nodes to improve solution times over traditional serial approaches.

Methodology

The TSP was modeled using an adjacency matrix within a **CityMap** structure, enabling efficient representation of city-to-city distances. A parallel algorithm was implemented, dividing the problem across multiple computing nodes, each tasked with exploring different routes via depth-first search (DFS). Key to this approach was the use of MPI for:

- Broadcasting the **CityMap** to all nodes, ensuring consistent data across the computational cluster.

- Aggregating the results through **MPI_Reduce** to determine the shortest route identified across all nodes.

Performance Evaluation

- ### Experimental Setup
    - The evaluation used a problem instance of 5 cities.

- ### Results and Analysis
    - The parallel TSP solver significantly outperformed the serial solver, reducing the computation time from 0.0167 seconds (serial) to 0.000047 seconds (parallel). This dramatic decrease highlights the efficiency of parallel computation, particularly for computationally intensive problems like the TSP.

- ### Scalability
    - While the parallel solver exhibited excellent performance for the given problem size, the scalability of this approach for larger problem instances remains a subject for further research.

Conclusion

Parallelizing the TSP using MPI offers substantial improvements in computation time over serial methods, underscoring the potential of parallel computing in addressing complex optimization problems. The success of this approach for a small problem instance encourages further exploration of parallel strategies for larger instances, although careful consideration must be given to the challenges of workload balance and communication overhead.