

## Assignment 2 – Travelling salesmen problem (TSP)

The purpose of this assignment is the following:

- Parallel programming experience, which includes communication, synchronization, data structures, converting sequential algorithms to parallel algorithms, debugging etc.
- Complexity of a NP-complete problem.
- Specific topics as dynamic load balancing and distributed termination.
- Written presentations, in English, of results that clearly emphasizes the results of the tests.

In TSP, a salesman is given a list of cities she needs to visit and a cost for travelling each pair of cities. Her problem is to visit each node once, returning to her hometown, and she must do this with the least possible cost.

The TSP problem is described in detail in the course book (chapter 6.2). This includes description of the problem, algorithms, parallel algorithms in MPI and OpenMP, results and discussions.

### The implementation

Your task is to implement a parallel TSP on AKKA using MPI. The focus should lie on dynamic load balancing and distributed termination.

*Do not focus in optimizing the sequential algorithm.*

### The following results must be included in your report.

For all tests and results, you should use AKKA and the implementation should be based on MPI. You are encouraged to (but it is not mandatory) follow the step-step working approach in this specification. Finally, you are allowed to work in pair.

The following results are mandatory, but other results could also be of interest.

1. *No load balancing is invoked*: Traverse the tree without cutting. Thus perfect (almost) load balancing is known in advance. Verify that the efficiency tends to 100% when the size of the problem increases. Use 4 cores.
2. *Testing the load balancing implementation*: Cut off part of the tree in a controlled manner. (That is, do not cut the path of the solution). How efficient is your load balancer? Use 2, 4, and 8 cores.
3. *Full algorithm*: Solve the TSP problem and present relevant figures (diagrams). Use 1,2,4,8 cores.

## Deadline

Will be decided during the course (see the scheme on the course home page)

## Assignment 2 – Step-by-step approach

A recommend step-by-step way of working in this assignment is the following:

1. Sequential algorithm. Learn from the problem, and study the iterative and recursive algorithms from the course book
2. Examine different methods for traversing the generated tree. From the course “data structure and algorithms” you have learned methods as in-order, pre-order and post-order.
3. It is important to store the data in an efficient way. Thus, the data structure used will be of great importance. It is sufficient to use integers. But what is to prefer among the use of list, stack, and trees?
4. Study the parallel algorithms from the course book.
  - a. How to generate a problem that you want to solve. How important is the size of the problem?
  - b. Data structures again. How to store all jobs?
    - i. How to get a good start?
    - ii. Are priority queues an alternative?
  - c. Load balancing is the focus in this assignment. Important issues include:
    - i. Receiver-oriented or sender-oriented?
      1. When to ask for jobs.
      2. When to send jobs.
    - ii. Mapping: ring, mesh or other?
    - iii. Should the size of local queue vary freely?
    - iv. How should you handle the bounding part?
  - d. Distributed termination must be implemented, of course. Ideas can be found in the course book.
  - e. MPI shall be used. How should that be done to make it as simple as possible?
5. How should you to present the result (see previous page)?
  - a. What is the focus in this assignment?