

## Assignment goals

### Objectives

The main goal of this exercise is introducing the student to the basic concepts of parallel programming. To do that, two different options are provided. The first one is more theoretical and is oriented to those students that do not have strong computer science background (i.e: programming, linux etc.). This option will let the student to propose a parallelization of a specific scientific problem. The second one is more practical and it is oriented to those students that have programming and system skills. In this second option the student will implement a parallel application using OpenMP and MPI+OpenMP.

### What it is expected

1) the sources for the developed code; 2) one document with the answers with the proposed questions, the performance evaluation and the comments that the student wants to make. The developments must be done in C and with the required extensions for OpenMP and MPI.

The maximum length for the document is 4 pages.

## Activities

### 1. The serial algorithm

The goal of this first part is to understand the nature of the smith-waterman algorithm: what it solves, what inputs it needs, what it generates, its computational cost and what parts of it can be potentially parallelized.

- 1.1 Describe what the smith-waterman algorithm does (include the references that you have used): inputs, outputs and pseudo-code describing the algorithm. The pseudo-code must include comments on
  - 1.1.1 Describe the computational complexity of the algorithm. It is expected to provide a description on the serial implementation (i.e:  $O(n^2)$ ,  $O(n \log n)$ )
  - 1.1.2 Describe what parts of the algorithm can be potentially parallelized.

### 2. Parallel implementation

The goal of this second part is to propose a pseudo-code parallel implementation for the parts identified in 1.3.

- 2.1 Describe in pseudo-code a potential parallel implementation for this algorithm:
  - 2.1.1 What strategy have you selected? (i.e: pipeline, shared memory, message passing etc.) Why?
  - 2.1.2 What other options could you use?
  - 2.1.3 Describe the pseudo-code including comments on why the different parallel parts were selected.
- 2.2 Describe the computational complexity of the algorithm. It is expected to provide a description on the serial implementation (i.e:  $O(n^2)$ ,  $O(n \log n)$ )

### 3. Performance projection

- 3.1 Given the pseudo-code proposed in 2.1 and 1.1 propose a theoretical model to project the speedup that the parallel implementation may show with respect to the serial implementation. (It's a model so it's not expected to provide 100% accuracy).
- 3.2 Provide a speedup analysis using the previous model for: 1, 2, 4, 16 and 32 threads.
- 3.3 Provide a description of what type of computational system would be better for the provided implementation and what components would be important to invest more:
  - 3.3.1 Software
  - 3.3.2 Hardware

## Evaluation criteria

Brevity and clear results, experiment setup and discussion and analysis.



## Format

One PDF document containing all the different answers for the selected option containing:

- The answers to the formulation questions (must not exceed 4 pages).
- All the different codes developed or scripts must be added as annex section at the end of the document (no limit)

