BE Systèmes Concurrents

07/12/2015

This BE is made of five different exercises of increasing difficulty for a total of 100 points:

Part 1 Schedules: an exercise on the usage of different scheduling types in the OpenMP parallel for worksharing construct. **15 points**.

Part 2 Tree traversal: an exercise on the parallelization of a postordered binary tree traversal. **20 points**.

Part 3 MergeSort: an exercise on the parallelization of the Merge Sort algorithm. **20 points**.

Part 4 ConjugateGradient: an exercise on the parallelization of the basic operations used in the Conjugate Gradient iterative method. **20 points**.

Part 5 PrefixScan: an exercise on the parallelization of the inclusive Prefix Scan operations. **25 points**.

For each exercise, detailed explanations and instructions are given in the `subject.pdf` file inside the corresponding directory.
All the exercises include some coding tasks: these tasks consist in writing, compiling and executing some OpenMP parallel code and are identified by the keyboard symbol ⌨.
Some exercises also include more theoretical questions that are identified by the pencil symbol ✎. Please write the answer to these questions in the `responses.txt` file in the topmost directory of the BE package. **Answers can be written in French**.

General advice:

- When implementing the parallelization, test your code on small data. When you're sure everything works fine, increase the size of data to evaluate performance.
- All the proposed parallel solutions have to work with any (reasonable) number of threads. This means that the parallel code has to work also in

the case where only one thread is used. Check your parallel code with one thread first; this case will be easier to debug in case of problems. Then test with more threads.

- The amount of coding required in each exercise is relatively small. If you find yourself writing a lot of code, you're probably on the wrong track.

Important

Once you have finished execute the `pack.sh` script like this:

```
$ ./pack.sh
```

This will generate a package containing the code you have developed and the responses you have provided and automatically send it to the supervisors.
Before leaving verify with the supervisor in your room that the package has been received.