TP - Shared-memory programming: OpenMP (III)

ING2-GSI-MI – Architecture et programmation parallèle

Academic year 2023–2024

Synchronisation

1 Write a program that computes the degree of each vertex in a graph. Write first a sequential version using the code `01-graphdegree_skel.c` provided in the *skeletons.zip* file. Then, add the parallelism. Measure the execution time for 1, 2 and 6 threads. Use the graphs from the following files:

- `https://algs4.cs.princeton.edu/41graph/tinyG.txt` to test that your code works.
- `https://algs4.cs.princeton.edu/41graph/largeG.txt` to measure the execution time.

A description of the format of those files can be found on `https://algs4.cs.princeton.edu/41graph/`. Does the time improve as you increase the number of processors? Why? What could you do to improve the performance? □

Tasks

2 Using `tasks`, write a parallel program in OpenMP to calculate, in a recursive way, the sum of the elements of an array of size N. Use 2 threads. For a size of N large enough, compare the execution time with the sequential version. □

3 Write a parallel version of the Mergesort (*tri fusion*) algorithm, using either `tasks` or parallel `sections`. □