OpenMP Report

Lorenzo Fresco

June 2019

1 Exercise 1

The exercise required us to compute an approximation of the value of Pi using OpenMP and a Montecarlo integral approximation.

2 Procedure and Results

For the execution of the exercise we were asked to implement four different function that cuold obtain the desired result

1. `serial` a serial implementation that does not use any kind of parallelization
2. `atomic` a parallel version that uses the `atomic` keyword for the reduction phase
3. `critical` a parallel version that creates a critical region in order to avoid race condition
4. `reduction` still a parallel implementation that uses a special keyword to perform the final summation

In order to see the different results and how these different functions behave we decided to fix `N` (the number of subintervals that we want to use in our approximation) to 1000000000 and to run the different functions with an increasing number of threads. The obtained result can be seen in the plots below.

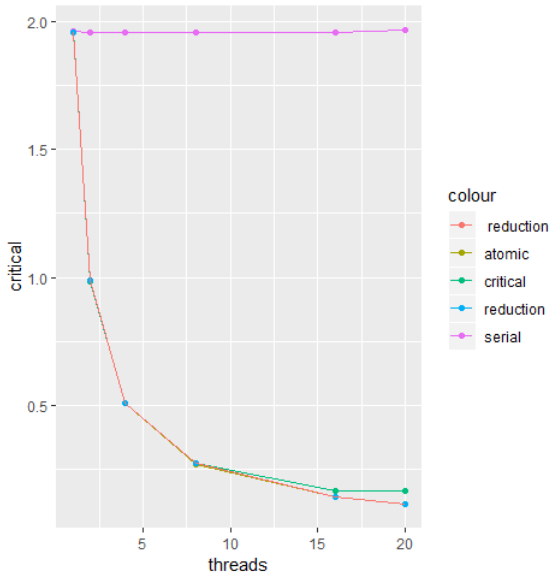

Figure 1: Number of threads vs Execution Time

So as we can see from the graph with all the different implementations of the parallel versions we obtain a good scalability and a much faster execution compared to the serial one. The fact that we are not able to see differences nbetween the various implementations may be due to the fact that we are just performing a single operation (the sum).

3 Exercise 2

In the second exercise we were asked to visualize different OpenMP schedules using different chunks. The result can be found inside the `openMP/LOOP_SCHEDULE/RESULTS` folder.