TP - Shared-memory programming: OpenMP (II)

ING2-GSI-MI – Architecture et programmation parallèle

Academic year 2023–2024

Calculate π

1. Using `parallel for` loops, rewrite the algorithm on Exercise 5 from the previous TP to approximate π in parallel. Measure the execution time for 2, 4, 8, 64 and 128 threads. Use a number of rectangles large enough to obtain execution times relatively long (at least, about 1 second). ☐

2. If a circle of radius R is inscribed inside a square with side length 2R, the ratio of the area of the circle to the area of the square will be $\pi/4$. Estimate the value of π using the **Monte Carlo method**. Then, write a parallel version using OpenMP. Calculate the *speedup* for 4 threads. ☐

Playing with matrices

3. Rewrite the code of the matrix-vector product that you wrote for the previous TP so that it uses `parallel for` loops. As before, you must parallelise your program using a row-wise partitioning. In this new code, how is the iteration space divided up by default among the available threads? Use a matrix big enough and, for 4 and 64 threads, compare the `static`, `dynamic` and `guided` scheduling strategies. Which strategy and with which number of threads the product yields the best results? **Note:** Use `#pragma omp for schedule(runtime)` in your code and a script to run all the different versions of your program. ☐

4. We have two matrices of doubles, A (size $M\ rows \times N\ columns$) and B (size $N\ rows \times P\ columns$). Write a parallel code in OpenMP to calculate the matrix product $A * B = C$. We will follow the simplest parallelisation strategy: B will be shared by all threads, whereas A will be divided up by blocks. Measure and compare the execution times of the sequential and parallel versions for 4 threads. Use the `04-mm.c` code in the `skeletons_2.zip` file. ☐

Splitting a job in sections

5 Write an OpenMP program with 2 threads in which, given an array of N integers, one thread calculates the sum of its elements while, **simultaneously**, the second thread calculates the product. □