TP - Shared-memory programming: OpenMP (I)

ING2-GSI-MI – Architecture et programmation parallèle

Academic year 2023–2024

OpenMP

Note : In an OpenMP program we can measure its execution time per thread by calling the `omp_get_wtime()` function:

```
double start = omp_get_wtime();
// My code
double end = omp_get_wtime();
printf("Execution time = %f \n", end - start);
```

1. Try the C codes from the course examples (`codes_OpenMP-1.zip` file and make sure that they work as intended. Compile them using the following line:
`gcc -fopenmp -Wall -Wextra -o myprogram myprogram.c` ☐

2. Write an OpenMP code that, using 4 threads, finds the maximum value in an array of doubles. Use the `02-exercise.c` code in the `skeletons.zip` file. ☐

3. We have a matrix of doubles, A (size $M\,rows \times N\,columns$) and a vector X (size N). Write a parallel code in OpenMP to calculate the matrix-vector product $A * X = Y$. Create as many threads as rows in the matrix so that *Thread 0* performs the computation for *row 0*, *Thread 1* for *row 1*, and so on. Start by writing the sequential code and modify it later to add the parallelism. Measure and compare the execution times of the sequential and parallel versions. Use the `03-mvp_seq.c` code in the `skeletons.zip` file. ☐

4 In the previous exercise, modify the code to do the matrix-vector product with only two threads, so that each thread handles half of the matrix. That is, *Thread 0* performs the computation for the first $M/2$ rows of the matrix, and *Thread 1* performs the computation for the rest.

1. Does the execution time improve compared to the previous exercise?
2. Choose a value of M big enough. Compare the sequential and parallel versions and measure the speedup for 2, 4, 8, 32 and 64 threads. As sequential version we typically use the parallel version with 1 thread.

□

5 It is well known that π can be approximated with the formula:

$$\int_0^1 \frac{4}{(1+x^2)} dx = \pi \tag{1}$$

In a discrete way, we can calculate an approximation as an addition of rectangles:

$$\sum_{i=0}^{N} F(x_i) \triangle x \approx \pi \tag{2}$$

where each rectangle has a width of $\triangle x$ and a height of $F(x_i) = \frac{4}{(1+x_i^2)}$ in the middle of the i interval. Write a sequential code to calculate π. Then, write the parallel version. □