

OpenMP Usage and Code Optimization

Pamela Salazar

May 2023

The provided serial PI program has a for loop that can be parallelized. We used OpenMP to explore different ways of make this:

- Usage for reduction and private OpenMP clauses.
- Usage for teams and distributed OpenMP clauses.
- Usage for num_threads, single and reduction OpenMP clauses.

The results for the experiments are shown in the figure 1. The figure 2 shows the execution time vs the thread's number, in this case our ideal threads number is 3.

```
pamela@pamela-Inspiron-5379:~/pamela_hpec/Homework2/source$ ./pi
pi with 100000000 steps is 3.141593 in 0.138698 seconds
pamela@pamela-Inspiron-5379:~/pamela_hpec/Homework2/source$ ./pi_omp_private
pi with 100000000 steps is 3.141593 in 0.046024 seconds
pamela@pamela-Inspiron-5379:~/pamela_hpec/Homework2/source$ ./pi_omp_teams
pi with 100000000 steps is 3.141593 in 0.133901 seconds
pamela@pamela-Inspiron-5379:~/pamela_hpec/Homework2/source$ ./pi_omp_threads
num_threads = 1
pi with 100000000 steps is 3.141593 in 0.143272 seconds
num_threads = 2
pi with 100000000 steps is 3.141593 in 0.063103 seconds
num_threads = 3
pi with 100000000 steps is 3.141593 in 0.046345 seconds
num_threads = 4
pi with 100000000 steps is 3.141593 in 0.048781 seconds
pamela@pamela-Inspiron-5379:~/pamela_hpec/Homework2/source$
```

Figure 1: Output of the original code and the parallelized codes

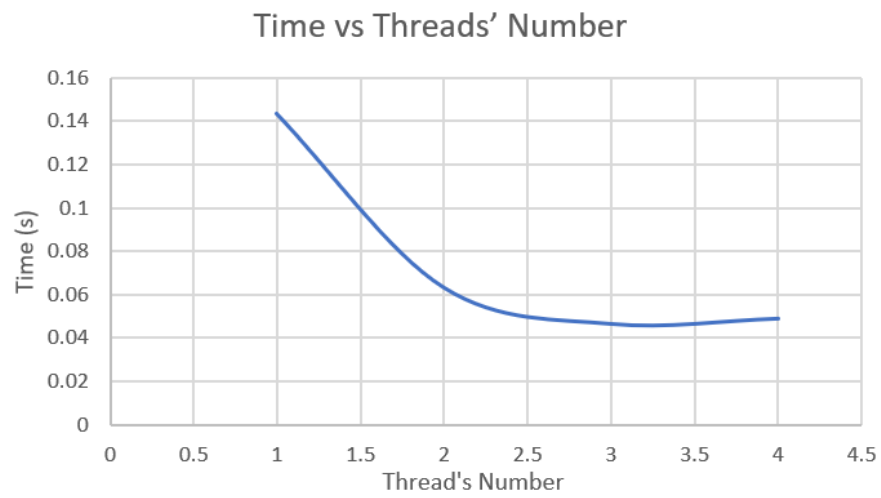


Figure 2: Time vs thread's number