

Simple OpenMP Experiment (Project # 0)

Mazen Alotaibi

Tell what machine you ran this on

I am running on my PC:

- CPU: **i5-4690K CPU**
- Operating System: **Arch Linux**
- GPU: **NVIDIA GeForce GTX 760**

To compile program, run `./runPro` and the output of the program would be in `output.txt`.

I have made the **array size** and **number of tries** to be fixed at **1e7** and **1000** respectively.

`output.txt`:

```
_NUMT(Number of Threads) = 1
|
.---_NUMS(Number of subdivisions) = 1
    Peak Performance =    461.52 MegaMults/Sec
Average Performance =    441.82 MegaMults/Sec
Average Elapsed Time = 22663.52 microseconds
-----

_NUMT(Number of Threads) = 4
|
.---_NUMS(Number of subdivisions) = 1
    Peak Performance =   1746.83 MegaMults/Sec
Average Performance =   1524.71 MegaMults/Sec
Average Elapsed Time =  6710.25 microseconds
-----

                        Speedup =    3.38
Parallel Fraction =    0.94
```

What performance results did you get?

For performance results, I have got a *peak performance*, an *average performance*, and an *average elapsed time* when using one thread, **461.52 Mega-Multiplies per Second**, **441.82 Mega-Multiplies per Second**, and **22,663.52 microseconds** respectively, and I have got a *peak performance*, an *average performance*, and *average elapsed time* when using 4 threads, **1,746.83 Mega-Multiplies per Second**, **1,524.71 Mega-Multiplies per Second**, and **6,710.25 microseconds** respectively.

What was your 4-thread-to-one-thread speedup?

For *speedup*, I have got **3.38**.

Why do you think it is behaving this way?

The speedup ratio is greater than 1 due to the average elapsed time when using one thread is greater than the average elapsed time when using four threads, which means that **when using four threads it takes less time to process compare to using one thread to do the same task.**

What was your Parallel Fraction, F_p ?

For *Parallel Fraction*, I have got **0.94**.