

CS 475/575 -- Spring Quarter 2017

Project #0

Simple OpenMP Experiment

Professor Mike Bailey

Xiao Tan

In this project, I ran this program in the server (flip.engr.oregonstate.edu), which belongs to OSU, and I got the following results while I set subdivision as 2 million, and number of tries as 50 times.

First result

```
flip2 ~/CS575/p0 168% ./proj
Using 4 threads
    Peak Performance = 1241.25 MegaMults/Sec
    Average Performance = 1229.68 MegaMults/Sec
Using 1 threads
    Peak Performance = 327.88 MegaMults/Sec
    Average Performance = 327.14 MegaMults/Sec
flip2 ~/CS575/p0 169%
```

Second result

```
Using 4 threads
    Peak Performance = 1247.40 MegaMults/Sec
    Average Performance = 1235.13 MegaMults/Sec
Using 1 threads
    Peak Performance = 331.20 MegaMults/Sec
    Average Performance = 324.85 MegaMults/Sec
flip2 ~/CS575/p0 184%
```

Third result

```
Using 4 threads
    Peak Performance = 1248.61 MegaMults/Sec
    Average Performance = 1234.49 MegaMults/Sec
Using 1 threads
    Peak Performance = 332.83 MegaMults/Sec
    Average Performance = 331.46 MegaMults/Sec
flip2 ~/CS575/p0 185%
```

According to equation, here I use average performance of first result to calculate, 4-thread-to-one-thread speedup is $S = 1229.68 / 327.14 = 3.76$.

Because it is executing $C[i] = A[i] * B[i]$, they will not affect each other, this program works like splitting one work to four independent parts to do, therefore, we can get these results which four threads will 4 times to one thread.

According to Parallel Fraction is $F_p = (4. / 3,) * (1. - (1. / S))$, I got the F_p is 0.98.

This is a super easy project. Mostly, I got right results in this program, however, I got some bad results sometimes, this is maybe machine I used problem, sometimes, I got 4-thread performance is twice to 1-thread performance.