Project #0

Simple OpenMP Experiment

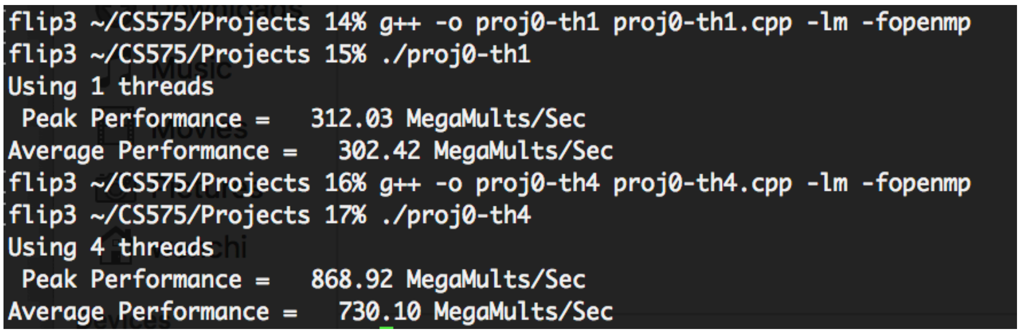
Name: Chi Wen

---

1. Tell what machine you ran this on

   I ran on school server, filp3.

2. What performance results did you get?

```
flip3 ~/CS575/Projects 14% g++ -o proj0-th1 proj0-th1.cpp -lm -fopenmp
flip3 ~/CS575/Projects 15% ./proj0-th1
Using 1 threads
 Peak Performance =   312.03 MegaMults/Sec
Average Performance =   302.42 MegaMults/Sec
flip3 ~/CS575/Projects 16% g++ -o proj0-th4 proj0-th4.cpp -lm -fopenmp
flip3 ~/CS575/Projects 17% ./proj0-th4
Using 4 threads
 Peak Performance =   868.92 MegaMults/Sec
Average Performance =   730.10 MegaMults/Sec
```

   1 threads:

   Peak performance = 312.03 Mega-Multiplies per Second

   4 threads:

   Peak performance = 868.92 Mega-Multiplies per Second

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

   S = (Execution time with one thread) / (Execution time with four threads)
   = (1/312.03)/(1/868.92)
   =(0.00320482)/(0.00115085)
   =2.77991066

4. Why do you think it is behaving this way?

   The code will start with executing in a single thread. When "#pragma omp parallel for" divides the for-loop to pass up among those 4 threads, each thread has different finish time. The implied barrier at the end will make each thread waits until all threads are done, so the finish time will be effect by the slowest thread.

5. What was your Parallel Fraction, Fp?

   Fp = 0.85370161