Project #0

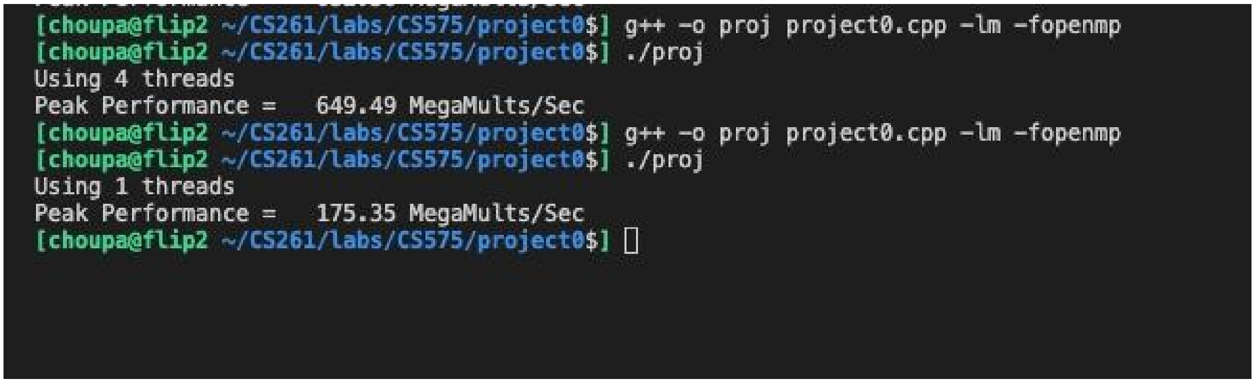
Simple OpenMP Experiment

PangFa Chou | choupa@oregonstate.edu

The program was tested on a Linux server. SIZE was set as 512*512, and NUMTRIES was set as 1000. As Figure 1.1 shows, I get 175.35 MegaMults/Sec for 1 thread, and 649.49 MegaMults/Sec for 4 threads. Therefore,

1. S(speedup) = 649.49 / 175.35 = 3.7039635
2. Fp(Parallel Fraction) = (4. / 3.)*(1. - (1. / **S**)) = 0.973358582

The 4-thread-to-one-thread speedup(S = 3.7039635) is less than 4.0. In my opinion, the perfect result for splitting 1 work into N parallel processors should be done in 1/N time. It means that the speedup is N. However, in the real world, there are many unpredictable factors that affect performance. Therefore, the speedup will never be an exact N.



```
[choupa@flip2 ~/CS261/labs/CS575/project0$] g++ -o proj project0.cpp -lm -fopenmp
[choupa@flip2 ~/CS261/labs/CS575/project0$] ./proj
Using 4 threads
Peak Performance =   649.49 MegaMults/Sec
[choupa@flip2 ~/CS261/labs/CS575/project0$] g++ -o proj project0.cpp -lm -fopenmp
[choupa@flip2 ~/CS261/labs/CS575/project0$] ./proj
Using 1 threads
Peak Performance =   175.35 MegaMults/Sec
[choupa@flip2 ~/CS261/labs/CS575/project0$] ▯
```

(Figure 1.1)