Name: Swathi Gangi
Mail: gangsw@oregonstate.edu


1. **Tell what machine you ran this on**

Ans: I ran the program on flip server (Oregon State University Linux server)

2. **What performance results did you get?**

Ans:

```
~ — gangsw@flip3:~ — ssh gangsw@access.engr.oregonstate.edu
flip3 ~ 108$ ./loop.bash
OpenMP version 201511 is supported here
For 1 threads, Peak Performance =   482.89 MegaMults/Sec
OpenMP version 201511 is supported here
For 4 threads, Peak Performance =  1709.00 MegaMults/Sec
flip3 ~ 109$
```

1 thread: 482.89 MegaMults/Sec
4 threads: 1709.00 MegaMults/Sec

3. **What was your 1-thread-to-4-thread speedup?**

Ans: S = 1709.00 /482.89 = **3.53**

4. **Your 1-thread-to-4-thread speedup should be less than 4.0. Why do you think it is this way?**

   Ans: The speedup is less than 4.0 because there's some overhead in managing threads and not all parts of the program can run in parallel. Also, when multiple threads access memory at the same time, they can slow each other down. So, perfect scaling isn't possible but a speedup around 3.5 is still a strong result.

5. **What was your Parallel Fraction, Fp? (Hint: it should be less than 1.0, but not much less.)**

Ans: Fp = 4. / 3. x (1. – (1. /3.53))

= 4. / 3. x (1. – 0.283)

= 4. / 3. x 0.717

= **0.956**

Parallel Fraction (Fp) ≈ 0.96, which means ~96% of the work was parallelizable