Lab report 3

Nhan Dao

October 28, 2020

1. MONTE CARLO METHOD FOR APPROXIMATING THE VALUE OF π IN OPENMP

a. Compile and run the template code with a single thread, and thus verify that the serial version is correct.

```
nhandao@nhandao-XPS-15-9560 ~/OneDrive/Curtin University/FifthYear/Concurrent
_Systems/concurrent-lab-3/code  master ./lab3part1 1 1000000000
Estimated pi: 3.141608e+00
Time = 2.77554e+01 seconds
```

Figure 1.1: Terminal output of execution of template code with a single thread

b. Parallelize the code for multi-threaded execution by ONLY adding OpenMP directives

```
1  # pragma omp parallel num_threads(thread_count) // OMP parallel
2     {
3        int my_rank = omp_get_thread_num();
4        unsigned seed = my_rank + 1;
5        long long int toss;
6        double x, y, distance_squared;
7
8  #     pragma omp for  reduction(+ : number_in_circle) // OMP for
9        for(toss = 0; toss < number_of_tosses; toss++) {
10          x = 2*my_drand(&seed) - 1;
11          y = 2*my_drand(&seed) - 1;
12          distance_squared = x*x + y*y;
13          if (distance_squared <= 1) number_in_circle++;
14       }
15    }
```

Listing 1: Parallizing the code for multi-threaded execution (line 34-48 in lab3part1.c)

b.1. At what point in the code new threads are forked and joined?

The code is serial up until *pragma*. At pragma, multiple threads are forked, line 1 in listing 1. After the structured block (line 2 - 15), the slaves are joined with the master thread and execution resumes on the master thread.

b.2. Which part of the code is executed by which threads?

The total number of threads are comprised of one main thread, and the remainder are slave threads. The main thread will run all code outside of the parallel enivornment. Master and slaves thread will run the section of code assigned by the OpenMP directives - line 2 - 15.

b.3. Where barrires if any in terms of thread synchronization would be encountered, and hence how the code achives parallelization

write something