COSC 420: High Performance Computing

Programming Assignment 1: Estimate of π by "Monte Carlo" Method

1) Understand estimating of π by "Monte Carlo" Method (A pseudo-code is given below)

Suppose we toss darts randomly at a square dartboard, whose bullseye is at the origin, and whose sides are 2 feet in length. Suppose also that there's a circle inscribed in the square dartboard. The radius of the circle is 1 foot, and it's area is π square feet. If the points that are hit by the darts are uniformly distributed (and we always hit the square), then the number of darts that hit inside the circle should approximately satisfy the equation

$$\frac{\text{number in circle}}{\text{total number of tosses}} = \frac{\pi}{4},$$

since the ratio of the area of the circle to the area of the square is $\pi/4$.

We can use this formula to estimate the value of π with a random number generator:

```
number_in_circle = 0;
for (toss = 0; toss < number_of_tosses; toss++) {
   x = random double between −1 and 1;
   y = random double between −1 and 1;
   distance_squared = x∗x + y∗y;
   if (distance_squared <= 1) number_in_circle++;
}
pi_estimate = 4∗number_in_circle/((double) number_of_tosses);
```

This is called a "Monte Carlo" method, since it uses randomness (the dart tosses).

2) Implement it using multithreading (such Pthreads or java threads)

3) Produce two charts that show the speedups and efficiencies of your program for various values of number of tosses and number of threads. And explain the curve treads shown in your charts.

4) Is your program scalable (strongly or weekly)? Explain it.