Assignment 4 - OpenMP

Parallel and distributed computing

This is Book programming assignment 5.2

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 are 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

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

since the ratio of the area of the circle to the area of the square is π = 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).
Write an OpenMP program that uses a Monte Carlo method to estimate π. Read in the total number of tosses before forking any threads. Use a reduction clause to find the total number of darts hitting inside the circle. Print the result after joining all the threads. You may want to use **long long int**s for the number of hits in the circle and the number of tosses, since both may have to be very large to get a reasonable estimate of π.