EG4387 Parallel Programming/EG6370 Parallel Processing
Project in lieu of the final Exam
Spring 2020

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 its 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

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

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);
```

Requirements:

1) Write an MPI program that uses the method described above to estimate π. Process 0 should read in the total number of tosses and broadcast it to the other processes. Use *MPI_Reduce* to find the global sum of the local variable *number_in_circle*, and have process 0 print the result. You may want to use long long ints 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. Your program should be able to handle any number of tosses even if they are not divisible by the number of allocated processes. You program should be executed using the following command line:

   mpirun -np <number_of_processes> ./ProgramName <number_of_tosses>

   If the user does not run the program using the above command line format, an error message should be displayed reminding him of the correct format of the command line before exiting the program.

2) Repeat question 1, but this time use OpenMP

Due date: Tuesday, May 5, 2020