Name : Ansari M.Saeem M.Saleem

Uid : 2019430001

Subject : NAD

Expt no : 4

Aim : High performance computation of Pi.

## Aim:

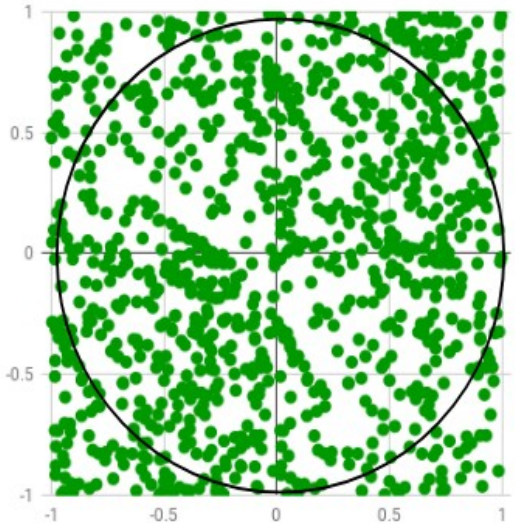
High performance computation of Pi.

## Objectives:

- To estimate the value of PI with parallel computation using openMP.

## Theory:

The idea is to simulate random (x, y) points in a 2-D plane with domain as a square of side 1 unit. Imagine a circle inside the same domain with same diameter and inscribed into the square. We then calculate the ratio of number points that lied inside the circle and total number of generated points. Refer to the image below:




If a circle of radius R is inscribed inside a square with side length 2R, then the area of the circle will be pi*R^2 and the area of the square will be (2R)^2. So the ratio of the area of the circle to the area of the square will be pi/4.

This means that, if you pick N points at random inside the square, approximately N*pi/4 of those points should fall inside the circle.

The idea is to picks points at random inside the square. It then checks to see if the point is inside the circle (it knows it's inside the circle if x^2 + y^2 < R^2, where x and y are the coordinates of the point and R is the radius of the circle). The program keeps track of how many points it's picked so far (N) and how many of those points fell inside the circle (M). Pi is then approximated as follows:

$$Pi = (4*M) / N$$

**Parallel Algorithm for estimating value of pi:**

1. Set the number of threads for the program.
2. The block of code beneath the #pragma omp parallel is run by each individual thread.
3. Each thread needs different seed variables for it to estimate properly.
4. A local copy of the count is kept in each thread. This is to avoid race conditions of the variable.
5. All local copies of count added together and stored in master thread.Reduction (+: count) tells the compiler to add the count variable in the end.

**Results:**

```
HPC-LAB@ubuntu:~/Exp4$ gcc -fopenmp Exp4.c -o exp4
HPC-LAB@ubuntu:~/Exp4$ ./exp4
Running thread 1
Running thread 0
Running thread 7
Running thread 6
Running thread 8
Running thread 5
Running thread 9
Running thread 10
Running thread 4
Running thread 11
Running thread 12
Running thread 3
Running thread 13
Running thread 14
Running thread 2
Count = 7990, Samples = 10000, Estimation of pi= 3.19600
```

```
HPC-LAB@ubuntu:~/Exp4$ ./exp4
Running thread 1
Running thread 0
Running thread 6
Running thread 7
Running thread 5
Running thread 8
Running thread 4
Running thread 9
Running thread 10
Running thread 3
Running thread 11
Running thread 12
Running thread 13
Running thread 2
Running thread 14
Count = 7870, Samples = 10000, Estimation of pi= 3.14800
HPC-LAB@ubuntu:~/Exp4$
HPC-LAB@ubuntu:~/Exp4$ ./exp4
Running thread 1
Running thread 6
Running thread 7
Running thread 8
Running thread 5
Running thread 9
Running thread 10
Running thread 4
Running thread 0
Running thread 11
Running thread 3
Running thread 2
Running thread 12
Running thread 14
Running thread 13
Count = 7825, Samples = 10000, Estimation of pi= 3.13000
```

**Conclusion:** Thus, we have successfully estimated the value of PI using parallel computation.