## Indian Institute of Information Technology Surat

****

# Lab Report on

# High Performance Computing (CS 602) Practical

**Submitted by**

### [RAHUL KUMAR SINGH] (UI21CS44)

**Course Faculty**

### Dr. Sachin D. Patil

## Department of Computer Science and Engineering

## Indian Institute of Information Technology Surat

## Gujarat-394190, India

**Jan-2024**

## Lab No: 3

**Aim: Write a program that generates np random points using any random function, computes the ratio q that approximates the probability P, and uses it to estimate the value of π.**

**Description:** In this task, we will estimate the value of π using the Monte Carlo method. Given a probability P, the Monte Carlo method relies on the generation of random samples (events) to compute a numerical approximation of P.

Consider a circle inscribed in a square. The method simply consists in generating random points (x, y) within the square range. Given the probability that the points lie within the circle, and the actual number of generated random points that do so, we can estimate π.

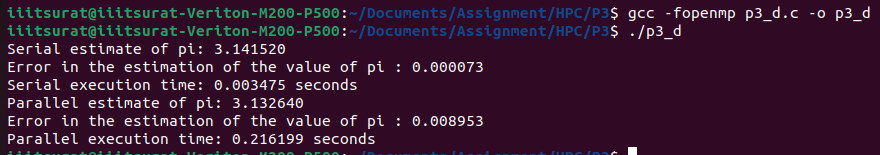
First, what is the probability of random points ending up within the area of the circle? The answer is to find the relationship between the geometry of the square and the circle. The area of the square is 4\*R^2 and the area of the circle is πR^2. Now, the ratio of the area of the circle over the area of the square is π/4. That is, the probability that a random point within the square lies also within the circle is P =π/4, and thus π = 4\*P.

In this task, we will generate np (1,00, 000) random points using any random function, compute the ratio q that approximates the probability P, and use it to estimate the value of π while maintaining parallel OpenMP construct.

## Source Code:



## Output:

****

## Conclusion:

* It can be observed how close the estimation is to the known value, acknowledging the stochastic nature of the method.
* Careful consideration was given to shared and private variables. Variables like count were marked for reduction to ensure thread safety, while x, y and z were designated as private to prevent data races.
* The program was parallelized using OpenMP to expedite the generation of random points.
* The Monte Carlo method is a probabilistic numerical technique that uses random sampling to estimate mathematical values.