

## Homework 4: Parallel and Distributed Computing

**DUE: See exact time on blackboard**

**ICSI 520**

### Problem

There are three parts in this homework

In this homework you will implement a serial and a parallel implementation to calculate the value of  $\pi$  as described in the following sections

- For the parallel version of Monte Carlo simulation, you will create two binaries
  - Part 1: One where you only use pthreads
  - Part 2: Other using only openmp
- Part 3: Convert homework 1's part 3 to use omp instead of pthreads. (use same tables from hw1)

### Serial and Parallel Implementations

You will do both serial and parallel implementations

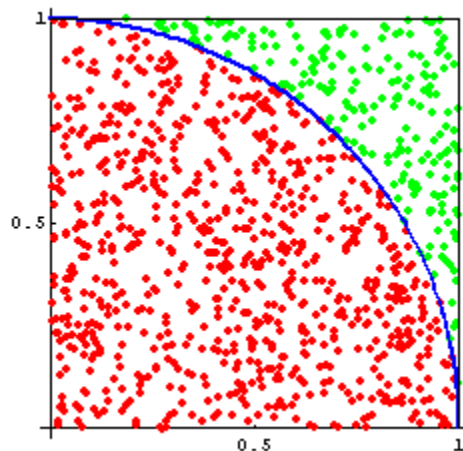
**Use timing code to measure timings for this code.**

### Description

#### Monte Carlo simulation

Monte Carlo methods can be thought of as statistical simulation methods that utilize a sequences of random numbers to perform the simulation. In a typical process one computes the number of points in a set  $A$  that lies inside box  $R$ . The ratio of the number of points that fall inside  $A$  to the total number of points tried is equal to the ratio of the two areas. The accuracy of the ratio  $p$  depends on the number of points used, with more points leading to a more accurate value.

#### Monte Carlo simulation of $\pi$



- Consider a circle inscribed in a unit square. Given that the circle and the square have a ratio of areas that is  $p = \pi/4$ , the value of  $\pi$  can be approximated using a Monte Carlo method
  - Draw a square on the ground, then inscribe a circle within it.
  - Uniformly scatter some objects of uniform size (grains of rice or sand) over the square.
  - Count the number of objects inside the circle and the total number of objects.
  - The ratio of the two counts is an estimate of the ratio of the two areas, which is  $\pi/4$  or  $p$ . Multiply the result by 4 to estimate  $\pi$ .

A simple Monte Carlo simulation to approximate the value of  $\pi$  could involve randomly selecting points  $\{(x_i, y_i)\}_{i=1}^n$  in the unit square and determining the ratio  $\rho = \frac{m}{n}$ , where  $m$  is number of points that satisfy  $x_i^2 + y_i^2 \leq 1$ .

$m$  = number of points inside the circle

$n$  = total points

In a typical simulation of sample size  $n = 1000$  if there were 787 points satisfying  $x_i^2 + y_i^2 \leq 1$

$$\rho = \frac{m}{n} = \frac{787}{1000} = 0.787 \text{ and } \pi \approx \rho * 4 = 0.787 * 4 = 3.148$$

### Results openmp

n	Number of threads (p)	Serial execution time	Parallel Execution time	Speedup
700	<decide the best number>			
1400				
2500				
5000				
10000				
<add more if needed>				

## Results pthreads

n	Number of threads (p)	Serial execution time	Parallel Execution time	Speedup
700	<decide the best number>			
1400				
2500				
5000				
10000				
<add more if needed>				

## Grading

Serial Implementation	10%
Parallel Implementation	25%
Correct execution Parallel	30%
Correct serial execution	10%
Speedup (greater than 1)	10%
Code clarity	5%
Timing competition	10%

## Timing competition

You are competing with your colleagues for these 10% of the points. Lowest timing gets 10 full points. Highest timing gets 0.

**Do not make up the speedup or the timing tables. You will get a 0 for all homework assignments (including previous).**

## Code clarity includes

- Use of variable names
- Function naming
- File names
- Formatting
- Comments where needed

Late submissions are subject to at least 20% grade deduction.

**Reminder:** If one or any of your binaries don't execute on student.rit.albany.edu, you get 0 points for this and any other homework where that happens.

**Submission:**

Your submission should include the following:

1. **Makefile** that builds all required binaries:
  - a. Part\_N\_type.out where N is the homework part and type can be serial/parallel
2. Serial code source file(s)
3. Parallel code source file(s)
4. Binary for both serial and parallel implementations
5. Document (word or excel) detailing the tabular results

All of the above should be zipped up into one file and uploaded on blackboard

If you are doing the bonus, add a comment in your code about it. (if applicable)

**Naming convention of your zipped file – If you do not follow this, you get zero points:**

FirstName\_LastName\_Homework\_<n>.zip

*Where n is the homework number.*