COMP 605: Introduction to Parallel Computing
Homework 3: Calculating $\pi$ with MPI

Mary Thomas

Department of Computer Science
Computational Science Research Center (CSRC)
San Diego State University (SDSU)

Due: 03/17/15
Posted: 03/04/15
Updated: 03/01/15
# Table of Contents

1. HW 3a: Using Monte Carlo Methods to Estimate $\pi$ (25 points)
2. HW 3b: Using Numerical Integration to Estimate $\pi$ (25 points)
3. Additional/General Instructions
Monte Carlo Methods for Integration

\[ I = \int \ldots \int_{\Omega} f(x_1, x_2, \ldots, x_n) \, dx_1 \ldots dx_n \]

Where \( \Omega \) is domain, which has a volume \( V \) given by:

\[ V = \int_{\Omega} d\mathbf{x} \]

MC method randomly samples \( N \) points:

\[ x_1, x_2, \ldots, x_n \in \Omega \]

to estimate the integral

\[ I \sim V \frac{1}{N} f(\mathbf{x}_i) \]

- Monte Carlo numerically computes multidimensional definite integrals
- Monte Carlo uses a non-deterministic approach: for the same input, an algorithm can exhibit different behaviors on different runs.
- Randomly chooses points at which the integrand is evaluated.
- Different sampling methods:
  - Uniform (a uniform distribution)
  - Stratified (population)
  - Importance (particular distribution)

Fig. Ref: http://en.wikipedia.org/wiki/Monte_Carlo_integration
Computing the area of a shape by random sampling: a simple way to estimate $\pi$

We estimate $\pi$ with the integral:

$$\pi = \int \int \left( \sqrt{x^2 + y^2} < 1 \right) P(x, y) \, dx \, dy$$

Where $P(x, y)$ is the probability that a point $(x, y)$ lies in a circle:

$$P(x, y) = \begin{cases} 
1 & 0 < x < 1 \text{ and } 0 < y < 1 \\
0 & \text{otherwise}
\end{cases}$$

Assume a set of darts are thrown with uniform probability of landing at any point in rectangular area. For $N$ large enough, the ratio of board hits to total throws will approach

$$\text{Ratio} \sim \frac{\text{Area(circle)}}{\text{Area(square)}} = \frac{\pi r^2}{2r^2}$$

which is about 0.79
Computing PI

Area of a Circle = $\pi r^2$

$= \frac{\pi d^2}{4}$

Circumference = $\pi \times$ diameter

$= \pi \times 2 \times r$

Area of Quarter Circle = $\pi r^2 / 4$

Point is in quarter circle if $x^2 + y^2 < 1$

$(0,0)$

Area of Square = $r^2$

$\pi = \frac{4 \times \# \text{ of Points in Qtr Circle}}{\text{(Total \# of Points)}}$
HW 3a: Using Monte Carlo Methods to Estimate $\pi$

Pacheco, Programming Assignment 4.2 (page 206)

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 area is $\pi$ 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:

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

since the ratio of the area of the circle to the area of the square is $\pi/4$.// We can use this formula to estimate the value of $\pi$ with a random number generator:

```c
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).
HW 3a: Additional Instructions

Write your own MPI program that uses a Monte Carlo method to estimate $\pi$.

- The main thread should read in the total number of tosses and print the estimate.
- You can use point-to-point or collective communications.
- Vary the number of darts (points) used: $N_{\text{darts}} = 10^n$, where $n = 3, 4, 5, N_{\text{max}}$
- You may want to use \texttt{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 $\pi$.
- Vary the number of PEs: $p = [1, 2, 4, 8, 16]$
- Time the job runs.
- You must run jobs on the queue.
HW 3a: Instructions (cont.)

- Find a reference value for $\pi$ to the limits of a **double precision** number.
- Estimate $\pi$ to the limits of a **double precision** number.
- Calculate the value for $\pi$ and the error of your estimate as a function of the number or areas used.
- Calculate the error of your estimate: $Err = \pi_{\text{ref}} - \pi_{\text{measured}}$
- Create summary tables of your test data and results.
- Plot the error as a function of the number of processors and number of points.
- Plot the runtime as a function of the number of processors and number of points.
- Reference key sources of information (Pacheco, lectures, Web, ).
Trapezoid Rule for Numerical Integration of a function

Solve the Integral: \[ \int_a^b F(x) \, dx \]

Where \( F(x) \) can be any function of \( x \): \( f(x^2), f(x^3), \ldots \)

Ref: Pacheco (2011), Ch3.
Using Numerical Integration to Estimate $\pi$

- Integral representation for $\pi$
  \[ \int_0^1 dx \frac{4}{1+x^2} = \pi \]

- Discretize the problem:
  \[ \Delta = \frac{1}{N} : \text{step} = \frac{1}{N_{\text{areas}}} \]
  \[ x_i = (i + 0.5) \]
  \[ \Delta(i = 0, \ldots, N_{\text{areas}} - 1) \]
  \[ \sum_{i=0}^{N-1} \frac{4}{1+x_i^2} \Delta \approx \pi \]

$\pi$ Formulae: http://en.wikipedia.org/wiki/Approximations_of_pi
Image: http://cacs.usc.edu/education/cs596/mpi-pi.pdf
Serial Code

```c
#include <stdio.h>
#define NAREA 10000000

void main() {
    int i; double step,x,sum=0.0,pi;
    step = 1.0/NAREA;
    for (i=0; i<NAREA; i++) {
        x = (i+0.5)*step;
        sum += 4.0/(1.0+x*x);
    }
    pi = sum*step;
    printf(PI = %f
,pi);
}
```
HW 3b: Instructions

- Use the method of *numerical integration* to estimate the value of $\pi$
- Design and write a parallel MPI version to estimate $\pi$ using the formula above.
- Explain how the formula works.
- See the *Trap* example discussed in Pacheco 2011, Ch 3.
- You can use point-to-point or collective communications.
- You must run jobs on the queue.
- Vary the number of areas used: $N_{\text{areas}} = 10^n$, where $n = 1, 2, 3, \ldots, N_{\text{max}}$
- Vary the number of PEs: $np = [1, 2, 4, 8, 16]$
- Time the job runs.
Find a reference value for $\pi$ to the limits of a double precision number.

Estimate $\pi$ to the limits of a double precision number.

Calculate the value for $\pi$ and the error of your estimate as a function of the number or areas used.

Calculate the error of your estimate: $Err = \pi_{\text{ref}} - \pi_{\text{measured}}$

Create summary tables of your test data and results.

Plot the error as a function of the number of processors and number of points.

Plot the runtime as a function of the number of processors and number of points.
All jobs run as batch jobs

Put homework into a directory: e.g.

\[ \text{HOME/ < your_username > /hw/hw3/hw3a} \]

include the source code(s), compiled binaries

Write a simple report (this can be TEXT, Word, PDF Doc).

**Turn in hard copy (condensed/minimal number of pages) at start of class.**

Reference key sources of information (Pacheco, lectures, Web, ).

---

Once the submission timeline has closed
DO NOT CHANGE THE FILE TIMESTAMPS!