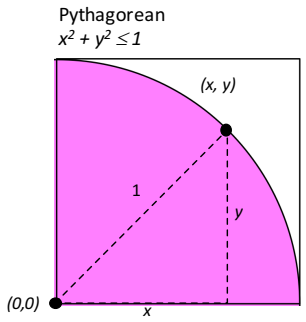
California State Polytechnic University, Pomona
Computer Science 3700-01, Spring 2024

Project 01: Assigned February 20, 2024 – Due February 27, 2024 @5:30pm

Objective: Monte Carlo using MPI

Monte Carlo method computes the number of points in a set **A** that lies inside box **R**. The ratio of the number of points that all inside **A** to the total number of points tried is equal to the ratio of the two areas. The accuracy of the ratio depends on the number of points used, with more points leading to a more accurate value.

Develop a MPI program in C/C++ or Python to estimate the value of PI using the Monte Carlo method as discussed in class.



Compute the ratio using Monte Carlo method, and multiple the ratio by 4 to obtain the value of PI

Random Number Generation

C/C++

```
#include <math.h>
srand((unsigned)(myid));
x = ((double)rand()) / ((double)RAND_MAX);
y = ((double)rand()) / ((double)RAND_MAX);
```

Python

```
import random
random.seed(myid)
x = random.uniform(0,1)
y = random.uniform(0,1)
```

Required SBATCH File for testing Python program

```
#!/bin/bash
#SBATCH --job-name=PI_MPI         # Job name
#SBATCH --output=PI_MPI_%j.log    # Log file name
#SBATCH --partition=compute       # Use computing cluster
#SBATCH --mem=1gb                 # Job memory request
#SBATCH --nodes=4                 # Number of computing nodes
#SBATCH --time=00:02:00           # Time limit HH:MM:SS

. /etc/profile.d/modules.sh

module load openmpi/2.1.2
module load python/3/mpi4py/3.0.0

/opt/openmpi-2.1.2/bin/mpirun python3.4 pi_mpi.py
```

Required SBATCH File for testing C/C++ program

```
#!/bin/bash
#SBATCH --job-name=PI_MPI         # Job name
#SBATCH --output=PI_MPI_%j.log    # Log file name
#SBATCH --partition=compute       # Use computing cluster
#SBATCH --mem=1gb                 # Job memory request
#SBATCH --nodes=4                 # Number of computing nodes
#SBATCH --time=00:02:00           # Time limit HH:MM:SS

. /etc/profile.d/modules.sh

module load openmpi/2.1.2

/opt/openmpi-2.1.2/bin/mpirun ./pi_mpi
```

Performance Report

You need to execute your program multiple times using the following configurations and collect output from your program according to the `proj01_log` excel spreadsheet. The ESTIMATED PI is the value of PI your program estimated. The DELTA is the difference from the actual PI constant up to four decimal places (C/C++ `M_PI` in `math.h/cmath`, Python `math.pi` in `math` library). The TIME is the elapsed execution time in seconds up to six decimal places.

Project Submission

1. Create a zip file of your `cs3700_proj01` directory and rename the zip file to `cs3700_proj01_`*`your-bronconame`*`.zip` (e.g., `cs3700_proj01_thuang.zip`)
2. Use secure copy (e.g., `scp` or WinSCP) to copy your zip to your local computer
3. Verify your zip file contains the right directory and all your lab files by unzipping it before submission.
4. **NOTE: It is important to name your directory and files exactly according to the instructions above, all in lowercase. Your zip file must unzip into a directory with your files inside. If I am unable to unzip or locate your files inside the unzipped directory, I won't grade your assignment.** For example, if your bronco-name is `thuang`, your zip file submission should look as follow.

```
$ ls cs3700_proj01_thuang.zip
cs3700_proj01_thuang.zip

$ unzip cs3700_proj01_thuang.zip
Archive:  cs3700_proj01_thuang.zip
   creating: cs3700_proj01/
 extracting: cs3700_proj01/monty_pi.py
 extracting: cs3700_proj01/run_monty_pi.sh
 extracting: cs3700_proj01/RUN_MONTY_PI_16821.log
 extracting: cs3700_proj01/proj01_log.xlsx
or
Archive:  cs3700_proj01_thuang.zip
   creating: cs3700_proj01/
 extracting: cs3700_proj01/monty_pi.cpp
 extracting: cs3700_proj01/compile_monty_pi.sh
 extracting: cs3700_proj01/run_monty_pi.sh
 extracting: cs3700_proj01/COMPILE_MONTY_PI_16821.log
 extracting: cs3700_proj01/RUN_MONTY_PI_16821.log
 extracting: cs3700_proj01/proj01_log.xlsx
```

5. Verify your zip file contains the right directory and all your lab files by unzipping it before submission. Your zip file should contain `python or C/C++ source, sbatch`, and `run log` files.
6. Submit your zip file and your `proj01_log` file by upload to Canvas.
   a. Login to Canvas
   b. Click `CS 3700.01->Assignments->Projects->Proj 01`
   c. Attach the zip file and click the Submit button.