CX 4220/CSE 6220 Introduction to High Performance Computing
Spring 2024
Programming Assignment 1
Due: February 12th, 11:59 PM

1 Problem Statement

Write a parallel program in `C/C++` to estimate the value of π using the Monte Carlo method specified below. The program should take n as input, which is the number of points to be used for the estimation. The program should then generate n random points in the unit square $[0, 1] \times [0, 1]$ and count the number of points that lie inside the unit circle.

Let n be a large integer, then estimated value of π is

$$\pi \approx \frac{4 \cdot \text{number of points in the circle}}{n}$$

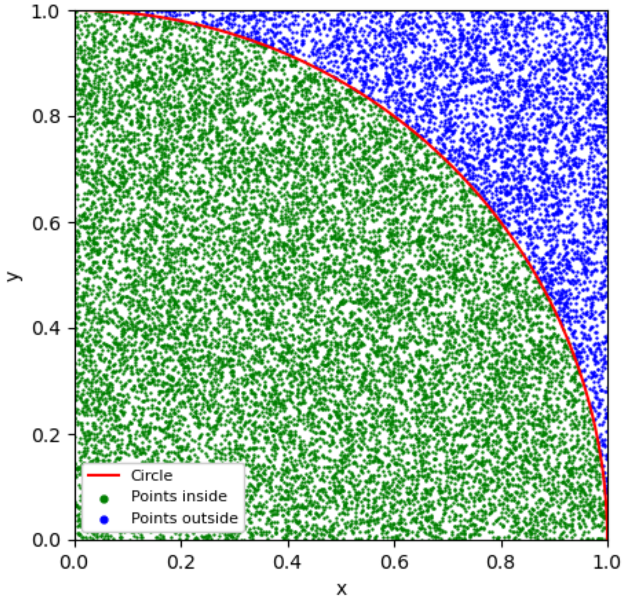

Figure 1: Estimating π using Monte Carlo method

2 Parallel Algorithm

The estimation can be easily parallelized by assigning each processor the responsibility for $\frac{n}{p}$ distinct points. You should use the following MPI functions:

- Have processor with rank 0 read n, and use `MPI_Bcast` to broadcast it to all processors.
- Use `MPI_Reduce` function to sum the number of points in the circle across all processors.

- Use MPI_Wtime to time the run-time of the program (measure on processor 0).

For the random number generator, you can use the rand() function in C/C++. However, you should make sure that the random number generator is seeded **differently** for each processor. You can use the processor rank for this purpose. For example, if the rank of the processor is i, then you can use srand(time(NULL) + i) to seed the random number generator on that processor.

3 Code framework

3.1 Input & Output Format

Your program should take n as the input using command line arguments and output the estimated value of π and the time taken to compute this value (comma-separated). For example, if the estimated value is s and the time taken is t, your output should be "s, t". (**Note:** s should be printed up to 12 decimal points). All output is done through processor with rank 0.

3.2 Deliverables

1. Create a Makefile for your program, and make sure the name of your output executable is "pi_calc". If you are not familiar with creating Makefiles, check resources below for help.

2. Write a "README.txt" briefly describing how your program works and the machine you used for generating the results.

3. For $n = 10^6$, plot a graph of run-time of the program vs. the number of processors for a few chosen values of p. This run-time should include all computations that contribute to the estimation, including any local computations and global reductions. Include your graph and observation regarding the speedup in a PDF file with name "report.pdf". **Make sure to list names of all your teammates at the very beginning of your report.**

4. Submit your a) "code.zip" in "Programming Assignment 1 - **Code**" on Gradescope. Your zip file should include all the .cpp files, Makefile and README.txt file; b) your report in "Programming Assignment 1 - **Report**" on Gradescope.

4 Resources

1. What is a Makefile and how does it work?: https://opensource.com/article/18/8/what-how-makefile

2. PACE ICE cluster guide: https://docs.pace.gatech.edu/ice_cluster/ice-guide/. Documentation for writing a PBS script: https://docs.pace.gatech.edu/software/PBS_script_guide/