Operating System-2

Programming Assignment 1: Parallel Monte Carlo technique for calculating π using Multi Threading in C++

Submission Date: 13th January 2023, 9:00 PM

Objective: The objective of this assignment is to calculate the value of π using parallel Monte Carlo technique efficiently by using multiple threads in C++.

Details: This assignment is problem 4.24 of 'Programming Problems' given in the 10th edition of the textbook (page 260 of the pdf). Please read the details given in the book carefully. Solve this problem in C++.

Input: The input to the program will be a file (named inp.txt) will contain two numbers. The first number will be n the total number of points to be used by the experiment while the second number be total of number of threads to use.

Example: inp.txt

1000000 32 // 1000000 is the total number of random points, 32 is the no. of threads

Output: The output will also be a file (named output.txt) will contain two parts: (a) Results: the time take to compute the value of π in microseconds and the value computed. (b) Log: The intermediate values computed by each thread.

Example: output.txt

Time: 800 μs

Value Computed: 3.14789

Log:

Thread1: Number of points inside the square, Number of points inside the circle
Points inside the square: point1, point2 .....
Points inside the circle: point1, point2 .....

Thread2: Number of points inside the square, Number of points inside the circle
Points inside the square: point1, point2 .....
Points inside the circle: point1, point2 .....

.
.
.

Report: As a part of this assignment, you have to prepare a report which will describe the low-level design of your program and give an analysis of its output. As a part of the report, you have to measure the performance of the algorithms. You will have to create two plots to measure the time taken:

- Plot1 - Time taken vs Number of threads: Here the y-axis will be time taken while the x-axis will be number of threads varying from 2 to 32 as 2 4 8 16 32. Fix the number of input points to be $6 * 10^6$ (6 million).
- Plot2 - Time taken vs Number of initial points: Here the y-axis will be time taken while the x-axis will be number of points $1 * 10^6$ to $5 * 10^6$. Fix the number of threads to be 32.

Note that each point in the above plots should be averaged over 5 times.

Deliverables:

1. Report describing the low-level design of your program and analysis of output and graph. This file should be named as Assgn1_Report_⟨Roll No.⟩.pdf
2. Prepare a README file that contains the instructions on how to execute your submitted file. The file should be named as Assgn1_README_⟨Roll No.⟩.txt
3. Name the source code file in the following format: Asgn1_Src_⟨Roll No.⟩.cpp
4. Combine the source code, report and README as a zip archive file and name it as Asgn1_⟨Roll No.⟩.zip. Upload this zip file.

Please see the instructions given above before uploading your file. Your assignment will NOT be evaluated if there is any deviation from the instructions posted there.

Grading Policy: The policy for grading this assignment will be

- Design as described in report and analysis of the obtained results: 50%
- Execution: 40%
- Code documentation and indentation: 10%.

Please note:

- All assignments for this course has the late submission policy of a penalty of 10% each day after the deadline till six days. After that it will not be evaluated.
- All submissions are subjected to plagiarism checks