2023-Batch-I-Set-2

Lab 4 - CSS 311 – Parallel and Distributed Computing

OpenMP – Synchronization Controls with Critical Section

----

1. Write an OpenMP program with C++ that estimates the value of **pi (*π*)** using a following function and apply **rectangle rule**.

$$Area = \int_a^b f(x)\ dx\ , where\ f(x) = \frac{4}{1+x^2} \quad a = 0, b = 1, n = 10, 50, 100, 500, 1000.$$

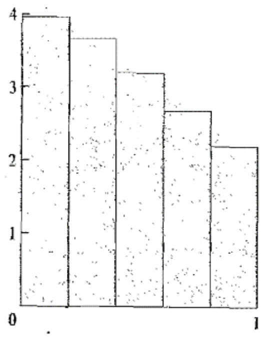

Figure 1: The rectangle rule is a simple way to approximate the area under a curve. The function is $4/\ 1{+}x^\wedge 2$, and the area under the curve between 0 and 1 is π.

Case 1: Value of x is the **starting point** in every rectangle. Calculate the *leftsum* of each rectangle on the curve and do sum of all. Observe the error value with actual pi (π).

Case 2: Value of x is the **end point** in every rectangle. Calculate the *rightsum* of each rectangle on the curve and do sum of all. Observe the error value with actual pi (π).

Case 3: Value of x is the **middle point** in every rectangle. Calculate the *midsum* of each rectangle on the curve and do sum of all. Observe the error value with actual pi (π).

Justify with error value in the above cases.

The following components are to be shown.

(a) Write the serial version program to estimate the value of pi (π). Test the result with classical integration value. Calculate the execution time by using the library function omp_get_wtime().

(b) Write the parallel version program to estimate the same. Test the result with classical integration value and by (a). It includes number of threads involved and the area calculated by which thread number. Calculate the execution time by using the library function omp_get_wtime().

(c) Identify the line of statement which leads the race condition. Race condition occurs when the multiple threads accessing a shared variable. If it exists how will you handle this problem? Use appropriate OpenMP directives/clauses such as **critical, atomic, reduction** and find the solution. Test the result with classical integration value and by (a) and (b). Calculate the execution time for **critical, atomic, reduction** clauses by using the library function omp_get_wtime().

Table 1: Summary for value of pi (π) for different cases

Actual value of pi (π) = 3.1415926535

Error = Actual-Estimated = A-E

Number of Rectangles (n)	**Case 1 (starting point)**	**Case 2 (end point)**	**Case 3 (middle point)**
n=			
Error=A-E			
n=			
Error=A-E			

You need to write your justifications for **Table 1.**

Table 2: Summary of Execution time in critical section and total time of the program

Number of Rectangles (n)	**OpenMP Directives/Clauses**	**Time taken in Critical Section**	**Total time taken of the program**
	omp critical		
	omp atomic		
	omp reduction		
	omp critical		
	omp atomic		
	omp reduction		

You need to write your justifications for **Table 2.**
