

2023-Batch-I-Set-2

Lab 2 - CSS 311 – Parallel and Distributed Computing

Questions

Part-A – Basic OpenMP programs

1. a) Write a simple C/C++ to print “*My Role Model is ******”
b) Write an *OpenMP* program to print “*My Role Model is ******” with parallel computation.
2. a) Write an *OpenMP* program to print “*My aim is to become ******” with parallel computation along with the corresponding thread id.
b) Write an *OpenMP* program to print number of threads for program 2. (a)
3. Write an *OpenMP* to print “*Welcome to IIIT Kottayam*” for 10 times using for loop and compute the execution time of sequential and parallel run.

Part-B – Matrix Multiplication and Sorting

4. Write an *OpenMP* program for **matrix multiplication** with serial and parallel computation and note the execution time of the same. Test the execution time with different dimension of the matrix (D). For example, $N=10, 100, 1000, \dots$ (Any Larger Number).

Test Case 1 – Basic 3×3 Matrices (Small Positive Numbers), $N=3$

Test Case 2 – Large Numbers

Test Case	Dimension (row, column)	Serial Execution time (sec)	Parallel Execution time (sec)
1	$N=3, 3 \times 3$ $N=10, 10 \times 10$ $N=1000, \dots$		
2	Similarly,		

4. Write an *OpenMP* program to implement **Quick Sort** on an array of integers. The program should:
 - a) Accept the number of elements and the array from the user.
 - b) Sort the array using Quick Sort (recursive divide and conquer approach).
 - c) Print the sorted array.

Show the asymptotic time complexities of best case, worst case and average case scenarios.

Test Case 1: Unsorted array - **Pivot=middle**

Test Case 2: Unsorted array - **Pivot=first** or **Pivot=last**

Test Case 3: Sorted array - **Pivot=first** or **Pivot=last**

Compare their execution times for different values of N (small to large number) and fill in a table similar to the above.
