Assignment 2

Rajdhani
DATE / /

Aim:-

Write a program to implement parallel bubble sort and merge sort using OpenMP. Use existing algorithms and measure performance of sequential and parallel algorithms.

Objectives:-

To understand and implement parallel bubble sort and merge sort using OpenMP.

Outcomes:-

Students will be able to implement parallel bubble sort and merge sort using OpenMP.

Pre-requisites:-

Students should know basic concepts of bubble sort and merge sort.
- 64 bit open source linux
- programming languages: C++/ Java/ Python/ R

Theory:-

What is sorting?

Sorting is a process of arranging elements in a group in a particular order i.e. ascending order, descending order, alphabet order, etc.

Characteristics of sorting are:-
- Arrange elements of list into certain order.

- Make data become easier to access.
- Speed up other operations such as searching and merging.

- Bubble Sorting:-

The idea of bubble sort is to compare two adjacent elements. If they are not in the right order, switch them. Do this comparing and switching until the end of the array is reached. Repeat this process from the beginning of the array 'n' times.

Bubble sort is a simple sorting algorithm that works by repeatedly swapping adjacent elements if they are in the wrong order. It is called 'bubble' sort because the algorithm moves the larger elements towards the end of the array in a manner that resembles the rising of bubbles in a liquid.

- Algorithm of bubble sort:-

Step 1: Start at the beginning of array.

Step 2: Compare the first two elements. If the first element is greater than the second element, swap them.

Step 3: Move the next pair of elements and

Rajdhani
DATE / /

repeat step 2.

Step 4: Continue the process until the end of array is reached.

Step 5: If any swaps were made in step 2-4, repeat the process from step 1.

Example of bubble sort:-

Here, we want to sort an array containing [8, 5, 1].
The following figure shows how we can sort this array using bubble sort. The elements in consideration are shown in bold 8, 5, 1.
Switch 8 and 5.

5, 8, 1	Switch 8 and 1
5, 1, 8	Reached end, start again
5, 1, 8	Switch 5 and 1.
1, 5, 8	No switch for 5 and 8.
1, 5, 8	Reached end, start again
1, 5, 8	No switch for 1 & 5.
1, 5, 8	No switch for 5, 8
1, 5, 8	Reached end.

Rajdhani
DATE / /

- Parallel Bubble sort algorithm :-

i) For k = 0 to n-2.
ii) If k is even then
iii) For i = 0 to (n/2)-1 do in parallel.
iv) If A[2i] > A[2i+1] then
v) Exchange A[2i] ↔ A[2i+1].
vi) Else.
vii) For i = 0 to (n/2)-2 do in parallel.
viii) If A[2i+1] > A[2i+1] ↔ A[2i+2]
ix) Exchange A[2i+1] ↔ A[2i+2]
x) Next k

- Merge sort :-

Merge sort is a sorting algorithm that uses a divide and conquer approach to sort an array or a list of elements.
The algorithm works by recursively dividing the input array into two halves, sorting each half and then merging the sorted halves to produce a sorted output.

Algorithm :-
1) Divide the input array into two halves.
2) Recursively sort the left half of array.
3) Recursively sort the right half of array.
4) Merge two sorted halves into a single sorted output array.
5) Divide step.

is

Rajdhani
DATE / /

6) Conquer step.
7) Combine step.

- Parallel Merge sort :-

- Parallelize processing of sub-problems.
- Max parallelization achieved with one processor per node.

Conclusion :-

In this way, we have implemented bubble sort and merge sort in parallel way using OpenMP. i also come to know how to measure performance of serial and parallel algorithm.