Assignment : HPC 4

Title : Parallel searching algorithms

Problem Statement :-
Design and implement parallel algorithm intilizing all available resources for.
- Binary search for sorted array
- Best first search or ~~DB~~ DFS or BFS.

Objectives :-
- To study and learn about parallel implementation of searching algorithm.
- To learn about MPI API in C/C++

Outcomes :-
We will be able to -
- learn about parallel computing searching techniques.
- learn about MPI

Software and Hardware requirements :
- OS : Fedora / windows (64 bit)
- GCC / C++ compiler
- Visual Studio / Text editor
- MPICC compiler using open MPI
- RAM - 4GB
- HDD - 500 GB

Teacher Signature

Theory:-

A) Binary Search:-

- Binary search, also known logarithmic search is an algorithm that finds the position of the target value within a sorted array
- It compares the target value with middle element of an array If they are not equal the half in which target element cannot ~~live~~ lie is eliminated and the search algorithms continues on the remaining half.
- If the search ends with remaining half during being empty the target is not in array.
- Binary search runs in logarithmic time in the worst case making O(log n) comparisons n = size of array.

B) ~~Breadth~~ Breadth first search

- BFS is the most common graph traversal algorithm
- It starts traversing from the source and traverse the graph ~~log~~ lengthwise ther exploring the neighbour nodes ~~first~~ first.
- A queue is maintained of neighbour nodes in each layer.

Open MPI.

- It is message passing interface library which provides extremely high and competitive performance.
- The OPEN MPI code has 3 major modules:-

1 OMPI : MPI code.
2 ORTE - Open Runtime Environment.
3. OPAL Open portable Access layer.

- mpi cc compiler is used to compile the C/C++ node. embedded with OPEN MPI.

Algorithm:-

A) Parallel Binary search.
parallel Binary Search (sorted_array)

1. Divide the array into M blocks of size n/M.
2 Apply one ~~se~~ step of comparison by the middle element of each block
3. I equality obtained return address and terminated.
4. Otherwise, identify the adjacent blocks and form a new block starting from the element following the one that signalled (>) and ending at the element preceding the one that signalled (<),
5. If they are same element, return index
6 Otherwise, parallel_binary_search (new_block)

Teacher Signature _____

B) Breath first search

```
BFS ( graph root , source S)
1 engue (S)
2 Make S as visited.
3. while ( Q is not empty)
    // remove the vertex from Q whose neighbur
     will be visited now
3.1   V = deque (Q)
  // processing all neighbour of V
  // w = neighbour of V # neighbour.
3.2   if (w is not visited)
    3.2.1 enque (W)
    33 endif
4. end while
```

Test case

Searching	input size	Seq. time	parallel time	Efficiency
Binary search (key=54)	n=1024	1.153	1.542377	0.7477
DFS.	n=1024	0.011	0.007	1.57

Conclusion:- Thus we have successfully implemented and understood concept of parallel searching algorithms using Open MPI

Teacher Signature