Assignment A4

Title : ~~B~~ Parallel Searching Algorithms.

Problem Statement :

Design and Implement parallel algorithm utilizing all available resources for

- Binary search for sorted array
- Depth First Search (DFS) OR Breadth First Search (BFS) OR Best First Search.

Objective :

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

Outcomes :

We will be able to -

- learn about parallel searching techniques
- learn about MPI

Software & Hardware Requirements :

- OS : Fedora 20 / Ubuntu (64-bit)
- GCC/G++ compiler
- Editor : gedit
- MPICC compiler using OpenMPI.
- RAM : 4GB
- HDD : 500 GB

Open MPI :

- It is a Message Passing Interface library which provides extremely high and competitive performance.
- The Open MPI code has 3 major code modules :
    1. OMPI - MPI code
    2. ORTE - Open Run Time Environment
    3. OPAL - Open Portable Access Layer.
- mpicc compiler is used to compile the C/C++ codes embedded with Open MPI.

Algorithms :

> Parallel Binary Search

parallel_binary_search (sorted_array)

1. Divide the array into M blocks of size n/M.
2. Apply one step of comparison to the middle element of each block.
3. If equality obtained, return address and terminate.
4. Otherwise, identify the adjacent blocks and form a new block starting from the element following the one that signalled (>) and endir, at the element preceeding the one that signalled (<)
5. If they are same e ement, return index.
6. otherwise, parall_binary_search (new_block)

Theory :

A> Binary search :

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

B> Breadth First Search :

- BFS is the most common graph traversal algorithm.
- It starts traversing from the source and traverse the graph layerwise, thus, exploring the neighbor nodes first.
- In sequential implementation, a queue is main of the neighbor nodes in each layer.

B> Breadth First Search :

```
BFS ( Graph root G , source S)

1.  enque (S)
2.  Mark S as visited.
3.  while (Q is not empty)
        // remove the vertex from Q whose neighbor
           will be visited now
        3.1.  V = deque (Q)
        // processing all the neighbors of V
        // W = neighbor of V ∀ neighbors.
        3.2.  if (W is not visited)
                  3.2.1.  enque (W)
        3.3.  end if.
4. end while.
```

Conclusion :

Thus, we successfully implemented parallel binary search and breadth first search using Open MPI.