Roll no:-41310.

* Assignment HPC4 *

* Title:- parallel searching algorithms.

* Problem statement:- For binary search based on existing sequential algorithms, design and implement parallel algorithm utilizing all resources available.

* Objective:-

  * To understend concept of Binary search using sequential algorithm.
  * To understand concept of parallel algorithm.
  * To compare performance by varifying number of processor used and also with sequential algorithm.

* Out Come:-
  * To ~~unde~~ display result for parallel binary search algorithms.
  * Analyze performance by varying number of processors.

* Software & Hardware apporatus:-

Open source O.S. or windows,
visual studio 2019
Key board, mouse computer.
Master slave parallel computation model.

Theory:-

Binary search is fast search Algorithm works on principle of divide& Conquer for this algorithm data must be in sorted form. It has time Complexity of O (logn)

Binory search looks for a particular item by Comparing the middle most item of the collection. If a match occurs, then idex of item is returned. If middle item is greater than the item, then item is searched in the Sub-array to left of middle term. otherwise item is searched for in the sub-array to the right of the middle term. This process Continues on the subarray as well until the size of the subarray reduces to zero.

```
mid = (low+high)/2

if mid is less than key
    low = mid+1
else
    hign = mid-1
```

Parallel Binary search algorithm:-

parallel Binary search algorithm does is move one step down in N binary search trees.

Parallel Binary search algorithm:-

1. Return if array is empty or Contains a single element.
2. Start traversing elements.
3. If Current element is not equal to next element then Store that element. Store the last element.
4. Modify original array.
5. Calculate starting time
6. Determine process identifier and number of processes.
7. using Quick sort, arrange elements in ascending order.
8. Distribute portion of array to each child processes.
9. perform a Standard mode send operation and returns when send buffer can be Safely reused.
10. Binary search for elements assigned to root process itself.
11. Collect Status from each slave process Slave process returns index of element if found in assigned segment to it else -1.
12. Calculate ending time.
13. Calculate time for parallel execution.
14. Display index at which element found
15. End.

Test Cases:-

N= 4096
arr[i] = rand() % 15000
element = 1 << 12

Result:-

modified N: 3567
average element : 3567
element found at index: 856
Execution time:- 0.034052 sec.

Conclusion:-

Binary search algorithm & Best first search algorithm is implemented parallely using mpi library.