Assignment A1:Binary Search Tree in a Cluster

1 Title:

Binary Search Tree in a cluster.

2 Problem Statement:

Using Divide and Conquer Strategies design a cluster/Grid of BBB or RasberryPi or Computers in network to run a function for Binary Search Tree using C /C++/ Java/Python/ Scala.

3 Learning objectives:

1. Basic Understanding of clustering and fromation of grid

2. To understand the use of OpenMPI for communication in a network.

4 Learning Outcomes:

1. Learnt about OpenMPI to communicate in networked cluster

2. Implementation of Binary Search Tree in a cluster using OpenMPI

5 Theory:

5.1 Divide and Conquer Strategy

Divide and Conquer is an algorithm design paradigm based on multi-branched recursion. A divide and conquer algorithm works by recursively breaking down a problem into two or more sub-problems of the same type , until these become simple enough to be solved directly . The solutions to the sub-problems are

then combined to give a solution to the original problem.
This divide and conquer technique is the basis of efficient algorithms for all kinds of problems, such as sorting, multiplying large numbers, syntactic analysis , and computing the discrete Fourier transform.
Understanding and designing these algorithms is a complex skill that requires a good understanding of the nature of the underlying problem to be solved. As when proving a theorem by induction, it is often necessary to replace the original problem with a more general or complicated problem in order to initialize the recursion, and there is no systematic method for finding the proper generalization.

5.2 Binary Search Tree:

A binary search tree is a rooted binary tree, whose internal nodes each store a key and each have two distinguished sub-trees, commonly denoted left and right. The tree additionally satisfies the binary search tree property, which states that the key in each node must be greater than all keys stored in the left sub-tree, and smaller than all keys in right sub-tree. The leaves of the tree contain no key and have no structure to distinguish them from one another. Leaves are commonly represented by a special leaf or nil symbol, a NULL pointer, etc.
Generally, the information represented by each node is a record rather than a single data element. However, for sequencing purposes, nodes are compared according to their keys rather than any part of their associated records. The major advantage of binary search trees over other data structures is that the related sorting algorithms and search algorithms such as in-order traversal can be very efficient; they are also easy to code.
Binary search trees are a fundamental data structure used to construct more abstract data structures such as sets, multisets, and associative arrays.

5.2.1 Message Passing Interface:

Message Passing Interface (MPI) is a standardized and portable message-passing system designed bya group of researchers from academia and industry to function on a wide variety of parallel computers.The standard defines the syntax and semantics of a core of library routines useful to a wide range of users writing portable message-passing programs.We can use it for creating programmers for distributed computing systems ($parallelcomputing$). MPI is a standard. There are different implementations of MPI for different platforms and programming languages.

6 Mathematical Model:

Let S be the system set:
$S = \{s, e, X, Y, F_{me}, DD, NDD, F_c, S_c\}$ where

s=start state i.e Machines connected in a network
e=end state i.e. search a word using BST in a cluster

X=set of inputs
$X = \{\{V, E\}, word\}$
where

V = vertices
E= edges
word=word to be searched

Y=set of outputs

1) word found
2) word not found

F_{me} is the set of main functions
$F_{me} = \{f_1, f_2, f_3, f_4\}$
where
f_1 = function to connect the machines in a cluster.
f_2 = function to communicate between different machines that are connected
f_3 = function to search the word
f_4 = function to display answer

DD= Deterministic Data
the binary search tree is deterministic in nature
NDD=Non-deterministic data
Communication between different connected machines

S_c = success case:
1)Work divided amongst the machines successfully to search a word in the BST

F_c =failure case:
1) System fails if network communication results in error

7 Conclusion:

Thus we have successfully learnt the use of MPI and implemented the binary search tree using OpenMPI in a cluster of computers.