WEB-BASE DISTRIBUTED QUERY PROCESSING SYSTEM BY USING REFERRAL RULE AND BINARY SEARCH TREE METHOD

Thet Wai San
Computer University, (Lashio)
pankabyar2008@gmail.com

ABSTRACT

Nowadays, as the amount of information has been increasingly rapidly, it becomes more difficult for users to get information which they require. In order to solve this kind of problems, various kind of information search systems have been developed. In this paper, an information retrieval system is presented to provide some information in which user's requirements based on interested location. In order to be effective support for Division and State, this paper implements the system called Distributed Query Processing System for Interesting Places based on referral rules.

In this paper, the proposed system can assists to get the information that users require with the help of referral rule and, Binary Search Tree (BST) algorithm is also improved in the system to accept the desired information as fast as possible. Additionally, the performance of the system is eventually maximized by the former BST algorithm.

1. INTRODUCTION

In the past few years, there has been a growing interest in using mobile agent technology for the next generation enterprise. This is because, the next generation enterprises will require the ability to efficiently retrieve, organize, manage and leverage information and knowledge from widely dispersed sources within and without virtual organization.

The current evolution of and active network in system and network management is based on this technology. A similar tendency is observed in the search and filtering of globally available information such as in the electronic marketplace, e-commerce, and information retrieval on the World Wide Web.

As the Internet constantly expands, the amount of information on the Internet expands as well. Thus, it becomes more difficult for users to get information which they require. In this paper, one of the main ideas for creating Distributed Query Processing System for Interesting Places based on referral rules is to assist the user to get some information in which user requirements based on their interested location by using referral rules abilities and techniques.

Besides, this system use Binary Search Tree algorithm to maximize the performance and to eliminate the time consuming behaviors of searching process. Searching is one of the most fundamental operations in the field of computing. It is used in any application where the users need to find out whether an element belongs to a list or, more generally, retrieve from file information associated with that element. The problem of searching a sorted sequence in parallel has attracted a good deal of attention since searching is an often-performed and time-consuming operation in most database, information retrieval, and office automation applications. Therefore, this system is able to show how problem-solving can be modeled as the process of searching for a sequence of actions that achieves a goal by using Binary Search Tree.

In section 2, the distributed Query processing system and Query Rules are briefly described. Section 3, presents searching process of system and section 4 contains design of the system. This proposed system is concluded in section 5.

2. DISTRIBUTED QUERY PROCESSING

2.1. Distributed Database Management System

To have a distributed database, there must be a database management system that coordinates the access to data at the various nodes. This system is called distributed DBMS. Although each site may have a DBMS managing the local database at that site, a distributed DBMS is also required to perform the following functions:

- Keep track of where data is located in a distributed data dictionary.
- Determine the location from which to retrieve requested data and the location at which to process each part of a distributed query.
- If necessary, translate the request at one node using a local DBMS into the proper request to another node using a different DBMS and data model, and return data to the requesting node in the format accepted by that node.
- Provide data management functions such as security, concurrency and dead-lock control, and failure recovery.
- Provide consistency among copies of data across the remote sites [6].

2.2. Query Rules

Query rules are the means for evaluations of distributed queries. Among sites, queries need to migrate, and migration rules for the flow of queries among sites are called query rules [4]. There are many query rules such as chaining rule, refined chaining rule, referral rule, proxying rule, recruiting rule, etc. A problem with chaining rule is that every server has to wait for the query to be evaluated before it can send the answer to the node that issued the query. Another problem with recruiting rule is over loaded. Therefore this paper is emphasized on referral rule [2].

2.3. Referral Rule

In referral rule, the client that originates the query has the full responsibility to answer the query. The client sends a query to a server to get the require data. If the server cannot directly answer the query, it will rewrite the query and send it back to the client. The client receives a part of the result data from this server and now has to deploy the new query to get the rest answer and migrate to another server. The process is repeated until the query ends up at server which has the last part of query result data. The client is now in charge of putting together the pieces of the final results, join the data locally and send the results back to the client. A big advantage of referral rule is that it makes the load on remote servers very light [1]. The proposed system is constructed by using referral rule.

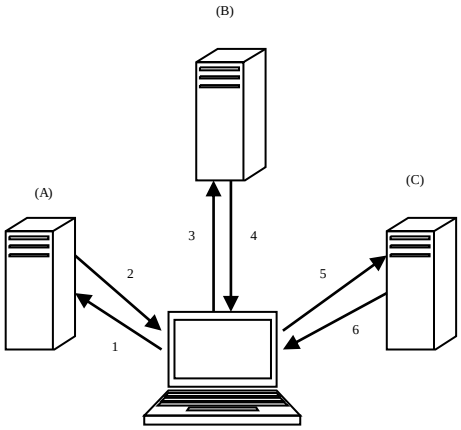


Figure 1. Referral Rule

3. SERACHING PORCESS OF SYSTEM

Searching is one of the most fundamental operations in the field of computing. The problem of searching a sorted sequence in parallel has attracted a good deal of attention since searching is an often-performed and time-consuming operation in most database, information retrieval, and office automation applications [3].

3.1. Binary Search Tree

A binary tree is either empty or consists of a node called the root and two node-disjoint binary trees, called the left and right sub trees of the root. The root of the left subtree is the left child of the root of the tree. Similarly, the root of the right subtree is the right child of the root of the tree. A binary search tree is a binary tree containing the items from a totally ordered set in its nodes, one item per node, with the items arranged in symmetric order, that is, all items in the left subtree are less than the item in the root, and all items in the right subtree are greater than the item in the root. Binary tree provide a way of representing static or dynamically changing sorted sets [5].

3.2. Binary Search Tree Operations

In this paper, Binary search trees can be used to support dynamic sets, i.e. data structures that change during lifetime, where an ordering relation among the keys is defined. They support many operations such as,
make - From Empty- initialize a new tree is empty
return - true if empty, false if not
search - return pointer to node in which key is found - otherwise return NULL
findMin - return smallest node value
findMax - return largest node value
insert - insert a new node into the tree maintaining BST property. All inserts are done at a leaf
remove - remove a node from the tree maintaining BST property
display - print a tree in an order traversal

3.3. Binary Search Tree Traversing

In proposes system query processes are based on tree traversing, which is a systematic way of examining the nodes in a binary search tree. The three principle tree traversal schemes are use these schemes are preorder, inorder and postorder traversal.

1. Inorder tree walk (visit the left subtree, the root, and right subtree)
2. Preorder tree walk (visit the root, the left subtree and right subtree)
3. Postorder tree walk (visit the left subtree, the right subtree and the root)

(1) Inorder-Tree-Walk(x)

```
1 if x ≠ NIL
2       then Inorde-Tree-Walk(left[x])
3       print key[x]
4       Inorder-Tree-Walk(right[x])
```

(2) Preorder-Tree-Walk(x)

```
1 if x ≠ NIL
2       then print key[x]
3       Preorder-Tree-Walk(left[x])
4       Preorder-Tree-Walk (right [x])
```

(3) Postorder-Tree-Walk(x)

```
1       if x ≠ NIL
2       then Postorder-Tree-Walk(left[x])
3       Postorder-Tree-Walk(right[x])
4       print key[x]
```

3.4. Searching Process

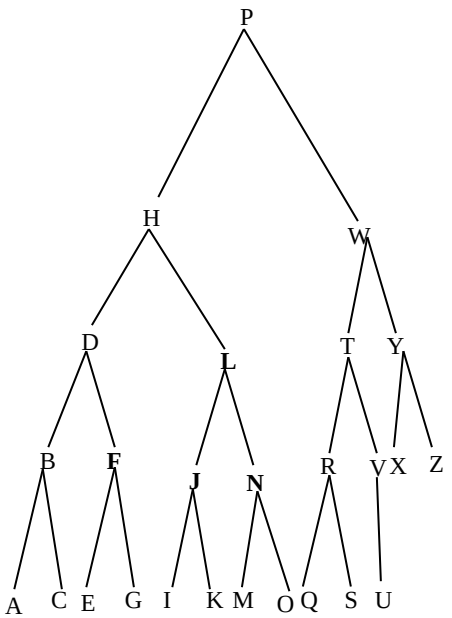


Figure 2. A Binary Tree

In this paper, the proposed system is intended to find interesting places in Myanmar based on user

interested location. This system uses Binary Search Tree algorithm to enhance the performance of system and to eliminate the time consuming behaviour of searching process.

For instance, assume that user interested location is 'Bamaw' (user's input is a city name). The search key is B, the first character in this name. In order to search first character, Figure 2, illustrates one possible binary search tree for the characters A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z (set S). The order here is ascending order. This tree traverses in in-order traversal.

Searching a binary tree for a specific value can be a recursive of iterative process. A recursive operation can call itself. If the tree is null, the value does not exist in the tree, otherwise, if the value equals the root, the search is successful. If the value is less then the root, search the left subtree. Similarly, if it is greater than the root, search the right subtree. This process is repeated until the value is found or the indicated subtree is null. If the searched value is not found before a null subtree is reached, then the item must not be present in the tree. Here is the binary search tree algorithm that our proposed system is applied:

```
def search_binary_tree(node, key)
if node is None:
    return None // key not found
if key < node.key:
    return search_binary_tree(node.left,
    key)
else if key > node.key:
    return search_binary_tree(node.right,
    key)
else:  // key is equal to node key
    return node.value  // found key
```

4. IMPLEMENTATION OF THE SYSTEM

In Figure 3 and 4, the design process of the system is described in detail. The proposed system is considered two level administrator and user, the system checks whether input is admin or user. This system uses State Database and Division Database

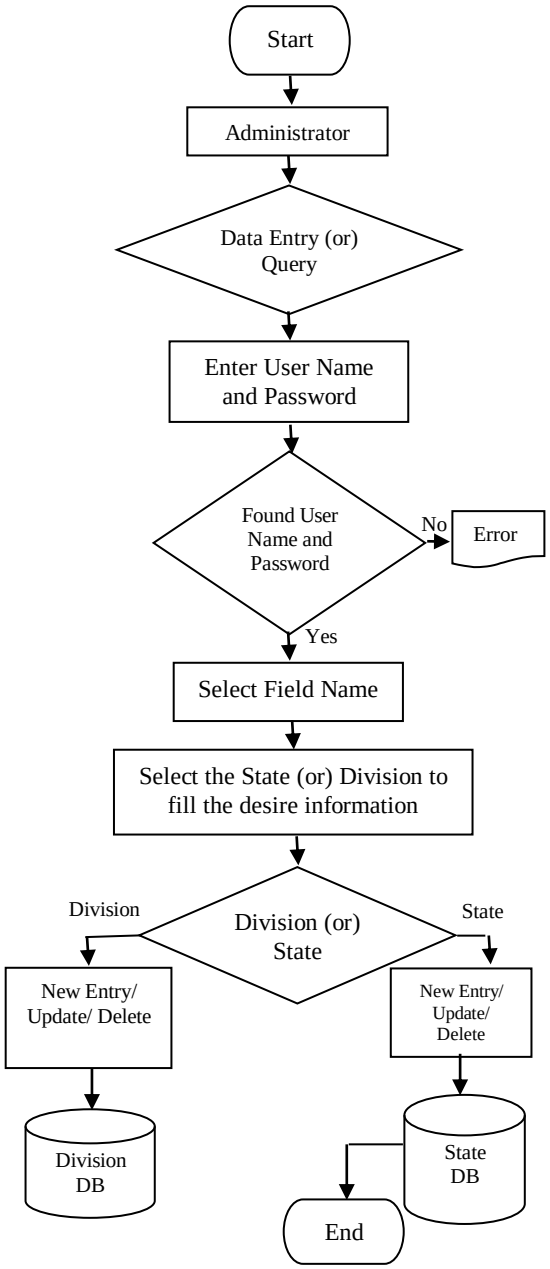


Figure 3. System Flow and Diagram for Administrator

including the name and related information of famous places in Myanmar (Hotel, Beach and Sightseeing and Pagoda etc.). After checking admin level (or) user level, the system finds the

relevant information that user wants to search using Binary Search method.

In this searching process, searches State (or) Division of user interested location and then finds the name of States and Divisions related to it using Database. Then searches famous places and pagodas that have in these states and Division. Finally, returns result information to the user.

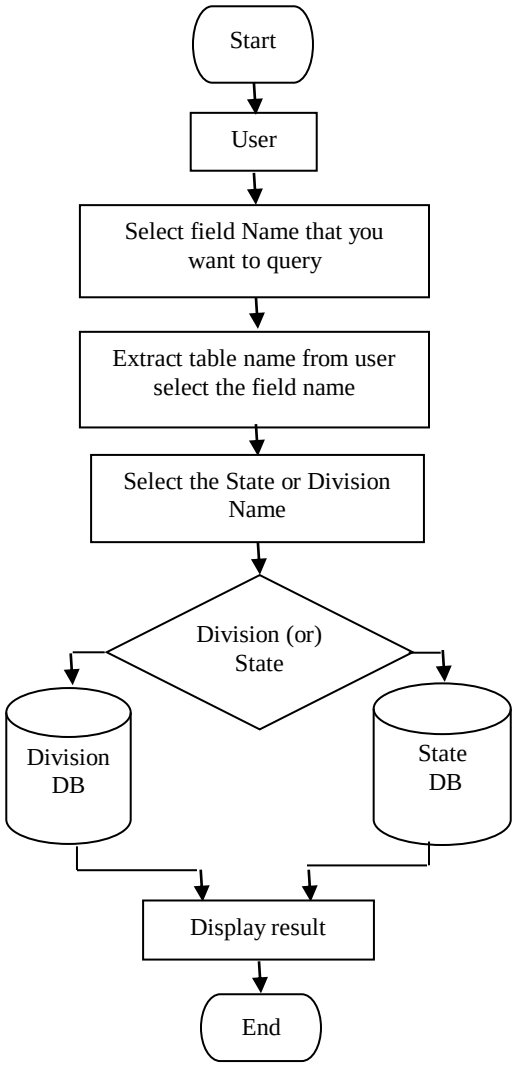
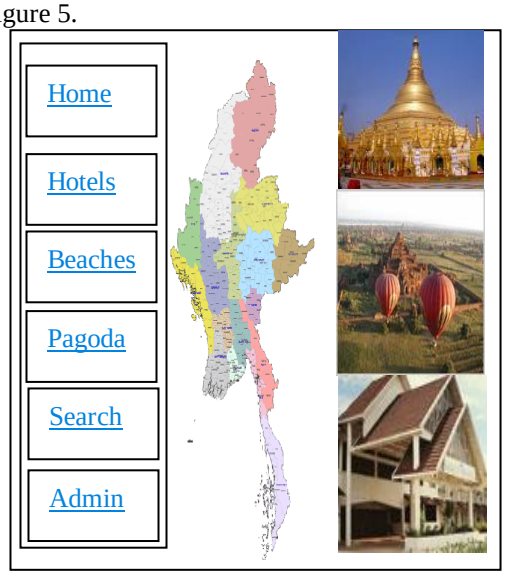


Figure 4. System Flow and Diagram for User

The main page of the System is presented in Figure 5.



5. CONCLUSION

In this paper, Distributed Query Processing System has been designed within the collaboration of referral rule for searching relevant information. The goal of are proposed system is to assist on the users to get the relevant information at short time according to the interested location.

The theories of referral rule, location-based services, information search and retrieval system and binary search tree algorithm are applied in this paper. Referral rule are effective choices for many applications and they are used already in a variety of Internet-based distributed computing application: web databases, cooperative environments, information gathering system, electronic commerce systems and so on.

The referral rules help the users to reduce their workload and save the time for searching some information that user required.

The benefits of using the proposed system are as follows:

- System provides the users to get the relevant information based on the users interested location.

- Using the concept of Binary Search Tree algorithm for necessary search, system can rapidly give the users some information that their required.
- Eliminate and facilitate the time consuming behaviour of the searching process by interpreting the concept of Binary Search Tree algorithm.
- System can assist on any users who do not know about interesting places and pagodas in Myanmar.
-

REFERENCES

[1] B. G. BABIN, "*Adaptiveness in Information Systems Integration"*, Ph.D, Thesis Sciences and Engineering Systems Department, Rensselaer Polytechnic Institute, Troy, pp-19-20 N.Y March 1998.

[2] B. Brewington, R.Gray, K. Moizuni, D/Kotz-, and E/Rus-*"Query Rules for Distributed Information Retrieval"* In Klusch-M (ed), Intelligent Information Rules; pp 355-395. Berlin: Springer-Verlag, Feb 1999.

[3] C. Coulouris, G. Dollimore, J and T. Kindberg, *" Distributed Systems, Concepts and Design"*, Wokingham: Addison Wesley, 3rd ed. Cited on page 737 Nov,2001.

[4] F. R.Fowler, "*Decentralized Object Finding Using Forwarding Address"* Ph.D. Thesis, University of Washington, Scattle, Cited in page 419, Dec 1999.

[5] G. G.IVES, ZACHARY *"Efficient Query Processing for Data Integration"*, pp 17-19 Dissertation March 2002, University of Washington.

[6] JEFFRER A. HOFFER MARY B. PRESCOTT FRED R. MCFADDEN *"Modern Database Management"* Third Edition, Vol1, No.2, pp. 19-25, Mar 1993.

[7] S. SAHUGUET, ARNAUD and TANNEN, VAL *"ubOL: A Distributed Query Language to Program Distributed Query Systems"*, Ph.D. Thesis, Computer and Information Science, University of Pennsylvania, 2002.

[8] V. VISINESCU, CATALIN *"Incremental Data Distribution on Internet-Based Distributed Systems"*, Ph.D, Thesis, Waterloo, Ontario, Canada, 2003.