ISSN:0975-9646

Meenu et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (5) , 2011, 2367-2368

LROR Oriented Tree Quorum Protocol

Meenu and Dr.Udai Shanker

Department of Computer Science and Engineering
Madan Mohan Malaviya Engineering College
Gorakhpur, India

Abstract- **In this paper, we are proposing a new replica control algorithm LROR Oriented Tree Quorum Protocol (LROR) for the management of replicated data in distributed database system. This algorithm imposes a logical structure of tree on the set of copies of an object. The proposed protocol provides a small read quorum as well as a small write quorum while guaranteeing fault-tolerance of write operations. With this algorithm read operation is executed by reading one copy in failure-free environment. In case of failure of sites, number of data copies required for read operation increases but remains constant for subsequent failure of the sites. The less number of data copies required for write operation provide low write operation cost and high write availability.**

I. INTRODUCTION

Replication is the technique of maintaining multiple copies of the data items at different sites. Replication increases the data availability. It means that we can access the data from any accessible site. It provides fault tolerance so that after failure of some sites; transaction can continue using different copy of data item. By imposing logical tree structure on the set of data copies there is no need of the reconfiguration. So failure and subsequent recovery of the sites do not cause any reconfiguration. In replication multiple copies of a data is stored at different sites. These multiple copies of a data item must appear as a single logical data copy to the transactions. This is called as one copy equivalence [4]. The replica control protocol ensures this equivalence.

The quorum is the set of the minimum number of data copies required for the successful execution of an operation. So, to execute a read and write operation, read and write quorum must be constructed respectively.

The quorums must be constructed in such a way that quorums follow the quorum intersection property. Quorum intersection property states that for any two operations op1(x) and op2(x) on a data item x, where at least one of them is write, the quorums must have a non-empty intersection. We note that tree structure is logical and does not have to correspond to actual physical structure of the network connecting the sites storing the copies.

II. RELATED WORK

There are various existing protocols for managing replicated data. In read one write all (ROWA) protocol read operation reads any one copy whereas all data copies are require for write operation. In ROWA, read cost is small and write cost is very high. Write operation cannot tolerate failure of any site in ROWA. In order to increase the fault tolerance of write operations in ROWA, voting protocol is proposed where write operations are not required to write all copies. Voting approach [3] is proposed to increase the fault tolerance of ROWA. Majority of votes of sites are required to make quorums. In this protocol, write operation need not to write all copies but the read operation reads several copies that increases the read cost. There are two versions of voting protocol namely static and dynamic voting protocol. In static protocol the size of quorums are predefined and fixed whereas in static voting the size of quorum vary according to the situation.

To overcome the problem of expensive red operation in voting protocols, several protocols have been proposed that use the network configuration information. As a result, read operation requires only a single copy. The tree quorum protocol tries to achieve the advantages of reconfiguration protocol i.e., that is low cost operation execution while maintaining availability. In tree quorum protocol [1], a logical tree structure is imposed on data copies. A read operation reads a single copy like ROWA in the failure free environment. Read operation require more copies in case of failure. Write operation tolerates failure and no reconfiguration protocol is used. Here, a write operation is required to write a majority of copies at all levels of the tree. Read operation can be executed by reading a majority of copies at any single level of the tree.

III. LROR ORIENTED TREE QUORUM PROTOCOL (LROR)

In this Section, we present a new protocol for the management of replicated data item in distributed database system. We assume that the tree has a well defined root. In this approach, quorums are constructed by using relationship of left node and right node with node in logical tree structure of data copies. For each node, we have defined Left node-Root-Right node (LROR) group. LROR group for a node consists the left node, root node and right node. We are describing a protocol that works by reading one copy of an object while guaranteeing fault-tolerance of write operations and still does not require any reconfiguration in case of a failure and subsequent recovery. This protocol provides a comparable degree of data availability too. Fig. 1 shows a tree of degree2 and of height5 having 10 nodes.

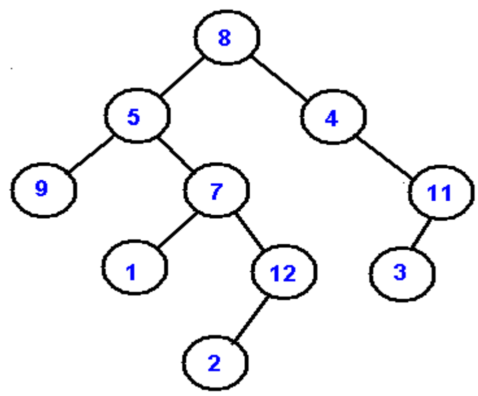

Figure 1. Tree Structure Imposed on Data Copies

A. Construction of Read and Write Quorum

For a read operation, the recursive function ReadQuorum is called with the root of tree as parameter. A read quorum can be formed by all the nodes of LROR group of any node. Fig. 2 and Fig.3 shows the algorithms for read and write quorum construction.

A transaction attempting to construct a write quorum calls the recursive function WriteQuorum with the root of the tree as parameter. By taking exactly one node from LROR group of each node, read quorum may be formed

```
Function ReadQuorum(trees): QUORUM;
var LROR Quorum, children: QUORUM;
var node: NODE;
if all nodes of LROR (node) are accessible then
return (each node of LROR (node));
else
begin
children= children of node;
for each node ε children
LROR Quorum = ReadQuorum(child Subtrees);
if all nodes of LROR (node) are inaccessible then
return ({ });
else
return (LROR Quorum);
end
end;
```

Figure2. Algorithm for Read Quorum Construction

```
Function WriteQuorum(Trees): QUORUM;
var LROR Quorum, children: QUORUM;
var node: NODE
if any node of LROR (node) is already taken then
for each child ε children
LROR Quorum = LROR  Quorum U
WriteQuorum (child subtree);
else if each node in LROR (node) is accessible then
begin
LROR Quorum = LROR Quorum U
(any node among node and its siblings);
children= children of node;
for each child ε children
LROR Quorum = LROR Quorum U
WriteQuorum (child subtree);
if unable to take any node from LROR (node)
return ({ });
else
return (LROR Quorum);
end
else
return ({ });
end;
```

Figure3. Algorithm for Write Quorum Construction

Before selecting one node from a LROR group, check all the nodes. If any node is already taken by our algorithm, there is no need to take a node from that LROR group.

An Example

For the tree in figure 1, possible read quorums may be {8, 4}, {9, 5, 7}, {2, 12} which is LROR group of node 4, 5 and 12 respectively. Write quorum is formed by taking at least one node from each distinct LROR group.

So, some possible write quorums are {8, 9, 2}, {4, 5, 2, {7, 12} etc. We can see that there is always a non-empty intersection between read and write quorum of the tree given in the figure 1.

V. CONCLUSIONS

In this paper, we have proposed LROR Oriented Tree Quorum Protocol for the management of replicated data in distributed systems. A logical tree structure is imposed by this protocol to increase operation availability and decrease operation cost. This protocol focuses on LROR relationships with the node. With LROR a read operation can be carried out by only a single root copy. Read quorum size does not increase with the increment in the number of site failures. Write operation requires one copy from LROR group corresponding to each node in the tree. So, the write operation cost of LROR is lower than all the three mentioned protocols. In both LROR and TQ, root node must be included in write quorum. So, the root node acts as a bottleneck for write operations. There is no need of reconfiguration for LROR in case of site failure and subsequent recovery. The logical structure of tree wiil be particularly beneficial if it is organized such that most reliable site is chosen as the root and the least reliable sites as the leaves. In this situation, the LROR gives very good performance in failure free environment as well as in failure environment.

REFERENCES

[1] Agrawal and A. El Abbadi, "The tree quorum protocol: An efficient approach for managing replicated data," Roc. VLDB,1990, pp.243-254.

[2] D. Agrawal and A El Abbadi., “An efficient solution to the distributed mutual exclusion problem In Proceedings of the Eighth ACM Symposium on Principles of Distributed Computing, pages 193-200, August 1989.