Symmetric Tree Replication Protocol for Efficient Distributed Storage System*

Sung Chune Choi[1], Hee Yong Youn[1], and Joong Sup Choi[2]

[1] School of Information and Communications Engineering
Sungkyunkwan University
440-746, Suwon, Korea +82-31-290-7952
{choisc,youn}@ece.skku.ac.kr
[2] Information Security Technology Division
Korea Information Security Agency
138-803, Seoul, Korea +82-2-405-5263
jschoi@kisa.or.kr

Abstract. In large distributed systems, replication of data and service is needed to decrease the communication cost, increase the overall availability, avoid single server bottleneck, and increase the reliability. Tree quorum protocol is one of the replication protocols allowing very low read cost in the best case but has some drawbacks such that the number of replicas grows rapidly as the level increases and root replica becomes a bottleneck. In this paper we propose a new replication protocol called symmetric tree protocol which efficiently solves the problems. The proposed symmetric tree protocol also requires much smaller read cost than the previous protocols. We conduct cost and availability analysis of the protocols, and the proposed protocol displays comparable read availability to the tree protocol using much smaller number of nodes. It is thus effective to be applied to survival storage system.

Keywords: replication protocol, tree quorum, availability, distributed system, symmetric tree

1 Introduction

In large distributed systems node and communication failures are likely to occur which can prevent the operations from being successfully carried out. This raises the need for introducing a sufficient level of fault tolerance into the distributed systems [1]. Other problems are increased communication cost as the system grows and bottleneck of a single server as many processes compete for the same service. Replication of data and services is one of the practical solutions to the problems. It can avoid the server bottleneck problem and increase overall availability [2]. However, the communication cost increases as the number of replicas increases. In order to minimize the

*This work was supported by Korea Research Foundation Grant (KRF-2002-041-D00421).
Corresponding author: Hee Yong Youn

P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2659, pp. 474–484, 2003.

communication cost, thus, the number of replicas should be kept as small as possible [3,4].

We consider a distributed system in which data are fully replicated and two types of operations are allowed on the replicated data, namely read and write. A read operation returns some value while write operation installs a new value. Proper synchronization is achieved if read operations return the value installed by the last write operation, and read and write operations or two write operations are not executed concurrently. Each node uses a centralized concurrency control scheme to synchronize the accesses to its local copy and a replication control protocol to coordinate the accesses to various replicas. The basic principle employed in maintaining consistency of replicated data is to require conflicting operations to lock at least one common copy.

A distributed system consists of a set of distinct sites that communicate with each other by sending messages over a communication network. No assumptions are made regarding the speed, connectivity, or reliability of the network. We assume that the sites are fail-stop. A distributed database consists of a set of objects stored at several sites. Users interact with the database by invoking transaction programs. Transactions are partially ordered sequence of read and write operations that are executed atomically. Execution of a transaction must appear atomic, i.e. a transaction either commits or aborts. In a replicated database, copies of an object may be stored at several sites in the network. Multiple copies of an object must appear as a single logical object to the transactions. This is termed as one-copy equivalence and enforced by replication control protocol.

In the literature there exist several replication algorithms aiming at different goals. The Read-One/Write-All algorithm [5] has minimum read cost and maximum read availability but has maximum communication cost for write operation. The quorum consensus [6] and dynamic voting [7] have good read and write availability but has a disadvantage of high read cost. For all these replication control protocols, making the read operation cheaper ends up with a more expensive write operation and vice versa. All the strategies have a cost of O(n), which means the operation cost linearly depends on the number of replicas in the system [8].

By assigning a logical structure to a set of replicas, it is possible to reduce the communication cost. Multi-Level-Voting-Protocol [9], Weighted Voting [10], Tree quorum protocol [11] and Grid protocol [12] are such type of replication protocols, but they still have relatively high operation cost especially when the number of replicas is large and some failures exist. The tree quorum protocol has read cost of 1 in the absence of failures and provides graceful degradation when failures exist. Hence in this paper we develop a new replication protocol which has low operation cost. We compare the replication protocols which use a logical tree structure and the proposed symmetric tree structure. The proposed symmetric tree protocol allows comparable read availability to the tree protocol with much smaller number of nodes. It is thus effective to be applied to survival storage system.

The rest of the paper is organized as follows. In Sect. 2 we describe the earlier protocols, and in Sect. 3 we propose a new protocol. We compare the new symmetric tree protocol with earlier protocols in Sect. 4. Section 5 concludes the paper.

2 Review of Earlier Protocols

2.1 Logarithmic Replication Protocol

The class of replication protocols discussed here is a generalization of the Tree Quorum Protocol. These replication protocols use a logical tree structure. All the algorithms have read cost of 1 in the absence of failures and provide graceful degradation when failures exist. The replication protocols which use tree structure have varying costs and availabilities according to fault condition, whereas other replication protocols have constant costs and availabilities.

Let n be the number of replicas which can be organized as a tree of height h. A non-leaf replica R_i has S_{Ri} descendants. For each replica except the leaves we define read quorum rq_{Ri}, and a write quorum wq_{Ri}. A tree consisting of 13 replicas organized in three levels is shown in Fig. 1.

Algorithm:
The algorithm is recursive:
• A tree of height 0 consists only of a leaf replica and can be locked by simply locking the replica for either read or write operation.
Read operation:
• Locking a tree of height h for a read operation means to lock the root replica or rq_{ROOT} of its subtrees with the rq_{ROOT} descendants of the root serving as new root replicas of the subtrees of height *h-1*.
Write operation:
• Locking a tree of height h for a write operation means to lock the root replica and wq_{ROOT} of its subtrees. Here, the wq_{ROOT} descendants of the root serve as new root replicas for the subtrees of height *h-1*.

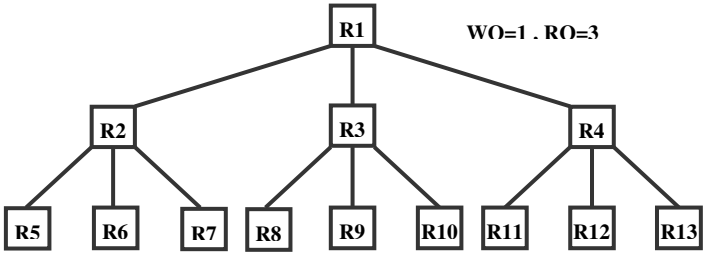

Fig. 1. A tree of height 3 with 13 replicas

Examples of valid read quorum sets (RQ) from Figure 1 are {R1}, {R2, R3, R4} when node-1 is not available, and {R5, R6, R7, R3, R11, R12, R13} when node-1, node-2, and node-4 are not available. Examples of valid write quorum sets (WQ) are {R1, R2, R5}, {R1, R3, R10}, and {R1, R4, R12}.

In order to maintain consistency of the replicated objects, the selected read and write quorums have to fulfill some requirements. To detect read/write conflict, any

valid read quorum set must intersect with valid write quorum set. This requirement can be satisfied by $rq_{Ri} + wq_{Ri} > V_{Ri}$ where V_{Ri} is the number of descendants. To detect write/write conflicts, any two valid write quorum sets must intersect. This is always satisfied because every valid write quorum set includes the root. So there exist many cases of valid RQ and WQ in the tree quorum protocol. After reviewing all possible cases regarding the provided cost and availability, the strategy with RQ = S and WQ = 1 is selected as the best choice. The case is called *Logarithmic Protocol*.

2.2 Grid Protocol

Here the replicas are logically arranged in a grid topology, and read and write operations are required to lock the rows and columns of nodes such that conflicting operations request a common node. In this protocol read quorum consists of a node from every column and write quorum consists of a node from every column plus a full column of nodes. A grid network which consists of 4 rows and 4 columns of 16 replicas is shown in Fig. 2.

Algorithm:
Read operation:
• Select and lock any single node from each column. This set of nodes is called Column-cover (C-cover).
Write operation:
• Choose a C-cover and lock any whole column.

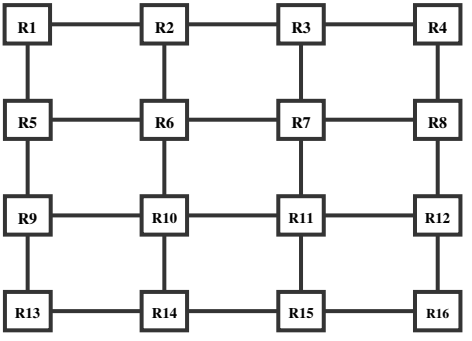

Fig. 2. A Grid network with 16 replicas

Examples of valid read quorum sets here are {R1, R6, R3, R12} and {R1, R6, R7, R8}, and examples of valid write quorum sets are {R1, R10, R3, R4, R2, R6, R14} and {R9, R6, R3, R8, R7, R11, R15}.

Here two write operations cannot be executed concurrently because each one locks a C-cover and all nodes in a column. If two operations have locked a C-cover, neither one can obtain write-locks from all the nodes in any column. Similar argu-

ment can be used to show that concurrent execution of read and write operations are not possible. The grid protocol has constant operation cost because the size of quorum set is always same. One disadvantage of this protocol is the operation cost is equally high regardless of failure condition. However, it has higher availability than tree quorum protocol. To solve the problems of the tree quorum and grid protocol, we next present the proposed symmetric tree protocol.

3 The Proposed Protocol

As mentioned in the previous section, tree quorum protocol and grid protocol have some drawbacks, respectively. In this section we propose a new replication protocol called symmetric tree protocol to solve the drawbacks of the previous protocols.

3.1 Motivation

In typical replication protocols almost all replicas need to be accessed for read and write operation. For example, in Quorum consensus protocol of 16 replicas, read quorum + write quorum need to be greater than 16. However, if we assign a logical structure to the replicas, then we can reduce the number of replicas because we can organize the structure in such a way that the tree and grid topology are used together.

As previously described, a major weakness of tree quorum protocol is that the number of replicas rapidly grows as the tree level grows. The tree quorum protocol has read cost of 1 in the absence of failures. As a result, root replica can become a bottleneck. To solve this problem, the proposed protocol has two root replicas using symmetric tree topology. Therefore, the proposed protocol can have the advantage of having minimum operation cost whether failures exist or not. Also it has the advantage of having not many replicas in high levels like grid protocol.

3.2 The Proposed Symmetric Tree Protocol

The proposed protocol solves the problem of other replication protocols of significantly increased read cost when failures exist using symmetric tree topology. It has the advantage of having minimum operation cost in the best case like tree structure protocol and also has the advantage of having not so many replicas in higher levels like grid protocol. In this protocol replicas are organized as shown in Figure 3, where two tree structures are joined whose leaf nodes are common. We call the two trees up-tree and down-tree, respectively.

Algorithm:
Read Operation:
• Read operation starts from either the root node of up-tree or that of down-tree. If the root replica is not available, all the descendants of it are read. Each the descendant

serves as a new root replica for the subtree to the *h/2*-level of the symmetric tree, where *h* is the number of levels of the tree. After the *h/2*-level, the next level is read if any node of the parent nodes is not available.

Write Operation:

• Write operation starts from any root replica of the two trees. Only one replica of the descendants of the root must writes. A descendant of the root serves as a new root replica of the subtree. The descendant of the parents must write from the *h/2*-level to the last level.

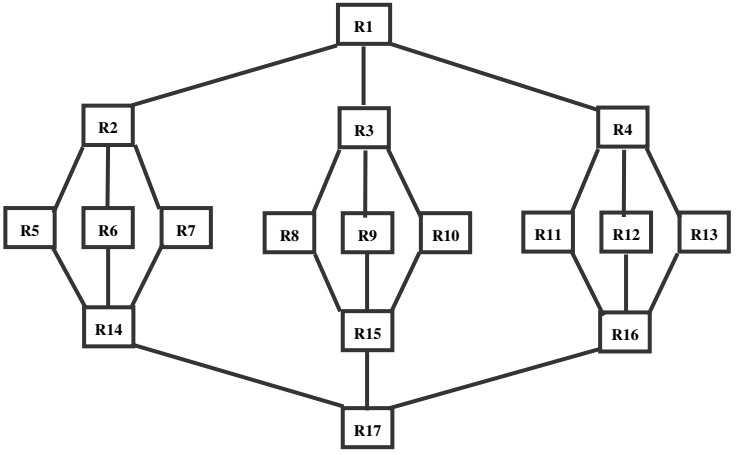

Fig. 3. A network for the proposed symmetric tree protocol with 17 replicas

From Fig. 3 examples of valid read quorum set are {R1}, {R2, R3, R4}, {R3, R4, R5, R6, R7}, {R3, R4, R14}, {R1, R8, R9, R10, R16}, {R14, R15, R16}, and {R17}. Examples of valid write quorum sets are {R1, R2, R5, R14, R17}, {R1, R3, R9, R15, R17}, and {R1, R3, R10, R15, R17}.

Consistency is maintained in selecting read, write quorum set to satisfy the requirement of detecting read/write conflict and write/write conflict. The Read/Write conflict can be detected because read operation should lock the whole descendants and write operation should lock at least one replica of whole descendants. The Write/Write conflict can also be detected because every write operation has to lock the root replica of the symmetric tree structure. Thus, if one replica is locked, then others cannot get the lock.

4 Performance Analysis

In this section cost and availability of the proposed protocol and tree protocol are analyzed.

4.1 Cost Analysis

For cost analysis, read cost and write cost are considered, which are computed by the number of sites involved in the operation. A read operation in both the protocols need to consult only the root replica which is very efficient if root replica is available. However, the root replica becomes a bottleneck if all operations are done on it. Therefore the algorithm is slightly modified as the level of the tree the operation will be performed is randomly selected. For this, uniformly distributed random variable X in the interval [0,1] and a parameter f in interval [0,1] are chosen. A random value x of X is generated and the top-level is chosen for performing read operation if $x \leq f$. Otherwise, a new x is generated and the next level is chosen if $x \leq f$, etc.

4.1.1 Logarithmic Protocol

Suppose there are h+1 levels. The average read cost computes to

$$\begin{aligned} C_{read} &= f + (1-f)\cdot f \cdot RQ + (1-f)^2 \cdot f \cdot RQ^2 + \cdots + (1-f)^{h-1} \cdot f \cdot RQ^{h-1} \\ &\quad + (1-(f + (1-f)\cdot f + (1-f)^2 \cdot f + \cdots + (1-f)^{h-1} \cdot f)) \cdot RQ^h \\ &= f \sum_{k=0}^{h-1} (1-f)^k \cdot RQ^k + \left(1 - f \sum_{k=0}^{h-1} (1-f)^k\right) \cdot RQ^h \end{aligned} \tag{1}$$

In the tree protocol, we consider Logarithmic protocol which is the best choice regarding the cost and availability. The Logarithmic Protocol is defined by a read quorum of RQ = S and write quorum of WQ = 1.

$$\min(C_{read}) = 1 \tag{2}$$

Average read cost is

$$C_{read} = f \frac{((1-f)S)^h - 1}{(1-f)S - 1} + ((1-f)S)^h \tag{3}$$

For write operation,

$$C_{write} = h + 1 \tag{4}$$

4.1.2 Symmetric Tree Protocol

For cost analysis, read cost and write cost are considered, which are computed by the number of sites involved in that operation. A read and write operation in both the protocols only need to consult the root replica which is very efficient if root replica is available.

Average read cost is computed by

$$
\begin{aligned}
C_{read} &= f + (1-f)\cdot f \cdot S + (1-f)^2 \cdot f \cdot S^2 + \cdots + (1-f)^{\frac{h}{2}} \cdot f \cdot S^{\frac{h}{2}} \\
&\quad + (1-f)^{\frac{h}{2}+1} \cdot f \cdot S^{\frac{h}{2}-1} + \cdots + (1-f)^h \cdot f \cdot S \\
&\quad + (1-(f + (1-f)\cdot f + \cdots + (1-f)^h \cdot f)) \qquad (5) \\
&= f\left(\left(\sum_{k=0}^{\frac{h}{2}} (1-f)^k \cdot S^k\right) + \left(\sum_{k=1}^{\frac{h}{2}-1} (1-f)^{\frac{h}{2}+k} \cdot S^{\frac{h}{2}-k}\right)\right) + \left(1 - f\sum_{k=0}^{h} (1-f)^k\right)
\end{aligned}
$$

$$\min(C_{read}) = 1 \qquad (6)$$

For write operation,

$$C_{write} = h + 1 \qquad (7)$$

Comparison of average read cost for both the protocols are shown in Fig. 4 for 121 nodes of tree quorum protocol and 161 nodes of symmetric tree protocol. Average write cost for them is depend on height of tree.

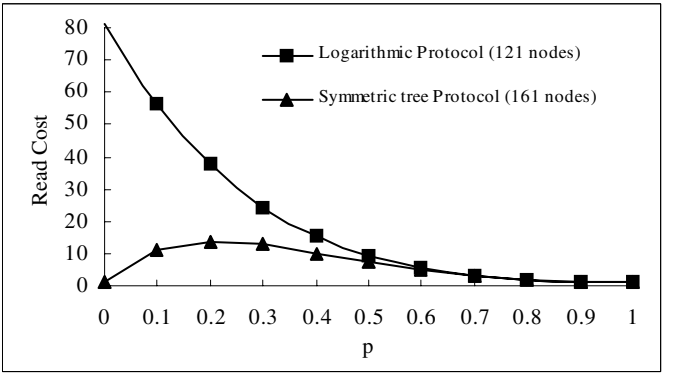

Fig. 4. Comparison of average read cost

The tree quorum protocol consists of 121 nodes of 5 levels and the last level has the maximal number of node. On the other hand, the symmetric tree protocol consists of 161 nodes of 9 levels and the fourth level has the maximal number of node. Also, the maximal number of node is the same. Observe from Figure 4 that read cost of the proposed protocol is always much smaller than logarithmic protocol in sprite of has more nodes. The difference between the two protocols gets more substantial when the failure of nodes is low.

4.2 Availability Analysis

4.2.1 Logarithmic Protocol

The availability of read operation for the tree quorum protocol performed on a tree of height $l > 0$ is

$$\wp_{read}^{(l)} = p + (1-p)\sum_{k=RQ}^{S}\binom{S}{k}\left(\wp_{read}^{(l-1)}\right)^{k}\left(1-\wp_{read}^{(l-1)}\right)^{S-k} \text{ with } \wp_{read}^{(0)} = p \tag{8}$$

So the availability of read operation for the logarithmic protocol is

$$\wp_{read}^{(l)} = p + (1-p)\left(\wp_{read}^{(l-1)}\right)^{S} \text{ since RQ = S} \tag{9}$$

The availability of write operation is

$$\wp_{write}^{(l)} = p\sum_{k=WQ}^{S}\binom{S}{k}\left(\wp_{write}^{(l-1)}\right)^{k}\left(1-\wp_{write}^{(l-1)}\right)^{S-k} \text{ with } \wp_{write}^{(0)} = p \tag{10}$$

So the availability of write operation for the logarithmic protocol is

$$\wp_{write}^{(l)} = p\left(1-\left(1-\wp_{write}^{(l-1)}\right)^{S}\right) \tag{11}$$

4.2.2 Symmetric Tree Protocol

The availability of read operation for the proposed protocol performed on height $h/2 \geq l > 0$ is

$$\wp_{read}^{(l)} = p + (1-p)\left(\left(\wp_{read}^{(l-1)}\right)^{S} + \left(1-\left(\wp_{read}^{(l-1)}\right)^{S}\right)p\right) \text{ with } \wp_{read}^{(0)} = p \tag{12}$$

The availability of write operation is

$$\wp_{write}^{(l)} = p^{2}\sum_{k=WQ}^{S}\binom{S}{k}\left(\wp_{write}^{(l-1)}\right)^{k}\cdot\left(1-\wp_{write}^{(l-1)}\right)^{S-k} \text{ with } \wp_{write}^{(0)} = p \tag{13}$$

Comparisons of read availabilities of the two protocols are shown in Fig. 5. For the comparison, we examine 3 cases, each one with a different number of replicas. They are 17 of level 5, 53 of level 7, and 161 replicas of level 9. As illustrated in Fig. 5, the read availability of the proposed protocol increases as the level of tree increases. Note that the read availability of symmetric tree protocol is higher than logarithmic protocol with much smaller number of nodes.

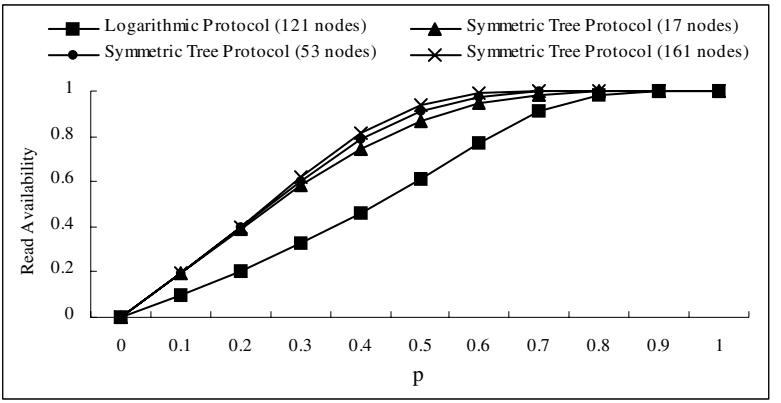

Fig. 5. Comparison of read availability

5 Conclusion

In this paper we have proposed a new symmetric tree replication protocol which is better than Logarithmic protocol in read cost and read availability using much smaller number of nodes. One of the main advantages of the proposed symmetric tree protocol is much smaller read cost than previous protocols even though the failure rate of the nodes increases. The choice of which replication protocol should be used depends on the primary performance criterion, and the new proposed symmetric tree protocol can be selected for low operation cost and high read availability.

As future work, we will analyze and compare the response time and throughput of the proposed protocol. Implementation of the protocols for survivable storage system will also be carried out.

Reference

[1] C. Amza,, A.L. Cox, W. Zwaenepoel, Data replication strategies for fault tolerance and availability on commodity clusters, *Proc. Int'l Conf on Dependable Systems and Networks (DSN)*, 2000, 459–467

[2] H.Y. Youn, B. Krishnamsetty, D. Lee, B. K. Lee, J.S. Choi, H.G. Kim, C.W. Park, and H.S. Lee, An Efficient Hybrid Replication Protocol for Highly Available Distributed System, *Proc. Int'l Conf on Communication and Computer Networks (CCN)*, Nov, 2002,

[3] K. Arai, K. Tanaka, M. Takizawa**,** Group protocol for quorum-based replication, *Proc. Seventh Int'l Conf on Parallel and Distributed Systems*, 2000, 57–64.

[4] G. Alonso, Partial Database Replication and Group Communication Primitives, *Proc. of the 2nd European Research Seminar on Advances in Distributed Systems (ERSADS'97)*, March 1997, 171–176.

[5] P.A. Bernstein and N. Goodman, An Algorithm for Concurrency Control and Recovery in Replicated Distributed Databases, *ACM Trans on Distributed Systems*, 9(4), 1984, 596–615.

[6] R.H. Thomas. A Majority Consensus Approach to Concurrency Control for Multiple Copy Data-based, *ACM Trans on Database Systems*, 4(2),1979, 180–207.
[7] D. Davcev, A Dynamic Voting Scheme in Distributed Systems. *IEEE Trans on Software Engineering*, 15(1), 1989, 93–97.
[8] D. Saha, S. Rangarajan, S.K. Tripathi, An Analysis of the Average Message Overhead in Replica Control Protocols, *IEEE Trans on Parallel and Distributed Systems*, 7(10), Oct. 1996, 1026–1034.
[9] B. Freisleben, H.H. Koch, and O. Theel, Designing Multi-Level Quorum Schemes for Highly Replicated Data. *Proc. of the 1991 Pacific Rim Int'l Symp on Fault Tolerant Systems, IEEE*, 1991, 154–159.
[10] D.K. Gifford, Weighted Voting for Replicated Data, *Proc. of the 7th ACM Symp on Operating Systems Principles*, 1979, 150–162.
[11] D. Agrawal and A. El Abbadi, The tree Quorum protocol: An Efficient Approach for Managing Replicated Data, *Proc of the 16th Very Large Databases (VLDB) Conf*, 1990, 243–254.
[12] S. Cheung, M. Ammar, and M. Ahamad, The Grid Protocol: A High Performance Scheme for Maintaining Replicated Data, *Proc of the 6th Int'l Conf on Data Engineering*, 1990, 438–445.