Diagonal Replication on Grid for Efficient Access of Data in Distributed Database Systems

M. Mat Deris, N. Bakar, M. Rabiei, and H.M. Suzuri

University College of Science and Technology,
Department of Computer Science, 21030, MengabangTelipot, Kuala Terengganu, Malaysia
{mustafa, suzuri}@kustem.edu.my

Abstract. Data Replication can be used to improve data availability in distributed database environments. In such a system, a mechanism is required to maintain the consistency of the replicated data. Grid structure technique based on quorum is one of the solutions for performing this while providing a high data availability of the system. It was shown in the previous study that, it still requires a bigger number of copies be made available to construct a quorum. So it is not suitable for the large systems. In this paper, we propose a technique called diagonal replication on grid (DRG) technique where a data will be replicated in a diagonal manner in a logical grid structure. In comparison to the Grid structure technique, DRG requires lower communication cost for an operation, while providing higher read/write availability, which is preferred for large distributed database systems.

1 Introduction

With the proliferation of computer networks, PCs and workstations, new models for workplaces are emerging [1]. In particular, organizations need to provide current data to users who may be geographically remote and to handle a volume of requests that might be high for a single server. Consequently, the availability and the consistency of data in the systems are extremely important. To maintain the consistency and integrity of data, expensive synchronization mechanisms are needed [2,4]. One way to provide access to such data is through replication.

One of the simplest techniques for managing replicated data is read-one write-all (ROWA) technique. Read operations on a data object are allowed to read any copy, and write operations are required to write all copies of the data object. The ROWA protocol is good for environments where the data is mostly read-only. This is because it provides read operation with high degree of availability at low communication overhead but vulnerable to the write availability. It is very popular and has been used in mobile and peer-to-peer environments [3] and database systems [6]. Dynamic quorum techniques have also been proposed to further increase availability in replicated databases [5,7]. Nevertheless, these approaches do not address the issue of low-cost read operations.

Recently, Computational Grids are becoming ubiquitous computing infrastructure for the coordinated and secure use of shared resources in a distributed environment. Based on this new computing platform, data replication technique proposed in [4]

provides high data availability of the system. However, this technique still requires that a bigger number of copies be made available to construct a read or write quorum, which is not suitable for the large systems.

Without loss of generality, the terms node and site will be used interchangeably. If data is replicated to all sites, the storage capacity becomes an issue, thus an optimum number of sites to replicate the data is required with non-tolerated read or write availability of the system. The focus of this paper is on modeling a technique to optimize the read or write availability of the replicated data in the distributed database systems. We describe the Diagonal Replication on Grid (DRG) technique considering only binary vote assigned with vote one to the primary site and its diagonal site, and zero otherwise. This assignment provides a higher availability of executing read or write operations in replicated database due to the minimum number of quorum size required. Moreover, it can be viewed as an allocation of replicated copies such that a copy is allocated to a site if and only if the vote assigned to the site is one.

This paper is organized as follows: In Section 2, we review the grid structure technique. In Section 3, the model and the technique of the DRG is presented. In Section 4, the performance of the proposed technique is analyzed in terms of communication cost and availability, and a comparison with other technique is given.

2 Review of Grid Structure (GS) Technique

In this section, we review the GS technique, which is then compared to the proposed DRG technique.

Maekawa [8] has proposed a technique by using the notion of finite projective planes to obtain a distributed mutual exclusion algorithm where all quorums are of equal size. Cheung et al. [9], extended Maekawa's grid technique for replicated data that supports read and write operations. In this technique, all sites are logically organized in the form of two-dimensional $n \times n$ grid structure as shown in Fig. 1 if there are $N = n \times n$ sites in the system. Read operations on the data item are executed by acquiring a read quorum that consists of a copy from each column in the grid. Write operations, on the other hand, are executed by acquiring a write quorum that consists of all copies in one column and a copy from each of the remaining columns. The read and write operations of this technique is of the size $O(n)$; this technique is normally referred to as the *sqrt(R/W)* technique. In Fig. 1, copies {1,2,3,4,5} are

Fig. 1. A grid organization with 25 copies of a data object

sufficient to execute a read operation whereas copies {1,6,11,16,21,7,13,19,25} will be required to execute a write operation. However, it still has a bigger number of copies for read and write quorums, thereby degrading the communication cost and data availability. It is also vulnerable to the failure of entire column or row in the grid [4].

3 Diagonal Replication on Grid (DRG) Technique

3.1 Model

A distributed database consists of a set of data objects stored at different sites in a computer network. Users interact with the database by invoking transactions, which are partially ordered sequences of atomic read and write operations. The execution of a transaction must appear atomic: a transaction either commits or aborts [10,11].

The correctness criteria for replicated database is one-copy serializability [11], which ensures the serializable execution of transactions. In order to ensure one-copy serializability, a replicated data object may be read by reading a quorum of copies, and it may be written by writing a quorum of copies. The selection of a quorum is restricted by the quorum intersection property to ensure one-copy equivalence: For any two operations $o[x]$ and $o'[x]$ on a data object x , where at least one of them is a write, the quorum must have a non-empty intersection. The quorum for an operation is defined as a set of copies whose number is sufficient to execute that operation.

Briefly, a site s initiates a DRG transaction to update its data object. For all accessible data objects, a DRG transaction attempts to access a DRG quorum. If a DRG transaction gets a DRG write quorum without non-empty intersection, it is accepted for execution and completion, otherwise it is rejected. We assume for the read quorum, if two transactions attempt to read a common data object, read operations do not change the values of the data object. Since read and write quorums must intersect and any two DRG quorums must also intersect, then all transaction executions are one-copy serializable.

3.2 The DRG Technique

All sites are logically organized in the form of two-dimensional grid structure. For example, if a DRG consists of twenty-five sites, it will logically organized in the form of 5×5 grid as shown in Fig. 2. Each site has a master data file. In the remainder of this paper, we assume that replica copies are data files. A site is either operational or failed and the state (operational or failed) of each site is statistically independent to the others. When a site is operational, the copy at the site is available; otherwise it is unavailable. The data file will replicate to *diagonal sites*. The circles in the grid represent the sites under the distributed database environment and $a, b, \dots$, and y represent the master data files located at site $s(1,1), s(1,2), \dots$, and $s(5,5)$ respectively.

Definition 3.2.1: Assume that a database system consists of $n \times n$ sites that are logically organized in the form of two dimensional grid structure. All sites are labeled

$s(i,j)$, $1 \leq i \leq n$, $1 \leq j \leq n$. The *diagonal site* to $s(i,j)$ is $\{s(k,l) \mid k=i+1, l=j+1, \text{ and } k,l \leq n, \text{ if } i=n, \text{ initialized } i=0, \text{ if } j=n, \text{ initialized } j=0\}$. A *diagonal set*, $D(s)$, is a set of diagonal sites.

Fig. 2. A grid organization with 25 sites, each of which has a master data file $a, b, \dots$, and y respectively.

For example, from Fig. 2, the diagonal site to $s(1,1)$ is $s(2,2)$, the diagonal site to $s(2,1)$ is $s(3,2)$, etc. Thus, based on this technique, sites in the diagonal set will have the replica copies in common. One such diagonal set is $\{s(1,1), s(2,2), s(3,3), s(4,4), s(5,5)\}$, and each site will have the same replica copies i.e., $\{a, g, m, s, y\}$. Also the number of diagonal sets in the system equals to n . Let $D^m(s)$ be the m^{th} diagonal set, for $m=1, 2, \dots, n$, then from Fig. 2, the diagonal sets are;

$$\begin{aligned} D^1(s) &= \{s(1,1), s(2,2), s(3,3), s(4,4), s(5,5)\}, \\ D^2(s) &= \{s(2,1), s(3,2), s(4,3), s(5,4), s(1,5)\}, \\ D^3(s) &= \{s(3,1), s(4,2), s(5,3), s(1,4), s(2,5)\}, \\ D^4(s) &= \{s(4,1), s(5,2), s(1,3), s(2,4), s(3,5)\}, \text{ and} \\ D^5(s) &= \{s(5,1), s(1,2), s(2,3), s(3,4), s(4,5)\}. \end{aligned}$$

Since the data file is replicated to only the diagonal sites, then it minimize the number of database update operations, misrouted and dropped out calls. Also, sites are autonomous for processing different query or update operation, which consequently reduces the query response time. The primary site of any data file and for simplicity, its diagonal sites are assigned with vote one and vote zero otherwise. A vote assignment on grid, B , is a function such that

$$B(s(i,j)) \in \{0, 1\}, 1 \leq i \leq n, 1 \leq j \leq n$$

where $B(s(i,j))$ is the vote assigned to site $s(i,j)$. This assignment is treated as an allocation of replicated copies and a vote assigned to the site results in a copy allocated at the diagonal site. That is,

$$1 \text{ vote} \equiv 1 \text{ copy}.$$

Let

$$L_B = \sum_{s(i,j) \in D(s)} B(s(i,j))$$

where, L_B is the total number of votes assigned to the primary site and its diagonal sites. Thus, $L_B = d$.

Let r and w denote the read quorum and write quorum, respectively. To ensure that the read operation always gets up-to-date values, $r + w$ must be greater than the total number of copies (votes) assigned to all sites. The following conditions are used to ensure consistency:

- 1.) $1 \leq r \leq L_B, 1 \leq w \leq L_B,$
- 2.) $r + w = L_B + 1.$

Conditions (1) and (2) ensure that there is a nonempty intersection of copies between every pair of read and write operations. Thus, the conditions ensure that a read operation can access the most recently updated copy of the replicated data. Timestamps can be used to determine which copies are most recently updated.

Let $S(B)$ be the set of sites at which replicated copies are stored corresponding to the assignment B . Then

$$S(B) = \{s(i,j) \mid B(s(i,j)) = 1, 1 \leq i \leq n, 1 \leq j \leq n\}.$$

Definition 3.2.2: For a quorum q , a *quorum group* is any subset of $S(B)$ whose size is greater than or equal to q . The collection of quorum group is defined as the *quorum set*.

Let $Q(B,q)$ be the quorum set with respect to the assignment B and quorum q , then

$$Q(B,q) = \{G \mid G \subseteq S(B) \text{ and } |G| \geq q\}$$

For example, from Fig. 1, let site $s(1,1)$ be the primary site of the master data file a . Its diagonal sites are $s(2,2), s(3,3), s(4,4)$, and $s(5,5)$. Consider an assignment B for the data file a , such that

$$B_a(s(1,1)) = B_a(s(2,2)) = B_a(s(3,3)) = B_a(s(4,4)) = B_a(s(5,5)) = 1$$

$$\text{and } L_{B_a} = B_a(s(1,1)) + B_a(s(2,2)) + B_a(s(3,3)) + B_a(s(4,4)) + B_a(s(5,5)) = 5.$$

Therefore, $S(B_a) = \{s(1,1), s(2,2), s(3,3), s(4,4), s(5,5)\}$.

If a read quorum for data file a , $r=2$ and a write quorum $w = L_{B_a} - r + 1 = 4$, then the quorum sets for read and write operations are $Q(B_a, 2)$ and $Q(B_a, 4)$, respectively, where

$$Q(B_a, 2) = \{\{s(1,1), s(2,2)\}, \{s(1,1), s(3,3)\}, \{s(1,1), s(4,4)\}, \{s(1,1), s(5,5)\}, \{s(2,2), s(3,3)\}, \\ \{s(2,2), s(4,4)\}, \{s(2,2), s(5,5)\}, \{s(3,3), s(4,4)\}, \{s(4,4), s(5,5)\}, \{s(4,4), s(5,5)\}, \\ \{s(1,1), s(2,2), s(3,3)\}, \{s(1,1), s(2,2), s(4,4)\}, \{s(1,1), s(2,2), s(5,5)\}, \{s(1,1), \\ s(3,3), s(4,4)\}, \{s(1,1), s(3,3), s(5,5)\}, \{s(1,1), s(4,4), s(5,5)\}, \{s(2,2), s(3,3), \\ s(4,4)\}, \{s(2,2), s(3,3), s(5,5)\}, \{s(2,2), s(4,4), s(5,5)\}, \{s(3,3), s(4,4), s(5,5)\}, \\ s(1,1), s(2,2), s(3,3), s(4,4)\}, \{s(1,1), s(2,2), s(3,3), s(5,5)\}, \{s(1,1), s(2,2), s(4,4), \\ s(5,5)\}, \{s(1,1), s(3,3), s(4,4), s(5,5)\}, \{s(2,2), s(3,3), s(4,4), s(5,5)\}, \{s(1,1), \\ s(2,2), s(3,3), s(4,4), s(5,5)\}\}$$

and

$$Q(B_a, 4) = \{\{s(1,1), s(2,2), s(3,3), s(4,4)\}, \{s(1,1), s(2,2), s(3,3), s(5,5)\}, \{s(1,1), s(2,2), \\ s(4,4), s(5,5)\}, \{s(1,1), s(3,3), s(4,4), s(5,5)\}, \{s(2,2), s(3,3), s(4,4), s(5,5)\}, \\ \{s(1,1), s(2,2), s(3,3), s(4,4), s(5,5)\}\}$$

4 Performance Analysis and Comparison

In this section, we analyze and compare the performance of the DRG technique with grid structure technique on the communication cost and the data availability.

4.1 Availability Analysis

In estimating the availability, all copies are assumed to have the same availability p . Let $SA(t)$ be the read/write availability of technique t . If the probability that an arriving operation of read and write for data file x are f and $(1-f)$, respectively, then the read/write availability can be defined as,

$$SA(t) = fSA(t_{\text{read}}) + (1-f)SA(t_{\text{write}})$$

4.1.1 The GS Technique

Let N be the number of copies which are organized as a grid of dimension $n \times n$. In the case of the quorum technique, read quorums can be constructed as long as a copy from each column is available. Then, the read availability in the GS technique, $A_{GS,R}$, as given in [4] equals to $[1-(1-p)^n]^n$. On the other hand, write quorums can be constructed as all copies from a column and one copy from each of the remaining columns are available. Let $A_{X,Y}$ be the availability with X technique for Y operation, then the write availability in the GS technique, $A_{GS,W}$, as given in [4] equals to

$$[1-(1-p)^n]^n - [(1-(1-p)^n-p^n)^n].$$

Then,

$$SA(GS) = fA_{GS,R} + (1-f)A_{GS,W} \quad (1)$$

4.1.2 The DRG Technique

Let p_i denote the availability of site i . Read operations on the replicated data are executed by acquiring a read quorum and write operations are executed by acquiring a write quorum. For simplicity, we choose the read quorum less than or equals to the write quorum. For example, from Fig. 2 the $D^1(s)$ has five diagonal sites, each of which has vote one, then $C_{DRG,R} = C_{DRG,W} = \lfloor 5/2 \rfloor = 3$.

For any assignment B and quorum q for the data file x , define $\varphi(B_x, q)$ to be the probability that at least q sites in $\Omega(B_x)$ are available, then

$$\begin{aligned} \varphi(B_x, q) &= \Pr \{ \text{at least } q \text{ sites in } \Omega(B_x) \text{ are available} \} \\ &= \sum_{G \in Q(B_x, q)} \left(\prod_{j \in G} p_j \prod_{j \in S(B_x) - G} (1 - p_j) \right). \end{aligned} \quad (2)$$

Thus, the availability of read and write operations for the data file x , are $\varphi(B_x, r)$ and $\varphi(B_x, w)$, respectively. Then,

$$SA(DRG) = f\varphi(B_x, r) + (1-f)\varphi(B_x, w). \quad (3)$$

Fig. 3. Comparison of the write availability between GS and DRG, for $N=25$

Fig. 4. Comparison of the read/write availability between GS and DRG for $p = 0.7$

4.2 Performance Comparisons

In this section, we will compare the performance on the read/write availability of the GS technique based on equations (1), and our DRG technique based on equations (2) and (3) for the case of $N = 25, 36, 49$, and 64 . In estimating the availability of operations, all copies are assumed to have the same availability.

Fig. 3 shows that the DRG technique outperform the GS technique. When an individual copy has availability 70%, write availability in the NRG is approximately 84% whereas write availability in the GS is approximately 60% for $N=25$. Moreover,

write availability in the GS decreases as N increases. For example, when an individual copy has availability 80%, write availability is approximately 86% for $N=25$ whereas write availability is approximately 80% for $N = 49$.

Fig. 4 shows that the DRG technique outperform the GS technique when the probability of an arriving operation is read, $f \leq 0.7$. This shows that the read/write availability is very sensitive to the read and write probability. For example, when $f = 0.5$, read/write availability in GS is approximately 69% whereas read/write availability in the DRG is approximately 87% for $N = 64$. Moreover, read/write availability in the GS decreases as N increases. This is due to the decreases of write availability as N increases. For example, when $f = 0.6$, read/write availability in GS is approximately 83% for $N=25$ whereas read/write availability is approximately 75% for $N = 64$.

5 Conclusions

In this paper, a new technique, called diagonal replication on grid (DRG) has been proposed to manage the data replication in the distributed database systems. The analysis of the DRG technique was presented in terms of read/write availability and communication costs. It showed that, the DRG technique provides a convenient approach to high availability for write-frequent operations by imposing a diagonal binary vote assignment to the logical grid structure on data copies. This is due to the minimum number of quorum size required. This technique also provides an optimal diagonal vote assignment which is less computational time required. In comparison to the Grid structure, DRG requires significantly lower communication cost for an operation, while providing higher read/write availability which is preferred for large distributed database systems.

References

1. J. Holliday, R.Steinke, D.Agrawal, A.El Abbadi, "Epidemic Algorithms for Replicated Databases", *IEEE Trans. On Knowledge and Data Engineering*, vol.15, no.5, (2003), 1218-1238.
2. O. Wolfson, S. Jajodia, and Y. Huang, "An Adaptive Data Replication Algorithm," *ACM Transactions on Database Systems*, vol. 22, no 2 (1997), 255-314.
3. Budiarto, S. Noshio, M. Tsukamoto, "Data Management Issues in Mobile and Peer-to-Peer Environment," *Data and Knowledge Engineering, Elsevier*, 41 (2002),183-204.
4. D Agrawal and A.El Abbadi, "Using Reconfiguration For Efficient Management of Replicated Data," *IEEE Trans. On Knowledge and Data Engineering*, vol.8, no. 5 (1996), 786-801.
5. J.F. Paris and D.E. Long, "Efficient Dynamic Voting Algorithms," *Proc. Fourth IEEE Int'l Conf. Data Eng.*, (1988), 268-275.
6. D Agrawal and A.El Abbadi, "The Tree Quorum technique: An Efficient Approach for Managing Replicated Data," *Proc.16th Int'l Conf. On Very Large Data Bases*, (1990), 243-254.
7. S.Jajodia and D. Mutchles, "Dynamic Voting Algorithms for Maintaining the Consistency of a Replicated Database," *ACM Trans. Database Systems*, vol 15, no. 2 (1990), 230-280.

8. M. Maekawa, "A $\sqrt{n}$ Algorithm for Mutual Exclusion in Decentralized Systems," *ACM Trans. Computer Systems*, vol. 3, no. 2, (1992), 145-159.
9. S.Y. Cheung, M.H. Ammar, and M. Ahmad, "The Grid Technique: A High Performance Schema for Maintaining Replicated Data," *IEEE Trans. Knowledge and Data Engineering*, vol. 4, no. 6, (1992), 582-592.
10. B. Bhargava, "Concurrency Control in Database Systems," *IEEE Trans. Knowledge and Data Engineering*, vol 11, no.1, (1999), 3-16.
11. P.A. Bernstein, V. Hadzilacos, and N.Goodman, *Concurrency Control and Recovery in Database Syatems*, Addison-Wesley, (1987).