Quorum-based Data Replication in Grid Environment

Rohaya Latip*, Mohamed Othman*, Azizol Abdullah*,
Department of Technology Communication and Network, Faculty of Computer Science and Information Technology,
University Putra Malaysia,
*Serdang, 43400, Selangor,Malaysia **
E-mail: {rohaya, azizol, Mothman}@fsktm.upm.edu.my
www.fsktm.upm.edu.my

Hamidah Ibrahim, Md Nasir Sulaiman±
Department of Computer Science, Faculty of Computer Science and Information Technology,
University Putra Malaysia,
Serdang, 43400, Selangor,Malaysia ±
E-mail: {hamidah, nasir}@fsktm.upm.edu.my
www.fsktm.upm.edu.my

Received: 04/01/09
Accepted: 06/09/09

Abstract

Replication is a useful technique for distributed database systems and can be implemented in a grid computation environment to provide a high availability, fault tolerant, and enhance the performance of the system. This paper discusses a new protocol named Diagonal Data Replication in 2D Mesh structure (DR2M) protocol where the performance addressed are data availability which is compared with the previous replication protocols, Read-One Write-All (ROWA), Voting (VT), Tree Quorum (TQ), Grid Configuration (GC), and Neighbor Replication on Grid (NRG). DR2M protocol is organized in a logical two dimensional mesh structure and by using quorums and voting techniques to improve the performance and availability of the replication protocol where it reduce the number of copies of data replication for read or write operations. The data file is copied at the selected node of the diagonal site in a quorum. The selection of a replica depends on the diagonal location of the structured two dimensional mesh quorum where the middle node is selected because it is the best location to get a copy of the data if every node has the equal number of request and data accessing in the network. The algorithm in this paper also calculates the best number of nodes in each quorum and how many quorums are needed for *N* number of nodes in a network. DR2M protocol also ensures that the data for read and write operations is consistency, by proofing the quorum must not have a nonempty intersection quorum. To evaluate DR2M protocol, we developed a simulation model in Java. Our results prove that DR2M protocol improves the performance of the data availability compare to the previous data replication protocol, ROWA, VT, TQ, GC and NRG.

Keywords: Data Replication, Grid, Data Management, Availability, Replica Control Protocol.

1. Introduction

A grid is a distributed network computing system, a virtual computer formed by a networked set of heterogeneous machines that agree to share their local resources with each other. A grid is a large scale, generalized distributed network computing system that can scale to internet size environment with machines distributed across multiple organizations and administrative domains [1, 2]. Ensuring efficient access to such a large network and widely distributed data is a challenge to those who design, maintain, and manage the grid network. The availability of a data on a large network is an issue [3, 4, 5, 6] because geographically it is distributed and has different database management to share across the grid network whereas replicating data can become expensive if the number of operations such as read or write operations is high.

Distributed computing manages thousands of computer systems and this has limited its memory and processing power. On the other hand, grid computing has some extra characteristics. It is concerned to efficient utilization of a pool of heterogeneous systems with optimal workload management utilizing an enterprise's entire computational resources (servers, networks, storage, and information) acting together to create one or more large pools of computing resources. There is no limitation of users or originations in grid computing. Even though grid sometime can be as minimum one node but for our protocol the best number of nodes should be more than five nodes to implement the protocol. This protocol is suitable for large network such as grid environment.

There are some research been done for replica control protocol in distributed database and grid such as Read-One Write-All (ROWA) [7], Voting (VT) [8], Tree Quorum (TQ) [9, 10], Grid Configuration (GC) [11, 12], and the latest research in year 2007 is Neighbor Replication on Grid (NRG) [13, 14, 15]. Each protocol has its own way of optimizing the data availability. The usage of replica is to get an optimize data accessing in a large and complex grid. Fig. 1 illustrates the usage of replica protocol in grid where it is located in the replica optimization component [16].

In this paper, a new quorum-based protocol for data replication namely DR2M protocol, is introduced to improve its ability to access data efficiently. The DR2M protocol imposes a logical two dimensional mesh structure to produce the best number of quorums and obtain good performance of data availability.

Quorums improved the performance of fault tolerant and availability of replication protocols [18]. Quorums reduce the number of copies involved in reading or writing data. To address the availability, replicated data are stored at the selected node from the diagonal site of the two dimensional mesh structured, which has been organized in quorums.

The paper is organized as follows. Section two discusses on the related works on replica control protocol. Section three, introduces Diagonal Replication in 2D Mesh (DR2M) protocol as a new replica control protocol. Section four, illustrates the implementation of DR2M. Section five, discusses the results by comparing DR2M with the existing protocols and the paper ends with a conclusion.

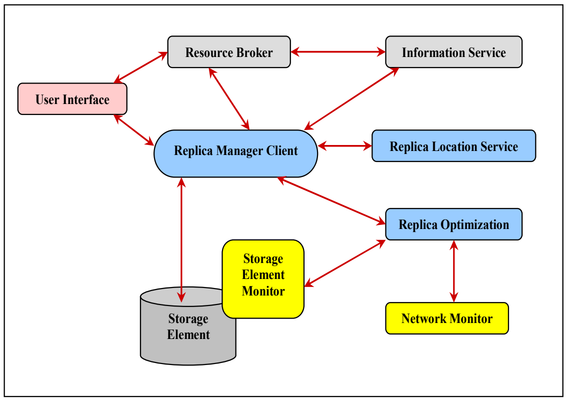

Fig. 1. Grid Components Interact with Replica Manager in Grid.

2. Related Works

2.1 *Read One -Write All (ROWA)*

Read One-Write All (ROWA) protocol is an approach proposed by Bernstein and Goodman in year 1984. In ROWA read operation needs only one copy. Meanwhile, a write operation needs to access a number of copies, *n*. ROWA is commonly used in distributed database to improve performance and availability of mostly read data. A read operation can be done from any replica in the system and therefore is very efficient and high in read availability. The communication cost for read operations are low because only one replica is accessed by the read operation [19].To ensure consistency of the data, write operation is done on all replicas thus ROWA has low in write availability. Due to this, the communication cost of write operation for ROWA is very high.

Write operation update data and all replicas have the same values when an update transaction commits. ROWA has a significant drawback which is if one of the replicas is unavailable then an update transaction cannot be terminated. As an example, ROWA is popular and has been used in mobile, peer to peer environment [20,21], database system [22], and grid [23, 9, 24, 25].

ROWA requires a read on any one of the copies and a write on all copies in order to confirm the read-write intersection property [8]. Therefore, the read and write availability of ROWA can be represented as one out of n and n out of n, respectively. Thus, the read availability of ROWA, $A_{ROWA,\ R}$ is as in Eq. (1). Where p is the probability of data file accessing between 0.1 to 0.9 and i is the increment of n.

$$A_{ROWA,\,R} = \sum_{i=1}^{n} \binom{n}{i} p^{i} (1 - p)^{n-i}$$

$$= 1 - (1 - p)^{n} \tag{1}$$

And the write availability $A_{ROWA,\ W}$ is as in Eq. (2).

$$A_{ROWA,\,W} = \sum_{i=n}^{n} \binom{n}{i} p^{i} (1 - p)^{n-i}$$

$$= p^{n} \tag{2}$$

If the probability that an arriving operation of read and write for the data file x are f and $(1\text{-}f)$ respectively, then the system availability of ROWA protocol, $SA(ROWA)$ is as in Eq. (3).

$$SA(ROWA) = f A_{ROWA,\,R} + (1 - f)\, A_{ROWA,\,W} \tag{3}$$

For example, with n = 36, p = 0.7 and f = 0.7, then $A_{ROWA,\ R}$ = 1, $A_{ROWA,\ W}$ = 2.652E-06 and $SA(ROWA)$ = 0.7.

2.2 *Voting Protocol (VT)*

This protocol was first presented by Thomas [26]. The idea of voting was later enhanced by Molina and Barbara in the year 1985. In their protocol, a certain number of votes, v, are assigned to each copy of a replicated data object. An invoking transaction has to obtain a read quorum of r votes to read a data object, and a write quorum of w votes to write the data object.

In particular, a quorum must obey the Quorum Intersection Properties rules [8] as follow:

i. $r + w > v$
ii. $w > v/2$

The first rule ensured that there is non empty intersection between read and write quorums. Each read quorum is guaranteed to have a current copy of the particular data object. A data object could be read and write by two transactions concurrently. Thus, the read and write are conflict is avoided. Meanwhile, the second rule prevents the write operations occur in two different partitions for the copies of the same data object. Write operations cannot occur concurrently. Therefore, the write-write conflict is avoided. Voting protocol obeys one-copy serializability due to the two rules above. Multiple copies of a data object appear as a single logical data object [27]. Thus, it fulfills the correctness criterion for replication database. In this protocol, the read availability is given a high priority where r votes are selected. The weakness of this protocol is that the write operation is fairly expensive because a write quorum must be larger than the majority votes. The sizes for both read and write operations can be the same in majority consensus such as the majority of the total number of votes assigned to the copies [28].

Weight Voting technique proposed by Gifford [29] works with any concurrency control algorithm, which produces serializable executions and the communication cost is reliant on the number of votes pre-selected for read and write operations. In this protocol, T is the last transaction that wrote a copy of data object d, T' is a transaction that comes after T. If the transaction T' reads a copy of data object d, then it can be concluded that it reads from previous transaction T. This is because, T wrote into a write quorum of d; T' reads from a read quorum of d; each of read and write quorum have a non empty intersection. In this protocol there is a significant drawback where multiple sites must be polled before every read transaction can be executed. Therefore, the workload with high proportion of read will not perform well under weighted majority quorum consensus [29].

Dynamic Voting protocol is an enhancement of Weight Voting proposed by Jajodia and Mutchler [30]. The protocol modifies the number of votes assigned to each replica. Long and Paris in their paper [31] mention that the Voting protocol suffers from some major drawback such as it requires at least three voting entities to improve upon the availability afforded by a single replica and Voting protocol also provides poor data availability compared to other replica control protocol. The Dynamic Voting protocol was complicated to be implemented and required complex metadata [31]. Thus, Long and Paris proposed the Voting protocol using cohort sets and requiring only $n + log\ (n)$ bits of state per voting entity.

By using cohort set, updates whenever a change in the availability of the replicas are detected. By using cohort sets, Voting protocol reduces the update cost of witnesses since witnesses would only need to be updated whenever a change in the accessibility of a replica is detected.

Unlike Weight Dynamic Voting, Dynamic Liner Voting does not rely on weight assignment to accommodate this situation and use an arbitrary but tie breaking rule. The sites that hold replicas are given a static linear ordering. Then when a tie occurs, if the group of communicating sites contains exactly one half the current replicas and the group contains the largest sites among the group of current replicas, the group is declared to be the majority block.

For n replicas, VT allows n choices for the read and the write quorums from read 1, write n to read n, write 1. When a small read quorum is selected, the read availability is high. However, the write availability becomes low. This demonstrates a trade-off between the read and the write availability in VT. To avoid the read availability becomes expensive and high, a read quorum k is selected (k is smaller than the majority quorum). In this case, $k = \lceil n/4 \rceil$ is selected by Mat Deris et al. [14]. Thus, the read availability of VT, $A_{VT,R}$ is as in Eq.(4) and the corresponding write availability, $A_{VT,\ W}$ is as in Eq. (5).

$$A_{VT,\ R} = \sum_{i=k}^{n} \binom{n}{i} p^{i}\ (1 - p)^{n-i},\ k \geq 1 \qquad (4)$$

$$A_{VT,\ W} = \sum_{i=n+1-k}^{n} \binom{n}{i} p^{i}\ (1 - p)^{n-i},\ k \geq 1 \qquad (5)$$

If the probability that an arriving operation of read and write for the data file x are f and $(1 - f)$ respectively, then the system availability of VT is as in Eq. (6).

$$SA(VT) = f A_{VT,\ R} + (1 - f)\ A_{VT,\ W} \qquad (6)$$

For example, if $n = 36$ and $p = 0.7$ and $f = 0.7$, then $A_{VT,\ R} = 1$, $A_{VT,\ W} = 0.4206$, $SA(VT) = 0.761$ when $k = 4$.

2.3 *Tree Quorum (TQ)*

Agrawal and El Abbadi proposed the Tree Quorum (TQ) protocol in the year 1990. This protocol is based on a logical tree structure over a network. Its read operation may access only one copy. Meanwhile the number of copies to be accessed by a write operation is always less than a majority of quorum. Fig.2 illustrates the tree quorum organized in nine copies of data object with 3 levels where {1} is at level 0 and {2, 3} is at level 1 and {4,5,6,7,8,9} is at level 2.

For the read operations, the root or the majority of the children of the root can form a read quorum. If any node in the majority of the children of that root fails, the majority of the children and so on recursively can replace it. In the best case, a read quorum consists of only a root. Example in Fig. 2, the best case is to have one root such as {1}.

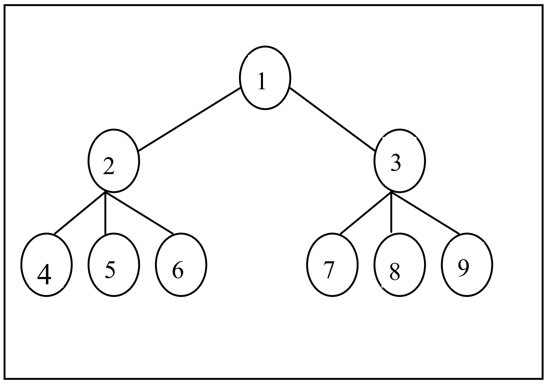

Fig. 2. Tree quorum organization of nine copies of data object

If the root at level 2 fails, a quorum is formed by the majority of the children at level 1, example the child at level 1 is {2, 3}. If site 2 fails but site 3 is available, the majority of the children replaces site 2. Thus, read quorums are {3, 4, 5} or {3, 5, 6} or {3, 7, 8} etc. If there is a failure at level 0 and 1, a quorum is formed by the majorities of the children of the selected majority at level 2. The set of copies {4, 5, 7, 9} or {5, 6, 8, 9} are the valid quorums. Thus, the size of a read quorum is at most equal to $|\{4, 5, 7, 8\}|$ or $|\{5, 6, 8, 9\}| = 4 = 2^2 = M^h$, h is the height of the tree. The degree of the copies in the tree is indicated by D. Meanwhile, $M = \lceil (D + 1) / 2 \rceil$ is the majority of the degree of copies. When the root is accessible, a read quorum size is equal to 1. As the root fails, the majority of its children replaced it. Thus, the quorum size increases to M. Therefore, for a tree of height h, the maximum quorum size is M^h [7].Therefore $C_{TQ,\ R}$ is in the range of $1 \le C_{TQ,\ R} \le M^h$.

The write operations in TQ must obtain a write quorum. It is formed from the root, a majority of its children, a majority of their children and so forth until the leaves of the tree are reached. Even though the size of the write quorum from a given tree is fixed, the members can be different. Example in Fig. 3, where if the size of write quorum is six thus the member of the write quorum could be {1, 2, 4, 5, 7, 8}, {1, 3, 5, 6, 8, 9},{1, 2, 3, 5, 7, 9}, {1, 3, 4, 9, 10, 11, 12}, which is to 6. Thus $\sum M^i$, $i = 0, \ldots, h$.

Agrawal and El Abbadi [7] found that the availability of write operations is significantly better than ROWA. Nevertheless, there is a limitation in TQ where if more than a majority of the copies at any level of the tree becomes unavailable, write operations cannot be executed.

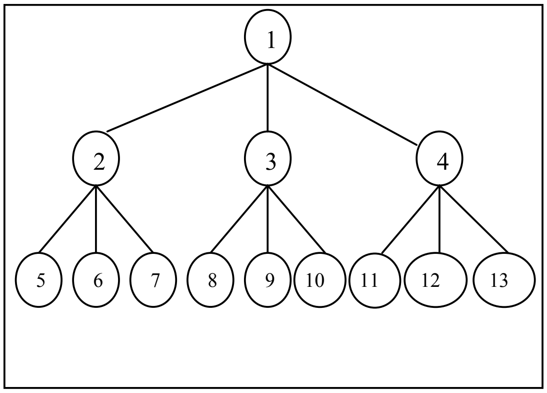

Fig. 3. A tree organization of 13 copies of a data object

The availability of read and write operations in the TQ can be estimated by using recurrence equations based on the tree height, *h,* [32]. Let $A_{TQ,R}$ and $A_{TQ,W}$ be the availability of read and write operations with a tree of height, *h*. *D* denotes the degree of sites in the tree and *M* is the majority of *D* as in Eq. (7).

$$A_{TQ,\ R_h} = p + (1 - p) \sum_{i=M}^{D} \binom{D}{i} A_{TQ,R_{h-1}}^{\ i} (1 - A_{TQ,\ R_h})^{D-i} \qquad (7)$$

The availability of a write operation for a tree of height *h* as given in Eq. (8).

$$A_{TQ,W_h} = p \sum_{i=M}^{D} \binom{D}{i} A_{TQ,W_{h-1}}^{\ i} (1 - A_{TQ,W_{h-1}})^{D-i} \qquad (8)$$

where *p* is the probability that a copy is available, and R_0 is equal to W_0 which is also equal to *p*.
If the probability that an arriving operation of read and write for the data file *x* are *f* and (1 - *f*) respectively, then the system availability of TQ protocol is as in Eq. (9).

$$SA(TQ) = f A_{TQ,R_h} + (1 - f) A_{TQ,W_h} \qquad (9)$$

For example, if *n* is equal to 13 where *D* = 3, *M* = 2 and *h* = 2, *p* is equal to 0.7 and *f* is equal to 0.7, then $A_{TQ,R_h} = 0.996$, $A_{TQ,W_h} = 0.401$, and $SA(TQ) = 0.818$.

2.4 *Grid Structure (GS)*

Grid Structure (GS) protocol was proposed by Maekawa [11] in the year 1992 where all quorums are in equal size. In year 1992, Cheung et al. [12] has extended Maekawa's grid structure protocol. In particular, the read and write operations are supported in managing replicated data. Through this protocol, *n* copies of a data object are logically organized in the form of two dimensional $\sqrt{n}$ x $\sqrt{n}$ grid structure. Fig. 4 illustrates 25 copies of data object.

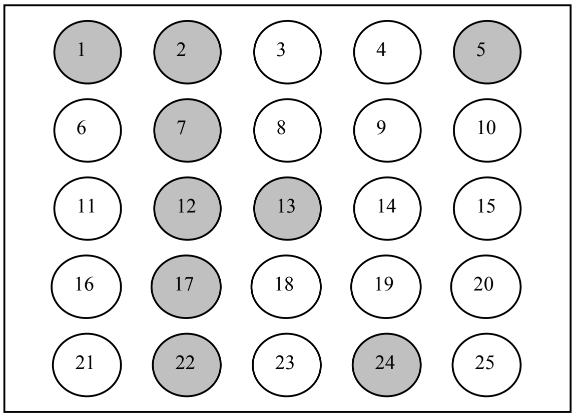

Fig. 4. A grid structure with 25 copies of a data object

Let *n* be the number of copies which is organized as a grid dimension $\sqrt{n}$ x $\sqrt{n}$. The read operations on the data item are executed by acquiring a read quorum. It considers a copy from each column. Meanwhile, write quorum is required for the write operations to be executed. It consists of all copies in one column and a copy from each of the remaining columns in grid as illustrated in shaded gray circles in Fig. 4. In particular the size of the read and write operations is $O(\sqrt{n})$. A grid quorum has a length of l and width w, and is presented as a pair $<l,w>$.

GS has its read and write quorum. The read quorums can be constructed as long as a copy from each column is available. The read availability in the GS protocol, $A_{GS,R}$ is as in Eq. (10) where n is the number of copies in the system and p is the probability of accessing the data which is between 0 to 1, as given by Agrawal and El Abbadi [7]and Ahmad [33].

$$A_{GS,R} = \left[1-(1-p)^{\sqrt{n}}\right]^{\sqrt{n}} \tag{10}$$

On the other hand, write quorums can be constructed as all copies from a column and one copy from each of the remaining columns available. Then, the write availability in the GS, $A_{GS,W}$ is as in Eq. (11).

$$A_{GS,W} = \left[1-(1-p)^{\sqrt{n}}\right]^{\sqrt{n}} - \left[(1-(1-p)^{\sqrt{n}} - p^{\sqrt{n}}\right]^{\sqrt{n}} \tag{11}$$

If the probability that an arriving operation of read and write for the data file x are f and $(1 - f)$, respectively, then the system availability of GS protocol is as in Eq. (12).

$$SA(GS) = f A_{GS,R} + (1 - f)\, A_{GS,W} \tag{12}$$

For example, if n is equal to 36 number of copies and p is equal to 0.7, whereas f is equal to 0.7, then $A_{GS,R}$ is equal to 0.526 and $A_{GS,W}$ is equal to 0.996 and $SA(GS)$ is equal to 0.855.

2.5 *Neighbor Replication on Grid (NRG)*

NRG proposed by Ahmad [33] in the year 2007 is treading a new path in replication that aims to maximize the system availability with low communication cost. This protocol considers only the neighbor can obtain a data copy by assigning votes to the neighbor as one. Quorum approach is deployed where read operations can only access the read quorums which assigned vote one to it and on the other hand, the write operations execute the write quorums only.

All sites are logically organized in the form of a two-dimensional grid structure. For example, if there are nine sites in the network, it will be logically organized in the form of 3 x 3 grids as shown in Fig. 5. Each site has a replica copy which is assumed as the master data file. 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.

In this protocol the availability of read and write operations for the data item x are $\varphi(B_x, R)$ and $\varphi(B_x, W)$, respectively. Since both read and write operations are assumed to be similar thus the availability for both operations is defined as $\varphi(B_x, q)$ as shown in Eq. (13) where B denotes as transaction, q as quorum for the data x and the probability that at least q sites in $S(B_x)$ are available [33].

$$\varphi(B_x, q) = Pr\ \{\text{at least } q \text{ sites in } S(B_x) \text{ are available}\}$$

$$= \sum_{G\in Q(B_x,q)} \left(\prod_{j\in G} p_j \prod_{j\in S(B_x)-G} (1-p_j) \right) \tag{13}$$

NRGs limitation is that the number of member in a quorum is five or three depending on which location of the grid it is located. Example illustrated in Fig. 5 has 2 neighbors for node 3, where its neighbors are node 2 and node 6. For node 5, it has 4 neighbors which are node 2, 4, 6, and node 8 but the data files are replicated at all the neighbor nodes and the node itself making it 5 numbers of data file in the network. Each neighbor is a quorum member of that particular node and every node has neighbors and connected thus the number of replicated data file in the whole network is high and is increasing if the network size is getting larger even though the quorum size is small. This will also increase the replication cost because this protocol needs to replicate at all of its neighbors.

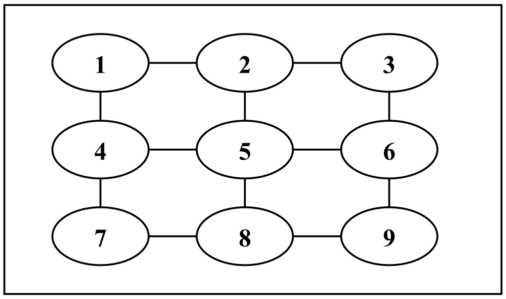

Fig. 5. A grid organization of 9 copies of a data object

Table 1 has the summarization of all the protocols in the replica control protocol for distributed database and grid environment. This has helped to develop a new protocol that has high data availability for read and write operation which will be discussed in the following section.

Table 1. Summarized existing data replica control protocol

Protocol	Approaches	Limitations
ROWA	-Read operation needs only one copy -Write operation needs to access numbers of copies(n).	-It reduces the availability of the database if one of the sites / database fail, the write / update cannot be done
VT	-Every copy of replicated data object is assigned certain number of votes. - If a read operation is to be done, the read transaction must collect a read quorum of r votes. - If a write operation is executed, the number of write quorum of w votes must be collected.	- writing an object is fairly expensive, because write quorum (w) of copies must be larger than the majority votes (v), $w > v/2$
TQ	- Logical tree structured over the network. - Read operation may access only one copy - Number of copies to access by write operation is always less than a majority of quorum - Write operations must access the write quorum, which can be the root, or their child. - The size of a quorum is fix but the member can be different - The root or child can also form a read operations.	If more than a majority of the copies in any level of the tree become unavailable, write execution cannot be done
GS	- It is logically organized in form of $\sqrt{n}$ x $\sqrt{n}$ grid - The black sites is the one that is down or not active, which can be placed anywhere in the structure. - Read operations are executed by acquiring a read quorum that consist of copy from each column -Write operation are executed by acquiring a write quorum that consists of all copies in one column and each copy from the remaining column	- Degrading the communication cost and data availability because of the protocol will require a bigger number of read and write quorum - For read operation to be executed, read quorum must be at every column -If write operations are executed, write quorum must exist at one of the entire column and at least once at other columns

Table 1. Summarized existing data replica control protocol (Continue)

Protocol	Approaches	Limitations
NRG	It is logically organized in a grid structure and only the node and its neighbors are the member of the quorum.	- The data file a replicated in a big number because every node has neighbors and connected to one another. - The quorum size is small but the number of quorums is high since every neighbor is a member to their neighbor's quorums. This has increase the replication cost.

3. Diagonal Replication in 2D Mesh (DR2M) Protocol

In DR2M protocol, all nodes are logically organized into two dimensional mesh structures. Assuming that the replica copies are in the form of data files and all nodes are operational meaning that the copy at the nodes is available. The data file is replicated to only one middle node at the diagonal site of each quorum. Selecting the middle node makes it easier to access the data file from all dimension.

This protocol uses quorum to arrange nodes in cluster. Voting approach assigned every copy of replicated data object a certain number of votes and a transaction has to collect a read quorum of *r* votes to read a data object, and a write quorum of *w* votes to write the data object. In this approach the quorum must satisfy two constraints in the Quorum Intersection Properties [8] as mention in Section 2.

Quorum is grouping the nodes or databases as shown in Fig. 6. This figure illustrates how the quorums for network size of 81 nodes are grouped by nodes of 5 x 5 in each quorum. Nodes which are formed in a quorum intersect with other quorums. This is to ensure that these nodes can communicate and read or write other data from other nodes which are in another quorum. The number of nodes grouped in a quorum, *q* must be odd so that only one middle node from the diagonal site can be selected such as site *s*(3,3) colored in black circles in Fig. 6. Site *s*(3,3) has the copy of the data file for read and write operation to be executed.

Definition 1. *Assume that a database system consists of n x n nodes that are logically organized in the form of two dimensional grid structures. All sites are labeled s(i,j), 1 ≤ i ≤ n, 1 ≤ j ≤ n. The diagonal site of s(i,j) is s(n,n), where n = 1, 2, ..., ∞.*

From Fig. 6, for q_1, the nodes of the diagonal site, *D(s)* is {*s*(1,1), *s*(2,2), *s*(3,3), *s*(4,4), *s*(5,5)} in each quorum and the middle node *s*(3,3) has the copy of the data file. This figure shows that 81 nodes have four quorums where each quorum actually intersects with each other. Node *y* in q_1 is actually node *u* in q_2, node *e* in q_3 and node *a* in q_4.

Since the data file is replicated only on one node for each quorum, thus it has minimizes the number of database operations. The selected node in the diagonal sites is assigned with vote one or vote zero. A vote assignment on grid, *B*, is a function such that, $B(s(i,j)) \in \{0, 1\}$, $1 \le i \le n$, $1 \le j \le n$ 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 vote ≡ 1 copy. Let $L_B = \sum B(s(i,j)), s(i,j) \in D(s)$

where L_B is the total number of votes assigned to the selected node as a primary replica in each quorum. Thus $L_B = 1$ in each quorum.

Definition 2. *For a quorum q, a quorum group is any subset of S(B) where the size is greater than or equal to q. The collection of quorum group is defined as the quorum set.*

Let $Q(B_m,q)$ be the quorum set with respect to assignment *B* and quorum *q*, then $Q(B_m,q) = \{G|\ G \subseteq S(B) \text{ and } |G| \ge q\}$.

For example, from Fig. 6, let site *s*(3,3) be the primary database of the master data file *m*. Its diagonal sites are *s*(1,1), *s*(2,2), *s*(3,3), *s*(4,4), and *s*(5,5). Consider an assignment *B* for the data file *m,* where $B_m(s(i,j))$ is the vote assigned to site *s(i,j)* and $L_{B,m}$ is the total number of votes assigned to primary database in each quorum which is (s(3,3)) for data file m, such that $B_m(s(1,1)) = B_m(s(2,2)) = B_m(s(3,3)) = B_m(s(4,4)) = B_m(s(5,5)) = 1$ and $L_{B,m} = B_m(s(3,3))$. Therefore, $S(B) = \{s(3,3)\}$.

As assumed, the read quorum for the data file m, is equal to write quorum. The quorum sets for read and write operations are $Q(B_m,q_1)$, $Q(B_m,q_2)$, $Q(B_m,q_3)$, and $Q(B_m,q_4)$, respectively, where $Q(B_m,q_1) = \{s(3,3)\}$, $Q(B_m,q_2) = \{s(3,3)\}$, $Q(B_m,q_3) = \{s(3,3)\}$, and $Q(B_m,q_4) = \{s(3,3)\}$. Therefore, the number of replicated data file m is four.

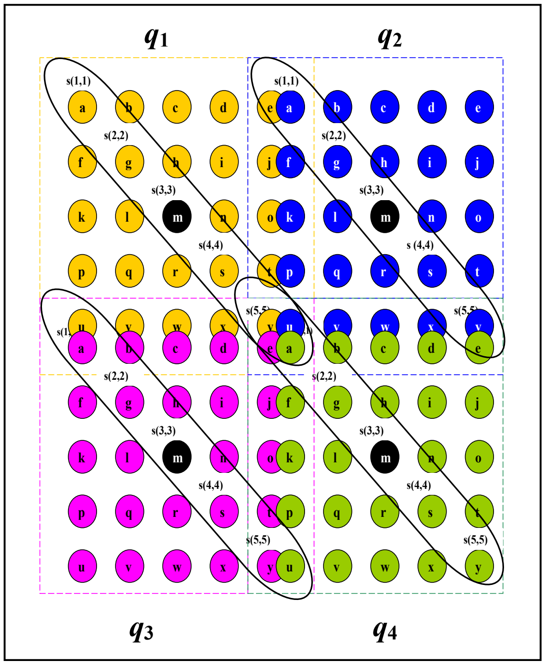

Fig. 6. A grid organization with 81 nodes, each of the node has a data file a, b,…, and y, respectively.

4. Simulation Design

DR2M was developed using Java. The algorithm of the model is as in Fig. 7. It illustrates how the algorithm is designed and implemented.

For the development of the simulation some assumptions were made such as the network has no failure and nodes are accessible.

To analyze the performance of read and write availability, below are the equations where n is the two dimensional mesh column or row size, example n is 7, thus 7 x 7 nodes of grid is the network size and p is the probability of data available which is between 0 to 1 whereas, q is the number of quorum for a certain operation such as read or write operation. Eq. (14) is to calculate the data availability.

$$\mathrm{Av},_{q} = \sum_{i=q}^{n} \binom{n}{i} (p^{i}(1-p)^{n-i}) \qquad (14)$$

```
Main
Input number of row or column, $N$
If $\sqrt{N}$ is odd integer then
        Find the number of quorum, $Q$
        $Q = \left\lfloor \sqrt{n} - \frac{n}{10} \right\rfloor$
        Find number of nodes in each quorum, $X$
        $X = n/Q$
        Get the next odd integer after $X$
        Select the middle replica, $R$
        Copy the data file at $R$
Else
        Add one virtual column and row ,
        $Col_{new} = Col + 1$ , $Row_{new} = Row + 1$
        Return new $N$ to Main
```

Fig. 7. Algorithm of DR2M protocol.

A larger network size has more quorums. The number of columns and rows in each quorum must be odd, to get the middle replica. DR2M protocol assumes all nodes have the same data access level and requested number of nodes.

The number of nodes in the network must be odd and if it is an even number then a virtual column and row is added meaning an empty node of rows and column are located to make the selection of the middle node easier. The pseudo code to add new column and row are shown in Fig. 8 where if the dynamic network increases the number of nodes it will be allocated at the existing empty nodes and if the existing empty nodes is lesser then the new number of nodes then new virtual column and row will be added.

```
If number of new nodes ≤ empty nodes then
  Assign new nodes to the empty Nodes
Else
  Add new column and row
End.
```

Fig. 8. Pseudo code to add new column and row

To find the best number of quorum, Q, for the whole network size, n x n, Eq. (15) is used where n is the number of nodes for row or column.

$$Q = \left\lfloor \sqrt{n} - \frac{n}{10} \right\rfloor \qquad (15)$$

For obtaining the best number of nodes for each quorum, X, Eq. (16) is used where the number of nodes must be odd to make the selection of the middle node easy and if X is not odd then X will be the next odd number after n/Q.

After the selected node is chosen, the data file is copied where it acts as a primary database for that particular quorum. Some protocol depends on the frequent usage of the data to select that particular node as the primary database. But for this protocol, an assumption is made where every node has the same level of data access and number of requested nodes.

5. Results and Discussion

In this section, DR2M protocol is compared with the results of read and write availability of the existing protocols, namely: ROWA, VT, TQ, GC, and NRG. Fig. 9 shows the results of read availability in 81 nodes of network size. ROWA protocol has the higher read availability about average of 2.425% for probability of data accessing between 0.1 to 0.9 even when the number of nodes is increased. This is because only one replica is accessed by a read operation for all n nodes of network size but ROWA has the lowest write availability.

$$X = {}^{n}\!/_{Q} \qquad (16)$$

Fig. 10 proves that the DR2M protocol has 4.163% higher of write availability for all probabilities of data accessing. This is due to the fact that replicas are selected from the middle location of all nodes in each quorum and by using quorum approach helps to reduce the number of copies in a large network.

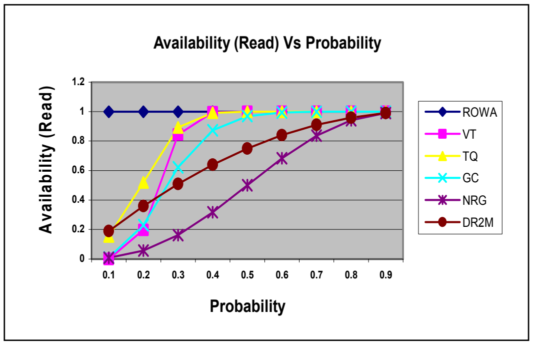

Fig. 9. Read availability results.

Fig. 10 illustrates the write availability for 81 numbers of nodes, where the probability is from 0.1 to 0.9.

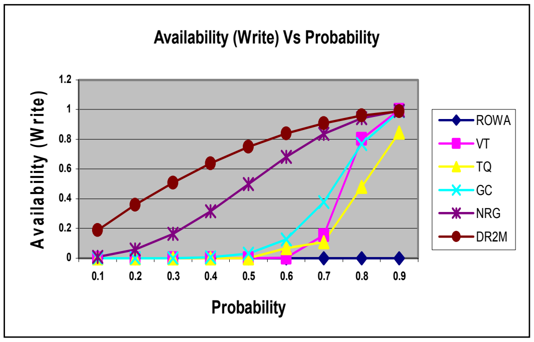

Fig. 10. Write availability results.

6. Conclusions

In this paper, DR2M protocol selects a primary database from the middle location of the diagonal site where the nodes are organized in structure of 2D Mesh. The middle location is selected because it is easy to access the data file from all dimension and assuming all nodes has the same priority for requesting or accessing the data file. By getting the best number of quorum and using the voting techniques have improved the availability for write operation compared to all protocols and has higher read availability compared to the latest technique, NRG. In the future, we will investigate the response time to access a range of data size in the grid environment and this investigation will be used to evaluate the performance of our DR2M protocol.

Acknowledgements

We would like to thank you the Malaysian Ministry of Science, Technology and Innovation (MOSTI) for supporting our project under the Fundamental Grant No: 02-01-07-269FR.

References

1. K. Krauter, R. Buyya, M. Maheswaran, A Taxanomy and Survey of Grid Resource Management Systems for Distributed Computing, *International Journal of Software Practice and Experience*, 32(2) (2002) 135-164.
2. I. Foster, C. Kesselman, J. Nick, S. Tuecke, Grid Services for Distributed System Integration, *Computer*, 35(6) (2002) 37-46.
3. K. Ranganathan and I. Foster, Identifying Dynamic Replication Strategies for a High Performance Data Grid, in *Proceedings of International Workshop on Grid Computing,* (Denver, 2001), pp 75-86.
4. H. Lamehamedi, B. Szymanski, Z. Shentu and E. Deelman, Data Replication Strategies in Grid Environment, in *Proceedings of ICAP'03*, (Beijing China, 2002), pp. 378-383
5. H. Lamehamedi, Z. Shentu, and B. Szymanski, Simulation of Dynamic Data Replication Strategies in Data Grids, in *Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, (Nice, France, 2003), pp. 22-26.
6. H. Lamehamedi, Decentralized Data Management Framework for Data Grids. Ph.D. thesis, Rensselaer Polytechnic Institute Troy, New York (2005).
7. D. Agrawal and A. El Abbadi, Using Reconfiguration for Efficient Management of Replicated Data, *IEEE Transactions on Knowledge and Data Engineering,* 8(5) (1996) 786-801.
8. M. Mat Deris, Efficient Access of Replication Data in Distributed Database Systems. Thesis PhD, Universiti Putra Malaysia (2001).
9. D. Agrawal and A. El Abbadi, The Generalized Tree Quorum Protocol: An Efficient Approach for Managing Replicated Data, *ACM Transactions Database System*, 17(4) (1992) 689-717.
10. D. Agrawal and A. El Abbadi, The Tree Quorum Protocol: An Efficient Approach for Managing Replicated Data, in *Proceeding 16th International Conference on Very Large databases*, (Brisbane, Australia, 1990) pp. 243-254.
11. M. Maekawa, A √n Algorithm for Mutual Exclusion in Decentralized Systems, *ACM Transactions Computer System.* 3(2) (1992) 145-159.
12. S.Y. Cheung, M.H. Ammar and M. Ahmad, The Grid Protocol: A High Performance Schema for Maintaining Replicated Data, *IEEE Transactions on Knowledge and Data Engineering.* 4(6) (1992) 582–592.
13. M. Mat Deris, D.J. Evans, M.Y. Saman and N. Ahmad, Binary Vote Assignment on Grid For Efficient Access of Replicated Data, *Int'l Journal of Computer Mathematics*. 80 (2003) 1489-1498.
14. M. Mat Deris, J.H. Abawajy and H.M. Suzuri, An Efficient Replicated Data Access Approach for Large Scale Distributed Systems, in *IEEE/ACM Conf. On Cluster Computing and Grid (CCGRID2004),* (Chicago, USA, 2004), pp 588-594.
15. N. Ahmad and M. Mat Deris, Managing Neighbor Replication Transactions in Distributed Systems, in *DCABES 2006,* (China, 2006), pp 95-101.
16. P. Kunszt, E. Laure, H. Stockinger and K. Stockinger, Advanced Replica Management with Reptor, *Parallel Processing and Applied Mathematics.* 3019 (2004) 848-855.
17. M. Mat Deris, J.H. Abawajy and A. Mamat, An Efficient Replicated Data Access Approach for Large-scale Distributed Systems, *Future Generation Computer Systems*. 24 (2007) 1-9.
18. R. Jimenez-Peris, et al., Are Quorums an Alternative for Data Replication?, in *ACM Transactions on Database System.* 28(3) (2003) 257-294.
19. M. Rabinovich and E. Lazowska, An Efficient And Highly Availability Read-One Write All Protocol For Replication Data Management, in *Proceedings Of The Second International Conference on Parallel and Distributed Information Systems*, (United States, 1993), pp 56-65.
20. S.K. Madria, M. Moania, B. Bharat, and B. Sourar, Mobile Data Transactions, *Journal of Information Sceince*. 141 (2002) 279-309.
21. S. Budiarto and N.M. Tsukamoto, Data Management Issues in Moblie and Peer To Peer Environement. *Data and Knowledge Engineering, Elsiver.* 41 (2002) 391-402.
22. W. Zhou, and A. Goscinki, Managing Replication Remote Procedure Call Transaction, *The Computer Journal.* 42(7) (1999) 592-608.
23. A. Kumar, and S.Y. Cheung, A high Availability $\sqrt{N}$ Hierarchical Grid Algorithm for Replicated Data, *Information Processing Letters*. 40(6) (1991) 311-316.
24. P.Z. Kunszt, E. Laure, H. Stockinger, and K. Stockinger, Next-Generation EU DataGrid Data Management Services. *Computing in High Energy and Nuclear Physics,* (La Jolla, California, 2003), pp 24-28.
25. P.Z. Kunszt, E. Laure, H. Stockinger, and K. Stockinger, File Based Replica Management, *Future Generation Computer System*. 21(1) (2005) 115-123.
26. R.H. Thomas, A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases, *ACM Transaction Database System*. 4(2) (1979) 80-229.
27. P. A. Bernstein and N. Goodman, An Algorithm for Concurrency Control and Recovery in Replicated Distributed Database, *ACM Transaction Database Systems.* 9(4) (1984) 596-615.
28. E. Ramez, and B. N. Shamkant, Fundamental of Database Systems (Addison Wesley, UK, 2003).
29. D.K. Gifford, Weighted Voting for Replicated Data, in *Proceeding 7th Symposium On Operating System Principles*, (California, USA, 1979), pp 150-162.
30. S. Jajodia and D. Mutchler, Dynamic voting algorithms for maintaining the consistency of a replicated database, *ACM transactions database systems*. 15(2) (1990) 230-280.
31. D. D. E. Long and J.F. Paris, Voting without version numbers, in *Proceeding of the International Conference on*

Performance, Computing, and Communications, (Arizona, USA, 1997), pp 139-145.

32. S.M. Chung, Enhanced Tree Quorum Algorithm For Replicated Distributed Database, in Proceeding of 3rd International Conference on Database System for Advance Application (DASFFA), (Tokyo, 1993), pp 83-89 .
33. N. Ahmad, M. Mat Deris, N.A. Ahmed, R. Norhayati,, M.Y.M. Saman, and M. Zeyad, Preserving Data Consistency through Neighbor Replication on Grid Daemon, *American Journal of Applied Science.* 4(10) (2007) 748-755.