ISBN 2320-2602

International Journal of Advances in Computer Science and Technology (IJACST), Vol.5 , No.3, Pages : 07-11 (2016)
*Special Issue of ICDCB 2016 – Held during April 14-15, 2016 in Kuala Lumpur, Malaysia*
*http://warse.org/IJACST/static/pdf/Issue/icdcb2016sp02.pdf*

# A Cluster-Based Data Replication Technique for Preserving Data Consistency in Data Grid

**Zulaile Mabni[1], Rohaya Latip[1], Hamidah Ibrahim[2], Azizol Abdullah[1]**

[1]Department of Communication Technology and Network,
Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Selangor, Malaysia.
[2]Department of Computer Science,
Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Selangor, Malaysia.
Email: zulaile@tmsk.uitm.edu.my,rohayalt@upm.edu.my

**Abstract*:*** Data grid has been developed to provide a scalable infrastructure for managing and storing data files and support data intensive applications. However, managing the huge and widely distributed data has raised some issues such as data consistency, data availability and communication costs. To address the issues, one of the commonly used techniques is data replication which can provide high availability and increase the performance of the system. Many replica control protocols have been proposed in distributed database and grid which achieved both high performance and availability. However, most of the previously proposed protocols perform well in small size systems and have a small number of replicas. As the network size grows, a larger number of replicas are required to be accessed in order to maintain data consistency, which is not suitable for a large scale system such as data grid. Thus, in this paper, we propose a new replica control protocol named Cluster-Based Replication (CBR) protocol for managing the data in data grid. We analyze the communication cost of the operations and compare CBR protocol with previously proposed tree-based replica control protocols namely Logarithmic protocol and Dynamic Hybrid protocol. A simulation model was developed using Java to evaluate CBR protocol. Our results show that for the read and write operations, CBR provides lower communication cost as well as maintains data consistency.

**Key words:** Data Grid**,** data replication, data availability, communication cost.

## INTRODUCTION

In recent years, with the emergence of large data collections such as high energy physics and computational genomics, efficient data management technique is needed to access and analyze this widely distributed huge data. Thus, data grid has been developed to provide a scalable infrastructure for managing and storing data files and support data intensive applications [1]. However, managing the huge and widely distributed data has raised some issues such as data consistency, data availability and communication costs. To address the issues, one of the commonly used techniques is data replication where many copies or replicas of an object may be stored at many sites in the network. Data replication has been shown to provide high availability and increase the performance of the system [2],[3].

In the literature, many replica control protocols have been proposed in distributed database and grid which achieved both high performance and availability. However, most of the previously proposed protocols perform well in small size systems and have a small number of replicas. As the network size grows, a larger number of replicas are required for the read and write operations to maintain data consistency, which is not suitable for a large scale system such as data grid [4]. Replicating data can become expensive if the number of operations for read or write operations is high [5],[6]. This is due to the communication cost depends on the number of replicas which have to be accessed.

Therefore, in this paper, we propose a new replica control protocol called Cluster-Based Replication (CBR) protocol to address the issue of data consistency and communication cost in large scale system such data grid. The proposed protocol employs a hybrid replication strategy where it combines the advantages of two common replica control protocols to improve the performance of earlier protocols. The proposed protocol groups nodes into clusters and organizes these clusters into a tree structure which enables the protocol to minimize the number of replicas for read or write operations. Thus, CBR provides low communication cost as well as maintains data consistency.

## RELATED WORKS

This section describes the replica control protocols that have been proposed in distributed database systems.

### Primary Copy Protocol

Primary Copy (PC) algorithm is a simple algorithm that designates one copy of a data object as primary copy [7]-[9]. The consistency of the object is maintained by the primary copy. Any other node which is known as a slave copy maintains a non-primary copy. A read and write operations are executed only at the primary copy. For the write operation, once the primary copy is updated, it will be propagated out to all slave copies. The communication cost for read and write operations are low because only one replica is accessed by the operations.

PC protocol is easy to implement and it is one of the most widely implemented replication techniques. However, it has a limitation where, if the node that maintains a primary copy fails, then an update operation cannot be executed until the node becomes available again.

### Tree Quorum Protocol

Tree quorum (TQ) protocol logically organized the replicas in a tree structure [10]. Fig. 1 illustrates the diagram of a tree quorum structure with thirteen replicas in a tree of *height* = 2 and degree of node *D* = 3. In this protocol, a read quorum consists of the root replica. If the root is inaccessible, then majority replicas of its children are added as members of this quorum. Furthermore, for every inaccessible replica, majority replicas of its children are added as members, and so forth. The examples of valid read quorums of Fig. 1 are {1} when root replica is accessible, and {2,3} when root replica is inaccessible.

On the other hand, a write quorum consists of the root, and any majority replicas of the root's children, and any majority replicas of their children, and so forth until the leaves are reached. In Fig. 1, the examples of valid write quorums are {1,2,3,5,6,8,9}, and {1,3,4,9,10,11,12}.

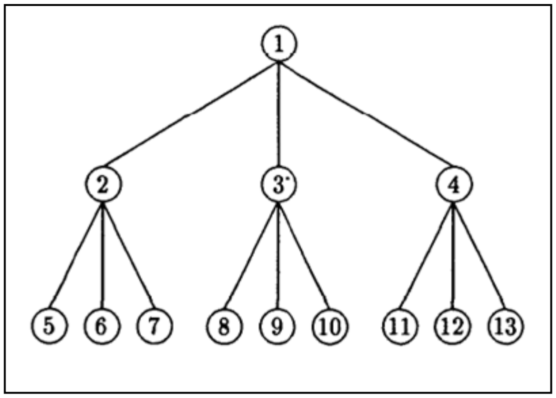

**Fig. 1:** A tree organization of 13 copies of data objects [10]

The strength of this protocol is that read operation may access only one replica. Thus, this protocol allows very low read cost. However, it has some drawbacks such as, as the level of the tree increases, the number of replicas grows rapidly, thus increases the communication cost.

### Logarithmic Protocol

The Tree Quorum protocol was generalized in [11] and named as Logarithmic Protocol. In this protocol, each node $R_i$ of the tree is assigned a value $wq_{Ri}$ ($rq_{Ri}$) which specifies the number of descendants of $R_i$ to be added as members of the write (read) quorum. In the tree structure of *height = 0*, the read operation reads only the root replica. Meanwhile, in the tree structure of *height h*, the read operation reads the root replica, or the $rq_{Root}$ of its subtrees if the root is not available. The $rq_{Root}$ descendants of the root serve as the new root replica of the subtree. The process is repeated until level *height* $h - 1$ is reached. Thus, the read cost is only 1, if the root replica is available. The examples of valid read quorums of Fig. 1 are {1}, and {2,3,4} when the root replica is not available.

In contrast, the write operation reads the root replica, $wq_{Root}$ replica of the root's descendants, $wq_{Ri}$ replica of these previously selected replicas' descendants and so forth until the leaves are reached. In Fig. 1, the examples of valid write quorums are {1,2,5}, and {1,3,9}. The Logarithmic Protocol has lower write cost as compared to the Tree Quorum protocol.

### Dynamic Hybrid Protocol

Dynamic Hybrid protocol combines the grid and tree structure, where the overall topology can be adjusted using the tree height, number of descendants and grid depth [6]. Fig. 2 illustrates the network of Dynamic Hybrid protocol with 31 replicas in (3,3,2) topology, where the three arguments represent the height *h*, number of descendants *s* and grid depth *g* respectively. In the tree structure of *height h*, the read operation reads the root replica or the *s* descendants of the root replica if the root is not available. The descendants of the root serve as the new root replica of the subtree. The process is repeated until level $h - 1$ is reached. Furthermore, in the grid network of depth *g*, read operation reads *s* replicas or go to the next level if one of the replicas is not available. Thus, the read cost is only 1, if the root replica is available. The examples of valid read quorums of Fig. 3 are {R0}, and {R1,R2,R3}, and {R2,R3,R4,R5,R6}.

Meanwhile, the write operation reads the root replica, one replica of the root's descendants, one replica of these previously selected replicas' descendants and so forth until the leaves are reached. Furthermore, in the grid network of depth *g*, write operation reads only one replica in each level down to the last level. In Fig. 2, the examples of valid write quorums are {R0,R1,R4,R13,R22}, and {R0,R2,R7,R16,R27}.

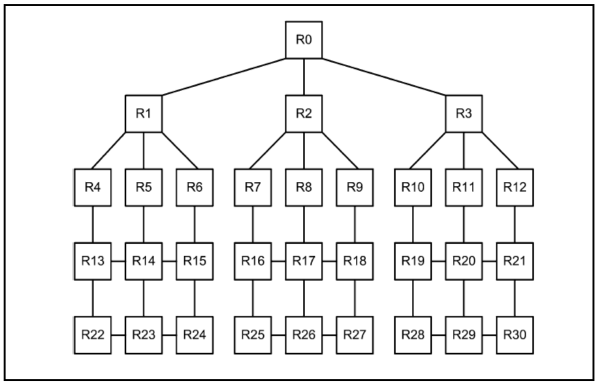

**Fig. 2**: A network for Dynamic Hybrid protocol of 31 replicas in (3, 3, 2) topology [6]

The strength of this protocol is that it combine the advantages of tree and grid protocol to allow low operation cost and high availability. However, this protocol has drawback where, as the network size grows, large number of replicas still need to be accessed to maintain data consistency and therefore, degrade the performance of the system.

## THE PROPOSED MODEL

In this section, we present the system model and algorithm for the proposed protocol called Cluster-Based Replication (CBR) protocol.

### System Model

The system consists of *N* sites that communicate with each other by sending messages over a communication network. We assumed that sites fail independently and communication links do not fail to deliver messages. In CBR protocol, the *N* sites in the network are logically grouped into several nonintersecting groups. We have divided the *N* sites into $\sqrt{N}$ disjoint groups with each group having approximately $\sqrt{N}$ sites [12],[13]. Each group is called a cluster. These clusters are logically organized as a tree of height *h* and descendants *s*. We defined the nodes in the tree to be a sequence of clusters $C_0$, $C_1$,… $C_i$, $C_{i+1}$, … $C_n$. We assume that the nodes in each cluster are logically organized into two dimensional grid structures. For example, if CBR protocol consists of 81 nodes, it will be divided into 9 clusters with 9 nodes in each cluster. The nodes in each cluster will be logically organized in the form of 3 x 3 grid. In Fig. 3, an example of a ternary tree of height = 2 with 81 nodes is presented. Each cluster designates the middle node of the cluster as the cluster head which is colored in black in Fig. 3 and has the replica or primary copy of the data object. The center of the cluster is selected because it is the shortest path to get a copy of the data from most of the directions in the cluster.

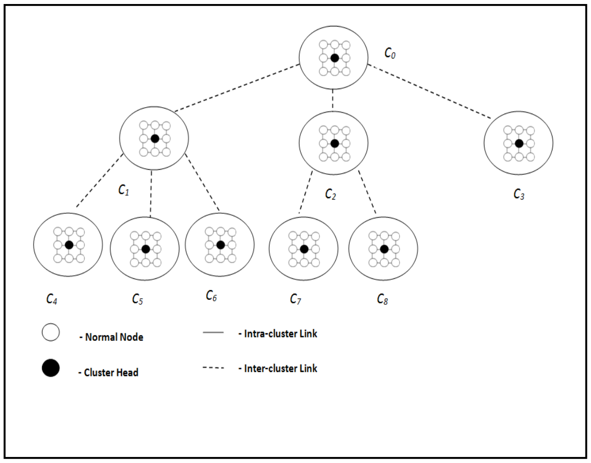

**Fig. 3:** System Model of CBH in a ternary tree of height = 2 with 81 nodes

## Proposed Algorithm

Our proposed CBR protocol combines the advantages of two common replica control protocols namely Logarithmic protocol (LP) protocol and Primary Copy (PC) protocol. In Fig. 3, we show a system with 81 nodes for which we use LP protocol on top of PC protocol as the replication strategy. In order to simplify the description, we assumed that every replica is assigned exactly one vote. Fig. 3 shows a system with 81 nodes where the nine clusters named $C_0$, $C_1$, …, $C_8$ are "logical replicas" which are managed by using LP protocol. The logical replica $C_0$ serves as the root cluster, whereas the logical replicas $C_1$,…, $C_8$ are its descendants. Each logical replica contains a cluster of physical nodes with one middle node called physical replica which has the replica or primary copy of the data object. The physical replica in the root cluster $C_0$ is called root replica. Thus, for a system with $N$ nodes, there will be $\sqrt{N}$ clusters, and $\sqrt{N}$ replicas. To illustrate the algorithm, the replication strategy involves two strategies: "Local Replication", where PC protocol is used for the replication strategy for managing the physical replica within a cluster and "Global Replication", where LP protocol is used as the replication strategy for managing the logical replicas between clusters.

## Read Operation

A read operation according to LP protocol, which is used as the global replication, can be performed by reading the root replica $C_0$ if $C_0$ is accessible or if the root replica is inaccessible then the descendants of the root replica are added as members of this quorum. Furthermore, for every inaccessible replica, all replicas of its children are added as members, and so forth. In CBR protocol, a logical replica can be read if a read operation can be performed on the physical replica which it contains, using the applied local replication strategy which is PC protocol. This means that the precondition for reading a logical replica is a read quorum of $RQ = 1$ if it's contained physical replica is accessible for read operation. Thus, in Fig. 3, assuming that the root replica is accessible, by employing LP Protocol, the read cost is only 1. However, if the root replica is not available, then all replicas of its children have to be accessed which results in the read cost of 3. The examples of valid read quorums of Fig. 3 are $\{C_0\}$ if the root replica is available and $\{C_1, C_2, C_3\}$ if the root replica is not available.

## Write Operation

A write operation according to LP protocol, which is used as the global replication, can be performed by reading the root replica $C_0$ and any one replica of the root's children, and any one replica of their children, and so forth until the leaves are reached. In CBR protocol, a logical replica can be written if a write operation can be performed on the physical replica which it contains, using the applied local replication strategy which is PC protocol. This means that the precondition for writing a logical replica is a write quorum of $WQ = 1$ if it's contained physical replica is accessible for write operation. Thus, in Fig. 3, assuming that the root replica is accessible, by employing LP protocol, we obtain a write cost of 3. An example of valid write quorum of Fig. 3 is $\{C_0, C_1, C_4\}$.

## Correctness of CBR Algorithm

Here, we demonstrate that the read and write quorums constructed by the CBR protocol will always have a non-empty intersection. In [11], the Logarithmic protocol was proven to satisfy the intersection property. Since the Logarithmic protocol was used in the global replication, the proof is as follows:

*Theorem:* The CBR protocol guarantees the intersection of read and write quorums.

*Proof:* The proof is by induction on the height of the trees.

*Basis:* The theorem holds for a tree of height zero, since there is only one physical replica in the tree.

*Induction Hypotheses*: Assume that the theorem holds for a tree of height *h*.

*Induction Step:* Consider a tree of height $h + 1$. The read quorum (*RQ*) and write quorum (*WQ*) constructed for this tree will be of the following form:

*RQ = {Root Replica} or {All physical replicas of sub trees of height h}*

*WQ = {Root Replica} and {Any one of physical replicas of sub trees of height h}*

## Consistency Maintenance

Consistency is maintained by ensuring that the selection of a read and write quorum must satisfy the quorum intersection property to ensure one-copy equivalence among the replicas and maintain their consistent state [14]. "The property stated that for any two operations *o[x]* and *o'[x]* on an object *x*, where at least one of them is a write, the quorums must have a nonempty intersection" [15].

For the proposed CBR, the read/write conflict can be detected because a read operation locks the whole descendants of the root replica while a write operation locks at least one logical replica of the descendants. As for the conflict between two write operations, it can be guaranteed to be detected since any two write operations have to share the root replica. For example, in Fig. 3, valid *RQ* are {$C_0$ }, and {$C_1$, $C_2$,$C_3$}, whereas, valid *WQ* are {$C_0$, $C_1$,$C_4$}, and {$C_0$, $C_2$,$C_7$}. Note that $C_1$ is included in both the valid *RQ* and *WQ* for detecting read/write conflict. On the other hand, $C_0$ is included in both valid *WQ* for detecting write/write conflict. Thus, the CBR protocol guarantees non-empty intersection between read and write quorums.

## PERFORMANCE ANALYSIS AND COMPARISON

In this section, we evaluate the communication costs of read and write operations for the proposed protocol (CBR) and compare them with Logarithmic Protocol (LP), and Dynamic Hybrid protocol (DH).

The communication cost of an operation is computed based on the number of replicas involved in the read or write operation.

For the Logarithmic Protocol, the minimum read cost as given in [11] is:

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

and the write cost for height *h* is:

$$C_{write} = h + 1 \qquad (2)$$

For the Dynamic Hybrid protocol, the minimum read cost as given in [6] is:

$$min(C_{read}) = 1 \qquad (3)$$

and the write cost that depends on the value of height *h* and grid depth *g* is:

$$C_{write} = h + 1 + \mathrm{g} \qquad (4)$$

In CBR protocol, the communication cost is based on the combination of cost for PC protocol [7]-[9] and LP protocol [11], thus, the minimum read cost is:

$$min(C_{read}) = 1 \qquad (5)$$

and the write cost for height *h* is:

$$C_{write} = h + 1 \qquad (6)$$

Table 1 shows the minimum and maximum read cost of the three protocols (LP, DH, and CBR) with 121 nodes in the system. Here, the LP is based on parameters *h = 4* and *s = 3*, whereas, the DH protocol is configured in the (4,3,3) topology. As for CBR, the parameters are *h = 2* and *s = 3*.

On the other hand, Table 2 illustrates the write communication cost of the three protocols (LP, DH, and CBR) for an example system with different total number of nodes, *n* = 49, 81, 121, 225 and 289. Here, the three protocols have the same number of descendants which is 3 but they differ in their heights based on the number of nodes. As an example, for 121 nodes in the system, the LP is based on height of 4, whereas, the DH protocol is based on height of 4 and grid depth of 3 which is configured in (4,3,3) topology and CBR is based on height of 2.

**Table 1:** Comparison for the minimum and maximum read costs of the protocols with 121 nodes

| Protocols | Read Cost | |
|---|---|---|
| | Minimum | Maximum |
| LP | 1 | 81 |
| DH | 1 | 27 |
| CBR | 1 | 7 |

Table 1 above shows that CBR, LP and DH have the same minimum read cost of 1, which is achieved by accessing only the root replica of the tree. However, among the three protocols, CBR has the lowest maximum read cost, whereas LP has the worst maximum read cost. In a system with 121 nodes, for maximum read cost, CBR, DH, and LP need to access 7, 27, and 81 replicas respectively in maintaining the consistency of data.

**Table 2:** Comparison for the write costs of the protocols

| Protocol | Number of Nodes | | | | |
|---|---|---|---|---|---|
| | N = 49 | N = 81 | N = 121 | N = 225 | N = 289 |
| LP | 5 | 5 | 5 | 6 | 6 |
| DH | 5 | 6 | 7 | 11 | 14 |
| CBR | 3 | 3 | 3 | 4 | 4 |

For the write operation as illustrated in Table 2 and Fig. 4, it is apparent that CBR has the lowest write communication cost when compared with LP and DH. It can be seen that CBR needs to access only 3 replicas on 121 copies for maintaining consistency. On the other hand, for LP and DH with 121 copies, the number of replicas that need to be accessed is 5 and 7 respectively, to satisfy the quorum intersection property as well as to ensure consistency.

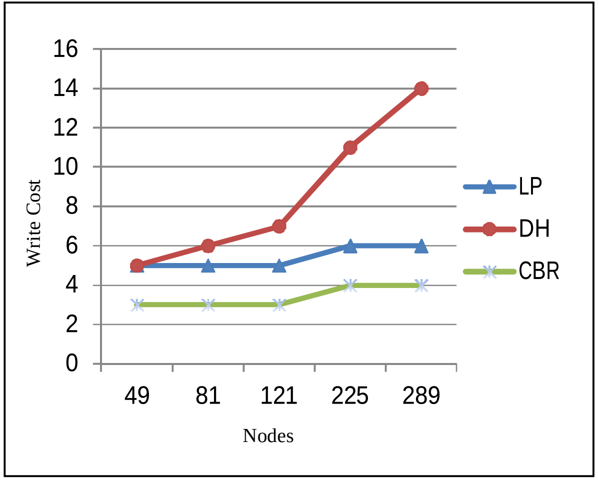

**Fig. 4**: Comparison for the write costs of the protocols with 49, 81, 121, 225 and 289 nodes.

Results in Fig. 4 shows that CBR has reduced the average write cost by up to 37% as compared to LP and 60% compared to DH.

## CONCLUSION

In this paper, we have proposed a new replica control protocol called Cluster-Based Replication (CBR) protocol for the management of replicated data in large scale distributed system in data grid. CBR employs a hybrid strategy that combined the advantages of two common replica control protocols which are Logarithmic Protocol and Primary Copy protocol. Its design goal was to minimize the communication cost while still maintaining the consistency of data objects. In CBR, by grouping the nodes into clusters and having only one replica in each cluster, this has resulted in a small number of replicas involved in maintaining the consistency of data for read and write operations.

We have presented the communication cost analysis of the read and write operations for the proposed protocol and compared it with Logarithmic Protocol (LP), and Dynamic Hybrid protocol (DH). The results show that CBR, LP, and DH have the same minimum read cost of 1, whenever the root replica is available. This is due to a read operation only needs to access the root replica. As for the write operations, our proposed protocol allows much smaller write communication cost than LP and DH protocol. CBR has minimized the communication cost by reducing the number of replicas that need to be accessed for maintaining consistency in the large scale distributed system such as data grid.

## ACKNOWLEDGEMENT

We wish to acknowledge the support of Malaysian Ministry of Higher Education (MOHE) under the Fundamental Research Grant Scheme (FRGS) No: 08-01-13-1220FR.

## REFERENCES

[1] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke, "The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets," *Journal of Network and Computer Applications*, vol.23, no.3, pp. 187-200, 2000.

[2] H. Lamehamedi, B. Syzmanski, Z. Shentu, and E. Deelman, “Data replication in grid environment,” in *Proc. 5th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP'02)*, pp. 378-383, 2002.

[3] Z.Mabni, and R. Latip, “A comparative study on quorum-based replica control protocols for grid environment,” *Springer Communications in Computer and Information Science*, 253, pp. 364-377, 2011.

[4] J.H.Abawajy and M. Mat Deris, “Data replication approach with consistency guarantee for data grid,” *IEEE Transaction on Computers*, vol.63, no.12, pp. 2975-2987, December 2014.

[5] R. Latip, H. Ibrahim, M. Othman, A. Abdullah, and M. N. Sulaiman, "Quorum-based data replication in grid environment," *International Journal of Computational Intelligence Systems (IJCIS)*, vol. 2, no. 4, pp. 386-397, 2009.

[6] S. C. Choi, and H. Y. Youn, "Dynamic hybrid replication effectively combining tree and grid topology," *The Journal of Supercomputing*, vol. 59, no. 3, pp. 1289-1311, 2012.

[7] M. Stonebraker, “Concurrency control and consistency of multiple copies of data in distributed ingres,” *IEEE Transaction on Software Engineering*, vol. 5, no. 3, pp. 188-194, 1979.

[8] M. Ahamad, M. H. Ammar, and S.Y. Cheung, "Replicated data management in distributed systems," *Readings in Distributed Computing Systems*, pp. 572-591, 1992.

[9] W. Zhou, and R. Holmes, “The design and simulation of a hybrid replication control protocol,” *Fourth International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN '99)*, pp. 210-215, 1999.

[10] D. Agrawal, and A. El Abbadi, "The tree quorum protocol: An efficient approach for managing replicated data," in *Proc. 16th International Conference on Very Large Databases*, pp. 243-254, 1990.

[11] H. Koch, "An efficient replication protocol exploiting logical tree structures," *The 23rd Annual International Symposium on Fault-Tolerant Computing*, pp. 382-391, 1993.

[12] S. Madhuram, and A. Kumar, “A hybrid approach for mutual exclusion in distributed computing systems,” *Sixth IEEE Symposium on Parallel and Distributed Processing*, pp.18-25, 1994.

[13] R. Latip, Z. Mabni, H. Ibrahim, A. Abdullah, and M. Hussin, “A clustering-based hybrid replica control protocol for high availability in grid environment,” *Journal of Computer Scie*nce, vol.10, no.12, pp. 2442-2449, 2014.

[14] P. A. Bernstein, and N. Goodman, "An algorithm for concurrency control and recovery in replicated distributed database," *ACM Transaction Database Systems*, vol. 9, no. 4, pp. 596-615, 1984.

[15] D. K. Gifford, "Weighted voting for replicated data," in *Proc. 7th Symposium on Operating System Principles*, pp. 150-162, 1979.