Novelty circular neighboring technique using reactive fault tolerance method

Ahmad Shukri Mohd Noor, Nur Farhah Mat Zian, Noor Hafhizah Abd Rahim,
Rabiei Mamat, Wan Nur Amira Wan Azman
Department of Computer Science, School of Informatics and Applied Mathematics,
Universiti Malaysia Terengganu, Malaysia

Article Info

Article history:

Received Dec 20, 2018
Revised Apr 18, 2019
Accepted Jun 11, 2019

Keywords:

Distributed computing
Fault tolerance
Replication
System availability

ABSTRACT

The availability of the data in a distributed system can be increase by implementing fault tolerance mechanism in the system. Reactive method in fault tolerance mechanism deals with restarting the failed services, placing redundant copies of data in multiple nodes across network, in other words data replication and migrating the data for recovery. Even if the idea of data replication is solid, the challenge is to choose the right replication technique that able to provide better data availability as well as consistency that involves read and write operations on the redundant copies. Circular Neighboring Replication (CNR) technique exploits neighboring policy in replicating the data items in the system performs well with regards to lower copies needed to maintain the system availability at the highest. In a performance analysis with existing techniques, results show that CNR improves system availability by average 37% by offering only two replicas needed to maintain data availability and consistency. The study demonstrates the possibility of the proposed technique and the potential of deploying in larger and complex environment.

Corresponding Author:

Ahmad Shukri Mohd Noor,
Department of Computer Science, School of Informatics and Applied Mathematics,
Universiti Malaysia Terengganu,
21030 Kuala Terengganu, Terengganu, Malaysia.
Email: ashukri@umt.edu.my

1. INTRODUCTION

Since distributed computing comprises of physical and virtual images in a large network [1], neither the volume, size, location, time processing and storage resources are detailed and restricted [2, 3]. The characteristics of distributed computing removes restrictions in using the system and traditional networks in providing its services to the user [1]. However, due to this, new problems, restrictions and challenges might arise for users and applications [3]. Therefore, on top of these adversity, it is important to safeguarding the ability of the computing system in handling unexpected faults and failures by providing a reliable fault tolerance mechanism [4]. With the presence of fault tolerance mechanism, system able to detect, identify and recover from failures without causing major damage to the final output of the distributed computing [5]. Hence, the distributed system is capable of having an optimum and acceptable performance in the presence of fault [6].

One of the way to implement fault tolerance in distributed computing is by applying reactive method [3] where when a fault occurs, application will restart at the last point onwards, replicates the data before migrating to other similar machine [7]. Data replication is a key practices used for disguising errors by preserving separate copies of data or object and synchronization of updating the data in its replica [6, 8]. It will deal with when and where to copy the data, resource optimization and growing or shrinking the replication tree [6]. The technique allows a prompt and locally accessed to shared data allowing the system to continue the availability of the application by ensuring alternate data access option is exist. Consequently, offering a data replication mechanism and control protocol that can promise data integrity, easy access, reliability and availability is highly desired to achieve a decent distributed environment especially if it is large in size or heavy in transaction number [4].

The rest of the paper is organized as follows: Section 2 discusses several replication techniques and presents the research background of the new circular neighboring technique. Section 3 describes the circular neighboring model with the possible maintenance protocol using reactive method. Section 4 presents a rough evaluation on the new technique. Conclusion and future works are given in Sections 5.

2. REPLICATION TECHNIQUES

Data replication comprises of two main types of solution; synchronous and asynchronous replication [9]. Asynchronous replication immediately captures the changes on the primary replica and then timely propagate the changes [9]. This resulted in a lower cost, flexible and scalable solution although dealing with the possibility of data lost and network bandwidth [9]. This type of solution work reasonably well for a single object updates but unreliable when involving multiples objects. For synchronous replication, it deals with quorum to update the replica at the same time and will roll back if one fails [10]. Synchronous replication breaks replica into several schemes providing the high availability, auto fail-over and minimal data loss [9]. However, network efficiency, scalability, cost and less flexibility issue could hinder the ability of this technique [1]. In this paper, we will focus on synchronous replication as the solution offers suitability in avoiding and resolving conflicts in replica access and updates which provides a mechanism that able to increase the data availability.

2.1. Read-once-write-all (ROWA)

This is the most common and straightforward policy [6] used in replicating the system which keeps multiple copies of replicas that allow anyone can be read and must all be updated. In this technique, a read operation needs only one copy and is allowed to read any copy of the data. At the meantime, a write operation is needed in order to write all copies of data. This operation require access in all n copies of data, if there are n copies (a copy in each replica) in the system [6, 11, 12].

ROWA will translates a logical read operation on a data item into one physical read on any of its replica and translates all its logical writes operation to physical writes operation one at each replica [1, 12]. The access to each replicas will be synchronize by the main concurrency controller thus makes this protocol is equivalent with serial execution where each replica that update the data item will update all of its copies or none at all. ROWA is good for the environments where the data is mostly read-only because it provides read operation with a high degree of availability at low communication overhead [11, 12]. However, the write operation has very high overhead as all replicas must be updated simultaneously and cannot be executed at the failure of any copy. This resulted in an imbalance of data availability and communication cost of read and write operations. The read operations have high availability and low communication cost while write operations in the other hand have low availability with higher communication cost [11, 12]. Despite offering a strong consistency and a high degree of fault tolerant, this techniques increases response time and low access concurrency [13, 14]. The other significant drawbacks of ROWA are this technique is rigid in selecting its read availability and inadequate in the ability of tolerating communication failures [6].

2.2. Quorum consensus (QC) or voting

The Quorum Consensus (QC) method generally allow writes operation to be recorded only at a subset (a write quorum) of the up sites, on condition that reads operation is made to query a subset (a read quorum) specifically proved will overlap with write quorum [11]. The read operation will be able to return its most recently written value whenever the quorum intersection condition is met and will be said as have voted for it, giving the QC method alternative name, voting. There are two states of voting, that is it is either static or dynamic [11, 12]. In static QC state, they are identified by votes that are assigned during the system start up time while for dynamic state is when the site are able to reconfigure its quorum specification in response to any systems events such as failure, load changes or else [11, 12].

Like ROWA, QC method also undergo few enhancement to overcome its weakness. Therefore each QC method utilizes different quorum membership designs, varying from the simplest number of quorum majority to an unambiguous membership listing of each possible quorum. QC was first applied in the replicated databases by using Uniform Majority Quorum Consensus Method [11, 12]. This method gives advantages on the flexibility to both site and able to tolerate network failures however, it required high read and update costs [11, 12].

2.3. Two-replica distribution technique (TRDT)

Shen. Chen. Zheng and Shi [12] have proposed Two-Replica Distribution Technique (TRDT) that introduced each node must has equal capacity of storage and all data have two replicas on different nodes as well as all nodes have two data replicas [11, 12]. For N number of nodes, it is divided to n set of nodes (N = 2n) where each of the set comprises of two nodes. The main drawback for this technique is TRDT increases resources used when adding second replica to the servers. TRDT also limits the operation if one of the replica set is absent, leaving the node exposed to double failure in case of both replicas are damaged or lost [11, 12].

2.4. Neighbor replica failure recovery (NRFR)

A.S Mohd Noor proposed Neighbor Replica Failure Recovery (NRFR) [15] that based on NRDT logical structure for data replication. NRFR logically organized in the form of two dimensional n x n grid structure when there are nodes in the system, N = n2 [16]. Each of the nodes owned its own master data file where updates will be applied on the primary data file before navigates to its neighbor copies within a separate transactions [15, 16].

2.5. Neighbor replication on grid (NRG)

Neighbor Replication of Grid has been introduced by N. Ahmad [11] by applying neighboring techniques in a two-dimensional grid structure. All the sites in this mechanism is logically organized in the form of n × n structure, by having a master data item stored in each sites.

Let r denote as read quorum, w denoted as write quorum and LB is the total number of votes, the consistency of data object will be controlled by ensuring there is a nonempty intersection with:

a. $1 \leq r \leq L_{\mathrm{B}}, 1 \leq w \leq L_{\mathrm{B}}$
b. $r + w = L_{\mathrm{B}} + 1$

The complexity of maintaining copies at different sites gave a challenged for NRG to handle its locking information especially in the current structure of distributed computing.

3. CIRCULAR NEIGHBORING MODEL AND METHOD

This section will discussed the newly proposed synchronous Circular Neighboring replication technique, CNR that use reactive method for fault tolerance policy [17]. The neighboring technique adopted a neighboring relationship that is identical to a circle where a number of replicas are related to each other back to back. Hence, each replica has two neighbors, the one preceding it and the one following. This relationship is true for replicas of three or more, whilst two replicas will solely be neighbors to one another [18].

A set of replicas is defined where N is the total number of replicas. As illustrated in a circle, for the base case of three replicas, each base case of three replicas, each member of the replica will have two neighbors i.e. has and as neighbors. For example, if one more replica will be added to the set, the replica is inserted to follow the previous replica. Thus, it breaks up the neighbor relationship between and consequently, replaced it with newly added replica. By iteration, adding more replicas to the set will undergo the same process and consequence. Therefore, it is proven that for any number of replicas of three or more, each of them has two neighbors preceding and following them, in a circular neighboring relationship. The CNR logical structure design example can be shown as shown in Figure 1. Assuming that the primary data file f_1 is located on the server site n_1, the primary data file f_2 located on the server site n_2 and so on.

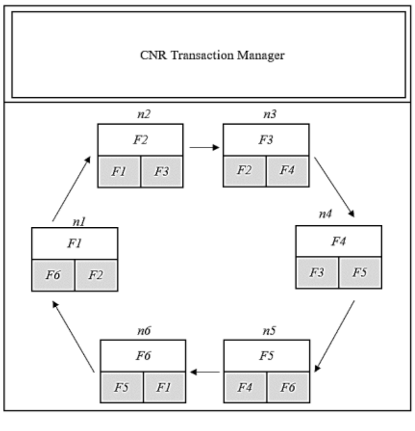

Figure 1. The arrangement of primary replicas in CNR with 6 nodes

Each site holds three copies of unique data with one of them is its own primary data and each individual data will has three copies anywhere on the system. Weigh against full data replication, this newly proposed technique offers minimization on the storage capacity without abandoning system availability.

Quorum is a number of replicas that listed in an agreement for any operation to take place on them. The mechanism must be able to manage two operations of read and write in the case of both operations run concurrently in order to avoid any conflicts except for read-read operations. Therefore, data synchronization is a crucial part for a replication technique. Data can be synchronize efficiently by adopting these two rules for quorum voting:

a. Read perimeter, R and write perimeter, W must be greater than certain non-negative number of votes, $R + W > v$
b. Write perimeter must be greater than non-negative number of votes divide by 2, $W > v/2$

The second rules only applicable if the version number is required to verify the most current copy of the replica. This two rules are important to avoid concurrent data writes and read as well as write operations occurred on the same data at the same time. As the result, read scores always reflect the most recent write because read quorum consist of at least one replica that was involved in the most recent write operations. For instance, quorum is modelled as a set of replicas; $C = \{C_1, C_2, C_3... C_n\}$ where $n = 1, 2, 3, ...,i$ are called the sequence numbers of these replicas. Each replica, C_i manage a set of data. In CNR, as discussed previously have only two (2) neighbors to each replica and only three (3) replicas contain the same copy of a data item. To ensure one copy serializability in CNR policy, read operation, r quorum will be defined as r = 1 and for write operation, w defined as w = 3. From Figure 2, any two adjacent replicas can be a read quorum. The write quorum is all three replicas that shares the mutual data with the two replicas in read quorum. Consider a primary data item of d_2. Based on CNR rules, assuming that n = 3 where each replica get one vote and all three replica contain the same copies of primary data d_2, d_1 and d_3 holds the copies of the primary data. Thus, in order to perform read on d_2 primary data, the valid quorum set are {d_1}, {$d2$}, {$d3$}, $\{d_2, d_1\}$, $\{d_1, d_3\}$, $\{d_2, d_3\}$ and $\{d_2, d_1, d_3\}$. The write quorum is {d_1, d_2, d_3}.

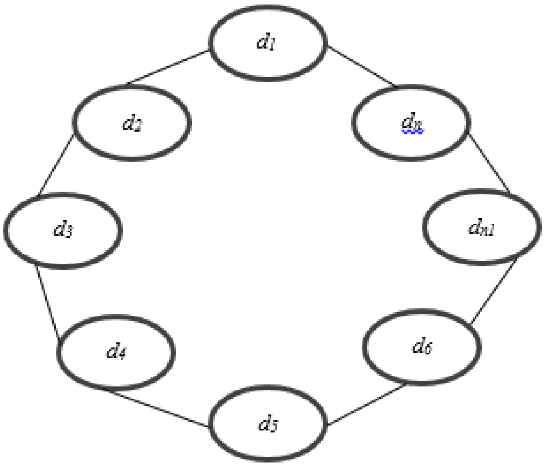

Figure 2. Replica organization of CNR of size *n* (any integer)

In each CNR replica, coordinating algorithm controls the primary replica while cooperating algorithm deals with non-primary replica. When the client sends its request, CNR transaction manager will manage the request from clients and locates the primary replica. Primary replica will execute coordinating algorithm upon receiving procedure call from transaction manager. These processes are divided into few phases. In the first phase, the primary replica asks related replicas to cast vote in order to form a quorum for the operation. If quorum is formed, it will return the value of 1 to transaction manager or else it will return 0 and abort the operation. In some cases, the quorum will return -1 in the case of partial commitment from the related replicas. During the second phase, the transaction manager will ask quorum to commit the operations i.e. lock the data. If there is any partial commitment involved, primary replica captures the details of which replica issued partial commitment and will use them to solve in conflict resolution. During operation execution, cooperating algorithm at the non-primary replicas' side will perform the operation and unlock the data when it is done.

4. CIRCULAR NEIGHBORING ANALYSIS AND EVALUATION

This section will further analyze the availability of the newly proposed replication technique [19]. Availability of a system refers to the probability of a system is completely working over a period of operating time [20]. Table 2 shows for each 4 techniques the number of replicas required for a data item or object in four system scenarios with different number of n nodes assuming that the replicas are under a normal circumstances without any fail-stop failure. Assume that during fail-stop failure, one replica is not available, as per listed on Table 3. Finally, Table 4 records all availability times for each techniques depending on the technique architecture based on Table 1 components' availabilities.

Table 1. The nine components of interdependent servers and its availabilities

Component	Availability
Web	0.95
Application	0.955
Database	0.95
DNS	0.97
Firewall	0.96
Switch	0.97
Data Centre	0.95
Application2	0.95
Manager	0.99
Total Availability	0.6956

Table 2. The comparison of the size of replicas under different set of n nodes for multiple replication technique

Replication	Number of replica for each data/object			
Technique	n = 12	n = 24	n = 35	n = 40
ROWA	12	24	35	40
TRDT	2	2	2	2
NRFR	5	5	5	5
CNR	2	2	2	2

Table 3. The comparison of the size of replicas during fail-stop failure for multiple replication technique

Replication Technique	Number of replica for each data/object			
	n = 12	n = 24	n = 35	n = 40
ROWA	11	23	34	39
TRDT	1	1	1	1
NRFR	4	4	4	4
CNR	2	2	2	2

Table 4. The comparison of improvements using different replication model for 9 nodes

Replication Technique	System Availability	Improvement (%)
No replication	0.6956	0.000
ROWA	0.9103	30.862
TRDT	0.9846	41.547
NRFR	0.9978	43.444
CNR	0.9995	43.691

Results show that the improvements in system availability in 9 nodes varied with replication technique. The different amount of replicated nodes in each technique contributes as the main factor in providing a highly available system. From Table 2 and Table 4, it is clear that replication technique with CNR achieves the highest system availability with the lowest replication required to maintain its availability. CNR requires at most 2 replicas in order to search for the nodes that hold the same data file of the failed node while for ROWA and NRFR requires additional time to recover due to the increment of the number of replicas for a data item due to the increase of number of nodes in the environment. While for TRDT that shares the same number of replicas with CNR scores lower system availability due to the arrangement of nodes in the technique architecture. Furthermore, in the event of fail-stop failure, TRDT only has one replica left for recovery, thus resulted in lower score of recovery latency which affected its system availability. Therefore, for a recovery procedure, it can be deduced that system availability improves further consistently to the number of replicas in the system.

From the results, ROWA has shown that as the complexity of the system increases, the technique become irrelevant to be use to recover the system as the number of data file grows as with the number of nodes in the system. Thus, it lowers the system availability. However, techniques that adopted neighboring policy that is NRFR, TRDT and CNR are not affected with the increase number of nodes in the system. CNR leads NRFR and TRDT in terms of higher degree fault recovery as it requires lower replicas than NRFR and in the event of fail-stop it is not limited to only one replica available per data item as in TRDT.

5. CONCLUSION

A unique circular neighboring technique based on reactive fault tolerance method has been presented. The technique provides lower data replication compared to other state of the art techniques. The model and methodology of the technique have been described along with analysis and evaluation of the technique.

CNR is an approach for data replication, by placing redundant copies of data in a circular form to improve the quality of service in a feasibly clutter distributed environment. CNR technique determines the most appropriate neighbor to perform data replication in a small quorum size to execute the procedures. The performance analysis of CNR compared with other techniques showed that CNR provides a better approach to highly available read and write operation for distributed system. CNR offers lower communication cost to replicate the data items as well as maintaining the higher score of recovery latency thus resulted in higher system availability compared with other techniques.

While comprehensive analysis and comparisons have been done for this work setting, further experiments are required before a completely decisive evaluation of the proposed algorithm can be achieved. However, the proposed technique showed that CNR able to increase a system availability by maintaining simplest replica distribution which able to reduce the time taken to recover a failed system compared with the existing techniques. These reassuring results show that the technique proposed in the paper able to improve fault management and contributes towards continuous service delivery in distributed systems.

ACKNOWLEDGEMENTS

This research is funded by Fundamental Research Grant Scheme (FRGS) with the Ref: FRGS/1/2018/ICT04/UMT/02/2. FRGS is a research grant from the Ministry of Education (MOE) Malaysia.

REFERENCES

[1] Noor A.S.M., Zian N.F.M., Shaiful Bahri F.N.M., "Survey on replication techniques for distributed system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9(2), pp. 1298-1303, 2019.

[2] J.A. Torkestani, "A Highly Reliable And Parallelizable Data Distribution Scheme For Data Grids," *Future Generation Computer Systems*, vol. 29(2), pp. 509-519, 2013

[3] N. Cheraghlou, et al., “A Survey Of Fault Tolerance Architecture In Cloud Computing,” *Elsevier Journal of Network and Computer Applications*, vol. 61, pp. 81-92, 2016

[4] Noor A.S.M., Deris M.M., “Failure recovery mechanism in neighbor replica distribution architecture,” *Lecture Notes in Computer Science*.Vol. 6377 LNCS, Issue M4D, 2010, Pages 41-48. Springer Nature. 2010.

[5] Y. M Teo at el. “Cost-Performance of Fault Tolerance in Cloud Computing”, *International Conference on Advance Computing and Applications (Special Issue of Journal of Science and Technology)*, vol. 49, pp 61-73, 2011.

[6] A.A. Helal, A.A. Heddaya and B.B. Bhargava, *Replication Techniques In Distributed Systems*, 2nd, Kluwer Academic, 1996.

[7] Mohd. Noor, A.S., Mat Deris, M. “Extended heartbeat mechanism for fault detection service methodology” *Communications in Computer and Information Science* Vol. 63, pp. 88-95. Springer Nature. 2009

[8] R.V. Renesse and R. Guerraoui., “Replication Techniques for Availability,” In *B. Charron-Bost, R. Pedone, and A. Schiper (Eds.)*: *Replication, Lecture Notes in Computer Science*, vol. 5959, pp 19-40, 2010

[9] A. Natanzon, E. Bachmat, “Dynamic Synchronous/Asynchronous Replication”, *ACM Trans-action on Storage*, vol. 9(3), 2013.

[10] Mohd Noor A.S., Yunus F., Mamat R., Sirajuddin E.A., Mat Zin N.F., "Enhancing availability of marine big data repository with a new fault tolerance technique,” *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9(3-5 Special Issue), pp. 83-87, 2017.

[11] A. Noraziah, “Managing Replication and Transactions Using Neighbour Replication on Data Grid Database Design,” Ph.D. Thesis. Universiti Malaysia Terengganu, 2007.

[12] Deris, M.M. "Efficient Access of Replicated Data in Distributed Database Systems." Ph.D. Thesis, Universiti Putra Malaysia, 2001.

[13] Dabrowski C., *Reliability in grid computing, Concurrency Computation: Practice and Experience*, Wiley InterScience, 2009.

[14] D.M. Bui, T.H. The, S.Y. Lee, *Fuzzy fault detection in IaaS Cloud Computing*, ACMIMCOM, 2016.

[15] Mohd Noor, A.S., Deris, M. and Saman, M.Y.,"Co-Existance Neighbourhood model availability prediction for Cloud Infrastructure as a service (IaaS) optimization in interdependent environment," *International Journal of Machine Learning and Computing*, vol. 4, pp. 85, 2014

[16] Mamat R., Deris M. and Jalil M., "Neighbor Replica Distribution Technique for Cluster Server Systems," *Malaysian Journal of Computer Science*, vol. 17(2), 2004.

[17] Siva S.S. and Babu K.S., "Survey of fault tolerant techniques for grid," *Computer Science Review, Elsevier Inc.*, vol. 4(2), pp. 101-120, 2010.

[18] Mamat, A, Deris, et al., “Managing Data Using Neighbor Replication on Triangular-Grid Structure,” vol. 3994. pp. 1071-1077, 2006

[19] Renesse, R.V.and Guerraoui, R., "Replication Techniques for Availability". In *B. Charron-Bost, R. Pedone, and A. Schiper (Eds.): Replication, Lecture Notes in Computer Science*, vol. 5959, pp. 19-40, 2010.

[20] Schmidt K. High, *Availability and Disaster Recovery: Concepts, Design, Implementation*, Springer-Verlag, 2006.