International Journal of Electrical and Computer Engineering (IJECE)
Vol. 9, No. 2, April 2019, pp. 1298~1303
ISSN: 2088-8708, DOI: 10.11591/ijece.v9i2.pp1298-1303

Survey on replication techniques for distributed system

Ahmad Shukri Mohd Noor [1], Nur Farhah Mat Zian[2], Fatin Nurhanani M. Shaiful Bahri[3]
[1,2]School of Informatics and Applied Mathematics, Universiti Malaysia, Malaysia
[3]Department of Mathematics, Faculty of Science, Universiti Teknologi Malaysia, Malaysia

Article Info

Article history:

Received Aug 26, 2018
Revised Oct 16, 2018
Accepted Nov 10, 2018

Keywords:

Distributed computing
Distributed systems
Fault failure recovery
High availability
Replication technique

ABSTRACT

Distributed systems mainly provide access to a large amount of data and computational resources through a wide range of interfaces. Besides its dynamic nature, which means that resources may enter and leave the environment at any time, many distributed systems applications will be running in an environment where faults are more likely to occur due to their ever-increasing scales and the complexity. Due to diverse faults and failures conditions, fault tolerance has become a critical element for distributed computing in order for the system to perform its function correctly even in the present of faults. Replication techniques primarily concentrate on the two fault tolerance manners precisely masking the failures as well as reconfigure the system in response. This paper presents a brief survey on different replication techniques such as Read One Write All (ROWA), Quorum Consensus (QC), Tree Quorum (TQ) Protocol, Grid Configuration (GC) Protocol, Two-Replica Distribution Techniques (TRDT), Neighbour Replica Triangular Grid (NRTG) and Neighbour Replication Distributed Techniques (NRDT). These techniques have its own redeeming features and shortcoming which forms the subject matter of this survey.

Corresponding Author:

Ahmad Shukri bin 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

A distributed computing system is a finite set of sites that are connected by communication links and is responsible for providing the main execution platform for High-Performance Computing (HPC). High availability is the most important elements of a reliable distributed system. Since sites and links are prone to failures, a property of distributed computing called fault tolerance enables the sites to carry on functioning even on the individual component has failed without terminating the entire system. With the present of fault in the system, it will disturb the normal execution and may cause the system to operate in an unusual way. Fault tolerance consists of two main components that are fault detection and fault recovery.

A fault is a defect at the lowest level of abstraction as defined in [1] and may cause an error, an internal data state. From the error, it may lead to externally visible deviation from the system correctness behaviour which called as a failure. Despite the fact that a fault need not result in an error or an error in a failure. When a fault is detected, the system either recovers their state by luck or by the designated fault tolerance, so as to ward off any failure.

Replication is a significant technique used for masking errors in the replicated component in order to achieve fault tolerant in distributed systems. It is a process of maintaining different copies of data or object and the synchronization of updating the data in its replica. It is not a backup method where data or object is not automatically overwritten whenever there are any changes to the original data and immediately lose any

historical state. It has to deal with when and where to copy the data, resource optimization and growing or shrinking the replication tree.

Replication consists of two types of solution that is synchronous and asynchronous replication. The synchronous solution will update two replicas at the same time and will roll back if one fails. The benefits of this type of solution are it is high availability, auto fail-over and minimal data loss encountered. However, this solution will have to deal with network efficiency, scalability, cost and it is less flexibility. For an asynchronous solution, changes on the primary replica will be captured immediately in a timely propagated. This solution offers a low cost, flexible and scalable solution but needs to deal with data loss and network bandwidth. Replication breaks into two schemes that are full replication (all-data-to-all-sites) and partial replication (all-data-to-some-sites). In this paper, the synchronous solution is chosen for its higher reliability and suitability to avoid conflicts based on its quorum execution or commitment protocol.

In a conventional way of computing, faults avoidance and faults removal techniques such as structured programming, software reuse, testing and so on are able to deliver dependability for the software by detecting and expunging the faults in the systems. However, in a large and dynamic distributed system, millions of computing devices connected to each other via communication links are prone to failures and these techniques are not enough to tolerate the failures. Distributed systems allow many users access to a common computing resource hence providing resources sharing causing a single fault can be inevitable in this computing environment.

The ability to tolerate failures while efficiently exploiting the computing resources in an accessible manner must be an essential part of distributed computing infrastructure. Therefore, a fault-tolerant approach will be useful in order to potentially prevent malicious sites that are distracting the overall performance of the environment. A distributed system is expected to be maintained in the presence of partial failures at the level of fault isolation or even higher level of fault tolerant. To achieve fault tolerance, we either reconfigure the service to take advantage of the new components or design a mechanism to mask failure on-the-fly by placing redundant resources. Replication techniques primarily concentrate on these two fault tolerance manners.

Replication enables the systems to be reconfigured in order to allocate more replicas to the systems, in ensuring flexibility of the system so as to preserve its dependability. It is capable to produce the aggregated computing power of all the replica sites to withstand on a single load category thus gives it a capacity to improve the computing performance. Theoretically, the computer system will be able to achieve higher reliability by using more reliable components which will be more robust as it is using more reliable parts or more suitable materials. In circumstances where using more reliable components is not an option, replication is the best approach to provide highly-reliable systems by using less reliable components. When an object is replicated, it will have several identical copies of the object called replicas. In the event of failure, the failure is masked by its other replicas thus availability is guaranteed in spite of the failure. The techniques in replication have been successfully implemented for distributed computing systems and allow such system to remain distributed, at the same time increasing their availability as well as performance in a large degree where the system is able to operate in the presence of fault without user intervention to tolerate the failures that may occur in the distributed computing environment.

2. READ ONE WRITE ALL (ROWA)

This is the most common and straightforward protocol use in replicating the system which keeps multiple copies of replicas that allow anyone can be read and must all be updated. This protocol will translate a logical read operation on a data item into one physical read on any of its replicas and translates all its logical writes operation to physical writes operation one at each replica. The access to each replica will be synchronized by the main concurrency controller thus makes this protocol is equivalent with a serial execution where each replica that update the data item will update all of its copies or none at all.

ROWA provides a simple and elegant technique which has the ability to process read operation regardless of any communication failures since one site will remain up and reachable. The characteristic of ROWA that provides read operation makes it suitable for the environment that most of its data is in read-only overhead. The significant drawbacks of ROWA is the protocol is rigid in selecting its read availability and will blocks all the writes operation if one site is down or unreachable until the failures is repaired. This eventually will cost increment in response time and decreases its performance.

3. 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. 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*.

The quorums can be static or dynamic depends on the assigned votes and the capability of the sites to reconfigure the quorum specification. QC is able to mask failures without any intervention until the failures is tolerated. However, this technique will cost the read operation fairly expensive since the implementation of the idea always a difficult challenge [2].

4. TREE QUORUM (TQ) PROTOCOL

Tree Quorum (TQ) was proposed by Agrawal and El-Abbadi [3] that applies replication in a logical tree structure over a distributed site as shown in Figure 1. From this structure, a read quorum is able to be performed by the root or the majority of its children while for write quorum it is formed from the root, a majority of its children and a majority of their children and so forth until it has reached the leaves of the tree.

In a best case, a read quorum will consist of only a root, {1}. If the root fails, a quorum is formed by the majority of the copies at level 1, e.g. {2, 3}, {3, 4} or {2, 4}. In the case of no majority are accessible at level 1 or only node 4 is accessible, nodes 2 and 3 will be replaced by their children in that order. In the event of all copies in level 0 and 1 failed, a quorum will be performed by the majority of the children of the selected majority at level 1. The size of the write quorum is fixed but the members may be different.

The advantage of this protocol is the write operation can access the number of copies always less than a majority of the quorum while for the reading operation it may access only one copy. For a read operation, the cost of executing is comparable with ROWA but for a write operation, it gave the much better result. Unfortunately, the write operation will be failed to be executed if more than a majority of the copies at any level of the tree become unavailable.

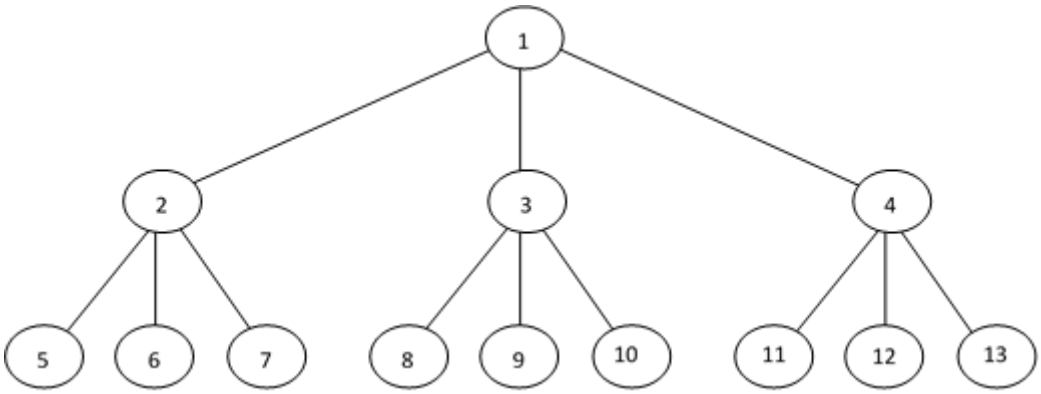

Figure 1. A tree quorum organization of 13 copies of data object

5. GRID CONFIGURATION (GC) PROTOCOL

This protocol is introduced by Maekawa [4] where all quorums are of equal size in order to obtain a distributed mutual exclusion algorithm which later extended by Cheung *et al.* [5] and Kumari and Meenu [6] for replicated data objects. This protocol introduced n copies of data objects are logically organized in the form of a $\sqrt{n} \times \sqrt{n}$ as depicted in Figure 2.

Read quorum consist of a copy from each column in the grid will be acquired in order to perform read ion on the data items. While for write operations to be performed, write quorum consists of all copies in one column and a copy from each of the remaining columns will be needed. This protocol introduced the read and write operation in the size of $O(\sqrt{n})$.

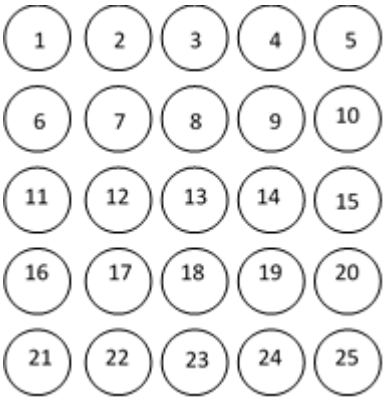

Figure 2. A grid configuration with 25 copies of data object

In the configuration depicted in Figure 2, to execute a write operation, copies {1, 6, 11, 16, 21, 7, 13, 19, 25} is required while to execute read operation copies {1, 2, 3, 4, 5} is sufficient enough. Generally, this protocol has a length of x and a width of y so can be presented as (x, y). A read operation is performed when retrieving a read quorum (x, y) that formed from x copies in each of y different columns. For a write operation, the write quorum is formed from x copies in each of y columns and any $\sqrt{n}–x+1$ copies in each of $\sqrt{n}+y+1$ column. The quorum intersection property between reading and write quorum is determined by the read grid quorum size where if the size (x, y) then write grid quorum must be ($\sqrt{n}–x+1$, $\sqrt{n}–y+1$).

The drawbacks of this technique are this structure degrade the communication cost and the availability of data as the number of copies for both read and write quorum is big as well as prone to failure of the entire row and column in the grid.

6. TWO-REPLICA DISTRIBUTION TECHNIQUE (TRDT)

Two-replica distribution technique (TRDT) has been proposed by Shen *et al.* [7]. This technique introduced that on each node has an equal capacity for storage and all data have two replicas on different nodes and all nodes have two data replicas. For N nodes, it is divided to n set of nodes (N=2n) where each of the set consists of two nodes as illustrated in Figure 3.

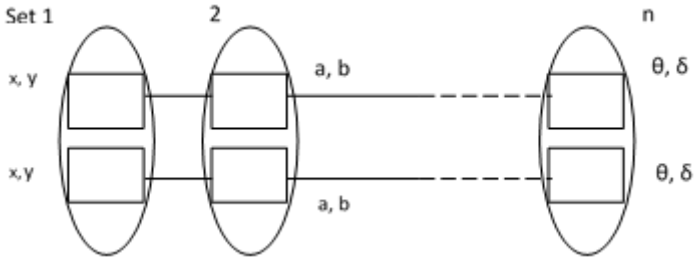

Figure 3. Data replica distribution technique when N=2n

Each rectangle represents the node while for each oval represents a set that consists of two nodes. Data x and y has two replicas, that is in nodes from set 1 and so as data from set b also has two replicas located in nodes from set 2 and so on. The replication is directly done on its replica whenever the primary node receives a request from the replication link and this replication technique uses asynchronous replication.

The weakness of this technique is even though the increment of the availability is not trivial, TRDT double up the resources as it added up to each of the serves with second replica [4]. TRDT also acquire the system to have replica-availability of more than 99% in order to achieve high reliability plus the operation cannot be completed if one of the sets is unavailable [8]. This technique also exposes the node to experience double faults in case of both replicas is damaged or lost.

7. NEIGHBOUR REPLICA TRIAGULAR GRID (NRTG)

Neighbour Replica Triangular Grid (NRTG) was proposed by Mamat *et al.* [9]. The concept of neighbour replication is partial replication (all-data-to-some-sites) where replicated data only copied to the node's neighbours. In this technique, all sites are logically organized in the form of the triangular grid structure. This structure is inherited from the binary-tree structure where the inner leaves of the tree are linked together as shown in Figure 4. Each of the sites in the structure has a master data file. If the site is operational, the copies will be available at the sites otherwise it is unavailable.

Each oval represents a node. Data from node 1 will be replicated to node 2 and node 3 which are neighbours to node 1. For node 5, it has four neighbours, which are nodes 2, 3, 8 and 9. Therefore node 5 has five replicas. This shows that each node in the tree required the maximum number of replications is 5 hence this technique minimizes the storage capacity as compared with full replication scheme. NRTG requires a minimum number of quorum sizes thus providing high availability of executing write operations in the replicated database.

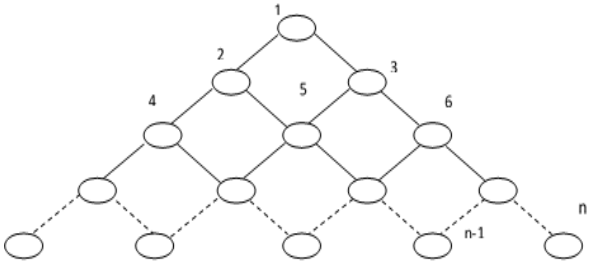

Figure 4. Triangular neighbor organization of n sites

8. NEIGHBOUR REPLICATION DISTRIBUTED TECHNIQUE (NRDT)

Neighbour Replication Distributed Technique (NRDT) was introduced by Mamat *et al.* [8]. In this technique, all nodes are logically organized in the form of two-dimensional nxn grid structure. It will logically organize in the form of n x n when there are nodes in the environment where N=n2.

From Figure 5 we assume that the environment consists of 9 nodes represented by the circles so that it will be logically organized in the form of 3x3 grid structure. Each of the nodes has its own master data file. In this technique, updates on the primary copy will be first committed at the primary node then will later be updated asynchronously to its neighbour copies within a separate transaction.

In NRDT, all data has some replicas on different nodes and all nodes have some data under the environment. The data is replicated to its neighbours from its primary nodes where the number of data replication from each node is less or equal to the number of neighbours +a data from the node itself. NRDT able to increase the availability of the distributed environment much better than TRDT model with the less overhead cost since in TRDT model the system is comprised from two redundant component [2]. Over time, the availability of the NRDT model reduces in a small percentage per year with only predicted to be 1.2% for the period of ten years.

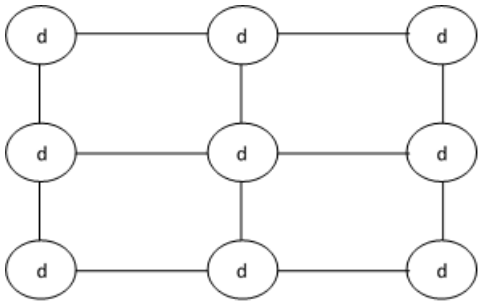

Figure 5. Distributed environment with 9 nodes

9. CONCLUSION

This paper presents a brief survey of replication techniques, its importance in distributed systems and the different techniques that can be used to implement the replication. The different techniques that are available in current works have been appropriately discussed under different headings. The description, concepts and illustration for each of the replication models have been analysed and presented.

REFERENCES

[1] Helal A.A., Bhargava B.K., Heddaya A.A., "Replication Techniques in Distributed Systems," Kluwer Academic Publishers, United States of America, 2002.
[2] Noor A.S.M., Herawan T., Deris, M.M., "Neighbor-Replica Distribution Technique Model for Availability Prediction in Distributed Interdependent Environment," *Int'l Journal of Cloud Applications and Computing*, 2013.
[3] Agrawal D. and Abbadi, A.E., "Using the Tree Quorum Protocol: An Efficient Approach for Managing Replicated Data," *Proc 16, Int'l Conf. on Very Large Database*, pp 243-254, 1998.
[4] Maekawa M., "A Vn Algorithm for Mutual Exclusion in Decentralized Systems," *ACM Transaction on Computer Systems*, Vol. 3 (2), pp 145-159, 1992.
[5] Cheung S.Y., Ammar M.H., Ahmad M., "The Grid Protocol: A High Performances Schema for Maintaining Replicated Data," *IEEE Transactions on Knowledge and Data Engineering*, Vol.4 (6), pp. 582–592, 1992.
[6] Kumari T., Meenu S., "Diagonal Replication with Intersection of Quorums in 2D Mesh (IQ2DM) Protocol for Grid Environment," *International Journal of Innovations & Advancement in Computer Science*, Vol. 4, 2015.
[7] Shen H.H., Chen S.M., Zheng W.M., Shi S.M., "Research on Data Replica Distribution Technique for Server Cluster," *IEEE Proc. 4th Int'l. Conference on Peformance Computing,* Beijing, pp. 966-968, 2000.
[8] Mamat R., Deris M.M., Jalil M., "Neighbour Replica Distribution Technique for Cluster Server Systems," *Malaysia Journal of Computer Science*, Vol. 17 (2), pp. 11-20, 2004.
[9] Mamat A., Deris M.M., Abawajy J.H., Ismail S., "Managing Data Using Neighbor Replication on Triangular-Grid Structure," *6th International Conference, Reading, UK*, pp 1071-1077, 2006.

BIOGRAPHIES OF AUTHORS

Ahmad Shukri Bin Mohd Noor Currently, he is an associate professor of computer science and Deputy Dean at School of Informatics Applied Mathematics, Universiti Malaysia Terengganu (UMT). He has published more than 50 research papers in various referred journals, conferences, seminars and symposiums. He also appointed as Chief Editor for various Scopus indexed journal and conferences. He reviews various papers in refereed journals such as IEEE and Elsevier. He was appointed as General Chair and technical committee for various iconferences. He received several invitations as invited speaker for various conferences. He is a Google Online Professional Certified and Cloud Professional Certified.

Nur Farhah Mat Zian has graduated from UMT, Terengganu in B.Sc. (Hons) Information Technology (Software Engineering) with CGPA of 3.69. She has been awarded with dean's list for 5 consecutive semesters and was a Best Graduate for Department of Computer Science, UMT, 2009 Software Engineering Course.She obtained MSc in Computer science in 2015. For her master, 4 papers have been written, 3 have been published and 1 under review of Journal's reviewer's committee. Currently, she is a Ph. D candidate a School of Informatics and Applied Mathematics, UMT.

Fatin Nurhanani M. Shaiful Bahri Currently, studied at Universiti Teknologi Malaysia, Johor Bahru in Bachelor of Science (Mathematics) with CGPA 3.15. She held her position as Crew Tun Fatimah College for four years. In addition, she holds the position of secretary in two major programs namely the Inspirational Women Symposium 2017, organized by the Tun Fatimah College, Universiti Teknologi Malaysia and the Academic Excellence Program of the Malaysian High School organized by the Federation of Malay Student Unions. Besides, she also held as the protocol unit in the Science Innovation Challenge Program (SIC), Faculty of Science.