**SURVEY ON REPLICATION TECHNIQUES**

**FOR DISTRIBUTED SYSTEM**

* **INTRODUCTION:**

We know that Distributed Computing Systems are finite set of sites that are connected by communication links and they play a significant role in providing the execution platform for **High Computing Performance (HCP)**. In a reliable distributed system, the most essential element is the High Availability. Failures are prone to happen in this type of systems, so a property called Fault Tolerance enables the system to continue functioning the entire system, even when a single component has failed. Whenever a fault is detected, the system recovers from its state or by using any designated **Fault Tolerance**. The important thing we need to consider is, how the fault tolerance can be achieved? The solution is to reconfigure the service from the component that was replaced in place of the failed one or the other approach is to use the masking approach that places redundant resources which contain enough functionality to operate under partial failures.

Here, we are going to have a brief look on Replication and its importance in distributed systems. Replication is a type of masking technique which is used to mask the errors in the replicated component to achieve fault tolerance in the distributed system. It maintains different copies of data or object and the synchronization of updating the data in its replica. This is not considered as a backup method, but deals with when and where to copy the data, optimizing the resources and growing or shrinking the replication tree. This replication consists of two types of solutions one is **Synchronous** and the other is **Asynchronous.**

Synchronous solution will update the two replicas at the same time and rollback if one fails. The benefits of this solution are: **High Availability**, **auto fail-over** and **Minimal data loss** encountered however this solution will have to deal with network efficiency, scalability and cost and it is less flexible. For Asynchronous solution, changes on primary replica will be captured immediately. This solution is flexible, offers low cost and scalable, but it must deal with network bandwidth and loss of data. Replication breaks into two schemes that is **Full replication** (all-data-to-all-sites) and **Partial replication** (all-data-to-some-sites). However, we discuss here only about the Synchronous solution for its higher reliability.

* **REPLICATION IMPORTANCE:**

In a distributed computing environment, many users can use common computing resources. This provides the resources sharing that cause a single fault to be affected to this computing environment. The important part of computing infrastructure is to access the computing resources in a manner by tolerating the failures. It is useful for preventing malicious sites that are degrading the overall performance of the system. Replication enables the system to maintain more replicas to ensure the flexibility of the system. A computer system will be able to achieve higher reliability by using more reliable components. This reliability will lead to a more robust system. Where in a system using more reliable components is not a good option, replication is the best approach to provide higher reliability of the systems by using less reliable components.

When an object is replicated, it will have several identical copies of objects called Replicas. The techniques in replication have been successfully introduced in distributed systems. It allows the system to remain distributed, while allowing the system to increase its availability and its performance. In this type, the systems can operate without the user intervention to tolerate the failures that may occur in the distributed computing environment.

Now, we shall discuss some of the **Replication Techniques** in detail as given below,

* **READ ONE WRITE ALL (ROWA):**

This is the most widely recognized and clear convention utilized in repeating the framework that keeps numerous duplicates of imitations, which can be perused by anybody. In any case, all the copies must be redesigned. This convention deciphers a consistent perused operation on an information thing into one physical read on any of its reproduction. It likewise interprets all its legitimate composes operation to physical composes operation one for every copy. The get to every copy is synchronized by the primary simultaneousness controller, consequently it makes this convention proportionate with the serial execution, where every reproduction that upgrades the information thing will overhaul most of its duplicates or none by any stretch of the imagination.

**ROWA** gives a straightforward and rich method which has the capacity to process read operation paying little respect to any correspondence disappointments since one site will stay up and reachable. The normal for ROWA that gives read operation makes it reasonable for environment that has the greater part of its information in read-just overhead. The critical disadvantages of ROWA is that the convention is inflexible in selecting its read accessibility and will hinder all the compose operation on the off chance that one site is down or inaccessible, until the disappointments is repaired. In the end will cost increase accordingly time and lessen its execution.

* **QUORUM CONSENSUS (QC) PROTOCOL:**

The **Quorum Consensus (QC)** strategy for the most part permits composed operation to be recorded just at a subset (a compose majority) of the up destinations, on a condition that the peruses operation is made to inquiry a subset (a read majority), which is particularly demonstrated that it covers with the compose majority. The read operation will have the capacity to give back its generally as of late composed esteem, at whatever point the majority convergence condition is met and it will be said as having voted in favor of it, giving the QC strategy an option name as **voting**.

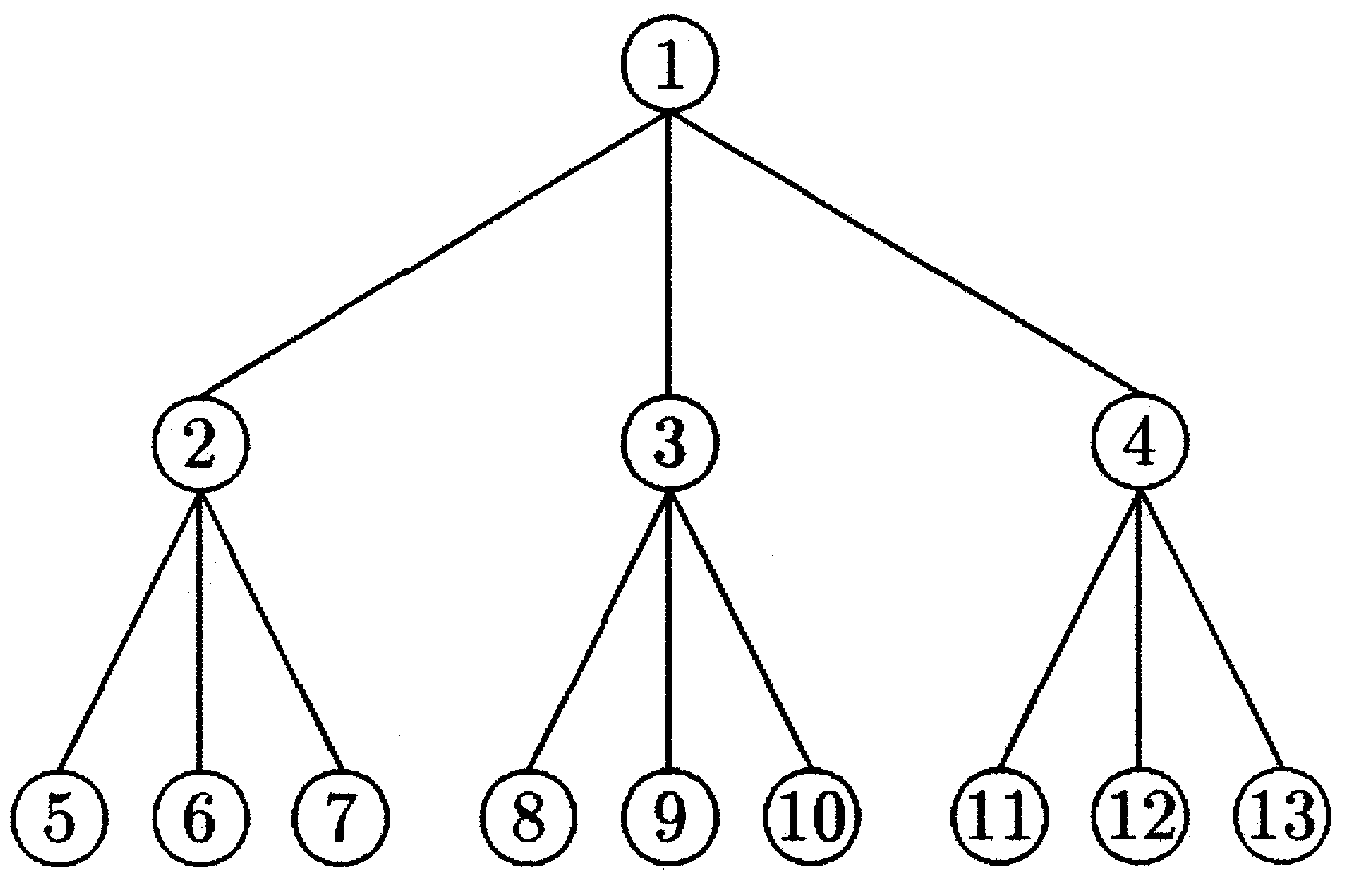
The majorities can be static or element relying upon the doled-out votes and the capacity of the destinations to reconfigure the majority. QC can cover disappointments

with no intercession until the disappointments are endured. Be that as it may, this procedure will cost the read operation to be genuinely costly since the execution of the thought is continuously a troublesome test.

* **TREE QUORUM (TQ) PROTOCOL:**

**Tree Quorum (TQ)**, proposed by Agrawal and El-Abbadi applies replication in a coherent tree structure over appropriated locale as appeared in Figure 1. From this structure, a perused majority can be performed by the root or the greater part of its kids. The compose majority is framed from the root that accordingly frames the greater part of its children what's more, trailed by the development of the lion's share of their children et cetera until it achieves the leaves of the tree. In a best case, a read majority comprises of just a root, {1}. If the root falls flat, a majority is shaped by most of the duplicates at level 1, e.g. {2, 3}, {3, 4} or {2, 4}. In cases, where no larger part is available at level 1 or just hub 4 is open, hubs 2 and 3 will be supplanted by their children in a specific order. In case of all duplicates in level 0 and 1 fizzled, a majority will be performed by most of the children of the chose larger part at level 1. The span of compose majority is altered, however the individuals might be distinctive

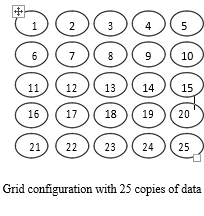
The upside of this convention is the compose operation can get to the quantity of duplicates that are constantly not exactly the lion's share of the majority, while for read operation it might get to one and only duplicate. For read operation, the cost of executing is equivalent with ROWA, however for compose operation it gives a superior result. Lamentably, the compose operation neglects to be executed if more than the lion's share of the duplicates at any level of the tree get to be inaccessible.



* **GRID CONFIGURATION (GC) PROTOCOL:**

This protocol is introduced by Maekawa. A quorum is defined as the minimum number of votes that a distributed transaction must obtain to be allowed to perform an operation in a distributed system. In this protocol to obtain a distributed mutual exclusion algorithm, all the quorums of same size are considered. This mutual exclusion algorithm uses c √N messages to create mutual exclusion in a computer network, where N is the number of nodes and c a constant between 3 and 5. This algorithm is symmetric and allows fully parallel operation. If any node is locked by a member, then no other node can use any of its members which results in a deadlock problem. Hence when more than one node simultaneously requests mutual exclusion, a node will yield to others if its priority is lower than any other conflicting requests. The request priority is based on sequence numbers. Message types like **REQUEST** and **RELEASE** are used to resolve the critical section problem.

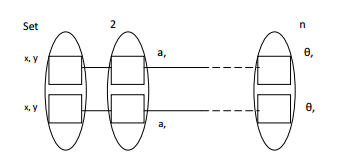
Later this protocol was later extended for replication data objects by Cheung et al. In this protocol we have n copies of data objects which are organized logically in form of a **√N x √N.**



**Read quorum** consists of a copy from every column in the grid which is fetched to perform read operation on the data item. To perform, a write operation, write quorum should have all the copies present in a column and one from each of the remaining columns. For example, if the read operation is performed on the above demonstrated figure copies {1,2,3,4,5} are required and to perform a write operation, we require the copies {1,6,11,16,21,7,13,19,25}. Both the read and write operations are performed within a size **complexity** of **O (√n)**. Usually, this protocol is represented as **(x, y)** where **x** represents the **length** and **y** represents the **width**. A read operation is performed when a read quorum (x, y) is fetched which is formed from x copies in each of y different columns. A write operation is performed when a write quorum is formed from x copies in each of y columns and any of √n-x+1 copies in each of √n+y+1 column. Based on the read and write grid quorum size, intersection property between read and write quorum is determined. If the read grid size is (x, y) and the write grid size should be (√n-x+1, √n-y+1). To increase the fault tolerance of write operations, reconfiguration grid protocol has been proposed. In such case, read and write quorum will have the values (√n/2+1, √n/2+1) i.e., both the operations are executed by accessing majority of copies in majority of the columns.

* The communication cost for **Read Operation** can be represented as
  + - **CGCR = √n.**
* The communication cost for **Write Operation** can be represented as
  + - **CGCW = √n + (√n-1) = 2√n -1.**
* **DRAWBACKS:**
* Communication cost will be degraded by the structure.
* Grid will be prone to a failure while the number of copies of read and write quorum is large.
* **TWO-REPLICA DISTRIBUTION TECHNIQUE (TRDT):**

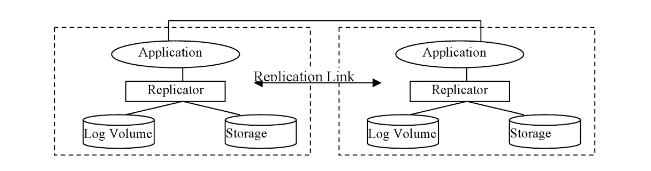
This technique has been proposed by Shen. Chen. Zheng and Shi. In this technique, each node will have equal capacity of storage and all data have two replicas on different nodes and all these nodes have two data replicas. Consider a scenario with N nodes which is divided into n set of nodes (N = 2n) and each set consists of 2 nodes which can be illustrated in the following diagram.



In the above diagram, each rectangle represents a node, and each oval represents a set which in turn consists of two nodes. Data x and y have two replicas which are represented by the nodes from set 1. In the same way data from set a also has two replicas which are located in the set 2 and the sequence continues. Whenever a primary node receives a request from the replication link, replication is done directly on the replica. These replication techniques use asynchronous replication.

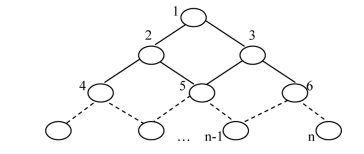
Asynchronous Replication is a technique that writes data to a primary storage array first and then depending upon the implementation approach, commits data to be replicated to memory or a disk-based journal. It then copies the data in real-time or at scheduled intervals to replication targets. This technique is a store and forward approach to data backup or data protection.

* **ARCHITECTURE OF A TYPICAL TWO-REPLICA DISTRIBUTION TECHNIQUE:**



* To achieve high reliability, this technique acquires the system to have replica-availability of more than 99%.
* **DRAWBACKS:**
* This technique would double up the resources and adds to each of the serves with the second replica even though the incremental value of availability is not trivial.
* If one of the sets specified in the diagram is not available, then the operation will be terminated.
* If both the replicas are lost/damaged, this technique would expose the nodes to experience double faults.
* **NEIGHBOUR REPLICA TRIANGULAR GRID (NRTG):**

This technique has been proposed by Ali Manmat, M. Mat Deris, J.H. Abawajy and Suhaila Ismail. A concept called Partial Replication (all-data-to-some-sites) is implemented in this technique where the replicated data is copied only to the neighbor nodes. By doing so, storage capacity will be minimized, and higher updates will be available. All the sites are arranged logically in the form of a triangular grid structure. In this grid structure, all the inner leaf nodes present are linked together resulting in a binary tree. A master data file is associated with every site present in the structure. Copies will be available at the site only if the site is operational else it will be unavailable.



To provide high availability of executing write operations on the replicated database, this technique requires minimum quorum size. This technique minimizes the storage, the high availability for update - frequent operations by imposing a neighbor binary vote assignment to the local grid structure on data copies and even data access time. This is because it requires minimum number of replication sites.

* **ADVANTAGES:**
* Low communication cost for an operation.
* Applicable for large systems because of higher system availability.
* **NEIGHBOUR REPLICATION DISTRIBUTED TECHNIQUE (NRDT):**

This technique has been proposed by R. Mamat, M. Mat Deris, and M. Jalil. All the nodes are logically organized in the form of a two-dimensional n x n grid structure. This would in turn be organized in the form of n x n when there are nodes in the environment where **N = n2.**

The main idea in this technique is, the update on the primary copy will be committed first at the primary node which would be updated to the neighbor nodes asynchronously within a separate transaction. In this technique, whatever the data which is present will have some replication the nodes present, and these nodes have some data within the environment. Data replication is done to the neighbor nodes only when the number of replications from each node is less than or equal to added value of neighbor nodes count and data from the current node. In this we also have recovery mechanism in some failure cases which is given below.

* **FAILURE RECOVERY MECHANISM:**

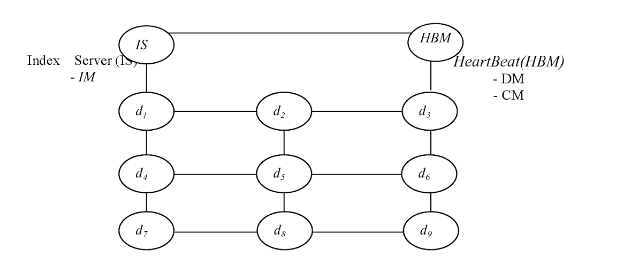


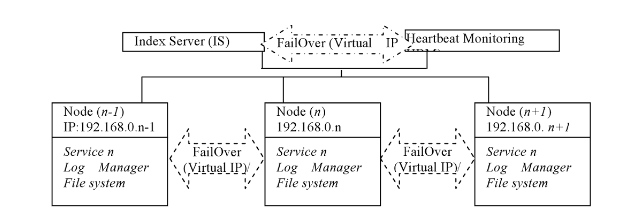
Fig: Architecture with IS and HBM

**Heart Beat Monitor** (HBM) would monitor and detect the occurrence of failures with in the application which are registered with AR. The faults will be monitored continuously, and the status will be updated in the Index Server (IS) and provide space for the happening of recovery operations. This recovery can be done in many ways like kill and restart, fail-over, virtual-IP and others. **Index Server (IS)** and **Heart Beat Monitor (HBM)** are included with in the distributed environment to improve reliability, fault-tolerance and failure recovery. Both IS and HBM back up each other as the data which is associated with them is their replica.

* Main purpose of the HBM is to detect the failure occurrence and the status need to be indexed with the HBM for further reference. HBM is comprised of **Two Components** namely:

1. **DETECTION MANAGER (DM):** This is used to detect if any failures are present in the participating nodes.
2. **COORDINATION MANAGER (CM):** In the presence of any failure node, this CM would decide a candidate node which would virtually act as a backup node for the failure node and render the functionalities of that node. This decision can be done by using Virtual IP.

Main component present with in the **Index Server (IS)** is **Index Manager (IM)**. Indexed content of the checksum data file is stored in the participating nodes with the help of Index Manager (IM). Once a replica has been elected, it then acts as a primary source. All the transferring of primary services from failure nodes to primary nodes comes under recovery process.



This technique has low overhead cost when compared to **TRDT** model because of the presence of two redundant components. This technique availability has been reduced in a small percentage with a predicted value of about 1.2% for a period of 10 years.

* **CONCLUSION:**

This report tells us about different replication techniques being used in the distributed systems and we have generalized the concept of each technique.

* **REFERENCES:**

1. A.S.M. Noor, T. Herawan and M.M. Deris, “Neighbor-Replica Distribution Technique Model for Availability Prediction in Distributed Interdependent Environment”, Int’l Journal of Cloud Applications and Computing, 2013.
2. H. H. Shen, S. M. Chen, M. M. Shen, and W. M. Zheng, “Research on Data Replica Distribution Technique for Server Cluster”. IEEE Proc. 4th Int’l. Conference on Performance Computing, Beijing, pp. 966-968, 2000
3. D. Agrawal and A. El Abbadi, “Using the Tree Quorum Protocol: An Efficient Approach for Managing Replicated Data”, Proc 16, Int’l Conf. on Very Large Database, 1998, pp 243-254.