I. J. Computer Network and Information Security, **2012, 10, 13-23**
Published Online September 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2012.10.02

Self Organized Replica Overlay Scheme for P2P Networks

Shashi Bhushan
Chandigarh Engineering College, Landran, Punjab, INDIA
shashibhushan6@gmail.com

Mayank Dave
Natioanal Institute of Technology, Kurukshetra Haryana, INDIA
mdave67@yahoo.com

R. B. Patel
G. B. Pant Engineering College, Pauri Garhwal, Uttarakhand, INDIA
patel_r_b@yahoo.com

Abstract **—** Peer-to-Peer (P2P) systems are widely used for data sharing applications in an autonomous and decentralized mode. P2P systems are suitable for large-scale distributed environments in which nodes can share resources other than data such as computing power, memory and network bandwidth. Some of important parameters that affect the performance of P2P systems are peer availability, data availability, network overhead, overlay structure, churn rate, and data access time.

In this paper a self organized replica overlay scheme “Improved Hierarchical Quorum Consensus” (IHQC) for P2P systems is proposed. This scheme organizes replicas in a Self Organized Hierarchical Logical Structure (SOHLS) that has special properties. The scheme improves performance of the system by reducing search time to form read/write quorums, reducing probability of accessing stale data, improving degree of intersection among consecutive quorums and reducing network overhead. This scheme is highly fault tolerant (tolerate up to $n-1$ faults) due to replication of data and inherits the best property of Read-One-Write-All (ROWA) protocol in a dynamic environment of P2P network. The architecture for IHQC is also proposed for implementing the scheme that supports improved performance of P2P systems. This scheme also maximizes the degree of intersection set of read and write quorums; hence, having higher probability to get updated data as compared to all other schemes. The mathematical correctness of the scheme is also presented in the paper. The results of simulation study of the proposed scheme also support and verify its better performance than Random and Hierarchical Quorum Scheme.

Index Terms **—** Data Replication, Hierarchical Quorum Consensus, Self-Organized Replica Overlay, Data Availability, Network Overhead, Fault Tolerant

I. INTRODUCTION

Peer-to-Peer (P2P) systems are very popular systems due to scalability, resource availability, node autonomy, fault tolerance, self-configuration and decentralized control. Examples include Napster, Gnutella, Freenet, BitTorrent etc. [1][2][3]. P2P systems provide an environment that overcomes limitations of the client/server based systems by avoiding bottleneck of scalability [4]. This advantage enables P2P systems to perform complex tasks with relatively low cost and without any need of powerful servers. P2P systems are used for sharing data and resources such as computing power, storage space, network bandwidth and all other attached resources in an autonomously decentralized manner [5]. The number of resources is increased for utilization by increasing in the number of connected peers to the P2P network. The applications which require high data availability and can compromise on the reliability can be implemented over P2P systems. A Distributed Data Base System (DDBS) can be implemented over an existing P2P system with very low cost of implementation as computing power and storage space is available free of cost. DDBS are known for their improved performance over conventional database management systems (DBMS). Data replication is a technique to improve the performance of DDBS [6-10] and make the system fault tolerant [11-13].

Replication improves the system performance by reducing latency, increasing throughput and increasing availability. However, data replication is the basic requirement for the DDBS [14] deployed on the networks that are dynamic in nature for example P2P systems [15]. In P2P systems (or networks) peers can join or leave the network at any time with or without prior information to the network. This is generally called “Churn Rate”. The churn rate of peers is observed to be high in P2P networks [16-19]. For such a highly dynamic environment, probability to access stale data from the replicas is higher as compared with the static environment where nodes do not leave the system.

Several protocols have been developed to solve the problem of accessing updated data items from replicas in dynamic environments. Examples include single lock, distributed lock, primary copy, majority protocol [20], biased protocol, and quorum consensus protocol [21, 22]. These protocols are used to keep data consistent and to access updated data items [19] by using multiple replicas maintained in the distributed system.

A group of replicas are accessed to get updated data items from the replicas. This group is generally known as “Quorum” [16] and depending upon the operation, quorum is said to be “Read Quorum” or “Write Quorum”. To get the updated data item, read-write quorums and two consecutive write-write quorums must intersect. The intersection is set of replicas which are common in read-write and two consecutive write-write quorums. This ensures that the read quorum always gets updated data from the system. This updated data can be propagated to all other replicas. The degree of intersection of two quorums makes the system resilient to churn rate of the peers.

The objective for this paper is to find a scheme through which database and its replicas can be distributed efficiently over dynamic network e.g., P2P. For this purpose a logical structure is to be identified which fulfill the above said objective with better response time, fast searching time of replicas for quorums, fault tolerant, having high degree of intersection among consecutive quorums generates minimum network traffic in the system. In this paper a self organized replica overlay scheme “Improved Hierarchical Quorum Consensus” (IHQC) is presented. This scheme reduces network traffic, maintain data availability in dynamic level while improving response time and access time of the quorum system. Our focus is on the hierarchical quorum consensus protocol. The proposed scheme also addresses some of the issues discussed above. This scheme takes advantage of overlay topology which is the logical topology deployed over the physical topology. The overlay topology can be changed without affecting the physical topology. To reduce the search time for generating read/write quorum and response time to access data, a logical structure is identified under IHQC named “Self Organized Hierarchical Logical Structure” (SOHLS).This scheme also produces maximum quorum intersection set and makes it fault tolerant.

Rest of this paper is organized as follows: Section 2 gives the related work. Section 3 explores System Architecture, Section 4 introduces proposed IHQC, SOHLS and mathematical correctness of proposed algorithm, Section 5 explores the results and discussions and paper is concluded in Section 6.

II. RELATED WORK

In the literature, many replication protocols have been suggested in [23, 24] for replica management protocol in a Binary Balanced Tree. The most simple replication protocol is the Read One Write All (ROWA) [25]. This protocol is suitable for static networks having fixed and dedicated servers for the replication. It has minimum read cost amongst other protocols and is highly fault-tolerant. This protocol has maximum communication cost for write operation. This communication cost increases with increase in number of replicas. In dynamic system update-all creates the problem of unlimited wait. A variation of this technique is known as Read One Write All Available (ROWAA). The scheme requires all replicas to be available to perform a write operation, which improves data availability for dynamic environments. [26]

The Dynamic Voting protocol [27] and Majority-Consensus protocol [28] perform better than ROWAA in dynamic environments. In both protocols the number of replicas is accessed in groups. These protocols have good read and write availability but have a disadvantage of high read cost. They have long search time to search the replicas as the replicas are stored randomly in the network.

Rather than storing replicas randomly, logical structures [29-31] have been proposed to store replicas over the dynamic network. These protocols reduce search time to make quorum from the replicas and reduce communication cost. The Multi Level Voting Protocol, Adaptive Voting [32], Weighted Voting, Grid protocol [33] and Tree Quorum protocol [34] are such replication protocols each with different operational process. The Multi Level Voting protocol is based on the concepts of the Hierarchical Quorum Consensus (HQC) strategy. HQC [23, 35, 36] is a generalization of the Majority Scheme. In this tree structure, replicas are located only in the leaves, whereas the non-leaf nodes of the tree are said to be as “logical replicas”, which in a way summarize the state of their descendants. The advantage of tree structure is it reduces the search time to find replicas from the structure as compare to the random structure. Tree structure also reduces the message transfer to find replicas; hence, it reduces the network traffic generated in the system. A disadvantage of Tree Quorum protocol is that the number of replicas grows rapidly as the tree level grows. In case of Adaptive voting and weighted Voting protocols the formed quorum satisfies some conditions which are (a) write and read quorums always made up of more than half replicas. (b) Write and read quorum must be such that they intersect with each other. The disadvantage of these protocols is the size of quorums grows with increase in number of replicas; hence, network overhead automatically increases in the system.

Bandwidth Hierarchy Replication (BHR) is proposed in [37]. BHR reduces data access time by avoiding network congestions in a data grid network. In [38] author proposed BHR algorithm by using three level hierarchical structures. The proposal addresses both scheduling and replication problems. Two replication algorithms Simple Bottom-Up (SBU) and Aggregate Bottom-Up (ABU) for multi-tier data grids are proposed in [39]. These algorithms minimize data access time and network load. In these algorithms replicas of the data should be created and spread from the root center to regional centers, or even to national centers. These strategies are applicable only to multi-tiered grids. The strategy proposed by Kavitha et. al. [40] creates replicas automatically in a generic decentralized P2P network. Their goal of proposed model is to maintain replica availability with some probabilistic measure. Various replication strategies are discussed in [41]. All these replication strategies are tested on hierarchical Grid Architecture. A different cost model was proposed in [38] to decide the dynamic replication. This model evaluates the data access gains by creating a replica and the costs of creation and maintenance for the replica.

There are several challenges to update and access replicated data items over a dynamic network like a P2P network. Data consistency, search time to find replica and fault tolerance are some of the identified problems. Any new proposals should support dynamic environment of P2P system and should have low search time, fast recovery from faults and access to updated data.

III. System Architecture

In this section proposed system architecture is presented for distributing and accessing replicas in the system. This architecture comprises of various modules, which helps the system work according to the requirement mentioned in section 2. The description of various modules in the architecture shown in Fig. 1 is as follows:

Query Optimizer: Queries are received from the requester by this module of the architecture. Query is divided in to number of subqueries. Subqueries are further optimized by the module Query Optimizer. This optimizes the subqueries to reduce the execution time of overall query. Query Optimizer module decides the order of subqueries to be executed by the system. Various conventional optimizing techniques may be used to optimize the subqueries.

Subquery Schedule Manager: Subqueries which is ready to execute, are scheduled to achieve the one copy serializability of the subqueries. Subquery Schedule Manager is responsible to rearrange the order of subqueries to achieve the above said property. Concurrency and data consistency is also maintained by this module. Concurrency control algorithms are also integrated within this module.

Quorum Manager: This module is responsible to decide the quorum consensus to access the data item. Quorum is decided such that the system got the acceptable availability of the replicas i.e., the number of replica to be accessed is increased if the availability of the peers storing replica are low. The number of replicas may be reduced if the availability of peers is high to reduce the overhead of the network. This module is responsible to maintain the availability of the replicas to the desired level. This module recognizes the replica to be accessed from the logical structure. The steps to form read/write quorums are implemented in this module as discussed in next section.

Replica Search Manager: This module is responsible for searching any replicas from the group of replicas. This group of replicas is arranged in a logical structure by Replica Overlay Manager module. The replicas for read/write quorums are searched by this module. The replica search is done according to the steps mentioned SOHLS. The performance of the system also depends upon the performance of search algorithm used. The time required to search replicas is reduced in case of maintained logical structure, as the structure reshuffle itself in case of any failure.

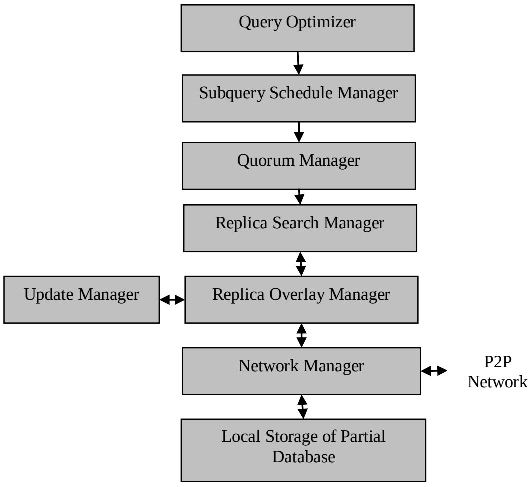

Figure 1. Proposed System Architecture of IHQC

Replica Overlay Manager: Response time and access time are used to measure the performance of the system. Efficient algorithm may reduce the search time of replicas from the system. Replicas are arranged in some logical structure, to increase the performance of the system. This module is responsible to place the replica in a logical structure to easily access the replicas. This module is used for actually identifying the replica for making quorum. This module also periodically maintains the logical structure, in which replicas are placed. Every time when any replica leaves the network, this module reshuffles the replicas by arranging the addresses of these replicas in the logical structure. All the readjustments are carried out according to the rules of SOHLS.

Update Manager: The major aim of this module is to maintain the freshness of data item. All update strategies are implemented by this module. The write through method is used to update the prioritized replicas, selected for the write quorum. Write back method is used for the remaining replicas existing in the system. This update is through the Replica Overlay Manager which is described above. This module implements the updation algorithm, so that data items stored at replica must contain the latest information. The algorithm is selected such that it updates the information on various replicas with the minimum time limits. The performance of the system is affected by the time consumed to update data items by the algorithm. The probability to access stale data from the system is minimized by minimizing the update time of the system.

Network Manager: Every logical address of the replicas are checked and converted in to the physical address, that replica are controlled through this module of the architecture. Data and control information is routed by this module. This module also maintains the connection between various peers. The complete information regarding underlay topology is maintained by this module. All replicas are accessed through this module.

Local Storage of Database Partition: This is an actual database to be accessed by the requester. All data items belonging to the database partition are physically stored in this memory space. This is the region provided by the owner of that peer. This memory region is shared among the network. The conventional data encryption techniques can be implemented to protect the data from any attack/misuse by any unauthorized person.

IV. Improved Hierarchical Quorum Consensus (IHQC) scheme

The primary objectives of the scheme are to provide data availability up to the acceptable level, fast response time, fast quorum formation time, high degree of intersection among consecutive quorums, reduced network traffic and should be highly fault tolerant. These are the hard objective to be achieved in the dynamic network e.g. P2P networks due to high churn rate of the replicas. To achieve these objectives in the P2P networks one has to focus on the logical structure in which replicas are arranged/distributed and accessed.

Session time of each replica can also be utilized to improve the system performance. The replicas with longer session time and actively participant in the system also have the high probability to hold updated data items. Prioritizing this type of replicas for generating the quorums can increase probability to access updated copy of data items.

Degree of Intersection in any two consecutive quorums is the number of common replicas accessed in both quorums. These common replicas provide and propagate the updated data items in to the system. Increase in number of such common replicas improves the system performance by increasing the probability to access fresh data items. Hence, performance of any quorum system can be improved by increasing the degree of intersection among read-write and write-write quorums. To increase the degree of intersection in the system, logical structure is accessed in special manner.

Logical topologies can be changed without affecting the physical topology, with this property of topology any logical structure can be modified any time. A self organized structure can also be generated using this property of the logical structure. To reduce the network traffic in P2P networks, directional forwarding or multicasting is recommended; thus, efficient logical structure play an important role to achieve the above said objectives.

The number of structure can be used to manage the peers in logical structure, e.g., tree, square, circle, cube etc. A Hierarchical Quorum Consensus (HQC) is proposed in [35]. In HQC all replicas are arranged in tree like logical structure and this is having reduced access time to form quorums. The proposed IHQC scheme is presented in this section. This is improved version of HQC. IHQC maintain a tree like logical structure named "Self Organized Hierarchical Logical structure" (SOHLS) to manage the replicas.

Initially, peers participated in overlay topology are checked for their average session time. The peers having highest average session time are selected to hold the replicas in the system. These replicas are arranged in tree like logical structure such that they can form almost complete tree. The replica having longest up session time in the system is selected to be the root of SOHLS. Similarly, other replicas are also arranged as left child and right child, according to the session time of each replica in the system. The idea for this scheme is to access replicas having updated copy of data items. All replicas having higher session time are placed on root side in the logical structure. These top replicas are further participated in every read/write quorums. Regular participation of these top replicas maintains updated copy of data items. Hence, the probability to access updated copy of data items is increased. The SOHLS is having some special properties e.g., each peer holding replicas are arranged in the tree such that the session time of each parent in SOHLS is greater than its child. The session time of left child is greater than right child of any parent. All peers participating in the system find the alternate path for all its ascendants. This includes all parent replicas comes across the path from leaf to root.

The replicas with highest session time are given priority over the smaller session time while included in the quorum from SOHLS. The quorum is formed by first taking the replica at the root of the SOHLS and then replicas at the branches of the SOHLS i.e. from top to bottom. The branches of parent node in SOHLS are accessed such that the left child is accessed before right child of that branch, i.e., from left to right. In the paper this pattern to access replicas from SOHLS is referred as Top-to-Bottom and Left-to-Right. Each read quorum can get the updated copy of data items, if the quorums are formed according to the rules specified in IHQC. The proof is presented in section 4. In IHQC all read-write and write-write quorum intersect each other; hence, every read quorum accesses the updated copy of data item. The maximized degree of intersection set from two consecutive read-write and write-write quorum ensures the access of fresh data items.

The problem of finding a path between a pair of source-destination peers in the overlay is the problem of finding a route between the source and destination peers in the underlay topology. Path between peer P_1 and P_2 is direct of one hop count distance and path between peer P_1 to P_4 is of two hop count distance as shown in Fig. 2. The identified route between source and destination in the underlay may or may not be the shortest path. However, a shortest path in the underlay will be advantageous for reducing communication cost [42].

The working of IHQC scheme is divided in to two independent parts, accessing the group of replicas and maintenance of logical structure. Both parts are executed in parallel to improve the efficiency of the system. The replicas from root to terminal nodes are included in quorums to maximize the degree of intersection set. The level up to which replicas are included depends upon the size of quorum, e.g., 1,2,3,4,5,6,7,8 are the replicas and generate SOHLS shown in Fig. 3. The replicas 1,2,3 are used for quorum of size three replicas and 1,2,3,4 are used to form a quorum of size four. Here the replicas with higher session time are given priority to form the quorums shown in Fig. 3.

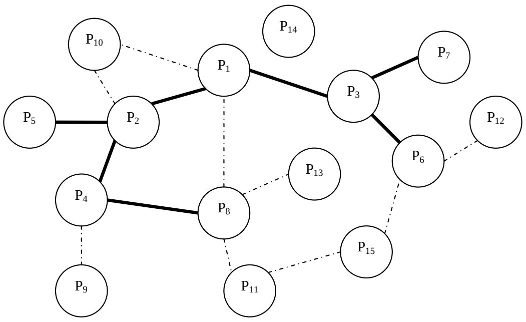

Figure 2. Diagram shows the arrangement of peers to make Self Organized Hierarchical Logical Structure (SOHLS) over the peers from underlay topology of P2P networks. Here the dotted line connectors show the connection between the peers in overlay topology. The dark line connectors show the connection between the peers in the replica topology in SOHLS. P_{14} is shown as isolated peer in the network

Every time replicas are accessed from same position from logical structure this increases the degree of intersection among consecutive quorums. The common replicas will increase the probability to access updated data items from the system. As these replicas are accessed form upper part of tree in Top-to-Bottom and Left-to-Right fashion, so time to search these replicas is minimum.

Proposed scheme provides better data availability as compare with HQC, due to priority given to the replicas having higher session time for quorum. All replicas having updated copy of data items are accessed on priority. IHQC also reduces the network traffic as compare to the HQC as the replicas are also placed on intermediate branches of the tree.

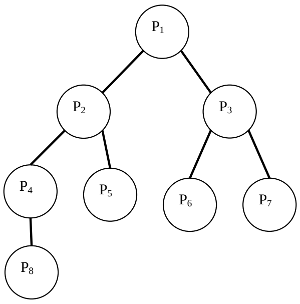

Figure 3. Showing replica arrangements in the Self Organized Hierarchical Logical Structure from Figure 2. Here the session time of P1 is greater than the P2 and P3. The order of the replicas according to session time from the SOHLS is P1, P2, P3, P4, P5, P6, P7, and P8

IHQC uses the self organized mechanism which ensures replicas with longest session time are always used to construct the quorums. The SOHLS triggers maintenance procedure for every leaved replica. Other replicas in the systems are adjusted according to their session time. The replica with next higher session time takes the position of leaved replica from the network. By default, the replicas with longer session time move in the upward direction with passage of time as discussed below.

The performance of the system remains same even in high churn rate of peers. With maximized overlapped replicas in read and write quorums, this scheme ensures the access of fresh data items from any number of replicas in the system. The IHQC also provides high fault tolerance. With self organization this scheme tolerate up to $n-1$ faults among n replicas in the system. Multicasting and directional forwarding is used to transfer the messages in the system. The IHQC performs better than HQC in respect of search time to form quorums, response time and probability to access updated data items even in dynamic environment of network.

A. Creation of Self Organized Hierarchical Logical Structure (SOHLS)

The SOHLS is a special type of logical structure similar to the complete binary tree [23] which is used for Replica Overlay. A SOHLS of size n is a binary tree of n nodes which satisfies the following two conditions:

(i) This binary tree is almost a complete tree, which means that there is an integer k such that every leaf of the tree is at level k or $k+1$ and if a node has a right descendant at level $k+1$then that node also has a left descendent at level $k+1$.

(ii) The key in the nodes are arranged in such a way that the content of each node is less than or equal to the content of its father. This means for each node$Key_i \le Key_j$, where j is the parent of node i .

A peer at level k holds the addresses of its connected peers at $k-1$ and $k+1$ levels in SOHLS i.e., each peer stores the addresses of its directly connected siblings and of its parent. Simultaneously each peer stores the addresses of all its grandparent peers come across the path from that peer to root of the SOHLS. All peers/replicas follow the rules of SOHLS to make this overlay logical structure or replica overlay. The session time of each peer/replica is used as key in the replica overlay. The use of session time as key results in the movement of replicas having longer session time towards the root. Each newly joined peer connect at the position of leaves in logical structure decided according to the rules of SOHLS. These peers also search the alternate path of each parent up to root. Each peer holding replicas transmit the beacon against its active status to all its directly connected peers. This addresses and beacon is used for making the connection in case of any failure.

The addresses of peers are used to access the peers in a particular sequence. IHQC reduces the search time in building the quorums by using minimum hop count. This reduction in search time is even less than that achieved by HQC.

B. Effects of replica leaving from the replica logical structure

When any replica leaves by informing or without informing the system, the following steps are taken to maintain the replica logical structure by the system:

Assuming that replica x is at level k going to leave the network, e.g., peer 2 going to leave the network shown in Fig. 4.

(i) The replicas at level $k+1$, which is directly connected with the replica x, tries to connect with its alive grandparent (addresses are stored at each peers joined in the system).

(ii) Alive grandparent compare the session time in case of multiple replicas approaches to connect. The replicas with highest session time will connect with the active grandparent and will take the position of leaved replica x in the logical structure as shown in Fig. 5. The remaining replicas join the system according to the rules of SOHLS toward downwards.

(iii) The replicas at level k under parent at level $k-1$ are adjusted according to conditions of SOHLS mentioned above.

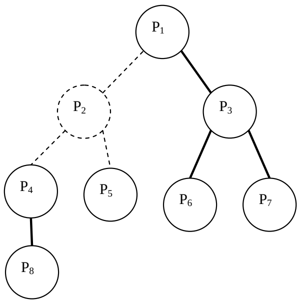

Figure 4. Replica arrangements in a SOHLS logical structure. Peer 2 which is shown by dotted lines is a peer leaving the network

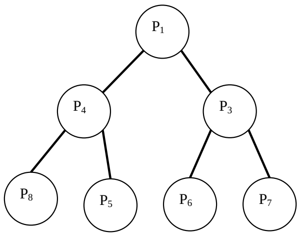

Figure 5. The SOHLS structure after leaving of Peer 2. Peer 4 takes the position of Peer 2 which already leaved the network. All other replicas in downlink are readjusted accordingly

C. Effects of replica joined in to the replica logical structure

When any replica rejoins the system, following steps are taken by system to maintain the replica overlay topology:

(i) Initially rejoined peer searches its position through ping pong messages starting from the root, assuming the position is at level k.

(ii) Replica establishes the connection with its parent peer and both save the addresses of each other.

(iii) Replica updates its data items by comparing the data items with its alive parent and update version number of data items.

(iv) Replica stores the addresses of its entire grandparents till root, through the path find message.

D. Creation of Read/Write Quorum from SOHLS

The following rules are defined to access the SOHLS to form the read/write quorums:

(i) $Size\,of\; Qr_i \leq Size\,of\; Qw_i$, the size of read quorum is always less than or equal to the size of write quorum.

(ii) Root is always included in the read/write quorum.

(iii) For any integer k, if the replica at k level is in the quorum then every replica from $k-1$ level must be in the quorum of the SOHLS. This rule is referred to as Top-to-Bottom in this paper.

(iv) If a replica from right descendent of a parent replica is in the quorum then there must be a replica from left descendent which also in the quorum. This rule is referred as Left-to-Right in this paper.

These rules are implemented in the proposed scheme by combining Top-to-Bottom and Left-to-Right to access replicas from the logical structure.

Read Write Quorum: The quorum size depends upon overall availability of replicas and network overhead of the network. The size of quorums may be increased in case of low availability and reduced in case of high availability of the replicas. The replicas included in quorums are selected according to the rules mentioned above. The size of quorums also affects the network traffic. The number of messages transferred to maintain the SOHLS increases with increase in quorum size. Therefore, network overhead due to network traffic is increased in the system. The quorum size is directly proportional to the network overhead and there is tradeoff between the network overhead and quorum size of the replicas.

The IHQC scheme uses the fixed number of replicas in quorums decided after considering all factors affecting the system for read/write quorums. The quorum size of read and write quorum may be different depending upon the requirement of system. The replicas are included in sequence from SOHLS according to session time to form the quorums.

Example: Considering quorum size equal to four, then all four replicas available at the top, starting from root to branch and left to right will be in read/write quorum of the IHQC. The peers P_1 and P_4 are used for quorum size two. The peers P_1, P_4 and P_3 are used in quorum of size three. The peers P_1, P_4, P_3 and P_8 are used in quorum of size four by considering the logical structure sown in Fig. 5.

All accessed replicas in read quorum are compared for fresh version of the data items. In the best case only the root can be accessed for fresh data. This is the property of ROWA protocol [25] which is the best identified protocol having minimum cost for read quorum.

Write quorums are decided same as the read quorums. The replicas from top to bottom and left to right are selected from the SOHLS to form quorum. Whenever write query is executed in the system, all the replicas in quorum are updated by write through method, i.e., write is committed after receiving acknowledgement from all available replicas in the quorum. The remaining replicas in the structure are updated with write back method. Here maximum queries are responded by the top most replicas of SOHLS structure having longer session time. The replicas which are not used in write quorums are updated in comparatively less time than the HQC due to extra links provided at each level. These extra links reduce time of update message to reach all replicas in the system.

D. Correctness of the Algorithm by the Mathematical Induction

We use Mathematical Induction to prove that the number of replicas accessed in the read quorums from IHQC has at least one replica having updated data items. Assuming replicas are organized in the SOHLS in height h.

Basis:

(i) Assuming the height of the SOHLS be 0 i.e., only one peer/replica is in the structure (placed at root). Since according to the algorithm rule (ii) read as well as write quorum must involve root peer in the quorum. Every read/write quorum includes this replica. Hence, every access gets the updated data items from the root. These quorums mathematically described as:

$Qw_0 = \{P_0\}$, $Qr_0 = \{P_0\}$, $\therefore Qr \subseteq Qw$, $Qw_0 \cap Qr_0 \neq \phi$ read quorum and write quorum intersect with each other. Therefore, the statement that read quorum gets the updated data items is verified.

(ii) Assuming height of the SOHLS be 1 i.e., SOHLS has maximum 2-3 peers. One replica is at the root and 1-2 in down link of the root. According to the algorithm, the size of write quorum is greater than or equal to the size of read quorum. The replicas in the quorum are selected through Top-to-Bottom and Left-to-Right. Write quorum Qw_1 can be constituted of the following sets:

$$Qw_1 = \{\{P_0\},\{P_0,P_1\},\{P_0,P_1,P_2\}\} \quad ... \quad (1)$$

Read quorum is constituted of the following set:

$$Qr_1 = \{\{P_0\},\ \{P_0,P_1\},\ \{P_0,P_1,\ P_2\}\} \quad ... \quad (2)$$

For every possible set of read quorum against write quorum, quorums intersect each other; hence, always get the updated information.

From (1) & (2)

$$\forall\ Qr_1, Qw_1 : Qr_1 \subseteq Qw_1,\ \therefore Qw_1 \cap Qr_1 \neq \phi$$

All possible read quorums always have at least one peer having updated replica. This implies that read quorum always accesses the fresh data item, as intersection is always non-empty. Hence, the above said statement is verified.

Hypothesis:

Assuming for SOHLS of height i and Qw_i, Qr_i the write and read quorums of size l and k respectively and defined as:

$$Qw_i = \{P_1,P_2,...,P_k,...,P_l\} \quad ... \quad (3)$$

$$Qr_i = \{P_1,P_2,P_3,...,P_k\} \quad ... \quad (4)$$

$l, k \le 2^i - 1$, and $k \le l$ where $2^i - 1$ is the total number of replicas up to i level of the logical structure. Replicas from the SOHLS are accessed in Top-to-Bottom and Left-to-Right fashion as mentioned in the algorithm. Assume that entire replicas up to P_k comes in the intersection set of write and read quorums according to the rules mentioned in algorithm, shown in (3) and (4),

$$Qw_i \cap Qr_i = \{P_1,P_2,P_3,...,P_k\} \quad ... \quad (5)$$

Therefore each read quorum accesses the updated replicas as intersection of write and read quorum is not empty.

Inductive Step:

We have to prove that this is also true for the SOHLS of height $i+1$. According to the algorithm, write quorum of size n is defined as:

$$Qw_{i+1} = \{P_1,P_2,P_3,...,P_k,...,P_l,...,P_n\} \quad ... \quad (6)$$

Read quorum of size m is defined as:

$$Qr_{i+1} = \{P_1,P_2,P_3,...,P_k,...,P_m\} \quad ... \quad (7)$$

If the size selected for the write quorum is n and size for the read quorum is m. Where $l \le n$ and $k \le m$. From this statement as the quorum is formed in a pattern of Top-to-Bottom and Left-to-Right, as mentioned in the algorithm.

$$Qw_i \subseteq Qw_{i+1}, Qr_i \subseteq Qr_{i+1} \quad ... \quad (8)$$

From (5) and (8)

$Qw_{i+1} \cap Qr_{i+1} \neq \phi$, at least this intersection set is equal to $\{P_1,P_2,P_3,...,P_k\}$ from (5). From above statement it is proved that every read quorum intersects with the write quorum; hence, every read quorum carries the updated information. The correctness of our proposed algorithm is thus proved.

Fault Tolerance: This scheme can tolerate up to $n-1$ faults among n number of replicas participating in the system.

Availability: It is the probability that at least one replica is available in the system and is given as

$$1-\prod_{i=1}^{n}(1-P_{r_i}) \quad ... \quad (9)$$

where P_{r_i} is the probability of i^{th} replica to stay alive and n is the number of replicas .

V. Simulation and experiment

For simulation, the network consists of 500 peers in underlay are used, 5%-20% of total peers are used in the overlay topology. A discrete-event simulator developed in C++ is used to simulate the network. Random peer placement, HQC and proposed IHQC topology is implemented for the simulation. Average search time, response time and average message transfer to maintain the system after executing write quorum are taken as performance metrics in the system. We have used the

Dijkstra's algorithm to find shortest path in all our scenarios.

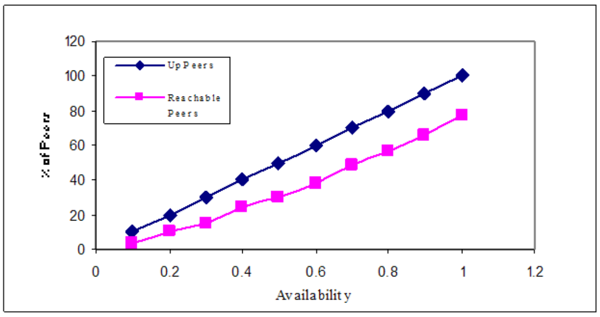

Figure 6. Reachability of peers under availability in the network

The reachability of a peer in the network is defined as the path exists from any source to that peer. Failure of any peer may cause network partitioning. The peers belonging to the network having more than one partition, are not reachable from the peer belongs to another partition. Peer availability is one of the factors responsible for the network partitioning. From simulation reachability of a peer is observed depending upon the peer availability. The network is partitioned with low availability peers. This is due to the small session time and small number of connections that peers have in the system. The number of active peers is also less in case of low availability. It is also observed from Fig. 6 that approximately 80% of peers are reachable at 100% availability. This reachability affects search time of replicas during the searching. The low reachability increases path length of the searched peer. Hence, it increases the cost to access the replica.

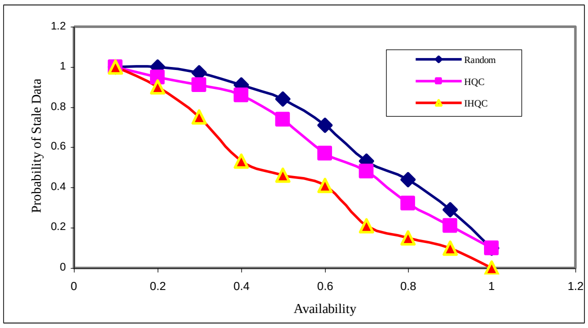

Figure 7. The comparison to access stale data under availability of peers in the system

The availability of peer is calculated as the total up time of peer over the total time of peer including down time. It is observed from the simulation that probability to access stale data decreases with increase in availability of the peers. In IHQC, the probability to access stale data is very less as compared to the Random and HQC. Here we have considered all subqueries accessing the data items from the quorum of replicas. The percentage of stale data is calculated as the number of accessed replica having stale data over total replicas accessed in the quorum. It is also observed that the probability to access stale data is in acceptable range with replicas having availability greater than 70%. The peers having availability more than 70% may be given priority to store the replica over other peers so that the performance of the system can be increased.

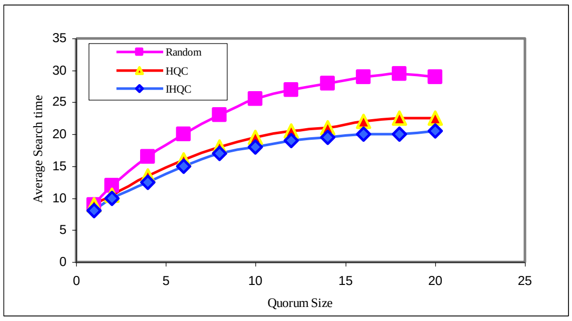

Figure 8. The comparison of average search time to form the quorum from the networks

The average search time is calculated as the average time taken by system to search replicas for completing the quorums. It is also observed that the search time to access quorum is increased with increase in quorum size. The average search time for random quorum consensus is comparatively more than that of IHQC. The search time also increases with increase in quorum size. In the case of IHQC, the slope of the search time graph is increases gradually and has comparatively less search time. This reduced search time is due to access of the peers available at proper position. The higher search time in Random Quorum Consensus is due to the higher time required to find the peer through flooding as compare to the structured.

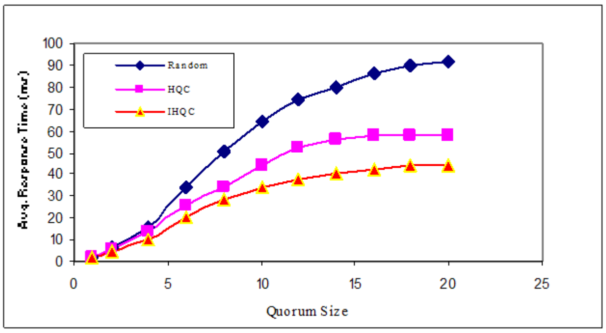

Figure 9. The comparison of average response time received from quorums in the networks

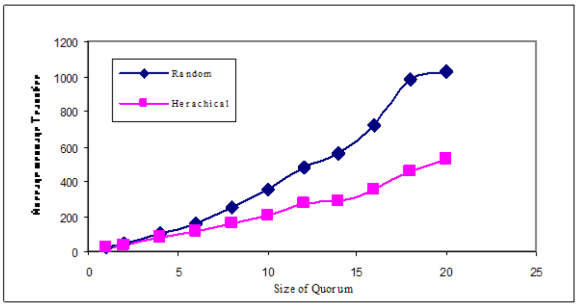

Figure 10. The comparison of average message transfer to maintain the system under quorum size

The network overhead is measured as the average number of messages transferred to update all replicas included in the quorum. It is observed from the simulation, that network overhead in case of Random Quorum Consensus increases rapidly as the size of quorum is increased. It is further observed that the network overhead in case of hierarchical quorum consensus is small as compared to the random quorum.

VI. Conclusions

In this paper an Improved Hierarchical Quorum Consensus Scheme (IHQC) was proposed. The IHQC places replicas in special SOHLS logical structure which self organizes itself after a peer leaves the system. The architecture and mathematical correctness of the proposed scheme is also presented in the paper. It is observed from the simulation results that average messages transfer in the P2P network is minimized through the directional search as compared to the random search. The average response time of IHQC is lower as compared to the HQC and Random Quorum Consensus. This scheme maximizes the degree of intersection set among the read and write quorums.

The probability to access updated data is higher in case of IHQC as compared to existing HQC and Random Quorum Consensus. To get better performance with respect to the probability to access updated data, the replicas may be stored on the peers having availability greater than 70%. This higher probability is due to the fact that IHQC always accesses the peers with the highest session time. IHQC has lesser search time as compared to the Random and HQC. This is due to searching fixed location in the logical structure. This scheme performs better than the other hierarchical schemes. In its best case this scheme access only one peer (from the root) of the SOHLS, which the minimum read cost reported till date with ROWA. This Scheme tolerate up to $n-1$ faults**.** In future we will try to simulate other aspects of the proposed scheme with some other parameters.

References

[1] D. Agrawal and A. E. Abbadi. An Efficient and Fault-Tolerant Solution for Distributed Mutual Exclusion. ACM Transactions on Computer Systems, 1991. 9(1): p. 1–20.

[2] Napster Website[EB/ OL] . http://www. napster.com.

[3] Gnutella Website[EB/ OL] . http://www.gnutella.com.

[4] T. Loukopoulos and I. Ahmad. Static and Adaptive Data Replication Algorithms for Fast Information Access in Large Distributed systems. IEEE International Conference on Distributed Computing Systems, Taipei, Taiwan, 2000. p. 385 – 392.

[5] S. Abdul-Wahid, R. Andonie, J. Lemley, J. Schwing, and J. Widger. Adaptive Distributed Database Replication Through Colonies of Pogo Ants. Parallel and Distributed Processing Symposium, IPDPS 2007. IEEE International, California USA, 2007. p. 358.

[6] J. Holliday, D. Agrawal, and A. E. Abbadi. Partial Database Replication Using Epidemic Communication. 22nd International Conference on Distributed Computing Systems, IEEE Computer Society, Vienna, Austria, 2002. p. 485–493.

[7] T. Loukopoulo and I. Ahmad. Static and Adaptive Distributed Data Replication Using Genetic Algorithms. Journal of Parallel and Distributed Computing, 2004. 64(11): p. 1270–1285.

[8] P. Francis, S. Jamin, V. Paxson, L. Zhang, D. Gryniewicz, and Y. Jin. An Architecture for a Global Internet Host Distance Estimation Service. IEEE INFOCOM '99 Conference, New York, NY, USA, 1999. p. 210-217.

[9] A. Vigneron, L. Gao, M. Golin, G. Italiano and B. Li. An Algorithm for Finding a k-median in a Directed Tree. In Information Processing Letters, 2000. 74(1, 2): p. 81-88.

[10] S. Jamin, C. Jin, T. Kurc, D. Raz and Y. Shavitt. Constrained Mirror Placement on the Internet. IEEE INFOCOM Conference, Alaska, USA, 2001. p. 1369 – 1382.

[11] I. Gashi, P. Popov, and L. Strigini. Fault Tolerance via Diversity for Off-the-Shelf Products: A Study with SQL Database Servers. IEEE Transactions on Dependable and Secure Computing, 2007. 4(4): p. 280–294.

[12] C. Wang, F. Mueller, C. Engelmann, and S. Scott. A Job Pause Service Under Lam/Mpi+Blcr for Transparent Fault Tolerance. IEEE in International Parallel and Distributed Processing Symposium, California USA, 2007. p. 1-10.

[13] Lin Wujuan, Bharadwaj Veeravalli. An Object Replication Algorithm for Real-Time Distributed Databases. Distributed Parallel Databases, MA, USA, 2006. 19: p. 125–146.

[14] Anirban Mondal, Masaru Kitsuregawa. Open Issues for Effective Dynamic Replication in Wide-Area Network Environments. Peer-to-Peer Networking and Applications, 2009. 2(3): p. 230-251.

[15] A. Bonifati, E. Chang, T. Ho, L. V. Lakshmanan, R. Pottinger and Y. Chung. Schema Mapping and Query Translation in Heterogeneous P2P XML Databases. The VLDB Journal, 2010. 19(2): p. 231-256.

[16] Ricardo JimeNez-Peris, Gustavo Alonso, Bettina Kemme, M. Patin O Martinez. Are Quorums an Alternative for Data Replication?. ACM Transactions on Database Systems, 2003. 28(3): p. 257–294.

[17] J. Kangasharju, K.W. Ross, and D. Turner. Optimal Content Replication in P2P Communities. Manuscript, 2002.

[18] Radded Al king, Abdelkader H. Franck M. Query Routing and Processing in Peer-to-Peer Data sharing Systems. International Journals of Database Management Systems (IJDMS), 2010. 2(2): p. 116-139.

[19] Jiafu Hu, Nong Xiao, Yingjie Zhao, and Wei Fu. An Asynchronous Replica Consistency Model in Data Grid. ISPA Workshops, LNCS 3759, Nanjing, China, 2005. p. 475 – 484.

[20] R.H. Thomas. A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases. ACM Transactions on Database Systems, 1979. 4(2): p. 180–209.

[21] A. Sleit, W. Al Mobaideen, S. Al-Areqi, and A. Yahya. A Dynamic Object Fragmentation and Replication Algorithm in Distributed Database Systems. American Journal of Applied Sciences, 2007. 4(8): p. 613-618.

[22] D. Agrawal, A. El. Abbadi. The Generalized Tree Quorum Protocol: An Efficient Approach for Managing Replicated Data. ACM Transactions on Database Systems, 1992. 17(4): p. 689-717.

[23] Hidehisa Takamizawa, Kazuhiro Saji. A Replica Management Protocol in a Binary Balanced Tree Structure-Based P2P Network. Journal of Computers, 2009. 4 (7): p. 631-640.

[24] Saha D, Rangarajan S, Tripathi SK. An Analysis of the Average Message Overhead in Replica Control Protocols. IEEE Transactions on Parallel Distributed Systems, 1996. 7(10): p. 1026–1034.

[25] Ahmad N, Abdalla AN, Sidek RM. Data Replication Using Read-One-Write-All Monitoring Synchronization Transaction System in Distributed Environment. Journal of Computer Science, 2010. 6(10): p. 1066–1069.

[26] P. A. Bernstein, N. Goodman. An Algorithm for Concurrency Control and Recovery in Replicated Distributed Databases. ACM Transactions on Database Systems, 1984. 9(4): p. 596–615.

[27] Jajodia, S., and Mutchler, D. Integrating Static and Dynamic Voting Protocols to Enhance File Availability. Fourth International Conference on Data Engineering (Los Angeles, Feb. 1988). IEEE, New York, 1988. p. 144-153.

[28] R. H. Thomas. A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases. ACM Transactions Database Systems, 1979. 4(2): p. 180–209.

[29] Oprea F, Reiter MK. Minimizing Response Time for Quorum-System Protocols Over Wide-Area Networks. International Conference on Dependable Systems and Networks (DSN), Edinburgh, UK, 2007. p. 409–418.

[30] Sawai Y, Shinohara M, Kanzaki A, Hara T, Nishio S. Consistency Management Among Replicas Using a Quorum System in Ad Hoc Networks. International Conference on Mobile Data Management (MDM), Nara, Japan, 2006. p. 128–132.

[31] Frain I,M'zoughi A, Bahsoun JP. How to Achieve High Throughput with Dynamic Tree-Structured Coterie. International Symposium on Parallel and Distributed Computing (ISPDC), Timisoara, Romania, 2006. p. 82–89.

[32] Osrael J, Froihofer L, Chlaupek N, Goeschka KM. Availability and Performance of the Adaptive Voting Replication. International Conference on Availability, Reliability and Security (ARES), Vienna, Austria, 2007. p. 53–60.

[33] S. Cheung, M. Ammar, and A. Ahamad. The Grid Protocol: A High Performance Scheme for Maintaining Replicated Data. IEEE Sixth International Conference on Data Engineering, Los Angeles, CA , USA, 1990. p. 438–445.

[34] Storm C, Theel O. A General Approach to Analyzing Quorum-Based Heterogeneous Dynamic Data Replication Schemes. 10th International Conference on Distributed Computing and Networking, Hyderabad, India, 2009. p. 349–361.

[35] A. Kumar. Hierarchical Quorum Consensus: A New Algorithm for Managing Replicated Data. IEEE Transactions on Computers, 1991. 40(9): p. 996-1004.

[36] Kevin Henry, Colleen Swanson, Qi Xie and Khuzaima Daudjee. Efficient Hierarchical Quorum in Unstructured Peer-to-Peer Networks. LNCS 5870, 2009. p. 183-200.

[37] Sang-Min Park, Jai-Hoon Kim, Young-Bae Ko, Won-Sik Yoon. Dynamic Data Replication Strategy Based on Internet Hierarchy BHR. Springer-Verlag Heidelberg, 2004. 3033: p. 838-846.

[38] A. Horri, R. Sepahvand, Gh. Dastghaibyfard. A Hierarchical Scheduling and replication strategy. International Journal of Computer Science and Network Security, 2008. 8: p. 30-35.

[39] Tang, M., Lee, B., Tang, X., and Yeo, C. The Impact of Data Replication on Job Scheduling Performance in the Data Grid. Future Generation Computing System, 2006. 22(3).

[40] Kavitha, R., A. Iamnitchi, and I. Foster. Improving Data Availability through Dynamic Model Driven Replication in Large Peer-to- Peer Communities. Global and Peer-to-Peer Computing on Large Scale Distributed Systems Workshop, Berlin, Germany, 2002.

[41] Ranganathan and I. Foster. Design and evaluation of Replication Strategies for a High Performance Data Grid in Computing and High Energy and Nuclear Physics. International Conference on Computing In High Energy And Nuclear Physics (CHEP'01), Beijing, China, 2001.

[42] Shashi Bhushan, R. B. Patel, M. Dave. Reducing Network Overhead with Common Junction Methodology. International Journal of Mobile Computing and Multimedia Communications, 2011. 3(3): p. 51-61.

Shashi Bhushan (1974) received his B. Tech. (Computer Technology) degree from Nagpur University, Nagpur INDIA in 1998. He received his M. Tech. (Computer Science and Engineering) degree from JRN Rajasthan Vidyapeeth, Rajasthan in 2005 and presently pursuing PhD in the field of Peer-to-Peer Networks from National Institute of Technology, Kurukshetra, INDIA. His fields of interest are Peer-to-Peer Networks, Sensor Networks and Mobile Computing.

Dr. Mayank Dave (1967) received PhD degree from IIT Roorkee, INDIA in 2002, M. Tech degree in Computer Engineering from IIT Roorkee, INDIA in 1991. He is presently working as an associate professor in Computer Engineering Department at NIT Kurukshetra, INDIA. He has more than 19 years experience in academic and administrative affairs. He had published more than 100 research papers in various International / National Journals and International / National Conferences and guided several PhDs, M. Tech and B. Tech students. His research interests include Peer-to-Peer Computing, Pervasive Computing, and Sensor Networks, QoS in Mobile Networks.

Dr. R. B. Patel (1966) received PhD from IIT Roorkee in Computer Science & Engineering, PDF from Highest Institute of Education, Science & Technology (HIEST), Athens, Greece, MS (Software Systems) from BITS Pilani and B. E. in Computer Engineering from M. M. M. Engineering College, Gorakhpur, UP. Dr. Patel is in teaching and Research & Development since 1991. He has supervised 50 M. Tech, 10 M. Phil and 10 PhD Thesis. He has published more than 130 research papers in International/National Journals and Refereed International Conferences. System and Method for Reliable and Flexible Mobile Agent Computing and System, Method, & Computer Program for Routing Data in Wireless Sensor Network are two patents in the credit of Dr. Patel. His research interests are in Mobile & Distributed Computing, Mobile Agent Security and Fault Tolerance, development infrastructure for mobile & Peer-to-Peer computing, Device and Computation Management, Cluster Computing, Sensor Networks.