IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 08, Issue 10 (October. 2018), ||V (II) || PP 64-75

IAFCM: Integrated and Adaptive File Consistency Maintenance in Peer-to-Peer Network

Bhuvaneswari R[1*], Ravi T N[2]
[1]*Research Scholar, Manonmaniam Sundaranar University, Tirunelveli*
[1]*Assistant Professor, Department of Computer Science, Periyar Arts College, Cuddalore, Tamil Nadu, India*
[2]*Assistant Professor, Department of Computer Science, Periyar EVR College (Autonomous), Trichy, Tamil Nadu, India*
Corresponding Author: Bhuvaneswari R

Abstract:- The Peer-to-Peer (P2P) based system suffers from a crucial problem of difficulty in the update transmission, as the peers in the network cannot obtain a comprehensive view of the replica locations. This paper presents an Integrated and Adaptive File Consistency Maintenance (IAFCM) algorithm to achieve high efficiency in the data replication and replica consistency maintenance at a low cost. Each replica server in the network chooses the replication and update sampling by dynamically adjusting to the file query and file update rates. This research work presents a hypothetical update that quickly transmits an update to all replicas in a pure P2P way. Each server determines the directions of replicas with maximum probability based on the local state of the server used for relocating the replica. Then, the update messages are transmitted in that direction. Experimental analysis demonstrates that the hypothetical update mechanism transmits an update to all replicas at a faster rate than the existing P2P-based approaches.

Keywords:- File Query Rate, File Update Rate, Hypothetical Update, Peer-to-Peer Network, Replica Consistency Maintenance

---

Date of Submission: 08-10-2018 Date of acceptance: 23-10-2018

I. INTRODUCTION

Replication is an effective mechanism to reduce the overall file transfer time and bandwidth consumption in the P2P network. The data access performance is improved by positioning the accessed data at the appropriate locations. P2P-based CDN is scalable due to the increase in the number of servers and client devices. The decentralized architecture is highly robust against the server failure. Due to the homogeneous functionality and independent operating capacity of each server, the Content Distribution System (CDS) keeps working efficiently despite of the server failures. P2P-based CDNs combine the commodity peers to provide a scalable service. While P2P-based architecture provides attractive features, it is highly difficult to propagate the updates to all replicas distributed over the network, as the peer does not maintain every location of the replica contents. The conventional approaches to the update propagation are Globule [1], structured overlay approaches [2-4], gossip protocols [5-8] and random walks [9-14].

In Globule, a centralized server holds a map of the current replica locations and propagates the update to all replicas according to the map. However, this approach reduces the fault resiliency of the P2P-based systems, as the centralized server may become a bottleneck and a single point of failure. The structured overlay approaches construct the trees or Distributed Hash Tables (DHTs) to determine the location of the replica without global information. But, it requires high maintenance overhead, as the structure should be updated every time with the change in the replica locations. Gossip protocols and random walks are appropriate for the large-scale P2P-based CDNs to propagate the update messages without requiring any structures or global information.

The gossip-based approaches flood the update messages on the network. Each server sends the received update message to all the servers that are directly connected to it. The server adjusts the probability to transmit an update based on the number of duplicate messages received. But, this approach generates many redundant messages that consume bandwidth. Random walk randomly identifies a next hop based on the Metropolis-Hastings algorithm. It can transmit an update at a lesser bandwidth consumption rate than the gossip protocols. But, it takes long time to converge to the desired server distribution [15]. To utilize the gossip protocols or random walks, the underlying overlap should be designed carefully as the overlay topology affects the relevance and coverage of the update propagation.

A hypothetical update is proposed for the quick propagation of the update to the most frequently accessed replicas. Each server determines the transmission directions of the replicas with high probability. The update messages are relayed hypothetically in that direction. The hypothetical update utilizes a small record of the query directions to determine the location direction of the replicas. It achieves quick and complete propagation of the updates, by transmitting the update message towards the probable replica locations. Thus, all the replicas are updated in a timely manner.The current standard of care for screening for lung cancer is computerized tomography (CT), as it has the ability to pick up very small pulmonary nodules. MRI has superior contrast resolution than CT, and performs well in several body parts including brain, abdomen and soft tissue. However MRI of the lungs is limited due to the poor spatial resolution and breathing artifacts. Moreover, MRI is not sensitive for the detection of small pulmonary nodules (<1 cm), and ground glass nodules (frequently seen in lung adenocarcinoma).

Data replication is highly necessary to maintain the uniformity between the data files and its replicas. IAFCM algorithm is applied to maintain the consistency of the data file replicas in the P2P network at a significant low cost. Each node decides to generate or delete a data replica and to sample for the update in a decentralized way based on the file query and file update rates. It replicates the highly queried files and polls at a high frequency rate for the frequently updated and queried files. By avoiding the unnecessary file replications and updates, the proposed IAFCM dynamically adapts to the file query and file update rates that vary with respect to time. Thus, the replica utilization, file query efficiency, replica consistency and reliability are improved. The hypothetical update is applied to ExaPeer [16] and evaluated on two synthetic networks. The simulation results show that the hypothetical update quickly transmits the updates to all replicas with less number of messages than the gossip protocols and random walks. The hypothetical update specially transmits the updates to the replicas accessed by the clients.

The organization of the paper is defined as follows: Section II presents an overview of the existing approaches to transmit the updates in the P2P-based CDN. The proposed work including the IAFCM algorithm, hypothetical update and ExaPeer model are explained in Section III. Section IV presents the simulation results of the proposed IAFCM algorithm using the King and PlanetLab datasets. Section V concludes the proposed IAFCM approach.

II. RELATED WORKS

In the P2P-based CDN, it is highly difficult to efficiently transmit the updates due to the lack of global information about the replica locations. Table 1 summarizes the classification of the existing approaches to propagate the updates in the P2P-based CDN. The centralized server approach assigns a dedicated server to preserve a map of current replica locations and transmit the updates. The master server maintains the Internet Protocol (IP) address of the server with the replicas. This enables the server to easily determine the replica locations, but it is difficult to cope with the network development. Though, the centralized server approach allows the CDNs to directly send an update to all replicas, it suffers from the scalability and single point of failure issues. The structured overlay approach makes the P2P-based CDNs more complex and adds extra development and maintenance costs for the creation of DHTs and trees.

The gossip-based approach overflows the update messages on the network. Each server sends the received update message to all the servers that are directly connected. The gossip-based approaches are simple and affordable, as they do not require a structure to maintain the state of the replica locations. It makes the update transmission mechanism to be scalable and resilient to the failures. But, these approaches generate redundant messages that consume bandwidth. Random walks can transmit an update by consuming less bandwidth than the gossip-based protocols. Unlike the gossip-based protocols, random walks do not require the structure to maintain the state of the replica locations. Random walk approach consumes more time to converge to the preferred server distribution.

Meng and Zhang [17] proposed a new replica consistency maintenance strategy for unstructured P2P networks based on the ant colony model. The actions and states of the ants are defined initially. The ants are used to update the file replicas in the forward walking direction and the pheromones of the files are updated during the returning process of ants. The simulation results have shown that the replica consistency maintenance strategy reduced the update cost and the impacts caused by the churn during the replica consistency maintenance. But, there is a need to optimize the pheromone computation for guiding the ants to the exact nodes.

Based on the swarm intelligence, Shen et al. [18] presented a mechanism for file replication and consistency maintenance in the structured P2P file sharing system. An update message is transmitted between the proximate nodes in a tree fashion from the top to the bottom. Swarm can improve the hit ratio and reduce the querying latency, number of replicas, consistency maintenance overhead when compared to the existing consistency maintenance approaches. However, this mechanism does not exploit the file popularity and update rate for efficient consistency maintenance.

Nguyen et al. [19] developed a new approach for the balanced data consistency maintenance in the P2P systems, where the nodes are organized into a tree to disseminate update processes. But, real-time maintenance of the tree depth is difficult and unnecessary, when the tree nodes are joining and leaving frequently. Yoichi and Sugawara [20] proposed a method for practically maintaining the consistency among the replicas of content items when a content item is updated. In this method, the peers that have the replicas of frequently accessed content items are previously selected. When an update occurs, the updating data is propagated to the peers on a priority basis.

The objective of the hypothetical update approach is to overcome the shortages of the existing approaches. The hypothetical update enables fast update transmission with minimum bandwidth consumption than the gossip protocols and random walks [15].

Table 1 Comparison Analysis of Existing Approaches for Update Transmission in P2P Based CDNS

Approaches	**Scalability**	**Fault resilience**	**Maintenance cost**	**Bandwidth consumption**	**Transmission time**
Centralized server	X	X	X	✓	✓
Structured overlay	✓	Moderate	X	✓	✓
Gossip protocols	✓	✓	✓	X	✓
Random walk	✓	✓	✓	✓	X
Hypothetical update	✓	✓	✓	✓	✓

III. PROPOSED WORK

3.1 P2P model

A P2P network comprises a set of peers that can be either a computing resource or storage resource. The computing resource allows the users to submit and execute the jobs and storage resource allows the users to store the data files. Each data file is produced by the top level site. The storage capacity of each peer is limited. The data files in the peer are divided as $D = \{D_1, D_2, \dots, D_p\}$. Each part of the data file is placed into each replica server [21].

In the proposed work, an IAFCM algorithm is applied for maintaining the consistency between the data. This algorithm involves sample reduction and sampling frequency determination. A linear increase/multiplicative decrease algorithm is used in the sampling frequency determination. In this algorithm, the Time-To-Refresh (TTR) value with each replica is set, if the file is changed frequently in the main server. In the sampling reduction method, each replica node considers the file query rate and update rate based on the file query rate. The TTR value is changed when a file changes frequently. Adaptive sampling is applied for maintaining the file consistency and providing file replication.

3.1.1 Sampling frequency determination

Let us assume the maximum update rate of a file be $1/\Delta t$. It means the file updates every Δt seconds in the highest update frequency. In this case, a replica node of the file can ensure that a replica is never outdated by more than Δt seconds by sampling the owner every Δt seconds.

Sampling refers to the condition where the timeliness of the peer is checked frequently. The system returns to a different task, if the peer is not ready. Busy waiting can cause wastage of the Central Processing Unit (CPU) cycles, when there is no communication. Thus, interrupt-driven Input/Output (I/O) is not efficient as an alternate to the sampling.

In determining the sampling frequency, the replica node can adapt its sampling frequency in response to the variations. The TTR value denotes the subsequent time instant to a replica node should sample the server to maintain the updated replica.

The TTR value is increased linearly, when each sampling data does not change.

$$TTR = TTR_{old} + \alpha \quad (1)$$

The data file has updated the previous sample, the TTR value is decreased by using a multiplicative factor.

$$TTR = TTR_{old} / \beta \quad (2)$$

The algorithm considers two input parameters: TTR_{max} and TTR_{min} that represents upper and lower limits on the TTR values. Values that fall outside these limits are set to

$$TTR = max\left(TTR_{min}, min(TTR_{max}, TTR)\right) \quad (3)$$

These limits ensure that the TTR is neither too large nor too small. TTR_{min} is set to Δt, since it is the minimum interval between the samples required to maintain the consistency assurance. The algorithm begins by initializing

$$TTR = TTR_{min} = \Delta t \quad (4)$$

3.2.2 Sampling Reduction

In the sample reduction, the file change rate and file query rate are the main factors to consider during the replica consistency maintenance. When a file changes frequently, if the replica node does not receive queries for the file during a time period, it is an overhead to sample the server for validation during the time period. When the file change rate is higher than the file query rate, there is no need to update the replica at the rate of file change event.

$if\ TTR \le T_{query}\ \ then$

$TTR_{sample}\ = T_{query}$

The file is queried at a higher rate than the file change rate. Then, the file should be updated based on the TTR value.

$if\ TTR > T_{query}\ \ then$

$TTR_{sample}\ = TTR$

IAFCM algorithm

If there is a query for the file 'f' **then**

Include a sampling request into the query

Else

Send a sampling request

If a validation reply is obtained from the file owner **then**

{

If the file is valid **then**

$[\alpha = 0.5, \beta = 1.5]$

$TTR = TTR_{old} + \alpha$

If the file is damaged **then**

$TTR = TTR_{old} / \beta$

Update file replica

}

If $TTR > TTR_{max}\ or\ TTR > TTR_{min}$ **then**

$TTR = max\left(TTR_{min}, min(TTR_{max}, TTR)\right)$

If $TTR \le T_{query}$ **then**

$TTR_{poll}\ = T_{query}$

Else

$TTR_{poll}\ = TTR$

}

The IAFCM algorithm is highly effective in minimizing the total number of data file replicas, file query delay and replica file consistency maintenance overhead. The performance of the proposed IAFCM mechanism is better than the existing algorithms in terms of low cost and file update rate [22], as the update replicas are received by the users.

3.2 ExaPeer

ExaPeer [15] is a P2P-based Content Addressable Network (CAN) [23] that dynamically repositions the replicas based on the demand fluctuations. ExaPeer uses the coordinates computed using the Global Network Positioning (GNP) devices [24] to construct a topologically-aware overlay network. GNP models the P2P network as a d-dimensional space and allocates d-dimensional coordinate to each server, so that the Euclidean distance between any pair of coordinates approximates the Round Trip Time. CAN is a DHT that assigns a virtual d-dimensional coordinate to each server and divides the d-dimensional space into the zones. One server is allocated in each zone, so that its coordinate is contained in the zone. CAN maps a key corresponding to the content to a specific point 'P' in the coordinate space. The CAN stores the content in the server responsible for the zone within the specific point is located.

A client obtains content from ExaPeer based on the CAN protocol. A query for the content is transmitted to the origin server through the intermediate servers. The origin server maintains a zone comprising the coordinate corresponding to the content. If a server on the path hosts a replica of the content, the replica is provided to the client instead of an origin server.

3.2.1 Replica Placement

ExaPeer dynamically relocates the replicas based on the demand fluctuations in a P2P fashion. The ExaPeer transmits a query so that the query becomes closer to the final destination. A server transmits a query from one direction to another direction where there is a high-demand area in the arrival direction of the query. A server provides the replica of the content, if there are many queries for content from one direction. In the ExaPeer architecture, a server monitors the degree of Access Path Convergence (APC) degree that indicates the number of shifted queries to determine hosting of a replica. ExaPeer applies a replica on the server Q to deal with the demand of clients X, Y and Z.

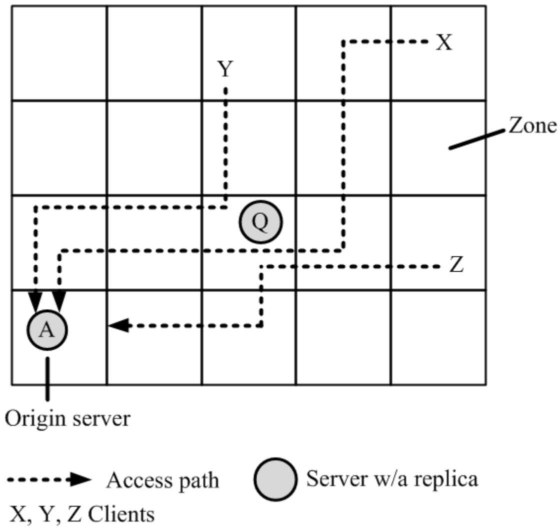

Fig.1 ExaPeer Structure

3.3 Hypothetical Update

Hypothetical update increases the update propagation of the replicas by determining the location of replica with the local state of the server used for replica relocation. Fig.2 shows the overview of the hypothetical update. Once a server determines the location direction of replicas, a hypothetical message is transmitted in that direction. A hypothetical message is transmitted in the above direction and an update is advertised to the replica on the message path. To efficiently advertise an update, the hypothetical update generates gossip-based update messages called as complementary messages with minimum Time-To-Live (TTL), after a hypothetical message reaches a replica site. The complementary messages efficiently advertise an update to the replica in a few hops.

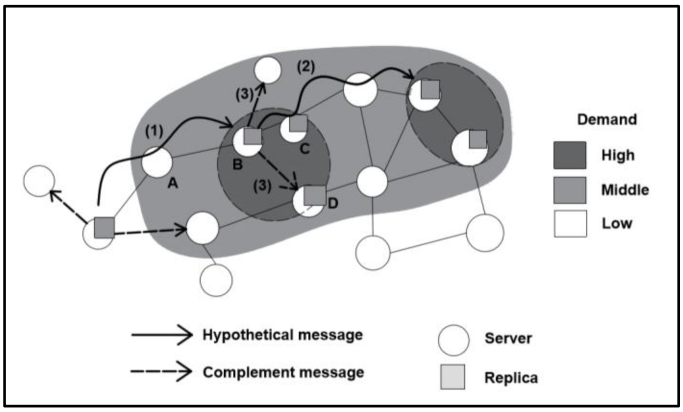

Fig.2 Overview of the hypothetical update

Initially, the origin server sends a hypothetical message to the server 'A' and complementary messages to its neighbors. Then, a hypothetical message reaches the server 'B' hosting a replica. B also attempts to determine the locations of replica and sends another hypothetical message to the server 'C'. Simultaneously, the server 'B' sends out the complementary messages to its neighbors. One of the complementary messages reaches the server 'D' hosting a replica.

3.3.1 Hypothetical Message

The hypothetical update initially computes a primitive vector to determine the transmission direction of a hypothetical message. A primitive vector represents the direction of replica locations in a rough way. A primitive vector is a simple vector sum computed by a server using the local state that is the APC degree in

ExaPeer of the neighbor servers. In the popular P2P-based CDNs, a client query identifies different access paths even if the same client requests the same content. Due to the flexibility of the routing protocol, transmission of a query to a failed server or a hot spot server is avoided. Hence, some queries reach the servers that are located far away from the clients. Hypothetical update acquires a cue from these queries to compute a primitive vector. It roughly determines that the replicas are positioned in the transmission direction of the query.

In ExaPeer, a primitive vector 'P_i' of server 'i' is calculated using the function

$$P_i = \sum_j \alpha_j x_{ij} \ such\ that \ |\alpha_j x_{ij}| = d_j \quad (5)$$

Where d_j represents the APC degree of the neighbor server 'j' and x_{ij} denotes a relative vector from the server 'i' and server 'j'. The hypothetical update computes the target vector based on the primitive vector. The target vector represents the replica location direction based on the primitive vector. Hence, it is more accurate than the primitive vector. Hypothetical update estimates an Identity (ID) so that the hypothetical message is transmitted in the direction of the target vector on the CDN. The destination of a hypothetical message is the ID.

There are two key ideas to compute the target vector. Initially, a server ignores the neighbor servers whose relative vector is the opposite direction of the primitive vector. This approximately adjusts the target vector in the location direction of multiple replicas. Next, a server applies a convert function that results in reversing the order of input values to the values that denote the local state of the neighbors. In the P2P-based CDNs, all the client queries in an area are assisted by the replicas located proximate to the clients. Few remaining queries take a round-about route to avoid the replicas for some reasons. It is inferred that the actual direction for positioning the replicas between the vectors that represent the transmission directions of remaining queries.

In ExaPeer, the target vector 'T' of the server 'i' is computed by the function

$$T_i = \sum_{j, P_i.x_{ij} > 0} \beta_j x_{ij}, \ such\ that \ |\beta_j x_{ij}| = \left(max(d_k) + min(d_k)\right) - d_j \, (k = 0,1, \ldots, N) \quad (6)$$

Where 'N' represents the number of neighbors. The basic mechanism of a hypothetical message precisely computes the location direction of the replicas. But, a server computes an inappropriate target vector.

3.3.2 Solution for Wrong Backtracking

A hypothetical message goes back to the replicas on which a hypothetical message has already passed. The server 'B' computes the target vector in the direction of server 'A' and sets a new destination of a hypothetical message based on the target vector. Then, the hypothetical message backtracks in the direction of the updated replicas. The backtracking algorithm enables to go back when the destination node is not reachable from the current node or when the path is intersected with the previously selected paths. The backtracking improves the routing efficiency in the structured P2P network.

3.3.3 Solution for Infinite Looping

When a hypothetical message enters an area with multiple replicas, there is a possibility of forming infinite loops between the replicas. This causes degradation in the effectiveness of the hypothetical update. To solve this issue, a server transmits a received hypothetical message without any modification even if the message reaches a replica. When a server with a replica receives a hypothetical message, it sends another hypothetical message preceding the target vector calculated by the received server. Along with the transmission of a new hypothetical message to the server 'C', the server 'B' transmits the hypothetical message sent by the origin server to the server 'D'. Thus, the hypothetical message can reach the replicas within another area. This looping solution increases the number of hypothetical message, there is a significant improvement in the coverage of the update propagation to the replica.

3.3.4 Complementary message

A hypothetical message quickly announces an update to the replicas located remote from an origin server. But, it is not good at advertising an update to all neighboring replicas. A complementary message enables the hypothetical update to transmit the update to the remote replicas and proximate replicas. A complementary message presents an update to all neighboring replicas. In the structured P2P-based CDN, an origin server receives the client queries from all directions on the logical overlay space. Hence, it is difficult for an origin server to send a hypothetical message in the direction with high accuracy. A complementary message can reach the replicas unsupported by a hypothetical message from an origin server. Furthermore, the servers received the complementary message sends out accurate hypothetical messages. Because they receive the client queries in the restricted directions due to the query routing protocol of the P2P-based CDN.

To limit the bandwidth consumption, a hypothetical update allocates a lower distribution probability to the complementary message. Also, the hypothetical update reduces the TTL of a complementary message. As all the complementary messages advertise an update located proximate to the replicas, while the gossip protocols should propagate the messages to all participants.

IV. RESULTS AND DISCUSSION

This section presents the simulation analysis of the proposed IAFCM algorithm using the Java Netbeans IDE tool. To confirm the behavior of the hypothetical update in the real-time environment, two datasets such as King dataset [25] and PlanetLab dataset [26]. The king dataset comprises latency measurement information of a set of 2048 DNS servers. PlanetLab comprises latency measurement from 105 PlanetLab nodes.

4.1 Performance Metrics

- The length of the time for update transmission shows the efficiency of update transmission.
- Success rate of the update transmission is the ratio of replicas that can receive an update message.
- The coverage of client queries is the ratio of client queries that reach an updated replica. It confirms whether the proposed mechanism transmits an update to the popular queries. More number of clients can obtain the updated content, if the coverage of the client queries is high.
- The number of transmitted messages is the total number of messages that are transmitted to the destination server per second in the network. The overhead of the update transmission is minimum, if a few number of update messages is transmitted.

4.2 Simulation Results

Fig.3(a) and Fig.3(b) show the time for update transmission for the King dataset and PlanetLab dataset. The X-axis shows the elapsed time from the initiation of update to the arrival of the update message to each server. Y-axis shows the ratio of replicas that received an update message. Hypothetical update completes an update transmission faster than the simple gossip-based protocol without TTL by 62% and the random walk by 89%. Hypothetical update enables quick update transmission in an early stage, when compared with the gossip-based protocols without TTL. The hypothetical update propagates in the direction of replica locations correctly. As the gossip-based protocol and random walk introduce more messages than the hypothetical update, they transmit an update at a faster rate than the hypothetical update. The gossip-based protocol and random walk take longer time when there is an increase in the number of servers. The results of the random walk vary extremely among the gossip-based protocols and random walk. Hypothetical update changes but it enables faster update transmission than the gossip-based protocols and random walk even in the worst scenario. There is a variation in the hypothetical update due to the flexibility of the routing protocol of a client query.

Fig.3 shows that the hypothetical update enables update transmission to all replicas even with the complementary messages with low TTL. Table 1 shows the time to update transmission for the King dataset and Table 2 shows the time to update transmission for the PlanetLab dataset. Because the hypothetical update uses the gossip protocol to transmit an update to the neighboring replicas. The hypothetical update transmits an update to all replicas, while the gossip-based protocol propagates an update of the replicas with the King dataset.

Table 1 Time to Update Transmission for the King Dataset

Elapsed time (sec)	Speculative update	Gossip (TTL=3)	Gossip w/o TTL	Random walk	IAFCM
0	0	0	0	0	0
5	0.14	0.02	0.06	0.1	0.16
10	0.14	0.02	0.06	0.1	0.16
15	0.14	0.08	0.08	0.1	0.16
20	1	0.14	0.7	0.14	1.2
25	1	0.4	1	0.26	1.2
30	1	0.5	1	0.3	1.2
50	1	0.52	1	0.52	1.2
100	1	0.52	1	1	1.2

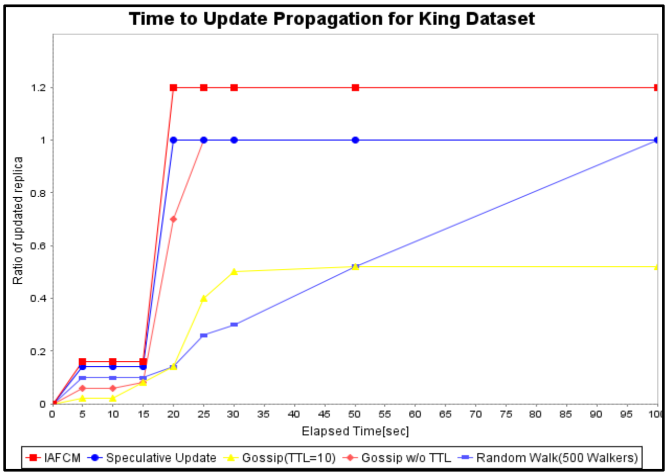

Fig.3(a) Time to update transmission for the King dataset

Table 2 Time to Update Transmission for the PlanetLab Dataset

Elapsed time (sec)	**Speculative update**	**Gossip (TTL=3)**	**Gossip w/o TTL**	**Random walk**	**IAFCM**
2	0.38	0.2	0.2	0.04	0.48
4	0.54	0.44	0.48	0.1	0.64
6	1	0.78	1	0.2	2
8	1	0.9	1	0.28	2
10	1	0.9	1	0.3	2
50	1	0.9	1	1	2

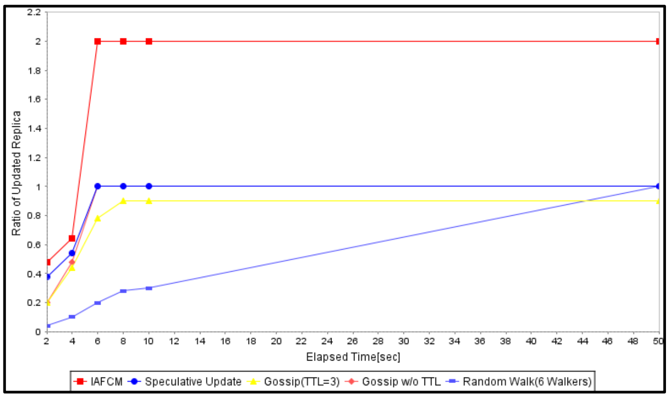

Fig.3(b) Time to Update Transmission for the PlanetLab Dataset

Fig.4 depicts the elapsed time until a client query reaches one of the updated replicas. Table 3 shows the length of time to transmit an update to a replica that receives client requests. The hypothetical update enables rapid update transmission to the popular replicas than the gossip-based protocols and random walk. The hypothetical update requires minimum time to access an updated replica than the gossip-based protocol and random walk by 42% and 82% respectively. Hypothetical update reduces the time of the clients to access an updated replica by 75% shorter than the random walk. Hypothetical update propagates an update 5% faster to the replicas that receive more client accesses. The result of the simple gossip-based protocol without TTL is similar to the hypothetical update. As there is a minimum number of servers in the PlanetLab dataset, the gossip-based protocol can reach all servers within a few hops.

Table 3 The Length of Time to Transmit an Update to a Replica that Receives Client Requests for the King dataset

Elapsed time (sec)	**Speculative update**	**Gossip (TTL=3)**	**Gossip w/o TTL**	**Random walk**	**IAFCM**
0	0	0	0	0	0

20	0.68	0.06	0.12	0.02	0.78
40	0.16	0.38	0.76	0.14	0.86
60	0.8	0.48	0.8	0.3	0.9
80	0.9	0.5	0.9	0.38	1
100	0.92	0.54	0.92	0.48	1.1
300	1	0.66	1	0.72	1.2
500	1	0.68	1	1	1.2

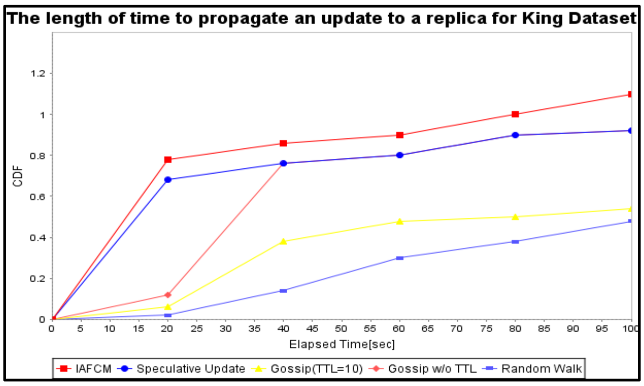

Fig.4 The Length of Time to Transmit an Update to a Replica that Receives Client Requests for the King dataset

Fig.5(a) and Fig.5(b) present the number of transmitted message for the King dataset and PlanetLab dataset. Table 4 and Table 5 show the number of transmitted messages for the King dataset and PlanetLab dataset. To evaluate the overhead of the hypothetical update, the number of transmitted messages per second on the network. With the hypothetical update, the servers transmit the least number of messages during update transmission. For the gossip-based protocol, the messages are rapidly increased as each server transmits the update messages to all the neighbors. When most messages are transmitted in the simulation, the nodes transmit minimum messages when compared with the gossip-based protocols without TTL by 80%.

In the case of random walk, the number of transmitted messages increases in the early stage of the simulation as the random walker sends the messages one by one due to the simulation environment. The number of transmitted messages with the random walks is constant. Though the random walk limits the bandwidth consumption, the hypothetical update requires minimum messages than the random walk.

Table 4 Number of Transmitted Messages for the King dataset

Elapsed time (sec)	**Speculative update**	**Gossip (TTL=3)**	**Gossip w/o TTL**	**Random walk**	**IAFCM**
0	0	0	0	0	0
10	250	800	800	800	150
20	100	1600	1000	1600	90
30	0	200	100	2250	0
40	0	100	100	2750	0
50	0	100	100	2500	0

Table 5 Number of Transmitted Messages for the PlanetLab dataset

Elapsed time (sec)	**Speculative update**	**Gossip (TTL=3)**	**Gossip w/o TTL**	**Random walk**	**IAFCM**
0	0	0	0	0	0
2	50	50	50	80	30
4	60	65	80	70	40
6	75	85	100	75	55
8	90	90	110	80	70
10	70	70	90	70	50
12	50	50	60	72	30
14	10	20	15	74	5

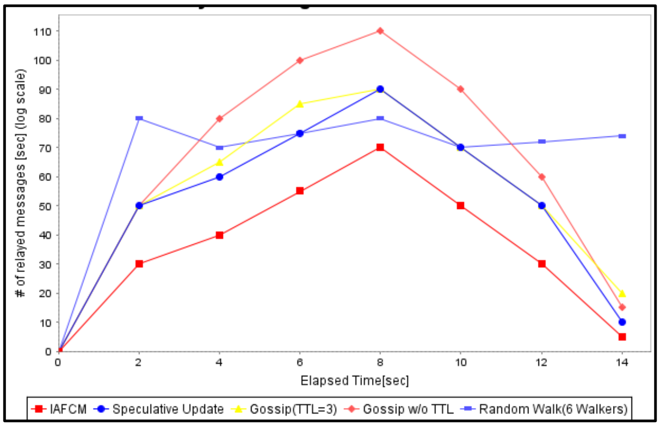

Fig.5(a) Number of transmitted messages for the King dataset

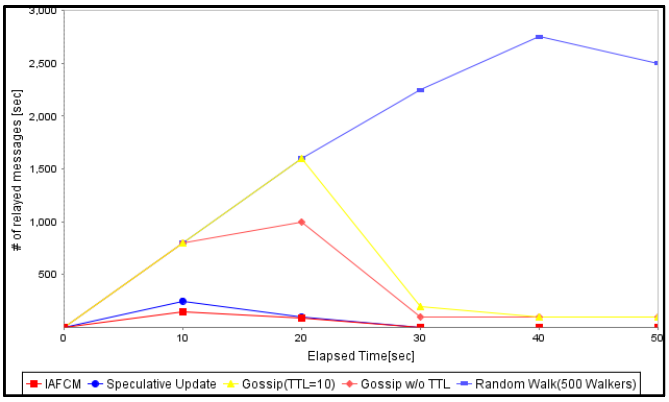

Fig.5(b) Number of transmitted messages for the PlanetLab dataset

Once a hypothetical message reaches the replica, the hypothetical update makes a server with a replica start transmitting an update with complementary message. The complementary message affects the increase of transmitted messages, but it helps the P2P-based CDNs to transmit an update quickly. On the other hand, the number of transmitted messages of the random walk converges to be constant and does not increase because only an origin server creates the update messages. However, the random walk takes longer time to complete transmission of update. In the random walk approaches, the servers relay most messages. The number of transmitted messages with the hypothetical update is lower of about 50%, 75%, 66.67% and 93.24% than the existing speculative update, gossip-based protocols and random walk, during most of the update transmission time. Thus, the proposed IAFCM algorithm requires minimum number of transmitted messages than the gossip-based protocols and random walk approaches.

V. CONCLUSION

In the P2P network, it is highly difficult to transmit the updates to all the replicas in the network, as the peer does not maintain the location of the replica contents. During data replication, there is a need to maintain the consistency between the files and replicas. To mitigate these issues, the proposed IAFCM algorithm achieves quick and completion transmission of all replicas in a timely manner.

The proposed IAFCM algorithm reduces the consistency maintenance overhead, unnecessary file replicas and file updates in the replica nodes. By leveraging the characteristics of P2P-based CDNs, hypothetical update determines the direction in which many replicas are positioned and rapidly propagates an update in that direction, even with fewer update messages. The hypothetical update quickly transmits an update to all replicas and reduces the number of relayed messages than the gossip-based protocols and random walk approaches. The IAFCM algorithm achieves high replica consistence maintenance efficiency at a lower cost in a replica node based on the file query rate and update rates. To leverage the structure of the P2P-based CDNs, the hypothetical update enables rapid update transmission to the popular replicas with minimum bandwidth consumption than the gossip-based protocols and random walk. In future, the hypothetical update is to be implemented with other networks.

REFERENCES

[1]. G. Pierre and M. Van Steen, "Globule: a collaborative content delivery network," *IEEE Communications Magazine,* vol. 44, pp. 127-133, 2006.

[2]. J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels*, et al.*, "Oceanstore: An architecture for global-scale persistent storage," *ACM SIGOPS Operating Systems Review,* vol. 34, pp. 190-201, 2000.

[3]. Z. Li, G. Xie, and Z. Li, "Locality-aware consistency maintenance for heterogeneous P2P systems," in *2007 IEEE International Parallel and Distributed Processing Symposium*, 2007, p. 22.

[4]. L. Ramaswamy, L. Liu, and A. Iyengar, "Cache clouds: Cooperative caching of dynamic documents in edge networks," in *Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference on*, 2005, pp. 229-238.

[5]. M. Deshpande, B. Xing, I. Lazardis, B. Hore, N. Venkatasubramanian, and S. Mehrotra, "Crew: A gossip-based flash-dissemination system," in *Distributed Computing Systems, 2006. ICDCS 2006. 26th IEEE International Conference on*, 2006, pp. 45-45.

[6]. P. T. Eugster, R. Guerraoui, S. B. Handurukande, P. Kouznetsov, and A.-M. Kermarrec, "Lightweight probabilistic broadcast," *ACM Transactions on Computer Systems (TOCS),* vol. 21, pp. 341-374, 2003.

[7]. H. C. Li, A. Clement, E. L. Wong, J. Napper, I. Roy, L. Alvisi*, et al.*, "BAR gossip," in *Proceedings of the 7th symposium on Operating systems design and implementation*, 2006, pp. 191-204.

[8]. R. Van Renesse, K. P. Birman, and W. Vogels, "Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining," *ACM transactions on computer systems (TOCS),* vol. 21, pp. 164-206, 2003.

[9]. A. R. Bharambe, M. Agrawal, and S. Seshan, "Mercury: supporting scalable multi-attribute range queries," in *ACM SIGCOMM computer communication review*, 2004, pp. 353-366.

[10]. B. F. Cooper, "Quickly routing searches without having to move content," in *International Workshop on Peer-to-Peer Systems*, 2005, pp. 163-172.

[11]. C. Gkantsidis, M. Mihail, and A. Saberi, "Random walks in peer-to-peer networks," in *IEEE INFOCOM*, 2004, pp. 120-130.

[12]. D. Loguinov, J. Casas, and X. Wang, "Graph-theoretic analysis of structured peer-to-peer systems: routing distances and fault resilience," *IEEE/ACM Transactions on Networking (TON),* vol. 13, pp. 1107-1120, 2005.

[13]. Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proceedings of the 16th international conference on Supercomputing*, 2002, pp. 84-95.

[14]. D. Tsoumakos and N. Roussopoulos, "Adaptive probabilistic search (aps) for peer-to-peer networks," 2003.

[15]. A. Hayakawa, M. Asahara, K. Kono, and T. Kojima, "A Strategy for Efficient update propagation on peer-to-peer Based Content Distribution Networks," in *IPSJ Online Transactions*, vol. 3, pp. 110-124, 2010.

[16]. M. Asahara, A. Shimada, H. Yamada, and K. Kono, "Finding candidate spots for replica servers based on demand fluctuation," in *Parallel and Distributed Systems, 2007 International Conference on*, 2007, pp. 1-10.

[17]. X. Meng and C. Zhang, "An ant colony model based replica consistency maintenance strategy in unstructured P2P networks," *Computer Networks,* vol. 62, pp. 1-11, 2014.

[18]. H. Shen, G. Liu, and H. Chandler, "Swarm Intelligence Based File Replication and Consistency Maintenance in Structured P2P File Sharing Systems," *IEEE Trans. Computers,* vol. 64, pp. 2953-2967, 2015.

[19]. H. H. C. Nguyen, H. M. Nguyen, and T. S. Doan, "The Method of Maintaining Data Consistency in Allocating Resources for the P2P Network Model," in *Context-Aware Systems and Applications, and Nature of Computation and Communication*, ed: Springer, 2017, pp. 155-165.

[20]. R. Yoichi and S. Sugawara, "Consistency preservation of replicas based on access frequency for content sharing in hybrid peer-to-peer networks," *International Journal of Space-Based and Situated Computing,* vol. 7, pp. 197-206, 2017.

[21]. M. Suguna and J. B. ME, "Maintaining Consistency during Data Replication in Grid Environment."

[22]. H. Shen, "IRM: Integrated file replication and consistency maintenance in P2P systems," *IEEE transactions on Parallel and Distributed systems,* vol. 21, pp. 100-113, 2010.

[23]. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, *A scalable content-addressable network* vol. 31: ACM, 2001.

[24]. T. E. Ng and H. Zhang, "Predicting Internet network distance with coordinates-based approaches," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2002, pp. 170-179.

[25]. Thomer M. Gil, Frans Kaashoek, Jinyang Li, a. Robert Morris, and J. Stribling. *p2psim*. Available: https://pdos.csail.mit.edu/archive/p2psim/

[26]. A. C. Bavier, M. Bowman, B. N. Chun, D. E. Culler, S. Karlin, S. Muir, *et al.*, "Operating Systems Support for Planetary-Scale Network Services," in *NSDI*, 2004, pp. 19-19.

IOSR Journal of Engineering (IOSRJEN) is UGC approved Journal with Sl. No. 3240, Journal no. 48995.

Bhuvaneswari R. "IAFCM: Integrated and Adaptive File Consistency Maintenance in Peer-to-Peer Network" IOSR Journal of Engineering (IOSRJEN), vol. 08, no. 10, 2018, pp. 64-75.