Middle-East Journal of Scientific Research 20 (4): 517-521, 2014
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2014.20.04.12033

Effective along with Incorporate Document Replication along with Persistence Preservation in P2P Devices

Kumaravel, M. Masthan

Department of Information Technology,
Bharath University, Tagore Engineering College, India

Abstract: In peer-to-peer file sharing systems, file replication and consistency maintenance are most widely used techniques for better system performance. Most of the file replication methods replicates files in requesters or near file owners or along a query path from a owner to a requester, leading to low replica utilization, creates redundant replicas and extra expenses for consistency maintenance. Most of the consistency maintenance methods promote update messages based on message spreading or structure based methods without considering file replication enthusiasm, leading to unsuccessful file update and obsolete file response. This paper presents an Adaptive file Replication and consistency Maintenance Mechanism (ARM) that assimilates file replication and consistency maintenance mechanism that achieves higher productivity in file replication and consistency maintenance at a low cost. In this mechanism, each node specifies the need for the file replication and update polling dynamically by adapting to time-varying number of request received for the file and updates performed in that file, which avoids redundant file replications and updates. For fast data access, the user interest model deletes the unused data from memory in order to reduce the memory size. It dramatically reduces expenses and yields sudden improvements on the ability of both file replication and consistency maintenance approaches.

Key words: File replication · Consistency maintenance · Peer-to-peer

INTRODUCTION

Peer-to-peer (P2P) system is a distributed application architecture that partitions workloads among peers. It is used for load balancing and to control the congestion. The P2P consists of all the participating peers as network nodes. P2P provides dedicated link that transfers data from one node to another node. The entire capacity of the channel is reserved for transmission between the nodes. P2P systems have turn into a standard and commercial way for file sharing and distribution where a requester sends a query for the file is forwarded to a file owner in a distributed manner [1-2]. If file owner receives many requests at a time, it becomes overloaded and cannot respond to queries quickly.

File replication is a powerful method to deal with the problem of overload condition due to sudden increase in demand of a particular file or competitive access to a particular file. It distributes load over replica nodes and improves file query efficiency by reducing file owner response actively and improve path length. File consistency maintenance is to preserve the compatible between a file and its replicas is necessary to file replication.

Thus file replication should reduce redundant replicas to reduce consistency maintenance expenses. The replication methods can be generally classified into three categories namely Server Side, Client Side and Path. Server Side replicates a file to the neighbours of file owner [3]; Client Side replicates a file only to the file requesters [4-5]; and Path replicates on the nodes along the query path from a file owner to a requester [6-8]. However, most of these methods either have low compelling on improving query efficiency or come at a cost of high expenses. These methods make it difficult to adjust the number of replicas to time-varying utilization of replicas and to ensure that all replicas are fully utilized. In EAD [8], traffic to each node which has dedicated peer-to-peer link only to the central controller and continually requesting requesters are chosen as replica nodes. These nodes

Corresponding Author: Kumaravel, M. Masthan, Department of Information Technology,
Bharath University, Tagore Engineering College, India.

periodically determine their query load to create replicas and remove inadequate replicas. More replicas lead to high consistency maintenance expenses and vice versa.

Most consistency maintenance methods update files depends on a structure [9, 10], or message spreading [7], Though these methods usually applied to all file replication methods, they cannot be utilized to their full capability without considering time varying and dynamic replica nodes. Message spreading produces high expenses due to redundant messages and cannot guarantee that all replica nodes receive an update. Structure based methods reduce the expenses but cannot guarantee timely consistency in nodes which can join and leave continuously and repeatedly.

Therefore, without taking the importance of file replication enthusiasm, consistency maintenance generates unnecessary expenses and cannot help to guarantee the accuracy of replica consistency.

This paper presents a mechanism an Adaptive file Replication and consistency Maintenance (ARM) that achieves high productivity in file replication and consistency maintenance at a significantly lower cost. It combines file replication and consistency maintenance in a cooperated and synchronized manner. Each node decides to create or delete a replica and requests for update based on request received for the file and updates performed in that file in a totally decentralized and autonomous manner. It replicates highly queried files and polls at a high frequency for continually updated and queried files. ARM avoids redundant file replications and updates by dynamically adapting to time varying request received for the file and updates performed in that file. It improves replica utilization, file query efficiency and accuracy in consistency. A significant feature of ARM is to achieve an optimized balance achieved between expenses and query efficiency as well as consistency guarantees.

Related Work: File Replication in p2p systems aims to release load in any location where Wi-Fi network access is made publicly available and to decrease file query latency. Several file replication methods are proposed and these are classified into three categories: Server Side, Client Side and Path.

In Server Side replication includes PAST [3] which replicates the file to the neighbour of file owner. It is an Internet-based global P2P storage utility with a storage management and caching system. It replicates files at the nodes whose nodeIDs match closely with the file owner's nodeIDs. File owner has limited number of neighbors, distributing its extra load to the neighbors may also overload them. PAST uses file caching along the lookup path to minimize query latency and balance query load.

In Client Side category, LAR [5] replicates file close to the requesters. LAR [5] is a lightweight, adaptive and system-neutral replication protocol. It specifies the overloaded degree of a server that a file should be replicated and replicates a file to a client. LAR also replicates file location hints along the lookup path Server compare their load to local maximum and desired load. High load causes a server to attempt creation of new replicas usually the sender of last message.

It creates replicas on the source of the query and adds pointers to the replica in caches along the path in the same network. A replica is used only when the replica node queries the file of the replica again and the replica cannot be shared by other requesters. The replicas have low utilization.

In the Path category, Serving DNS Using a Peer-to-Peer Lookup Service [6] Replicate files along a query path between file requester and file owner. Difficult to adjust the number of replicas to the time-varying utilization of replicas and to ensure that all replicas are fully utilized. Large number of replicas needs more updates. It has high expenses for redundant file replication and consistency maintenance.

Along with the file replication methods, numerous file consistency methods have been proposed. Generally they are classified into two categories: structure based [9], [10] and message spreading based [7], the work in SCOPE [9] constructs a tree for update propagation. It will generate high expenses. In a tree structure, if a parent leaves or fails, its children cannot receive the update until the broken link is recovered. The Efficient and Scalable Consistency Maintenance for Heterogeneous Peer-to-Peer Systems [4]. It constructs a hierarchical structure with locality consideration. It builds one structure for each replicated file. If any node in the structure fails, some replica nodes are no longer reachable. Structure may not be able to recover in time which may lead to unsuccessful update notification. DUP [10] propagate update along a routing path. In message spreading based methods, hybrid push/poll algorithm [7] flooding is replaced by rumor spreading to reduce communication expenses. In these methods updates are not guaranteed to be propagated to each replica and redundant message will generate high expenses. Free Net [11-12] replicates

file on the path from the file requester to the target routes and it routes an update to other nodes based on key closeness.

ARM: Adaptive File Replication and Consistency Maintenance Mechanism: ARM assimilate file replication and consistency maintenance by allowing each node to specifies the need for file replication and consistency maintenance based on number of request received for the file and updates performed in that file. File replication needs to minimize the number of replicas to reduce the expenses of consistency maintenance. Consistency maintenance helps file replication to keep the compatible between a file and its replicas. Combination of these two techniques improves their mutual interactions and avoids their conflicting or incompatible behaviors. The basic idea of ARM is to use request received for the file and updates performed in that file for direct file replication and consistency maintenance. The update is based on the number of request received for the file and updates performed in that file. in order to provide high productivity. It avoids unnecessary expenses in both file replication and consistency maintenance. ARM replicates file in the nodes which have large number of connections to guaranteed high replica utilization and remove underuse replicas and expenses of consistency maintenance. ARM aims to guarantee consistency maintenance at a low cost with file replication enthusiasm. ARM uses adaptive polling to ensure timely update operation and avoids redundant updates.

Issues addressed in ARM

File Replication:

- Where to replicate files so that the request for the file can be significantly extricated and the replicas can be fully utilized?
- How to remove underuse file replicas so that the expenses for consistency maintenance are minimized?

Replica Consistency Maintenance:

- How to determine the state of replication that a replica node examines a file owner in order to guarantee timely update for the file?
- How to minimize the number of polling actions to save cost and at the same time provide the accuracy of consistency guarantee?

Overview of ARM: In P2P file sharing systems, some nodes carry more query traffic load than others. The reasons are

- There will be more query traffic along the query paths from the requesters to the file owner.
- Request for the file is non-uniform and time-varying.
- In P2P network, nodes are situated in different places and it may have distinct neighbors.

File Replication: An ARM aim is to achieve high query efficiency and low file replication expenses. In this method, the node which has large number of connections is selected as replica node. If the node has same number of connections then it will consider the number of request received for the file in those nodes. When a node receives queries for a file continually or itself queries a file continually, placing a replica of that particular file in the node can improve the query efficiency and at the same time replicas are used fully. When a replica node doesn't receive queries for its replica continually or itself doesn't query its replica continually, it removes the replica. Based on this principle, a node adaptively polls file owner for update based on the number of request received for the file and updates performed in that file avoid unnecessary overload.

Replica Consistency Maintenance: In ARM consistency maintenance, each replica node requests its file owner to check whether its replica is updated and updates its replica accordingly. Thus, a replica node does not need to update its replica that won't be requested before its next updating. Based on the number of requests for the file, a replica node can know whether its replica should be updated or not. All replica nodes can get the update successfully. The replica nodes directly request the server and there is no delay for replica updating because of enthusiasm. A server does not need to keep track of replica nodes for updating and there is no redundant update message.

The modules are

- Network Construction
- Server
- Identification of Replica node
- Dynamic data Transfer
- User Interest model

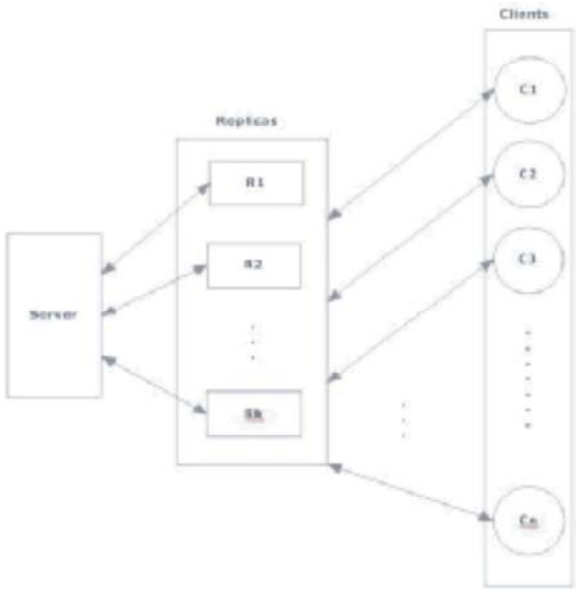

Fig 1: System Architecture

Network Construction: This module is developed in order to create a dynamic network. In a network, nodes are interconnected and the resources can be shared among them. For the successful data transfer the network must be properly controlled and handled.

This module is designed in order to develop a controlled network traffic environment.

Server: The server will have the entire data. The server selects the replica node for data distribution. It distributes the data to client through the replica node. Whenever a server gets any new updating of data, immediately it will update all the replica nodes if a replica need the updated data means it will send the request to the server. The server receives the request and the server will send the copy of data to the replica node.

Identification of Replica Node: To avoid the server overload, the server selects replica nodes in some region. The replica node will reduce the load of the server. The replica node will act as server in the region where it is located. The Identification of the Replica node is achieved by identifying the node which has more number of connections. The node which has highest connection will be selected as replica node. Once the Replica node is selected, a copy of data from the server will send to the replica node. The replica node will serve the request from the requester in the region where the replica node is located.

Dynamic Data Transfer: Whenever the server adds any new data or updates any data, it immediately update the same in all the replica nodes if a replica need data means it will request the server and the server will give the data to the replica node. The updating of the data is carried by the server at the periodic time, So that the replica node will have the updated data when a node sends the request to it. In this way the replica node can work very efficiently. This process will do by the server in a dynamic manner. The dynamic data transfer will be of useful to any node in getting updated data from the replica node.

User Interest Model (UIM): It will calculate the frequent data utilization by the clients and the request passed by the user to the Replica node. Replica node will maintain the data access records. The Replica node will dynamically delete the data which are unused for a period of time. Replica node will assign the time frame for data access. If a file access crosses a time frame, the Replica node dynamically deletes the unused data from its memory in order to reduce its memory size for fast data access.

CONCLUSION

To develop file replication and file consistency maintenance in p2p systems, there has been very little experimentation committed to tackle both challenges simultaneously. File replication needs consistency maintenance to keep the productivity between a file and its replicas and on the other hand, the expenses of consistency maintenance are determined by the number of replicas. Connecting these two components will increase the system performance.

This paper proposes an ARM that achieves high efficiency at a significantly low cost.

It chooses query traffic hubs and frequent requesters as replica nodes to guarantee high utilization of replicas and high query efficiency. Instead of passively accepting replicas and updates, nodes autonomously determine the need for file replication and validation based on request received for the file and updates performed in that file. It guarantees high utilization of file replicas, query efficiency and consistency. At the same time, ARM reduces redundant file replicas, consistency maintenance expenses by polling approach and unnecessary file updates. ARM's update rate decreases with the increase of the query interval time. The numbers of ARM update messages decrease as the query interval time

increases. Its low expenses and high effectiveness are particularly attractive to the deployment of largescale P2P systems.

REFERENCES

1. Shen, H., 2010. "IRM: Integrated File Replication and Consistency Maintenance in P2P Systems," Proc. IEEE Trans. Parallel and Distributed Systems, 21(1): 100- 113.
2. Saroiu, S., P. Gummadi and S. Gribble, 2002. "A Measurement Study of Peer-to-Peer File Sharing Systems," Proc. Conf. Multimedia Computing and Networking (MMCN).
3. Rowstron, A. and P. Druschel, 2001. "Storage Management and Caching in PAST, a Large-Scale, Persistent Peer-to-Peer Storage Utility," Proc. ACM Symp. Operating Systems Principles (SOSP).
4. Xie, G., Z. Li and Z. Li, Dec. 2008. Efficient and Scalable Consistency Maintenance for Heterogeneous Peer-to-Peer Systems, IEEE Trans. Parallel and Distributed Systems, 19(12): 1695-1708.
5. Gopalakrishnan, V., B. Silaghi, B. Bhattacharjee and P. Keleher, 2004. Adaptive Replication in Peer-to-Peer Systems," Proc. 24th Int'l Conf. Distributed Computing Systems (ICDCS).
6. Cox, R., A. Muthitacharoen and R.T. Morris, 2002. Serving DNS Using a Peer-to-Peer Lookup Service," Proc. First Int'l Workshop Peer-to- Peer Systems (IPTPS).
7. Datta, A., M. Hauswirth and K. Aberer, 0000. "Updates in Highly Unreliable, Replicated
8. Abou-Deif, M.H., M.A. Rashed, M.A.A. Sallam, E.A.H. Mostafa and W.A. Ramadan, 2013, Characterization of Twenty Wheat Varieties by ISSR Markers, Middle-East Journal of Scientific Research, 15(2): 168-175.
9. Kabiru Jinjiri Ringim, 2013. Understanding of Account Holder in Conventional Bank Toward Islamic Banking Products, Middle-East Journal of Scientific Research, 15(2): 176-183.
10. Muhammad Azam, Sallahuddin Hassan and Khairuzzaman, 2013. Corruption, Workers Remittances, Fdi and Economic Growth in Five South and South East Asian Countries: A Panel Data Approach Middle-East Journal of Scientific Research, 15(2): 184-190.
11. Sibghatullah Nasir, 2013. Microfinance in India: Contemporary Issues and Challenges, Middle-East Journal of Scientific Research, 15(2): 191-199.
12. Mueen Uddin, Asadullah Shah, Raed Alsaqour and Jamshed Memon, 2013. Measuring Efficiency of Tier Level Data Centers to Implement Green Energy Efficient Data Centers, Middle-East Journal of Scientific Research, 15(2): 200-207.