ISSN (e): 2250 – 3005 || Vol, 04 || Issue, 12 || December – 2014 ||
International Journal of Computational Engineering Research (IJCER)

An Efficient approach of Integrated file Replication and Consistency Maintenance In peer-to-peer systems

R.Sakunthala jenni[1], Dr.T.Pandikumar[2],G.S.Aiswarya[3]

1. Assistant Professor, Department of Computer Science and Engineering, Kalaivani College of Technology, Coimbatore, India-641105,
2. Associate Professor, Dept. of Computer and Information Technology, Defence University, Ethiopia.
3. Assistant Professor, Department of Electronics and communication Engineering, Tamizhan College of Engineering and Technology, Kanyakumari, India-629304,

ABSTRACT:

Peer-to-Peer is a decentralized system which is well known for its high scalability and reliability. So applications on a P2P system are used widely now days. For high system performance of this peer-to-peer file sharing systems, file replication and consistency maintenance are widely used techniques. These two techniques are intimately connected to each other. File replication needs consistency maintenance to keep the consistency between a file and its replicas, and on the other hand, the overhead of consistency maintenance is determined by the number of replicas. Connecting the two important components will greatly enhance system performance. Traditional file replication and consistency maintenance methods either are not sufficiently effective or incur prohibitively high overhead. To overcome these, IRM (Integrated file Replication and Consistency Maintenance inP2P systems) can be used which will achieve high efficiency at a significantly lower cost. Instead of passively accepting replicas and updates, nodes autonomously determine the need for file replication and validation based on file query rate and update rate. It guarantees the high utilization of replicas, high query efficiency and fidelity of consistency. IRM reduces redundant file replicas, consistency maintenance overhead, and unnecessary file updates.

Keywords: *Consistency Maintenance, File Replication, Replica Node, Peer-to-Peer.*

I. INTRODUCTION

Now in a P2Pfile sharing system, every node has a routing table regarding its neighbors. So when a node requests a file the request will be forwarded to the file's destination and then the file will be sent back to the requester. In P2P file sharing systems, the file access is highly repetitive. So if a node becomes a hot spot, there will be a delayed response. File replication is one solution to deal with such problems. Its replicates a file to some other node in order to distribute the query load among the number of nodes and to avoid a hot spot so the file query efficiency can be enhanced. File replication means file consistency maintenance in order to keep the consistency between file and its replicas .For example, a file is changing all its replicas should be updated corresponding.

File replication is an effective method to deal with the problem of overload condition due to flash crowds or hot files. It distributes load over replica nodes and improves file query efficiency. File consistency maintenance to maintain the consistency between a file and its replicas is indispensable to file replication. In most current file replication methods, file owners rigidly specify replica nodes and the replica nodes passively accept replicas. The methods were designed without considering the efficiency of subsequent file consistency maintenance. The number of replicas has a significant impact on the overhead of file consistency maintenance. Large number of replicas needs more updates hence high consistency maintenance overhead and vice versa. So the methods lead to high overhead for unnecessary file replications and consistency maintenance.

IRM integrates file replication and consistency maintenance in a harmonized and coordinated manner. IRM achieves high efficiency in file replication and consistency maintenance at a significantly lower cost. Each node actively decides to create or delete a replica and to poll for update based on file query and update rates in a totally decentralized and autonomous manner. IRM improves replica utilization, file query efficiency, and consistency fidelity. It avoids unnecessary file replications and updates by dynamically adapting to time varying file query and update rates. A significant feature of IRM is that it achieves an optimized trade-off between overhead and query efficiency as well as consistency guarantees.

IRM is good for P2P systems due to a number of reasons. 1) IRM does not require a file owner to keep track of replica nodes. So, it is resilient to node joins and leaves, and thus suitable for highly dynamic P2P systems. 2) Since each node determines its need for a file replication or replica update autonomously, the decisions can be made based on its actual query rate, eliminating unnecessary replications and validations. 3) IRM enhances the guarantee of file consistency. It offers the flexibility to use different replica update rate to cater to different consistency requirements determined by the nature of files and user needs. 4) IRM ensures high possibility of up-to-date file responses.

II. RELATED WORKS

File replication in P2P systems is targeted to release the load in hot spots and at the same time decrease file query latency. Generally, the methods replicate files near file owners, file requesters or along a query path from a requester to a owner.

[2]PAST, [3]CFS, and Backslash replicate each file on close nodes near the file’s owner. Backslash also pushes cache one hop closer to requesters as soon as nodes are overloaded. In LAR and Gnutella, overloaded nodes replicate a file at requesters. Freenet replicates files on the path from a requester to a file owner. [3]CFS, [2] PAST,LAR cache routing hints along the search path of a query.

[4]Cox et al. studied providing DNS service over a P2P network as an alternative to traditional DNS. The caches index entries, which are DNS mappings, along search query paths. Overlook deploys client-access history to place a replica of a popular file on a node with most lookup requests for fast replica location. [21] Less Log determines the replicated nodes by constructing a lookup tree based on IDs to determine the location of the replicated node.

In Ocean Store, files are replicated and stored on multiple servers for security concern without restricting the placement of replicas. Ocean Store maintains two-tier replicas: a small durable primary tier and a large soft-state second tier. Other studies of file replication investigated the relationship between the number of replicas, file query latency, and load balance in unstructured P2P systems. In most of these methods, file owners rigidly determine replica nodes and nodes passively accept replicas. They are unable to keep track replica utilization to reduce underutilized replicas and ensure high utilization of existing replicas. Thus, unnecessary replicas lead to a waste of consistency maintenance.

[5]Yang et al. proposed Parity Replication in IPNetwork Storages (PRINS). PRINS replicates the parity of a data block upon each write operation instead of the data block itself. The data block will be recomputed back at the replica storage site upon receiving the parity. PRINS trades off high-speed computation for communication that is costly distributed storages.

An efficient and adaptive decentralized [6] File replication algorithm in P2P file sharing systems called EAD. In the method, traffic hubs that carry more query load and frequently requesters are chosen as replica nodes.[7]Lv et al. and Cohen and Shenker showed that replicating objects proportionally to their popularity achieves optimal load balance but has varied search latency, while uniform replication has the same average search latency for all files but causes load imbalance.

[8]Tewari and Kleinrock showed that proportional replication can optimize flooding-based search, download time, and workload distribution. They also showed that local storage management algorithms such as Least Recently Used (LRU) automatically achieve near-proportional replication and that the system performance with the replica distribution achieved by LRU is very close to optimal. APRE adaptively expands or contracts the replica set of a file in order to improve the sharing process and achieve a low load distribution among the providers.

III. IRM: INTEGRATED FILE PLICATION AND CONSISTENCY MAINTENANCE MECHANISM

IRM achieves high efficiency in both file replication and consistency maintenance. File replication places replicas in frequently visited nodes to guarantee high utilization of replicas, and meanwhile reduce underutilized replicas and overhead of consistency maintenance. Consistency maintenance in turn aims to guarantee file fidelity of consistency at a low cost with file replication dynamism consideration. IRM aims to guarantee that a file is the updated file when visited. A node adaptively polls file owner for update based on file query rate and update date to avoid unnecessary overload. When a node receives queries for a file frequently or itself queries a file frequently, placing a replica in the node can improve the query efficiency and meanwhile make full use of replicas. When a replica node doesn't receive queries for its replica frequently or itself doesn't query its replica frequently, it removes the replica.

IV. FILE REPLICATION

The replication algorithm achieves an optimized trade-off between query efficiency and overhead in file replication. File replication component by addressing two main issues in file replication:

1) To replicate files so that the file query can be significantly expedited and the replicas can be fully utilized

2) To remove underutilized file replicas so that the overhead for consistency maintenance is minimized.

There are three ways for replicating the file namely Determination of Replica nodes, Creation of Replica, Adaptation of Replica.

Frequent requesters of a file and traffic junction nodes (i.e., hot routing spots) in query paths should be the ideal file replica nodes for high utilization of file replicas. Because in structured P2P systems, some nodes carry more query traffic load while others carry less.

IRM sets a threshold for query initiating rate, denoted by T_q. IRM sets a threshold for query passing rate, denoted by T_l. File destination receives the query, if it is overloaded, it checks if the file query has additional file replication requests. If so, it sends the file to the replication requesters in addition to the query initiator. Or, it replicates file f to its neighbors that forward the queries of file f most frequently.

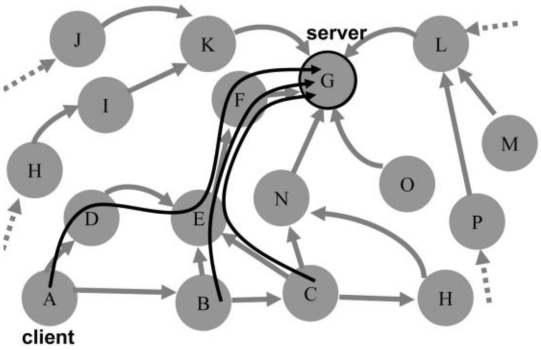

Figure 1: File querying in a file sharing system.

If a file is no longer requested frequently, there will be no file replica for it. IRM lets each replica node periodically update their query passing rate or query initiating rate of a file. If the rates are below their thresholds, the node removes the replica.

3.2 Consistency Maintenance

In IRM poll-based consistency maintenance, each replica node polls its file owner or another node to validate whether its replica is the up-to-date file, and updates its replica accordingly. IRM addresses two main issues in consistency maintenance:

1) To determine the frequency that a replica node probe a file owner in order to guarantee timely file update

2) To reduce the number of polling operations to save cost and meanwhile provide the fidelity of consistency guarantees.

There are two ways for maintenance for file consistency namely Frequency Determination of Polling and Reduction of Poll.

IRM associates a time-to-refresh (TTR) value with each replica. The TTR denotes the next time instant a node should poll the owner to keep its replica updated. The TTR value is varied dynamically based on the results of each polling.

The value is increased by an additive amount if the file doesn't change between successive polls

$$TTR = TTR_{old} + \alpha$$

The TTR value is reduced by a multiplicative facto

$$TTR = TTR_{old} / \beta$$

Values that fall outside these bounds are set to

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

TTRpoll should be calculated based on the following formula:

$$TTR_{poll} = \begin{cases} T_{query} & TTR \leq T_{query}, \\ TTR & TTR > T_{query} \end{cases}$$

Pseudo-code for the IRM adaptive file

consistency maintenance algorithm

```
==operation at time instant Tpoll

if there is a query for the file then
include a polling request into the query for file f
else

send out a polling request

if get a validation reply from file owner then
{
if file is valid then
$TTR = TTR_{old} + \alpha$

if file is stale then{

$TTR = TTR_{old} / \beta$

update file replica
}
if TTR > TTRmax or TTR < TTRmin then
$TTR = \max(TTR_{min}, \min(TTR_{max}, TTR))$

if TTR ≤ Tquery then

TTRpoll = Tquery
else
TTRpoll = TTRg
```

V. COMPARISON STUDY ABOUT IRM

IRM integrates file replication and consistency maintenance in a harmonized and coordinated manner. File replication helps to minimize the number of replicas in order to minimize the overhead of consistency maintenance. At the same time, consistency maintenance helps to guarantee the fidelity of consistency among replicas in file replication dynamism. This principle helps IRM achieve high efficiency and effectiveness in both file replication and consistency maintenance.The Impact of File Replication on Consistency Maintenance. The Impact of consistency Maintenance on File Replication. IRM minimizes the number of replicas while maintaining high efficiency and effectiveness of file replication. It produces much less replicas than the traditional file replication methods while still keeping high utilization of replicas. This significantly reduces the overhead of replica update in consistency maintenance phase, and hence enhances the scalability and efficiency of P2P file sharing systems.

IRM consistency maintenance is to guarantee that a replica file is up to date when being provided. The query rate is also used for consistency maintenance. Based on the query rate, a replica node can know if its replica should be updated or not.

VI. PROPOSED WORK

IRM associates a time-to-refresh (TTR) value with each replica. The TTR denotes the next time instant a node should poll the owner to keep its replica updated. The TTR value is varied dynamically based on the results of each polling. But in case there is no change in file for more than 5 or 10 TTR, there is a waste of querying the owner and increase the burden of network. messages based on message spreading or a structure without considering file replication dynamism, leading to inefficient file update and hence high possibility of outdated file response.

An Integrated file Replication and consistency Maintenance mechanism (IRM) that integrates the two techniques in a systematic and harmonized manner. It achieves high efficiency in file replication and consistency maintenance at a significantly low cost. It reduces overhead and yields significant improvements on the efficiency of both file replication and consistency maintenance approaches.

REFERENCES

[1]. H. Shen, "IRM: Integrated File Replication and Consistency Maintenance in P2P Systems," Proc. IEEE 2010.

[2]. A. Rowstron and P. Druschel, "Storage Management and Caching in PAST,a Large-Scale, Persistent Peer-to-Peer Storage Utility,"Proc. ACM Symp. Operating Systems Principles (SOSP), 2001.

[3]. F. Dabek, M.F. Kaashoek, D. Karger, R. Morris, and I. Stocia,"Wide Area Cooperative Storage with CFS," Proc. ACM ymp .Operating Systems Principles (SOSP), 2001.

[4]. R. Cox, A. Muthitacharoen, and R.T. Morris, "Serving DNS Usinga Peer-to-Peer Lookup Service," Proc. First Int'l Workshop Peer-to-Peer Systems (IPTPS), 2002.

[5]. Q. Yang, W. Xiao, and J. Ren, "PRINS: Optimizing Performance of Reliable Internet Storages," Proc. 26th Int'l Conf. Distributed Computing Systems (ICDCS), p. 32, 2006.

[6]. H. Shen, "EAD: An Efficient and Adaptive Decentralized File Replication Algorithm in P2P File Sharing Systems," Proc. Eighth Int'l Conf. Peer-to-Peer Computing (P2P '08), 2008.

[7]. Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and Replication in Unstructured Peer-to-Peer Networks," Proc. 16th Int'l Conf. Supercomputing (ICS), 2001.

[8]. S. Tewari and L. Kleinrock, "On Fairness, Optimal Download Performance and Proportional Replication in Peer-to-Peer Networks," Proc. IFIP Networking, 2005.

[9]. I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications," IEEE/ACM Trans. Networking, vol. 11, no. 1, pp. 17-32, Feb. 2003.

[10]. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," Proc. ACM SIGCOMM, pp. 329-350, 2001.

[11]. G. Xie, Z. Li, and Z. Li, "Efficient and Scalable Consistency Maintenance for Heterogeneous Peer-to-Peer Systems," IEEE Trans. Parallel and Distributed Systems, vol. 19, no. 12, pp. 1695- 1708, Dec. 2008.

[12]. X. Chen, S. Ren, H. Wang, and X. Zhang, "SCOPE: Scalable Consistency Maintenance in Structured P2P Systems," Proc. IEEE INFOCOM, 2005.

[13]. P. Druschel, M. Castro, A.-M. Kermarrec, and A. Rowstron, "Scribe: A Large-Scale and Decentralized Application-Level Multicast Infrastructure," IEEE J. Selected Areas in Comm., vol. 20, no. 8, pp. 1489-1499, Oct. 2002.

[14]. M. Roussopoulos and M. Baker, "CUP: Controlled Update Propagation in Peer to Peer Networks," Proc. USENIX Ann. Technical Conf., 2003.

[15]. L. Yin and G. Cao, "DUP: Dynamic-Tree Based Update Propagation in Peer-to-Peer Networks," Proc. 21st Int'l Conf. Data Eng. (ICDE), 2005.

[16]. A. Datta, M. Hauswirth, and K. Aberer, "Updates in Highly Unreliable, Replicated Peer-to-Peer Systems," Proc. 23rd Int'l Conf. Distributed Computing Systems (ICDCS), 2003.

[17]. I. Clarke, O. Sandberg, B. Wiley, and T.W. Hong, "Freenet: A Distributed Anonymous Information Storage and Retrieval System," Proc. Int'l Workshop Design Issues in Anonymity and Unobservability, pp. 46-66, 2001.

[18]. J. Lan, X. Liu, P. Shenoy, and K. Ramamritham, "Consistency Maintenance in Peer-to-Peer File Sharing Networks," Proc. Third IEEE Workshop Internet Applications (WIAPP), 2003.

[19]. M. Theimer and M. Jones, "Overlook: Scalable Name Service on an Overlay Network," Proc. 22nd Int'l Conf. Distributed Computing Systems (ICDCS), 2002.

[20]. K. Huang, T. Huang, and J. Chou, "LessLog: A Logless File Replication Algorithm for Peer-to-Peer Distributed Systems," Proc. 18th Int'l Parallel and Distributed Processing Symp. (IPDPS), 2004.