An Efficient Caching Scheme and Consistency Maintenance in Hybrid P2P System

E.Kalaivani
PG Scholar,Dept of CSE(PG)
Sri Ramakrishna Engineering College
Coimbatore

J.Selva Kumar
Assistant Professor, Dept of CSE(PG)
Sri Ramakrishna Engineering College
Coimbatore

***Abstract* - Peer-to-peer overlay networks are widely used in distributed systems. P2P networks can be divided into two categories: structured peer-to-peer networks in which peers are connected by a regular topology, and unstructured peer-to-peer networks in which the topology is arbitrary. The objective of this work is to design a hybrid peer-to-peer system for distributed data sharing which combines the advantages of both types of peer-to-peer networks and minimizes their disadvantages. Consistency maintenance is propagating the updates from a primary file to its replica. Adaptive consistency maintenance algorithm (ACMA) maintains that periodically polls the file owner to update the file due to minimum number of replicas consistency overhead is very low. Top Caching (TC) algorithm helps to boost the system performance and to build a fully distributed cache for most popular information. Our caching scheme can deliver lower query delay, better load balance and higher cache hit ratios. It effectively relieves the over-caching problems for the most popular objects.**

Key words: - hybrid, peer-to-peer systems, overlay network, over caching, structured, unstructured P2P.

I. INTRODUCTION

In the recent years, the evolution of a new wave of innovative network architectures labeled *"peer-to-peer (p2p)"* has been witnessed. A peer-to-peer (P2P) network [1] is a distributed system in which *peers* employ distributed resources to perform a critical function in a decentralized fashion. Nodes in a P2P network normally play *equal roles*; therefore, these nodes are also called peers. The P2P participants join or leave the P2P system frequently; hence, P2P networks are *dynamic* in nature. Each link in a P2P overlay corresponds to a sequence of physical links in the underlying network [1], [6].

The flexibility of the overlay topology and the decentralized control of the peer-to-peer network make it suitable for distributed applications. It can also be used for distributed computing which utilizes the idle resources in the network for huge computing tasks [1].

Based on whether a regular topology is maintained among peers, peer-to-peer networks can be divided into two categories: *structured* peer-to-peer networks in which peers are connected by a regular topology, and *unstructured* peer-to-peer networks in which the network topology is arbitrary. Hence, neither structured peer-to-peer networks nor unstructured peer-to-peer networks can provide efficient, flexible, and robust service alone [11].

In this paper, we propose a hybrid peer-to-peer system for distributed data sharing which combines the structured and unstructured peer-to-peer networks. In the proposed hybrid system, a structured ring-based core network forms the backbone of the system and multiple unstructured peer to peer networks are attached to the backbone and communicate with each other through the backbone. The core-structured network provides an accurate way to narrow down the queried data within a certain unstructured network, while the unstructured networks provide a low cost mechanism for peers to join or leave the system freely.

The main contributions of this paper can be summarized as follows:

- Propose a hybrid peer-to-peer system for distributed data sharing. It utilizes both the efficiency of the structured peer-to-peer network and the flexibility of the unstructured peer-to-peer network, and achieves a good balance between the efficiency and flexibility [1].
- To maintain consistency, using file consistency algorithm for hybrid P2P system so that periodically the file owner to update the file due to number of replicas consistency overhead is very low.

- To boost the performance of hybrid P2P, Top Caching (TCS) algorithm is used to build a fully distributed cache for popular information in P2P systems. It effectively relieves the over-caching problems for the most popular objects.

The rest of the paper is follows: In section II, we reviewed some related papers. In section III, we present a new hybrid P2P system with caching algorithm and consistency algorithm. . In Section IV, we propose the future work and concluding remark is given.

II. REVIEW OF PREVIOUS PAPER

Many peer-to-peer networks have been proposed for different applications in the literature, see, for example [1], [3], [5], [7], [9]. In this paper, we focus on peer-to-peer networks for efficient distributed data (file) sharing among peers. There have been several approaches to cope with network heterogeneity. The most popular way is to cluster peers, and select a super peer in each cluster as a local server to manage the cluster as well as to index objects in the cluster. Intra-cluster communication and lookup can therefore be efficiently done via the super peer of a cluster. The super peers also form an overlay to facilitate inter-cluster communication. The overlay is typically unstructured, e.g., KaZaA, Gia (Chawathe et al., 2003), and recent versions of gnutella.

A. Envoy [3] is a two-layer P2P network where a structured overlay is build on top of an unstructured one. The purpose of using the two-layer architecture is to combine the advantage of each structure and create synergy. Structured overlay, on the other hand, guarantees every search to be completed in bounded steps, typically in logarithmic of the network size. Therefore, by combining the two structures, both popular and rare/distant objects can be effectively and efficiently located.

B. BitTorrent [7] is a centralized unstructured peer-to-peer network for file sharing. When a peer has received the complete file, it should stay in the system for other peers to download at least one copy of the file from it. Since BitTorrent uses a central server to store all the information about the file and the peers downloading the file, it suffers so called “single point of failure” problem, which means that if the central server fails, the entire system is brought to a halt.

C. YAPPER [9] combines both structured peer-to-peer networks and unstructured peer-to-peer networks to provide a scalable lookup service over an arbitrary topology. Both data keys and peers are hashed to different buckets or colors. Data is stored in the peer in the same color. Finally, all the peers in the same color will be checked. However, YAPPERS is designed for efficient partial lookup that only returns partial values of data. For a complete lookup, YAPPERS still needs to flood the request to all peers that are in the same color as the data.

D. In [5], the authors propose a hybrid peer-to-peer system, which treats rare and popular data items differently. Some “ultrapeers” form a structured peer-to-peer network, which is responsible for caching the rare data. Each ultrapure has multiple attached leaf peers. Data lookup is first performed through the conventional flooding method. If not successful, the query is reissued to an ultrapeer as a DHT data lookup. It is somewhat similar to the hybrid system proposed in this paper. However, the main difference is that in [16], the structured overlay was used as a supplement for unsuccessful flooded data lookup.

E. In [1], they proposed a hybrid peer-to-peer system that combines both the structured peer-to-peer network and the unstructured peer-to-peer networks to form a two-tier hierarchy to provide efficient and flexible distributed data sharing service. The hybrid peer-to-peer system can utilize both the efficiency of the structured peer-to-peer network and the flexibility of the unstructured peer-to-peer network and achieve a good balance between them. However, a disadvantage is that they did not focused on the consistency and caching the data.

F. In [12], a hybrid peer-to-peer (P2P) system uses flooding and DHT are both employed for content locating. The decision to use flooding or DHT largely depends on the population of desired data. By dynamically detecting the content popularity, PASH properly selects search methods and efficiently saves query traffic cost and response time.

G. IRM file consistency techniques [2] which is generally used to integrate file replication and consistency maintenance by letting each node autonomously determine the need for file replication and update based on actual file query rate and update rates. However, they are based on the chord P2P system.

III.OUR IDEA

In this section, we first give an overview of the new hybrid peer-to-peer system. We describe how to maintain the peer-to-peer network topology when peers join and leave the system. Then, we describe how to insert and look up data items in the system. Finally describe the consistency and caching scheme in the hybrid P2P system.

A. *Construction of Hybrid P2P System:*

The new hybrid peer-to-peer system is composed of two parts: a core transit network and many stub networks, each of which is attached to a node in the core transit network. The core transit network, called t-network, is a structured peer-to-peer network, which organizes peers into a ring. We call peers in the t-network t-peers. Each t-peer is assigned a peer ID (p_id). Each t-peer maintains two pointers, which point to its successor and predecessor, respectively.

A stub network, called s-network, is a Gnutella-style unstructured peer-to-peer network. The topology of an s-network is arbitrarily formed. Each s-network is attached to a t-peer and this t-peer belongs to both the t-network and the s-network. One thing to mention about the s-network is that the topology of an s-network is a tree instead of a mesh. Fig. 1 shows the overview of the proposed hybrid peer-to-peer system.

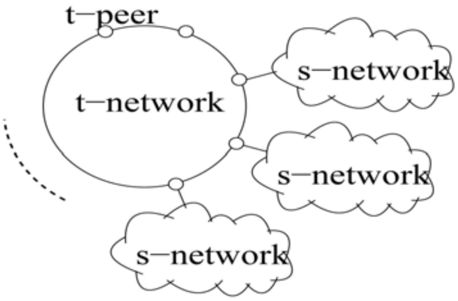

Fig 1: Construction of Hybrid P2P systems

The basic idea behind the hybrid peer-to-peer system is that the t-network is used to provide efficient and accurate service while the s-network is used to provide approximate best-effort service to accommodate flexibility. Peers can join either t-network or s-network directly. The hybrid system can effectively reduce the topology maintenance overhead caused by peer joining or leaving.

In this paper, we focus on applying the hybrid peer-to-peer system to distributed data sharing. A data item is represented by a (key, value) pair. A key is a label or name of the data, such as a file name, while a value is the content associated with the key, such as a file. A peer uses operation store (key, value) to insert the data item into the system and operation lookup (key) to obtain the value of the data item. Here, we only consider exact-match data lookup.

B. *Consistency Algorithm:*

In the distributed data sharing [2], [4] [10], [13], the consistency of the data needs to be focused because there are two different networks are built on single. Maintaining consistency between frequently updated or even infrequently updated files and their replicas is a fundamental reliability requirement for a P2P system. P2P systems are characterized by dynamism, in which node join and leave continuously and rapidly. Moreover, replica nodes are dynamically and continuously created and deleted. For consistency maintenance, we introduce an algorithm for hybrid network, which is known as Adaptive File Consistency Algorithm (AFCA).

1) Polling frequency Determination: AFCA employs a linear increase multiplicative decrease algorithm in which frequently modified files is polled more frequently than relatively static files.

We assign the time-to-refresh (TTR) value with each replica [2]. The TTR denotes the next time instant a node should poll the owner to keep its replica updated. The value is increased by an additive amount if the file doesn't change between successive polls

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

where α,α>0 is an additive constant. In the event the file is updated since the last poll, the TTR value is reduced by a multiplicative factor:

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

where β,β> 1, is the multiplicative decrease constant. In this proposed algorithm takes as input two parameters: TTR_{min} and TTR_{max}, which represent lower and upper bounds on the TTR values. Values that fall outside these bounds are set to

$$TTR=\max (TTR_{min}, \min (TTR_{max},TTR)) \ \text{---------} \ (3)$$

Generally, the algorithm begins by initializing

$$TTR = TTRmin = \Delta t \quad \text{------------} \ (4)$$

2) Adaptive polling reduction: In addition to the file change rate, file query rate is also a main factor to consider in consistency maintenance. However, most current consistency maintenance methods neglect the important roles that file query rate plays in reducing overhead. In AFCA, combines file query rate into consideration for poll time determination. We use TTR_{query} and TTR_{poll} to denote the next time instant of corresponding operation of a file.

AFCA Algorithm

//operation at time instance T_{poll}

```
if there a query for the file then
        include a polling request into the query for a file f
 else
        send out the polling request
if get a validation reply from file owner then
{
        if the file is valid then
            $TTR = TTR_{old} + \alpha$
        if the file is stale then {
                $TTR= TTR_{old} /\beta$
                Update the file replica}
    if  $TTR> TTR_{max}$ or $TTR< TTR_{min}$ then
      $TTR=\max(TTR_{min},\min (TTR_{max},TTR))$
        if $TTR<T_{query}$ then
                $TTR_{poll} =T_{query}$
        else
                $TTR_{poll} = TTR$}
```

C. Caching Algorithm:

A Hybrid P2P caching system [6], [8], [12], [14], [15] should take into account dynamic characteristics of peers. Unlike static dedicated caches, peers may join or leave a P2P network dynamically. Therefore, the system should minimize the management overheads and the performance degradation caused by dynamic participation of peers.

Top-caching (TC) Algorithm:

```
While k ≤K and X has not obtained j:
    1.  X uses substrate to determine i, the $k_{th}$ place
        winner for j
    2.   X requests j from i.
        • Node i update $\lambda_j$ (i).
        • If node i already has j, node i sends j to X;
        stop.
        • If node i does not have j but it should, i
    gets j, stores j and evicts files if necessary. Node i
    send j to X.
    3.   k = k + 1
```

The Top Caching (TC) algorithm is a fully distributed, adaptive content management algorithm that is, for all practical purposes, optimal for DHT-based file sharing systems. Some of the parameters used are Failure Rate, join latency and lookup latency.

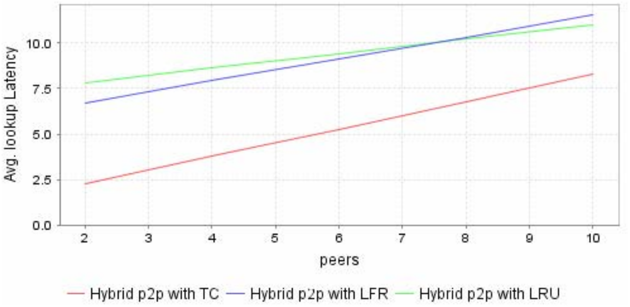

Fig 2: Join Latency Comparison

Finally, the system will perform well by Consistency and Caching schemes and also boost the system performance.

IV. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a hybrid peer-to-peer system that combines both the structured peer-to-peer network and the unstructured peer-to-peer networks to provide efficient and flexible distributed data sharing service. Hence, the hybrid system has less lookup latency and higher data lookup efficiency. Top Caching (TC) algorithm is used for caching the most popular and rare data items. Nevertheless, it also helps to boost the system performance. Our caching scheme can deliver lower query delay, better load balance and higher cache hit ratios. It effectively relieves the over-caching problems and to balance the load of the hosting peer when many peers request popular data.

REFERENCES

[1] Min Yang, Yuanyuan Yang., "An Efficient Hybrid Peer-to-Peer System for Distributed Data Sharing " IEEE transaction on Computers, Vol. 59, no.9, September 2010.

[2] Haiying (Helen) Shen, "IRM: Integrated File Replication and Consistency Maintenance in P2P Systems", IEEE trans on parallel and distributed systems, vol. 21, no. 1, Jan 2010.

[3] Yuh-JzerJoung, Zhang-WenLin,"On the self-organization of a hybrid peer-to-peer system", ELSEVIER, Journal of Network and Computer Appln 33 (2010).

[4] Z. Li, G. Xie, Z. Li, "Efficient and scalable consistency maintenance for heterogeneous peer-to-peer systems", TPDS (2008).

[5] B.T. Loo, R. Huebsch, I. Stoica, and J.M. Hellerstein, "The Case for a Hybrid p2p Search Infrastructure," Proc. Workshop Peer-to-Peer Systems (IPTPS '04), pp. 141-150, Feb. 2004.

[6] V. Gopalakrishnan, B. Silaghi, B. Bhattacharjee, P. Keleher, "Adaptive replication in peer-to-peer systems", in: Proc. of ICDCS, 2004

[7] P2P traffic is booming, "BitTorrent The Dominant Protocol". http://torrentfreak.com/p2p-traffic-still-booming-071128/ .

[8] P. Linga, I. Gupta, and K. Birman. Kache: "Peer-to-peer web caching using kelips". In submission, June 2004.

[9] P. Ganesan, Q. Sun, and H. Garcia-Molina, "YAPPERS: A Peer-to-Peer Lookup Service over Arbitrary Topology," Proc. IEEE INFOCOM '03, pp. 1250-1260, 2003.

[10] J. Lan, X. Liu, P. Shenoy, K. Ramamritham, "Consistency maintenance in peer-to-peer file sharing networks", in: Proc. the IEEE Workshop on Internet Applications, WIAPP, 2003.

[11] E. Cohen, S. Shenker," Replication strategies in unstructured peer-to-peer network's", in: Proc. of ACM SIGCOMM, 2002.

[12] A. Rowstron, P. Druschel, "Storage management and caching in PAST, a large scale, persistent peer-to-peer storage utility", in: Proc. of SOSP, 2001.

[13] V. Duvvuri, P. Shenoy, and R. Tewari. "Adaptive Leases: A Strong Consistency Mechanism for the World Wide Web", In Proceedings of the IEEE Infocom'00, Tel Aviv, Israel, March, 2000.

[14] S. Iyer, A. Rowstron, and P. Druschel, "Squirrel: A Decentralized, Peer-to-Peer Web Cache," Proc. 21st Ann. ACM Symp. Principles of Distributed Computing (PODC), 2002.

[15] M. Nelson, B. Welch, J. Ousterhout" Caching in the sprite network file system", IEEE/ACM Transactions on Networking (1) (1988).

AUTHORS PROFILE

E.Kalaivani received B.E Degree from Anna University in 2010. She is currently pursuing M.E Degree in Software Engineering in Anna University. Her area of interest in Computer Networks and Software Engineering.

J. Selvakumar received the B.E degree in Computer Science & Engineering from the Madras University in May 2001 and the M.E degree in Computer Science & Engineering from the Bharathiyar University in December 2002. He is currently working towards the Ph.D degree at the Anna University, Chennai .He is currently an Assistant Professor. His current research interest is Requirements Engineering.