IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 1, March, 2013
ISSN: 2320 - 8791
www.ijreat.org

An Efficient Hybrid Peer-to-Peer System for Distributed Data Sharing

[1]P.Elamathi, [2]S.Saranya, [3]E.Elamathi,
Department of CSE,
[123]Apollo Priyadarshanam Institute Of Technology, Tamilnadu.

Abstract
Peer-to-peer overlay networks are widely used in distributed systems. 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 areconnected by a regular topology, and unstructured peer-to-peer networks in which the topology is arbitrary. Structured peer-to-peer networks usually can provide efficient and accurate services but need to spend a lot of effort in maintaining the regular topology. On the other hand, unstructured peer-to-peer networks are extremely resilient to the frequent peer joining and leaving but this is usually achieved at the expense of efficiency. 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. The proposed hybrid peerto-peer system is composed of two parts: the first part is a structured core network which forms the backbone of the hybrid system; the second part is made of multiple unstructured peer-to-peer networks each of which is attached to a node in the core network. The core structured network can narrow down the data lookup within a certain unstructured network accurately, while the unstructured networks provide a low-cost mechanism for peers to join or leave the system freely. A data lookup operation first checks the local unstructured network, and then, the structured network. This two-tier hierarchy can decouple the flexibility of the system from the efficiency of the system. Our simulation results demonstrate that the hybrid peer-to-peer system can utilize both the efficiency of structured peer-topeer network and the flexibility of the unstructured peer-to-peer network and achieve a good balance between the two types of networks.

Index Terms—Peer-to-peer systems, P2P, structured peer-to-peer, unstructured peer-to-peer, hybrid, overlay networks

1. INTRODUCTION

Last few years have witnessed a rapid development of peer-to-peer networks. Research has shown that a large fraction of traffic in the Internet is occupied by peer-to-peer applications [2]. A peer-to-peer (P2P for short) network is a logical overlay network on top of a physical network. Each peer corresponds to a node in the peer-to-peer network and resides in a node (host) in the physical network. All peers are of equal roles. The links between peers are logical links ,each of which corresponds to a physical path in the physical network. The physical path is determined by a routing algorithm and composed of one or more physical links .Logical links can be added to the peer-to-peer network arbitrarily as long as a corresponding physical path can be found, that is, the physical network is connected .The flexibility of the overlay topology and the decentralized control of the peer-to-peer network make it suitable for distributed applications. For example, it can be used for distributed data (file) sharing, where peers announce the data (files) they have and exchange data (files) from each other through a loosely formed peer-to-peer network, or for collaborative Web caching in which Web pages are cached in collaborative peers to reduce network delay for URL requests, or for application layer multicast in which peers are group members and the peer-to-peer overlay network is a multicast tree. It can also be used for distributed computing which utilizes the idle resources in the network for a huge computing task. Finally, it can be used to provide communication anonymity in which the sender's identity is concealed .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. Structured peer-to-peer networks build a distributed hash table (DHT) on top of the overlay network. The hash table supports efficient data insertion and lookup. Given a key of the data item, the corresponding value of the data item can be inserted or found by transforming the key to a hash value by a hash function. The hash value is the index of the data item and all the hash values form the key space. In DHT, the key space is divided among peers. Each peer is responsible for one partition of the key space. Peers are connected by an overlay network through which the requests of data insertion and lookup are delivered .Structured peer-to-peer

networks can provide efficient and accurate query service but need a lot of efforts to maintain the DHT, which makes it vulnerable to frequent peer joining and leaving, also known as churn. Churn is a common phenomenon in peer-to-peer networks. Measurement studies of deployed peer-to-peer networks show a high rate of churn [21], [22]. Unstructured peer-to-peer networks organize peers into an arbitrary network topology, and use flooding or random walks to look up data items. Each peer receiving the flooding packets or random walk pack checks its own database for the data item queried. This approach does not impose any constraint on the network topology. It can perform complex data lookup and support peer heterogeneity. Unstructured peer-to-peer networks are resilient to churn while they usually achieve this goal by sacrificing the data query efficiency and accuracy. Hence, neither structured peer-to-peer networks nor unstructured peer-to-peer networks can provide efficient, flexible, and robust service alone. The motivation of this paper is to combine the two types of peer-to-peer networks and provide a hybrid solution which can offer efficiency and flexibility at the same time. To achieve this goal, the solution should inherit the advantages of both types in such a way that their disadvantages are minimized .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 peerto-peer networks are attached to the backbone and communicate with each other through the backbone. Data is generated and distributed among the peers. The corestructured 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 summarizedas 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. Give a theoretic analysis on the system performance, and derive quantitative results on the improvement of the join latency and data lookup latency. Evaluate the proposed scheme through extensive simulations. Simulation results match the theoretic analysis and show that the proposed scheme achieves the original design goal.The rest of the paper is organized as follows: In Section 2,we briefly describe some related work. We present the new hybrid peer-to-peer system in Section 3 and analyze its performance in Section 4. We give some enhancements to the hybrid peer-to-peer system in Section 5. In Section 6, we present the simulation results and discuss the performance of the system. Section 7 gives the concluding remark and future work.

2.EXISTING SYSTEM

A hybrid peer-to-peer system for distributed data sharing which consists of two parts combines the advantages of both types of peer-to-peer networks and minimizes their disadvantages. The first part is a structured core network which forms the backbone of the hybrid system. The second part is made of multiple unstructured peer-to-peer networks each of which is attached to a node in the core network. This two-tier hierarchy decouples the flexibility of the system from the efficiency of the system. If server is crashed, the entire systems are collapsed. At the same time more than one members giving the request to sever means, automatically the server performance ratio minimized .If we set the public or private keys for data sharing means, it will easily detect by intruder .If we use client-server model has load imbalance problem

3.PROPOSED SYSTEM

In the proposed hybrid peer-to-peer system the work is done to minimize the failure ratio and thus improving the system performance. This can be done by designing a caching scheme. The caching scheme distributes the load among as many peers as possible in such a way that no peers are overloaded .It minimizes the problem of concentrating the workload on a single entity.

4.SYSTEM IMPLEMENTATION

The scope of the system to solve the scalability problem in multicast communications. Since a large group is divided into many small groups. Each subgroup is treated almost like a separate multicast group with its own subgroup key. All the keys used in each subgroup can be generated by a group of KGCs in parallel. The intuitively surprising aspect of

this scheme is that, even the subgroup controller aborts, it does not affect the users in this subgroup. Because every user in the subgroup can act as a subgroup controller. From the security analysis we can see that our scheme satisfies both forward and backward secrecy.

The main security properties of multicast are:

1.Group Key Secrecy guarantees that it is computationally infeasible for a passive adversary to discover any group key.
2.Backward Secrecy is used to prevent a new member from decoding messages exchanged before it joined the group. This property guarantees that a passive adversary who knows a subset of group keys cannot discover the previous group keys.

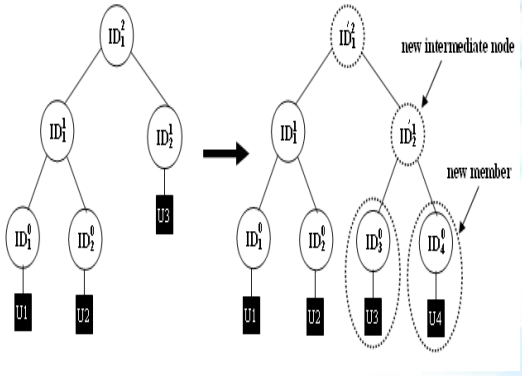


Fig.1U4 is added to the group

After U4 joined the group, all the private keys from U3's parent to the root are changed. Since U4 does not know the private keys in the previous identity tree, So we cannot compute the group key in the previous session. Thus the backward secrecy is satisfied.

- Forward Secrecy is used to prevent a leaving user or expelled group

member to continue accessing the group communication. This property guarantees that a passive adversary who knows a subset of old group keys cannot . Discover the subsequent group keys.

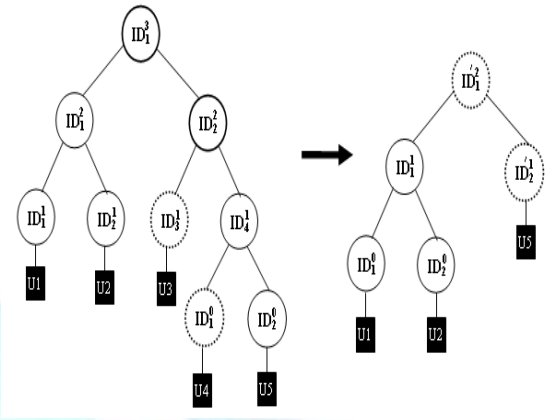
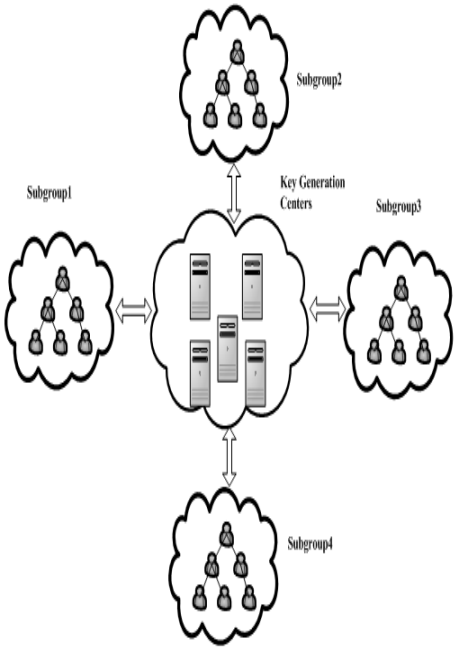


Fig.2 U3 and U4 are deleted from the group

After U3 and U4 leave the group, all the private keys known by U3 and U4 are changed. So U3 and U4 cannot compute the group key in the future. This means that our scheme satisfies forward secrecy.

6.SYSTEM ARCHITECTURE:



7. MODULE DESCRIPTION

1. Login Module

2. Active Node in Dynamic root

3. Find Group Key

4. Relieve Node in Dynamic Root

5. Data Object Sharing

7.1 Login Module

a. Authentication checking

In this module checks whether the user is authenticated or not if the user is authenticated then they have the permission to process further transactions otherwise they cannot access any transaction in this system.

b. Registration process.

If the new user to this system first they must registered in the register module after they have continue to process in the system. During the registration the user must enter the valid information for create new user name and password if only valid user. Once user registered after they have authorized user of this system. Recalibrate

7.2Join Node in Dynamic route

In our communication group have number of client nodes are interconnected in the server. Each group has the separate group key for communication in the group. When a new member joins or leaves the communication group, only it's reflecting for local group. Peer-to-peer networks are highly dynamic and autonomics stems in which peers can join or leave the systems at any time. Peers that want to join the system first contact a well known server to obtain an arbitrary existing peer in the system.

7.3 Leave Node in Dynamic root

Unstructured peer-to-peer networks process join or leave requests in a more flexible way than structured peer-to-peer networks largely because structured peer-to-peer networks have to maintain the network topology after peers join or leave. Thus, when a member joins or leaves the communication group, it joins or leaves only its local subgroup. As a result, only the local subgroup communication key needs to be refreshed and the scalability problem is greatly mitigated.

Data is generated and inserted to the system by peers. The peer generating the data item first hashes the key into this space. If the id lies in the range of the current network, the data item is inserted to its database and the data insertion is completed. If the id does not lie in the range, the data item is sent to the peer of the current network. Then, it is forwarded along the network until it reaches the peer in charge of the ID range covering id. Then, the data item is inserted into the database of the t-peer.

7.4. Data Sharing

In cryptography encryption is the process of transforming information referred to as plaintext using an algorithm called cipher to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key. The result of the process is encrypted information in cryptography, referred to as ciphertext. In many contexts, the word encryption also implicitly refers to the reverse process, called decryption. In our multicast communication group mainly concentrates on enabeling the data transfer among the server and multiple clients in the network communications. The server sends encrypted data and clients receive the decrypted data.

7.4.1 Encryption

Encryption is the conversion of data into a form, called a cipher text that cannot be easily understood by unauthorized people. The translation of data into a secret code. Encryption is the most effective way to achieve data security. In our communication protocol each sub group maintain the separate group key for the communication among the network. Server send data transfer to multiple clients in encrypted data because in between the data transfer unauthorized people cannot see easily what the server sending data to clients.

7.4.2Decryption

Decryption is the process of converting encrypted data back into its original form. The process of decoding data that has been encrypted into a secret format. Decryption requires a secret key or password. Server send data transfer to multiple clients in encrypted data because in between the data transfer

unauthorized people cannot see easily what the server sending data to clients. After receiving encrypted data back into original form in client side. Converting encryption and decryption only for security purpose.

8.BLOWFISH Encryption / Decryption algorithm:

Blowfish is a keyed, symmetric block cipher, designed in 1993 by Bruce Schneier and included in a large number of cipher suites and encryption products. Blowfish provides a good encryption rate in software and no effective cryptanalysis of it has been found to date. However, the Advanced Encryption Standard now receives more attention.

Blowfish is capable of strong encryption and can use key sizes up to 56 bytes (a 448 bit key). The key must be a multiple of 8 bytes (up to a maximum of 56). This example will automatically pad and unpad the key to size. Because Blowfish creates blocks of 8 byte encrypted output, the output is also padded and unpadded to multiples of 8 bytes.

Blowfish has a 64-bit block size and a variable key length from 32 up to 448 bits. It is a 16-round Feistel cipher and uses large key-dependent S-boxes. It is similar in structure to CAST-128, which uses fixed S-boxes.

Blowfish is a symmetric block cipher that can be used as a drop-in replacement for DES or IDEA. It takes a variable-length key, from 32 bits to 448 bits, making it ideal for both domestic and exportable use. Blowfish was designed in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms.

The decryption algorithm of a block cipher should be identical to encryption algorithm step by step in reverse order. But for Blowfish cipher, the encryption algorithm is so well designed, that the decryption algorithm is identical to the encryption algorithm step by step in the same order, only with the sub-keys applied in the reverse order.

9.CONCLUSION

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 peertopeer networks are attached to the backbone and communicate with each other through the backbone. Data is generated and distributed among the peers. 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 lowcost mechanism for peers to join or leave the system freely.

FUTURE WORK

Our future work includes how to design a caching scheme for the hybrid peer-to-peer system to improve the system performance. In the case that some extremely popular data are requested by a large amount of peers,the peer hosting the data may be overwhelmed by the large amount of requests. The goal of the caching scheme is to balance the load of the hosting peer when popular data are requested by many peers. The idea is to distribute the load among as many peers as possible so that no peer is overwhelmed. The challenges include how to choose some surrogate peers to redirect the requests to, which data should be cached and how long the data should be cached.

10.REFERENCES

[1] E.K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A
Survey and Comparison of Peer-to-Peer Overlay Network Schemes," IEEE Comm. Surveys and Tutorials, vol. 7, no. 2,
pp. 72-93, Mar. 2005.
[2] S. Sen and J. Wang, "Analyzing Peer-to-Peer Traffic Across Large
Networks," Proc. Internet Measurement Workshop, Nov. 2002.
[3] Y. Chu, S. Rao, and H. Zhang, "A Case for End System Multicast,"
Proc. ACM Sigmetrics '00, pp. 1-12, June 2000.
[4] E. Brosh and Y. Shavitt, "Approximation and Heuristic Algorithms
for Minimum Delay Application-Layer Multicast Trees,"
Proc. IEEE INFOCOM '04, Mar. 2004