The Management Strategy of Metadata in Large-scale Network Storage System

Haifeng Zhong[1] and Jianying Xiong[1]

[1]*Jiangxi Police Institute, Nanchang Jiangxi, P.R. China*

Abstract. The wan Internet storage system based on Distributed Hash Table uses fully distributed data and metadata management, and constructs an extensible and efficient mass storage system for the application based on Internet. However, such systems work in highly dynamic environments, and the frequent entry and exit of nodes will lead to huge communication costs. Therefore, this paper proposes a new hierarchical metadata routing management mechanism based on DHT, which makes full use of the node stabilization point to reduce the maintenance overhead of the overlay. Analysis shows that the algorithm can effectively improve efficiency and enhance stability.

1 Introduction

In recent years, along with the proliferation of Internet-based applications, the rapidly increasing number of users share files and data over the network [1]. However, files such as multimedia and high-definition video contain more data than ever before. Therefore, the storage subsystem requires a large amount of storage capacity, which often becomes an extension bottleneck of the entire application system.Various applications impose higher requirements on the performance of the storage subsystem in the following aspects: 1) Capacity, 2) High Performance, 3)Scalability, 4)High Availability, 5)Manageability. Naturally, with the continuous increase of computer performance and network bandwidth, it is a necessary choice to use distributed technology to build a network distributed storage system with massive storage space, so as to improve the capacity, reliability and scalability of the application system [2].

The virtualized file systems based on structured overlay networks and Distributed Hash Table provide an effective large-scale network storage system model consisting of personal computers and file servers distributed over a wide area Internet network. There is no master-slave relationship with each other. Each node has a part of system files and metadata dispersedly stored, and is integrated into a single large file system through file system virtualization technology. Typical systems in this area are Oceanstore, Farsite and CFS.

The large-scale network storage system is an open system which contains a large number of nodes. And the nodes in such system can randomly join and exit the system, which makes it difficult to maintain a cascading network structure. So updating of the topology often results in huge communication cost when nodes join and quit frequently, which seriously affects the scalability of the system [3]. Therefore, how to properly organize the nodes to reduce the overheads of maintenance and improve the efficiency of the entire system is a key issue to be solved in such systems.

In this paper we presents a new DHT-based Hierarchical metadata routing management mechanism called HMRM, which reduces the maintenance overhead of the overlay network by distinguishing node functions. The HMRM uses stable nodes to manage the addition and exit of member nodes to improve system update efficiency and enhance stability.

The remainder of this paper is structured as follows: We first discuss some related work in section 2. Then in section 3 illustrate the main design of HMRM. And in section 4 we justify our approach through analysis. Finally, we draw conclusions from our aforementioned work.

2 Related work

The early distributed file system was more concerned with access performance and data reliability under the constraints of network resources and computing resources. And the goal of the systems is to provide the upper layer applications with remote file access based on standard protocols. Generally acknowledged, these systems have basically reached the requirements of building a network-based distributed storage system, and thus have been widely used such as CFS, AFS, and Cassandra [4]. However, in many aspects such as storage capacity, data availability, system scalability, data manageability, file access performance, etc., these systems are still far from the actual requirements of mass distributed storage services based on the Internet.

The rapid development of modern computer network technology and the increasing transmission bandwidth

[a] Corresponding author: billpaper@sina.com.cn

provide the possibility to use the network to build a storage system with large capacity, high reliability and scalability. Thus, recently large-scale network storage systems based on structured overlay networks and distributed hash tables have become a new trend in the development of distributed storage technologies [5]. Because such systems often contain a large number of nodes and data, how to efficiently store and retrieve the required metadata in the system to locate the corresponding nodes and data is the cornerstone and key to the construction of the system. And it is also a difficult point for research [6,7]. In general, there are mainly three resource location model:

(1) Central directory model

Despite the advantages of simple and high efficiency, the main problem of this kind of file sharing model in application is that with the increase of scale, the central directory server must become the single point of failure and performance bottleneck of the system, which greatly limits the scalability of the system.

(2) Flooded requests model

The flooding request mode is a fully distributed model without any central server. And every node in the system knows its neighbor nodes directly connected to itself. Thus the main drawback of flooding request mode is that query overhead is large. With the increase of nodes, network communication has increased rapidly, resulting in poor scalability of the system.

(3) File routing model

File routing model is a hot topic in the research of large-scale distributed resource location algorithms. In this model, the system assigns a PID to each node according to the Hash method. At the same time, each node communicates with each other to establish the routing table of the node according to a certain protocol. When shared files are published on the system, FID is generated according to the name of the file, and Its metadata is placed on the specific PID according to the protocol. Therefore, there is no single point of failure in the file routing mode, and network bandwidth consumption is small, which is more efficient than centralized and flooded. However, if the nodes in the network frequently join and quit, the large-scale failure of the adjacency relationship will affect the query efficiency. Typical system based file routing model are CAN, Chord [8] and so on.

3 System design

In this section, we introduce the basic design of our systems. The distributed storage system consists of a large number of independent nodes distributed on the network. With the cross_disk file management capabilities of the distributed file system, a large amount of data information stored on different storage nodes is integrated to achieve massive information storage. The system structure shown in Figure 1 can be divided into three layers:

(1) Physical resource layer: As the lowest level in the model, it is responsible for physical connection,data storage and transmission between nodes. It usually transfers or forward data through protocol messages such as 802.11, TCP and UDP.

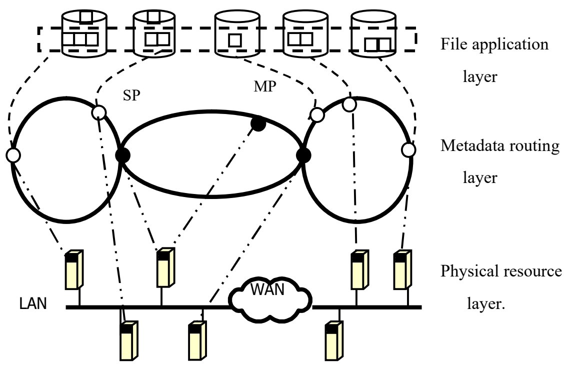


Figure 1. The Architecture of large network storage system.

(2) The metadata routing layer: The metadata routing layer is the foundation of the distributed storage system, which is in the middle of the model. It manages the dynamically joining and exiting of the system nodes through the collaboration among nodes, and provides the query location service of the resources in the overlap network. The metadata routing layer is an overlay based on the IP network. Each node on the overlay stores the local routing information of the system through the routing table and maintains the topology correctness of the overlay.

The system assigns each node a m_bit PID according to the Hash method. At the same time, each node establishes the routing table of the node by communicating with each other according to the chord_like protocol. When the shared file is released to the system, an m_bit FID is generated based on the attribute of the file, such as the name Hash, and the metadata is stored in its successor node according to the protocol. The successor node denoted as PID=successor(FID) is the first node whose PID is greater than or equal to the FID.

(3) File application layer: Responsible for organizing the data stored on the physical nodes in the network to form a virtual file system to provide external file services. The application layer uses the metadata routing layer to publish and locate file data, and uses the free storage space of nodes on the Internet to construct a storage and information sharing application system for massive data. At the same time, the application layer provides users with a user interface.

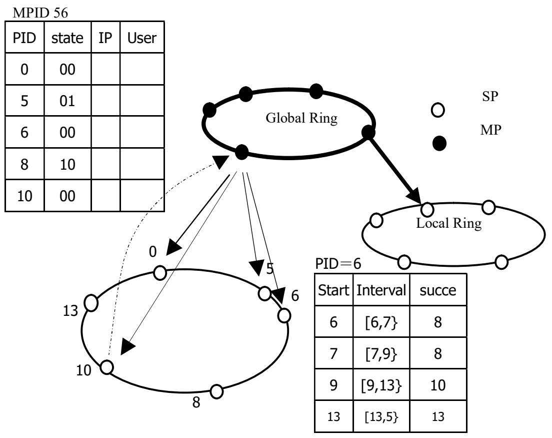
When the user needs to access the File A, the application layer issues an instruction read (File A) and at the same time initiates an object location query find (File A) to the metadata routing layer. After receiving the instruction, the metadata routing layer converts the file name File A into FID=H (File A), and sends out the instruction lookup (FID). This layer finds the IP address of the next-hop node PID according to the local routing table. The physical resource layer issues an instruction to the next hop node to forward send (lookup(FID), IP). After receiving the metadata routing layer forwarding instruction, the physical resource layer will call the lower layer address mapping service to complete the mapping of the IP address to the MAC address, and the next hop node will forward the lookup (FID) until obtaining the IP address of the PID corresponding to the FID and returning it to the application.

To support the Large-scale Network Storage System, all nodes in the HMRM system are divided into a management node MP and a storage node SP.The geographically adjacent SP is organized into clusters, each of which is managed by a stable and powerful MP, and MP provides services such as registration, authentication, cross cluster queries, etc. for all nodes in the system, but does not store system data and metadata. At the same time, the SP provides system file services and data services. It not only publishes files and metadata to other SPs in the system, but also responds to requests from other SPs for queries, storage, and downloads.

Table 1. The definition of the state value.

State value	Corresponding status change
00	Keep working
01	New join node and keep working
10	Working state becomes invalid
11	Keep invalid

Each MP in the HMRM algorithm saves a node status table to record changes in node status. And the definition of the state value is shown in Table 1. The MP periodically sends all collected member node state change information to the valid child nodes in the state table to help the associated SP update the routing table. The unit format of the message is <PID, state, PID.Successor>, where PID represents the node number of the state change, state represents the result of the change, and PID.Successor represents the successor node of the PID. Each message contains a number of elements in the format, indicating the set of nodes whose state has changed and their changes. The principle is shown in Figure 2.



PID	state	IP	User
0	00		
5	01		
6	00		
8	10		
10	00		

Start	Interval	succe
6	[6,7}	8
7	[7,9}	8
9	[9,13}	10
13	[13,5}	13

Figure 2. HMRM algorithm mechanism.

The SP node periodically checks the predecessor node and the successor node and sends a message to notify the MP node of the status change. When a failure is found, the PID of the node is notified to the MP to modify the node status table of the MP. When each SP receives the MP's node status change message, it checks every R(i) of the routing table to determine if it needs to be changed. For the new node PID, check the R(i) entry value corresponding to the PID. If it is greater than the PID, replace the old value with the PID. For the R(i) entry value corresponding to the PID of the failed node, if it is equal to PID, the PID's successor node updates the old value. The HMRM information maintenance policy reduces the maintenance overhead of each message forwarding from the original O (lg2 N) to O (k), which greatly reduces the number of times the user node maintains the maintenance message on the overlay network.

For example, assume that the node PID=8 exits the system and its successor node PID=10 detects that the node 8 has failed. Then node 10 will report to the management node MID=56. After the MP receives the message, it will update the state value of the node 8 to ten. And if the node status does not change in the next refresh cycle, the status value becomes 11. And it is transferred to the back of the MP status table. In the next refresh period, the MP broadcasts the group node status change message to all storage nodes on the node state table. After comparing the routing table, node 6 updates the successor containing node 8 entries to 10, and repairs the routing table error caused by the failure of node 8.

4 Evaluation

The failure recovery strategy of the HMRM system topology is to periodically send messages to perform fault node detection and routing table update, and to measure the overhead required for topology maintenance by the total number of required messages. We use the method of paper[9] to establish the analysis model for the maintenance cost of chord-like system and HMRM system according to the behavior of online nodes in the system. And to evaluate the traffic that is generated in the analyzed overlay architectures we make the following definitions

Table 2. The Parameter used in the analysis.

p	Number of peers per group
n	Total number of peers in the system
Ts	Peer runs the node failure detection procedure periodically every Ts seconds.
Tf	Peer runs the routing table update procedure periodically every Tf seconds

The total number of messages required for topology maintenance falls into two categories: node failure detection and routing table updates. We first analyze the number of messages required for node failure detection. The node needs three messages to execute the failure detection procedure. The initiating node sends a REQUEST message to the predecessor node to determine whether it is valid. And the predecessor responds with a RESPONSE message RESPONSE message indicates that the response is valid. Then the initiating node responds to the ACK. confirm. Here, we use Ms1 and MS2 respectively to represent the number of messages required by the Chord-like and HMRM system nodes to perform node failure detection procedures.

$$M_{s1} = M_{S2} = \frac{3}{T_s}$$

Then, we analyze the number of messages needed to update the routing table. In the Chord-like protocol, each node has a logn entry in the routing table, and the update requires an overhead of log2n/2. We use Mf1 to represent the cost of routing table updates.

$$M_{f1} = \frac{\log^2 n}{2T_f}$$

In the HMRM algorithm, the management node MP update overhead is 2 logn, and the storage node SP periodicity and the MP switching node status require four messages: the SP sends the pre-dweller node status message to the MP and receives acknowledgement, and the MP sends the node status of this sub-group. Change information to all nodes and receive confirmation. We use Mf2_np and Mf2_mp to represent the storage node and management node overhead, respectively.

$$M_{f2\_np} = \frac{4}{T_f}$$

$$M_{f2\_mp} = \frac{2\log\frac{n}{p}}{T_f}$$

Finally, the total system traffic caused by topology maintenance is the sum of the messages generated by each node. Here, we use M1 and M2 to represent the system maintenance traffic of Chord-like system and HMRM system, respectively.

$$M_1 = \sum_{all\_peer} M_{s1} + \sum_{all\_peer} M_{f1} = n\left(\frac{3}{T_s} + \frac{\log^2 n}{T_f}\right) \quad (1)$$

$$M_2 = \sum_{normal\_peer} M_{f2\_np} + \sum_{management\_peer} M_{f2\_mp} + \sum_{all\_peer} M_{s2}$$

$$= n\frac{3}{T_s} + \left(n - \frac{n}{p}\right)\frac{4}{T_f} + \frac{n}{p}\frac{2\log\frac{n}{p}}{T_f}$$

$$= n\frac{3}{T_s} + n\frac{4}{T_f} + \frac{n}{p}\frac{2\log\frac{n}{p} - 4}{T_f} \quad (2)$$

Using Equations 1 and 2, we calculated the total network traffic per Ts of the foregoing two system, respectively. Here, we assume that Ts = Tf and p=10, 100, 200 and use M1, M2, M3 and M4 to represent the system maintenance traffic of Chord-like, p=10, 100, 200, respectively. The results are shown in Figure 3. The figures show that the total network traffic in M2, M3 and M4 slightly grow as the number of nodes increases, which presents a great improvement in the total network traffic. What is more, the total network traffic for M4 is smaller than M3 and M2, which shows that HMRM does take advantages of more powerful and stable management peers.

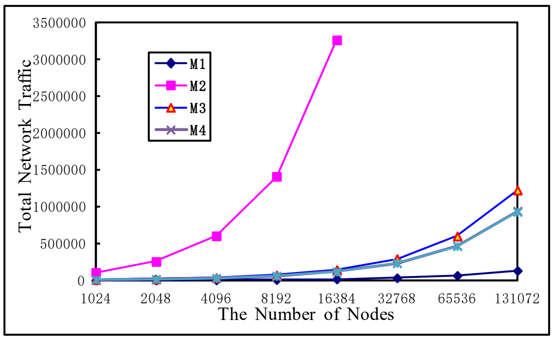


Figure 3. Topology maintenance overhead

Compared with the traditional DHT topology maintenance mechanism, the node MP of the HMRM algorithm bears more system maintenance load. Therefore, it should also consider the ratio of the maintenance load of the node MP in the total load and the ratio of the MP number of the node in the total node. So here we define the load ratio of the management node, denoted by R_{MP}.

$$R_{MP} = \left(\left(\frac{n}{p}\frac{3}{T_s} + \left(n - \frac{n}{p}\right)\frac{2}{T_f} + \frac{n}{p}\frac{2\log\frac{n}{p}}{T_f}\right)\Big/M_2\right)\Big/\frac{\frac{n}{p}}{n}$$

$$= \left(\frac{n}{p}\frac{3}{T_s} + \left(n - \frac{n}{p}\right)\frac{2}{T_f} + \frac{n}{p}\frac{2\log\frac{n}{p}}{T_f}\right)p\Big/M_2$$

$$= \left(\frac{3n}{T_s} + (np - n)\frac{2}{T_f} + \frac{2n\log\frac{n}{p}}{T_f}\right)\Big/M_2 \quad (3)$$

Figure 4 shows the management node load ratio of the HMRM algorithm. It is not difficult to see that the relative load of the management nodes of the HMRM algorithm is heavy, because the management nodes need to manage and send state information of the nodes in the cluster. And the load ratio reaches the maximum in a certain interval. When the size of the cluster is extended, the load ratio decreases gradually and the overhead of the management node becomes smaller, which means that the HMRM algorithm adapts to the hierarchical P2P network with the large-scale network storage system.

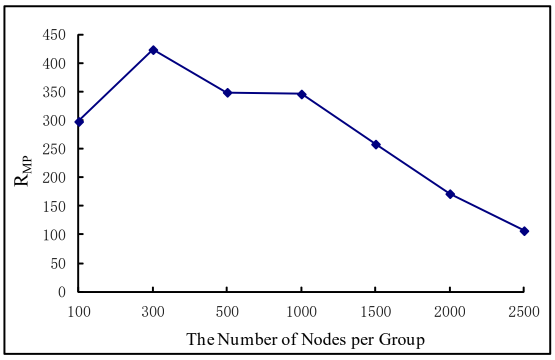


Figure 4. management node load ratio in HMRM.

5 Conclusions

In this paper, we presented a two-layer structure HMRM to reduce maintenance costs in network storage system. The main idea of HMRM is to make full use of the heterogeneity and management nodes which are more powerful and stable to achieve better updating efficiency. The analysis results also show that HMRM reduces total network updating load dramatically, therefore, improves system scalability generally.

Acknowledgements

This work is supported by Science and technology research project of Jiangxi Province Education Department under Grant No. GJJ14771 and Research on Dynamic Forensics Mechanism in Cloud Storeage Environment.

This work is supported by Science and technology research project of Jiangxi Province Education Department under Grant No. GJJ12765 and Research on Data Layout Technologies for Power-efficient Video Surveillance Storage

This work is supported by the research topic of teaching reform in Jiangxi higher schools under Grant No. JXJG-16-19-4 and construction and evaluation of collaborative training mode for network security and law enforcement major

This work is Supported by the Opening Project of Collaborative Innovation Center for Economics crime investigation and prevention technology, Jiangxi Province under Grant No. JXJZXTCX-006 and Research on Key technologies of distributed cooperative economic crime detection system

References

1. Jiachen Yang, Shudong He, Multimedia cloud transmission and storage system based on internet of things, Multimedia Tools and Applications, volume 76, Issue 17, pp.17735–17750 (2017).
2. Xu Cheng , Coordinate Live Streaming and Storage Sharing for Social Media Content Distribution, IEEE Transactions on Multimedia, volume 14, Issue 6, pp.1558-1565 (2012).
3. Xianfu Meng, A churn-aware durable data storage scheme in hybrid P2P networks, The Journal of Supercomputing, volume 74, Issue 1, pp.183-204 (2018).
4. L. Avinash and M. Prashant, Cassandra - A Decentralized Structured Storage System, *3rd ACM SIGOPS International Workshop on Large Scale Distributed Systems and Middleware*, pp.81-89 (2009).
5. YahyaHassanzadeh-Nazarabadi, Decentralized and locality aware replication method for DHT-based P2P storage systems, Future Generation Computer Systems, volume 84, pp.32-46 (2018).
6. Luo HF, Deng L, Research on a P2P supper node selection mechanism based on trust model. In: *The 8th International Conference on Computer Science and Education*, pp.851–854 (2013).
7. H. Shen, Z. Xu, Hash-based proximity clustering for efficient load balancing in heterogeneous DHT networks , Parallel distributed Compute, volume 68, pp.686-702 (2008).
8. Stoica I, Morris R et al, Chord: a scalable peer-to-peer lookup protocol for internet applications. IEEE/ACM Transactions on Network, 11(1), pp.17-32 (2003).
9. S. Zoels, W. Kellerer, Z. Despotovic, On hierarchical DHT systems-An analytical approach for optimal designs, Computer Communications, volume 31, Issue 3, pp.576-590 (2008).