The Study of Data Copies in Cloud Environment

ZONG Zhong
Jiangsu Transportation Institute Group, Nanjing 210017
zongzhong@gmail.com

Keywords: Data Copy, Redundancy, Overhead, Cloud Environment

Abstract. The redundancy of data copies is the key problem of replica management for the storage of copies of data in cloud environment. This paper proposes a tactic of dynamic copies redundancy based on the heat of document. The heat is calculated by the access frequency in different period. The redundancy of document copies is based on the heat. It is also dynamic changing by the heat. This paper analyses the max and minimum number of copies, and proposes to measure the minimum number of copies by the reliability of users, to calculate the max number of copies by the consistency maintenance overhead of the system.

Introduction

Through cluster applications, grid technology and distributed file system, cloud storage organizes different types of storage devices together in network and provides data storage and business access [1]. When the core business of cloud computing system is big data storage and management, the cloud computing system reverses to a cloud storage system. As a result, the cloud storage system is a kind of cloud computing system which is focus on the data storage and management [2].

For the security issue, the cloud storage uses not only the encryption algorithm but also the data redundancy. The cloud storage system makes several copies for one piece of data. Each copy is stored on different devices to avoid data unavailable during the downtime of one device. This method of storage strategy guarantees the accessibility and security of data.

The diversity variability of cloud storage leads the quantity of each copy in the system should be unique and dynamic changing. The redundancy of copies should be changed based on the access of users. This could ensure the reliability and security of the data, and also reduce the delay of user access and balance the load of system.

The default policy of HDFS for the creation of copy

When data is stored on the cloud not local, the data is controlled by the service provider [5]. Hadoop Distributed File System (HDFS) uses full backup policy [6]. When data is damaged or lost, the full backup data can be used directly from other copies by the recovery mechanism. By default, HDFS creates three copies for each file, and each copy is stored separated.

The policy of HDFS copy satisfies the security and efficiency issue of system [7]. Because of the diversity of each file and the difference of access frequency, the same redundancy policy of each file is not an optimized solution. It not only hides the heat of files but also wastes storage resource. To avoid this insufficient, this paper provide a tactic which changes the quantity of copies dynamically based on the heat and store location to achieve the easy-access target and to enhance the security. This policy also can not only realize the reducing of waste copies, to save the storage resources, but also improve system performance.

The section headings are in boldface capital and lowercase letters. Second level headings are typed as part of the succeeding paragraph (like the subsection heading of this paragraph).

File-based dynamic heat redundancy policy

This paper uses the heat to evaluate the popularity of the file. The more popular file is, the more copies it should have. For the location of copies, in principle it is the nearest to achieving the fast-access. This policy would reduce the network occupation, improve the system performance and decrease the quantity of copies. This policy is not only ensuring the security of data, but also reducing the redundancy.

(1) The calculation method of file heat

The system need to create more copies for popular files. To add up the popularity of file, this paper proposer the heat attribute of file.

HDFS uses the centralized management. The requirement of client should pass the Namenode first. Because of this Namenode can provide the access of each file. If a monitoring process is set to calculate the access of each file on the Namenode in different times, the weight could be set based on the access frequency in different periods.

The heat of file is the weighted average access frequency in each cycle. It is represented by File_Ht. From the moment of created, the start of the file life cycle, each file is monitored by the process to calculate the access. The heat of file is calculated by access frequency in its life cycle. There are two factors which influence the heat value of the file. One is the heat value of the file in last cycle. The other is the access frequency in this cycle. The formula is as follow:

$$File\_Ht'=\alpha File\_Ht+\beta\ Ff \qquad (1)$$

In formula 1, File_Ht' is the heat of the file in the end of this cycle. File_Ht is the heat of the file in last cycle. The File_Ht is access frequency of the file of this cycle. α factor is refer to the previous access. The access is called β factor in cycle. α and β are greater than 0 and $\alpha+\beta=1$. If α is close to 1, it means system concentrates on the total access of files, and the influence of access frequency to the new file heat is little in the cycle. If α is close to 0, it means system concentrates on the recently access of files and the influence of access frequency to the new file heat is great in the cycle. The value of α and β could be set according to the system flexibly.

Assume $File\_Ht_i$ is the heat of file at the end of cycle i. The Ff_i refers to the access frequency of the cycle i. The relation between Ff_i and Ff_i is as follow:

$$File\_Ht_n=\alpha File\_Ht_{(n-1)}+\beta\ Ff_n\ (n>0) \qquad (2)$$

The file heat of n-1 cycle is $File\_Ht_{(n-1)}=\alpha File\_Ht_{(n-2)}+\beta Ff_{(n-1)}$. Bring it into formula 2 can get the formula 3 as below:

$$File\_Ht_n=\alpha^2\ File\_Ht_{(n-2)}+\alpha\beta Ff_{(n-1)}+\beta\ Ff_n \qquad (3)$$

Based on the formula 3, the formula 4 could be get:

$$File\_Ht_n=\alpha^n File\_Ht_0+\sum_{i=0}^{n-1}\alpha^i\beta * Ff_{n-i} \qquad (4)$$

File_Ht0 represents the initial heat when the file is created. This paper considers the initial heat value to be 0. The formula 4 can be simplified as follow:

$$File\_Ht_n=\alpha^n File\_Ht_0+\sum_{i=0}^{n-1}\alpha^i\beta * Ff_{n-i} \qquad (5)$$

Assume the number of times a file d accessed in n cycles is N1，N2，…，Nn . The heat of the file at the end of n cycle would be as:

$$File\_Ht_n=\frac{\beta}{T}\sum_{i=0}^{n-1}\alpha^i N_{n-i} \qquad (6)$$

Assuming the cloud storage system has m copies of the file, the Namenode distribute all user requirement to each copy as equal as possible. The heat of each copy could be calculated as:

$$File\_Ht_{single}=File\_Ht/m \qquad (7)$$

Cloud storage system stores a huge number of objects. Through formula 6 and formula 7, the real time access heat of file can be calculated. Although this calculation would cost system resource, the dynamic heat value of heat would better represent the access of users and would help cloud storage system to better appropriate response.

(2) Copy creation process

The time of creating the copy under the dynamic mode is very important. This article set an appropriate value for the heat $File\_Ht_{single}$ of file copy, called the copy increasing value and the copy decreasing value. When the $File\_Ht_{single}$ value is over the copy increasing value, it means each copy carries too much access requirement. This may cause the network bottlenecks in the system. Namenode can start the copy increasing order as the administrator. It would choose the next datanode of copy to write data based on the copy layout policy. When work is finished, the Datanode will feedback the Namenode. Namenode will change the number of copies and recalculate the heat value of each copy until the heat value is not bigger than the copy increasing value or at the max of number of copies.

The copy deleting process is similar to the creating process. When the value File_Htsingle is lower than the copy decreasing value, it means there is too less access requirement. To avoid the waste of system resource, copies should be deleted to save system resource. In this situation, the Namenode will start the copy deleting process. It finds the last created copy and deletes. When the deleting work is finished, Datanode will feedback to the Namenode. The Namenode make the number of copies of the file minus 1. It also recalculates the heat of each copy, until the value is not less than the copy decreasing value or reaches the minimum number of copies. The process is shown in Fig. 1.

错误!不能通过编辑域代码创建对象。

Fig. 1 The creating and deleting of copies

(3) The number range of copy

To set the minimum number of copy need to ensure the reliability and validity of the file. The minimum number is set based on the reliability of users' requirement. The calculation of minimum number is as follow:

Assuming the user's reliability requirement is Avail, the value 0.9 means the successful access of users is 90%. To achieve the reliability requirement of users, it is need to create minimum number of copies for the object at least.

F is the failure rate of storage node. Here assuming when the node is available, the copy of file is also usable. F is also called the failure of single copy. The rate of one of copies failure is as follows.

$$1\text{-}F^{min} \tag{8}$$

Considering the environment failure rate RE caused the other uncertain factors, such as network congestion, data package lost, and so on, the rate of file successful accessed can be represented as below:

$$P= (1\text{-}RE)*(1\text{-}F^{min}) \tag{9}$$

The user's reliability requirement Avail and the minimum number of copies have the approximate relationship as follows:

$$(1\text{-}RE)(1\text{-}F^{min})\geq Avail \tag{10}$$

In certain cloud storage system under special network, the RE and F can be easily calculated. The Avail is set by users and the minimum number of copies can be calculated. Assuming the RE is 5%, F = 30%, Avail = 80%, through formula 3.9, the minimum is 1.533, rounding up to give 2. If one file is more important than others, the Avail is 90%, then the minimum is 2.445, rounding up to give 3.

To ensure the maximum number of copies is to control the consistency maintenance cost of copies. The bigger the maximum value is, the more the consistency maintenance cost. With the augmentation of the cost, the performance of system would decrease. This is not the purpose of the copy mechanism.

The cost of the consistency maintenance is the data flow which is caused by the renovation of copies. This paper proposes to measure the cost of the consistency maintenance by data flow caused by the renovation of copy in a unit of time. Assuming the maximum number of copy is max, the renovation frequency of the copy is p, the size of new copy is S. the data flow, L, which is used for the renovation is as bellow:

$$L=S*max \tag{11}$$

Considering the frequency of user's renovation, the data flow in a unit time for the consistency maintenance is:

$$L=S*max*p \tag{12}$$

The cost of consistency maintenance and the number of copies, max, have the linear relationship. The more it cost, the more copies it has. Obviously there is the limitation of the number of copies. If the allowed cost is R, then:

$$R\geq S*max*p \tag{13}$$

It is said that the limitation of copies is:

$$\max \leqslant R/(S*p) \tag{14}$$

The simulation and performance analysis

Build the Hadoop platform [8] and modify the HDFS to adjust the simulation. The all platform is organized by five computers. One is used for the Namenode. The other fours are used for Datanode. The operation system is Redhat. The java environment is jdk1.7-Linux. The version of Hadoop is Hadoop 0.21.

To compare the original policy of HDFS and the heat based policy proposed by this article, the chosen file f uses these two policies one by one. All parameters are set as: MIN=2，MAX=6，α=0.5，β=0.5. The original number of copies of HDFS is three. In the simulation, users of cloud storage access to file on the system randomly. To show the dynamic of the number of copies, the file access frequency is increasing gradually. Comparing the average number of copies in different access frequency and the expect result the performance of system in different policy is shown.

The Fig. 2 shows the trend that the change of number of copies when the access frequency is increasing. The experiment reveals that the more often file is accessed, the more copies the system makes for the file.

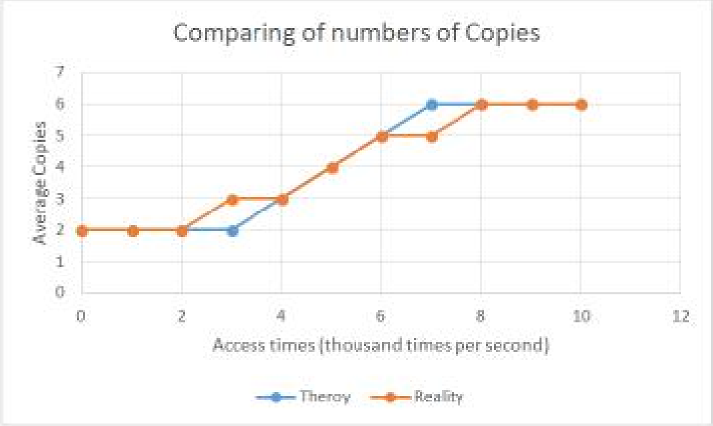

Fig. 2 The comparison of number of copies

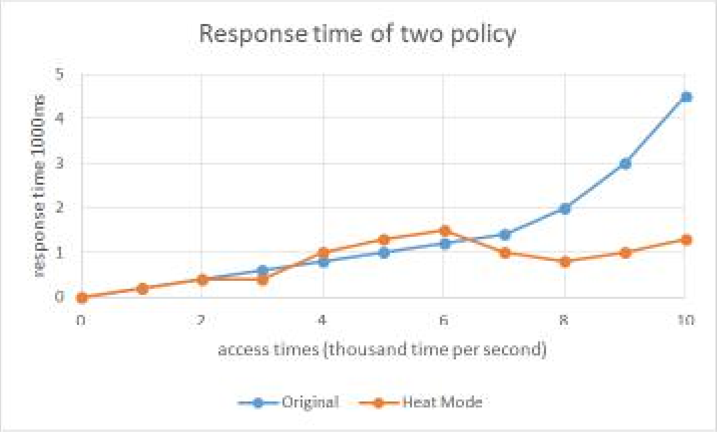

Fig.3 The response time of two policy

Fig. 3 shows the system response time in two different policies. In Fig. 3, it is seen that the more operations, the longer the system response takes. Under the original policy, when the access frequency is low, the response time increases slowly. When the access frequency is getting high, the response time dramatically increase. Under the heat policy, the augmentation of the response time increases gradually, reaching the peak and down. When the access frequency increasing, the heat of file is augmenting meanwhile. When it reaches the preset value, the system starts to create copies. In the beginning, it takes time and resource to create copies, so the system response becomes slow even slower than the original policy. But when copies are ready, the response cost falls back. This experiment shows that this heat policy is do more efficient than the original policy.

Conclusions

The redundancy of data copies is the key problem of copy management. This paper proposes a policy based on the heat of file. This policy measures the heat of file to show the popularity of the file. It gives the calculation of the heat. When the heat value is bigger than the default value, the number of copy augments. Otherwise the number of copy decreases. This paper also defines the process of copy creating, defines the number of copy and proposes the method of calculating the minimum value and maximum value based on the users' reliability requirement and the consistency maintenance cost.

References

[1] ZHOU Ke, WANG Hua, LI Chunhua, The application of cloud storage technology [J]. ZTE Communication Technology, 2010, 16(4):25-27.
[2] what is the Cloud storage [EB/OL] http://www.dqjsw.com.cn/wulianwang/102539.html. 2011.12
[3] LOHR S. Google and IBM join in 'cloud computing' research [OL]. New York Times. http://www.csun.edu/pubrels/clips/Oct07/10-08-07E.pdf. 2012.
[4] CAI Lin, Large-scale distributed storage system copies Management and Implementation [D]. Chengdu University of Electronic Science and Technology of China 2008
[5] GUO Dong, DU Yong, HU Liang, HDFS cloud-based data backup system [J]. Journal of Jilin University (Science Edition). 2012(1):101-105.
[6] Xuhui Liu, Jizhong Han, Yunqin Zhong, etal. Implementing WebGIS on Hadoop: A case study of improving small file I/O performance on HDFS [C]. Proceedings of the IEEE International Conference on Cluster Computing and Workshops (CLUSTER '09), Beijing, China. 2009:1-8.
[7] Mayur R. Palankar, Adriana Iamnitchi, Matei Ripeanu, etal. Amazon S3 for science grids: a viable solution? [C]. Proceedings of the 2008 international workshop on Data-aware distributed computing (DADC '08). New York, NY, USA. 2008:55-64.
[8] ZENG Datan, ZHOU Aoying Translation, Hadoop:the Definitive Guide [M] Beijing: Tsinghua University Press 2010.