Advances in Intelligent Systems Research, volume 163
8th International Conference on Management, Education and Information (MEICI 2018)

An Adaptive Fault-Tolerant Strategy in Library Cloud Storage System

Meilin Zeng[1] and Qiangqiang Xiong[2]

[1.]Jiangxi Vocational Technical College of Industry Trade Nanchang Jiangxi Province China 330038

[2.]Nanchang institute of technology Nanchang Jiangxi Province China 330013

Keywords: Cloud library; Cloud storage; Fault-tolerant; Copy replication; Erasure code; Adaptive switching

Abstract. The advantage of cloud storage is to attract a large number of libraries to use cloud platforms to store collections of literature resources. In order to avoid the increase of failure probability of cloud storage nodes caused by the increase of data volume, the design of fault-tolerant technology is particularly important. Due to the advantages and disadvantages of common fault-tolerant technologies, this paper proposes an adaptive switching fault-tolerant strategy, which can switch between copy replication fault-tolerant technology and erasure code fault-tolerant technology on the basis of literature access frequency and literature size. The experimental results show that the proposed scheme saves 43% storage space compared with the copy replication fault-tolerant technology, and improves 52% data recovery time compared with the erasure code fault-tolerant technology.

Introduction

Cloud storage has more advantages, and a large number of libraries build cloud platforms to store the massive digital resources of the library collection, so the traditional storage methods are gradually replaced[4,5]. However, there are thousands of common storage nodes in the cloud storage system, so the risk of data failure is very high. In order to avoid the immeasurable loss caused by data failure to some precious collection literature, the data fault-tolerant design of cloud storage system is extremely important. A good fault-tolerant design can ensure the efficiency of literature access and realize the rapid recovery of data after failure.

There are two commonly used data fault-tolerant technologies: one is the copy replication technology. It is simple to implement, but the storage space needed is relatively large. The other is the erasure code technology. Only the redundant data is needed to be stored, so the advantage of this technology is that the storage space needed is relatively small, but the data recovery needs to be implemented by decoding, and the implementation is more complex[8,9]. The above fault-tolerant technologies have their own advantages and disadvantages. It is difficult to meet the needs of different types of users at the same time by using only one fault-tolerant technology. Therefore, the adaptive switching fault-tolerant technology has more practical value.

The adaptive switching fault-tolerant algorithm proposed in this paper takes the resource access frequency and literature size as the basis to carry out adaptive switching between copy replication fault-tolerant technology and erasure code fault-tolerant technology. The application object of the copy replication fault-tolerant technology is the high access literature, which can realize efficient access and data recovery. The application object of the erasure code fault-tolerant technology is the low access literature, so as to reduce the occupation of storage space. By building a fault-tolerant framework in the library cloud platform, this paper verifies the fault-tolerant performance of the adaptive switching algorithm.

Adaptive Switching Fault-Tolerant Algorithm in Library Cloud Storage System

Due to the large difference in the access frequency of different files in the library cloud storage system, taking the access frequency as the basis of the files to select the fault-tolerant strategies can achieve better overall fault-tolerant performance. Usually, the proportion of files with high

frequency of access is not high. For these files, copy replication technology can be used for fault tolerance, resulting in little increase in storage capacity, while providing users with higher access timeliness. However, the remaining large number of files have the characteristics of low access frequency, so it is appropriate to adopt erasure code technology for fault-tolerant processing, so as to avoid the huge storage space requirements caused by copy replication technology. Since the access frequency of files in the system may change greatly over time, the fault-tolerant technology adopted by each file is not invariable, and the fault-tolerant strategy needs to be adaptively selected according to the change of access frequency.

Selection of Switching Methods.

The widely used switching algorithms in computer systems are of the following types: One is the FIFO, and the earlier the object gets in, the sooner it switches out. Its advantage is easy to achieve. The second is the LRU. It switches depending on the access time. The third is the LFU, and it switches depending on the access frequency[13,14].

The higher the frequency of files being accessed in the library cloud storage system, the higher the probability of file failure, and the most important factor influencing the user experience of using the system is the overall access quality, therefore, this paper chooses the switching algorithm taking the access frequency as the switching basis: LFU algorithm.

The following basic requirements should be met according to the switching of files in library cloud storage system:

(1) If an article of literature has a high access frequency in the early period and the access frequency in the near future decreases rapidly, but the short-term cumulative number of visits of other literature still does not exceed the cumulative number of visits of the literature, the literature can not be switched from the frequency table.

(2) If the number of visits to an article of literature is small in the early stage, the frequency of recent visits increases greatly, but the cumulative number of visits of the literature in the short term is still not more than that of other literature in the frequency table, the literature can not be switched into the frequency table.

In this paper, the LFU algorithm is used to divide the whole time period T into N time slices, which is recorded as $\{T_1 T_2 ... T_N\}$. The switching fault-tolerant strategy is based on the weight of access frequency within each time slice, so as to avoid the influence of local access frequency. In addition, the size of the literature will also affect the choice of fault-tolerant strategy. The larger literature occupies more storage nodes, so the risk of failure is also greater, resulting in increased access delay during the recovery period. Based on the above analysis, the adaptive switching algorithm in this paper considers both the literature size and the access frequency, and defines the literature with both large size and high access frequency as the high-access literature.

Adaptive Switching Description

Relevant Definitions

(1) Time slice division of life cycle T: $T = \bigcup_{i=1,2,\cdots N} \{T_i\}$;

(2) Literature: $D_\kappa = \langle S_k | F_{ki} \rangle$ S_k represents the size of the literature D_k, F_{ki} represents the access times of the literature D_k; ’

(3) Literature data set: $T = \bigcup_{i=1,2,\cdots N} \{D_k\}$, in which D_k represents the literature;

(4) High access literature data set: $Z = \bigcup_{j=1,2,\cdots N} \{Z_j\}$;

(5) Number of literature visits(Single time slice): $C_{ki} = F_{ki} - F_{ki-1}$;

(6) Switching threshold:w;

(7) Number of visits varied(Adjacent time slice): $R_{ki} = C_{ki} - C_{ki-1}$;

(8) Ascending set Z_{up}: $Z_{up} = \{D_k | R_{ki} < 0\}$, which is the subset of the high access literature data set;

(9) Descending set Z_{down}: $Z_{dowm} = \{D_k | R_{ki} < 0\}$, which is the subset of the high access literature data set.

Because of the locality of user access behavior, the probability of re-visiting the recently

visited literature is higher, so the statistics of access frequency AF can not be simply accumulated in a specific time range, but is the weighted accumulation after the recently access frequency is given a larger weight, as shown in Formula (1). The interval number is represented by N_T as the statistical time range:

$$AF\left(D_k\right)=\sum_{i=1}^{N_T}\left(C_{ki}\times 2^{-\left(N_T-i+1\right)}\right) \quad (1)$$

Since the data recovery time and normal access time of the literature are proportional to its size, The AF is multiplied by a weight value when calculating the final amount ofliterature access $A\left(D_k\right)$, as shown in Formula (2):

$$A\left(D_k\right)=S_k\times AF\left(D_k\right) \quad (2)$$

Switching Algorithm

When accessing the*k*-thliterature outside the high access scale, if the variable quantity in number of visits within the adjacent time slices $R_{ki}>0$, it indicates that the number of visits to the literature in the adjacent time slices increases, and the first judgement condition for switching the literature is satisfied. If the second judgement condition is satisfied at the same time, that is $A\left(D_k\right)>w$, switching can be made. That is to say, the necessary condition for switching is $R_{ki}>0$ and $A\left(D_k\right)>w$.

If the literature *k*meets the necessary conditions for switching, that is to say its $R_{ki}>0$ and $A\left(D_k\right)>w$, it is also necessary to judge whether the number of visits to the literature exceeds all the literature in the high access scale. In the process of comparison of the amount of the above literature access, in order to improve the speed of comparison, the literature in the high-access scale is divided into two parts according to the change trend of the amount of literature access: The first one is the ascending set Z_{up}. There is a upward trend in the amount of access to literature in this set. The second one is the descending set Z_{down}. There is a downward trend in the amount of access to literature in this set. Therefore, the literature *k* is first compared with the literature in Z_{down}, and then compared with the literature in Z_{up}. The specific algorithm flow is as follows:

(1) Search in the Z_{down} set to find the literature with minimum access amount, which is written as $D_{k''}$;

(2) If $A\left(D_k\right)>A\left(D_{k'}\right)$ is satisfied, $D_{k''}$ is removed from the set Z_{down}, and D_k is added to the set Z_{down} at the same time;

(3) If $A\left(D_k\right)<A\left(D_{k'}\right)$ is satisfied, search in the set Z_{up} to find the literature with minimum access amount, which is written as $D_{k''}$;

(4) If $A\left(D_k\right)-w>A\left(D_{k''}\right)$ is satisfied, $D_{k''}$ is removed from the set Z_{up}, and D_k is added to the set Z_{up} at the same time.

Fault-Tolerant Framework for Library Cloud Storage System

Fig.3 is an architectural diagram of adaptive switching fault-tolerant strategy in this paper. When the client reads the literature data, if there is a block failure, the fault-tolerant strategy is based on whether the literature is in the high-access list or not, and then the copy replication strategy is selected for data recovery, otherwise the erasure code strategy is selected for data recovery.

Metadata server is the core component of the whole architecture. To ensure reliability, its fault-tolerant strategy is implemented by hardware, and the hot standby mode is used.Two servers that work simultaneously are used to respond to client requests.Normally, the master server is responsible for the work, and the slave server is in a slave state. When the master server fails, the slave server is responsible for the work.

The metadata server includes two modules: one is the metadata management module, which is responsible for data block processing.The second is the fault-tolerant management module, which is responsible for fault-tolerant policy switching, and takes the amount of literature access as the basis for switching. If the amount of a document access reaches a set threshold and exceeds the amount of literature access contained in the high-access scale, the erasure code corresponding to this literature is deleted and it is copied in three copies, and the copies are save in the “complete copy” frame. If the amount of a document access decreases and is deleted from a high-access set, the data block containing its copy is deleted, and then the erasure code is generated and saved to the “erasure code” frame.

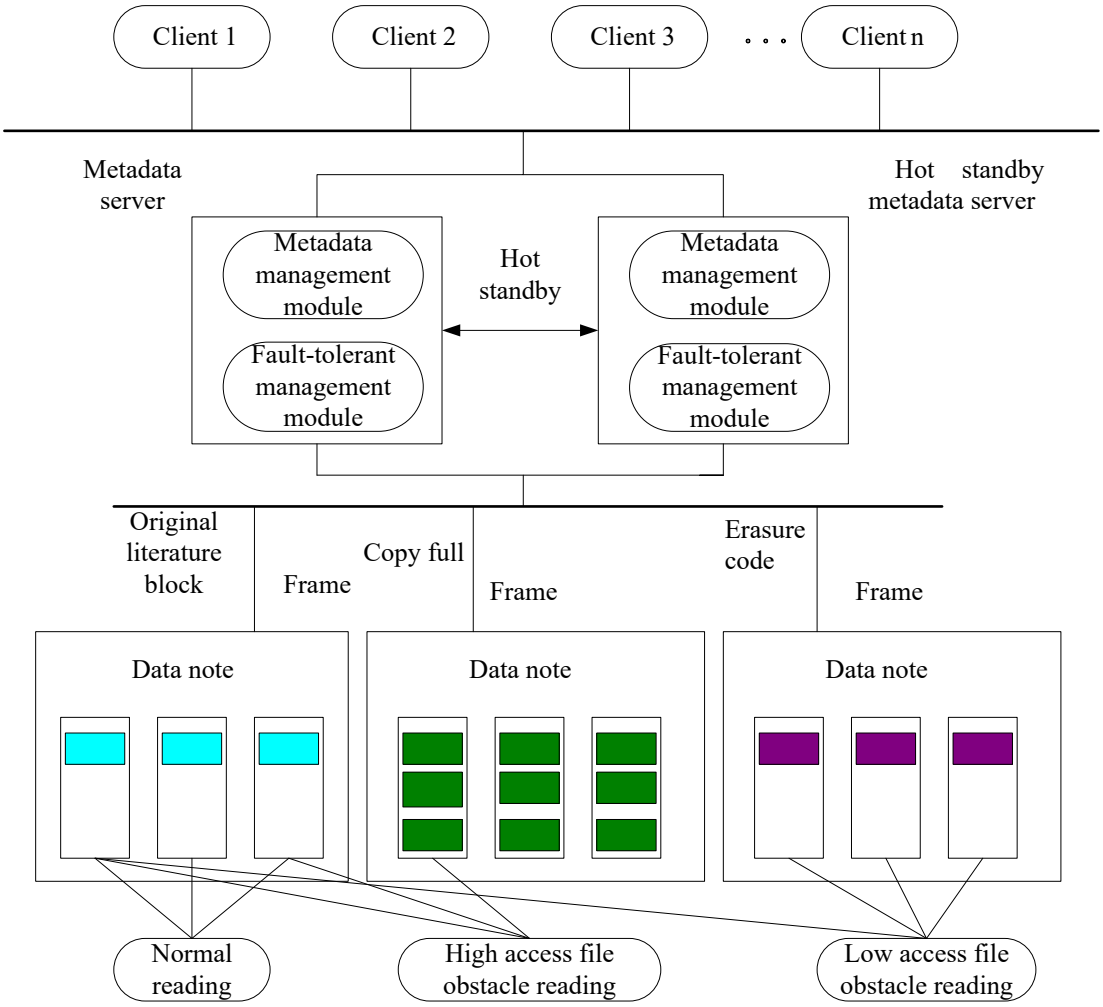

Figure 3.Cloud storage fault-tolerant architecture based on copy replication and erasure code adaptive switching

Experimental Performance Evaluation

In this paper, a fault-tolerant system was built on the private cloud platform of a college library to verify the feasibility of the scheme. The fault-tolerant system server had 2 named nodes, and 12 data nodes. The hardware configuration was as follows: The CPU model was Xeon E5-2609 v3 with a main frequency of 1.9GHz, the memory size was 4GB and the type was DDR4, the hard disk size was 1TB, and the network bandwidth was 100Mbps. Its operating system was ubuntu-11.10, the 6u30-linux-i586 version of JDK was installed and the hadoop cluster platform was configured.

In the test, the copy replication technology was implemented by three copies replication, and the erasure code technology was implemented by the standard RS coding of the R(n,k)=R(9,6). The total literature capacity was 300 GB, the high access scale capacity was 1000, and the access record time span was 3 months which were equally divided into 20 time slices. Through the Hadoop benchmark SWIM, the storage space occupied by different fault-tolerant strategies and the data

recovery time are obtained for comparative analysis. Testing benchmark program SWIM through hadoop, the storage space occupied by different fault-tolerant strategies and data recovery time were obtained for comparative analysis.

Comparative Analysis of Storage Space

When testing, the size of the data block was 64MB, and Fig.4 is a comparison of the storage space occupied by the three fault tolerance strategies. The storage space occupied by fully using the copy replication strategy was 900GB, reaching 3 times the total capacity of the literature. The storage space occupied by fully using the erasure code strategy R(9,6) was 450GB, reaching 1.5 times the total capacity of the literature. The average storage space occupied by the adaptive switching strategy in this paper was only about 43% of that when the copy replication strategy was completely adopted.

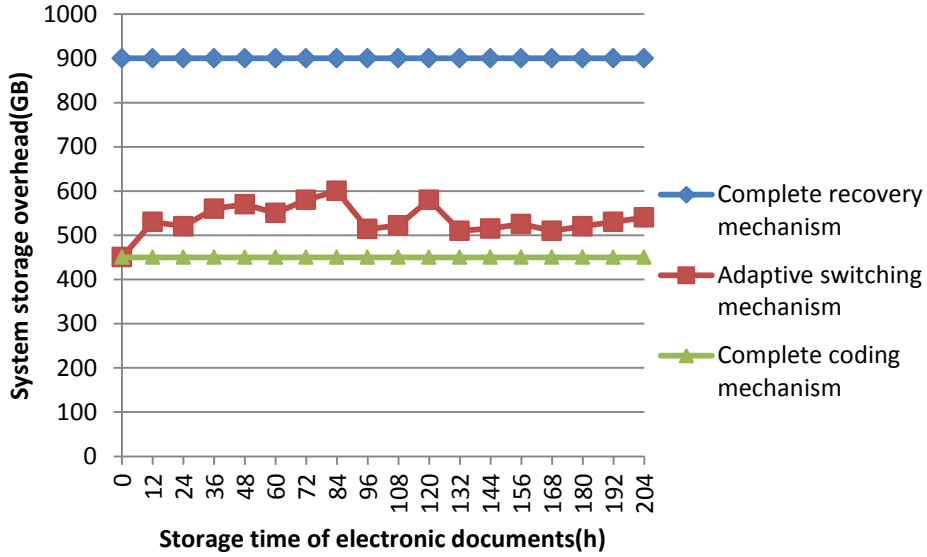

Figure 4. Comparative analysis of storage space

Fault-Tolerant Aging Comparative Analysis

This paper simulated two-node faults by stopping the power supply of the nodes. After the access records were rebuilt, 1000 records were selected randomly to simulate the access. Data blocks of various sizes were selected to calculate the average access delay of the three fault-tolerant strategies respectively. The comparative results are obtained in Fig.5.Completely using the copy replication strategy to recover data was to assign values directly to the data stored in the surviving nodes, and the average access delay was the shortest. Complete use of erasure code strategy required complex algebraic operations, but also read data from the number of nodes, and the average access delay was the longest. In this paper, the access delay of adaptive switching strategy was increased by about 52% than that of the erasure code strategy.

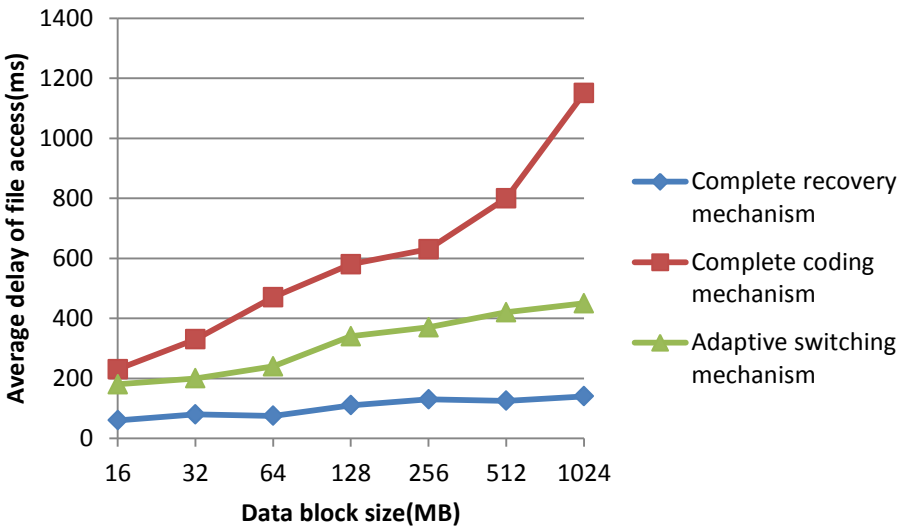

Figure 5.Comparative analysis of fault-tolerant aging

Conclusions and Prospects

It has become a popular trend for libraries to use cloud platforms to store electronic literature, so the fault tolerance of cloud storage systems becomes particularly important. This paper analyzes the advantages and disadvantages of two widely used fault-tolerant technologies, and puts forward an adaptive fault-tolerant strategy for library cloud storage system. With the literature size and access frequency as the switching basis, it takes up the storage space close to the erasure code technology, and its data recovery performance is close to the copy replication technology.

This paper describes an adaptive switching fault-tolerant framework in detail and tests it on a library private cloud platform. Experiments show that although the scheme has a slightly higher CPU occupancy rate and memory occupancy rate, it only needs to use 43% of the total storage space by copy replication technology, and its average access delay is 52% higher than that by erasure code technology, thus achieving a good balance between storage cost and repair performance. The next step of this paper is to give the corresponding weight to the importance of the literature, and to optimize the performance of the switching algorithm.

References

[1] Wenbin Yao, Si Han and Xiaoyong Li: Secure Sharing Mechanism for Ciphertext in Cloud Storage Environment[J], Journal on Communications,2015,36(10):1-8.

[2] Yan Yuan: Security Analysis of Private Cloud Storage in University Library[J], A Vast View on Publishing, 2015(12): 46-47.

[3] Jiping Jiang and Zhenli Yuan: Analysis of the Current Situation of Library Cloud Storage in China[J], Library and Information Studies, 2017(1): 59-64.

[4] Du S, Li C, Mao X, et al. The Optimization of LRU Algorithm Based on Pre-Selection andCache Prefetching of Files in Hybrid Cloud[C]. International Conference on Parallel andDistributed Computing, Applications and Technologies. IEEE, 2017:125-132.

[5] DU Z, HU J，and CHEN Y, et al. Optimized Qos-aware replica placement heuristics andapplications in astronomy data grid[J]. Journal of Systems and Software, 2011, 84(7): 1224-1232.

[6] CHEN Y, GANAPATHI A, GRIFFITH R, et al. The case for evaluating mapReduceperformance using workload suites[C]. 19th Annual IEEE International Symposium on Modelling,Singapore, 2011:390-399.