Preservation Mechanism of Electronic Record Based on Erasure Code and Multi Copies in Cloud Storage

Yongjun Ren[1,2](✉), Lin Zhou[1,2], Yepeng Liu[1,2], and Xiaorui Zhang[1,2]

[1] Jiangsu Collaborative Innovation Center of Atmospheric Environment
and Equipment Technology (CICAEET), Nanjing University of Information
Science and Technology, Nanjing 210044, China
renyj100@126.com
[2] School of Computer and Software,
Nanjing University of Information Science and Technology, Nanjing, China

Abstract. With the rapid growth of cloud storage center, the cumulative volume of data reaches EB and even ZB from PB. As a result, both network size and the number of storage nodes continue to grow explosively, while the data failure rate is still increasing. Cloud storage centers encode the raw data into erasure codes, to save the system overhead as much as possible meanwhile guarantee the reliability of data. However, the state-of-art erasure codes techniques still rely on a conventional centralized model which results in unaffordable encoding/decoding cost, and thus cannot adapt to the data-intensive processing requirements for distributed cloud storage environments. In the paper, the preservation mechanism of combining erasure code and copy backup is proposed, to improve the reliability of electronic records in cloud storage. This paper focuses on the erasure code archiving of electronic documents and puts forward the ability aware erasure code filing of electronic documents. Moreover, the corresponding implementation algorithm and steps are described.

Keywords: Cloud storage · Electronic record · Multi copies

1 Introduction

With the acceleration of information digitization and network service, digital resources have developed rapidly, and extensive electronic records have been produced. The acquisition, use, and sharing of electronic documents have unique advantages over traditional documentary resources, making them play an increasingly important role in people's lives, learning, and work. However, people have found that the long-term preservation of electronic records is very tricky. First of all, electronic documents are stored on a physical carrier in the form of digital codes, mainly based on light, electricity, and magnetism. The transport of these materials has a very high requirement for the storage environment. The effects of high temperature, humidity, and magnetic fields all contribute to the loss of information, and the longevity of these carriers is far shorter than that of the traditional carrier papers, which is hundreds of years. The magnetic carrier storage time is about ten years, and the storage time of the optical disk is 10–20

X. Sun et al. (Eds.): ICCCS 2018, LNCS 11065, pp. 174–183, 2018.
https://doi.org/10.1007/978-3-030-00012-7_16

years [1–3]. Second, the reading of electronic records depends on computer software. As technology continues to evolve, operating systems and software upgrades will no longer support traditional record formats, which will lead to embarrassing situations in which electronic records cannot be used. Electronic files stored in computers are bound to be subject to security threats. Nowadays, electronic records have reached a tremendous amount and are continuing to develop, which makes it impossible to estimate the cost of saving electronic records [4]. Moreover, the preservation cost of electronic records not only includes the physical space and environmental control costs required for the preservation of traditional paper records but also relates to other expenses necessary to ensure the reproduction of electronic files, such as technical updating and digital migration. Also, the long-term preservation of electronic records still lacks public standards and legal difficulties [5, 6].

Long-term preservation of electronic records can only be guaranteed by a secure storage environment to ensure the long-term validity and availability of records. The proposal of cloud storage provides a new possibility for the long-term preservation of electronic records. Compared with traditional preservation strategies, cloud storage has a reliable storage architecture, perfect backup measures, and an efficient migration mechanism that reduces initial investment, saves management costs, reduces maintenance expenses. So cloud storage is suitable for large-scale digital storage, providing a one-to-one portable service which can efficiently guarantee the continuity of service, and the speed of website access response is fast. It is a better way for current electronic records to be stored for a long time [7, 8].

However, under the cloud storage environment, electronic records are stored in the cloud server for a long time. The electronic record is completed under the control of the cloud server. The record manager entirely loses the physical control of the electronic record. Once a problem occurs with the cloud service provider, the organization may not be able to retrieve electronic records and electronic records. In 2014, the international CodeSpace cloud company was hacked. All the records in the Apache Subversion collection and Elastic Block collection on the company's cloud service platform were all permanently deleted, and the records could not be recovered. This situation is disastrous for electronic records and archives that require permanent preservation and retention as human history [9, 10]. With the aim of enhancing the reliability of electronic records, this paper proposes a combination of erasure codes and copy backup for electronic records stored in the cloud. Based on this, the technical implementation problem was studied.

2 Related Work

Currently, magnetic disks are the key and core of electronic record storage systems in both centralized and distributed electronic record storage systems. However, due to the limitations of the mechanical characteristics of the disk itself, although some researchers have proposed RAID (Redundant Arrays of Inexpensive Disks) technology, its reliability has not been substantially improved. However, in large-scale electronic record storage systems, disk failure or storage node failure has become a regular behavior. For example, in an electronic record center with a scale of approximately

4,000 nodes, an average of four disks will fail each day. Google researchers counted disk corruption in its electronic record center, and about 1.7% to 8.6% of disks in the system would fail each year [11]. According to statistics from Carnegie Mellon University, the annual replacement rate of the disk in some systems is about 13% [12]. Each year, dozens or even hundreds of disk corruptions are commonplace for a PB-class system consisting of tens of thousands of disks. For larger EB-class storage systems, tens or even tens of thousands of disks are destroyed every year. At the same time, the magnetic media on a portion of the surface of the disk platter is often damaged or occurs read and write errors, resulting in inaccessibility or loss of data on some sectors of the disk. Net App has made statistics on disk sector errors. Within 32 months, the proportion of such errors in the disk system with a size of 1.53 million reached 3.45% [13].

In addition to regular disk damage, the storage node's network card is damaged, and memory, CPU, and other hardware are damaged. Alternatively, the whole rack in the storage system is damaged due to the system power off, and the electronic records in the entire rack are temporarily unavailable. There are also unreliable electronic records due to system software errors. Kroll Ontrack conducted a systematic statistical study of the reasons for the loss of electronic records, of which system failures or hardware device damage caused approximately 56% of electronic record losses; about 26% was due to human-induced system failures. Software failures or virus intrusion cause the resulting loss of about 16% of electronic records; about 2% of electronic records are lost due to natural causes such as earthquakes and tsunamis. That is, one out of every 500 electronic record centers have an electronic record disaster [14].

On the one hand, the exploding storage capacity of electronic records has increased the demand for primary storage devices. On the one hand, it is the frequent failure of large-scale mass electronic record storage systems. On the other hand, the loss of electronic records to their owners and users is enormous. All of this makes the reliability of the electronic record storage system an critical challenge.

Increasing redundancy is a standard way to realize the reliability of electronic records. When an electronic record partially fails, customers can satisfy their own needs by accessing redundant data. Under the distributed storage environment such as GFS, HDFS and Amazon S3, three-replica redundancy is used, which can well meet the reliability of electronic records and load balancing requirements. The original intention of the three-copy strategy adopted by GFS/HDFS is to ensure that no more electronic record lost under the condition that keeps the node hardware performance [15].

The electronic record storage cluster mainly consists of the following components: cluster manager nodes, access nodes, and storage nodes. The cluster management node is responsible for the metadata information in the system such as the configuration of the cluster and the system's namespace. When a block of electronic records in a cluster needs to undergo a block redundancy change, the cluster management node also concurrently manages the work of encoding record blocks. The visiting node is mainly responsible for responding to the I/O access request sent by the user. After the user request arrives at the access node, the access node first interacts with the management node to obtain the state information of the accessed record block and the address information of the record block on the production node. Then returns corresponding information from the corresponding storage node according to the address information of the electronic record block to the user. The storage node is responsible for the actual storage of the data and saves the original content and the associated verification data. A large number of storage nodes are deployed on multiple racks and interconnected by switches. The three copies of the electronic record block are distributed in the cluster in a rack-aware, random layout, specifically, two copies are placed on two nodes in the same rack, and a third copy is placed on the other rack.

When the cluster size is small, the consumption of storage space of three copies is not particularly significant. However, in a large-scale application cluster, the utilization rate of nodes is often too high, and the needs of cost control cannot be entirely satisfied only by increasing the storage space of the nodes. More importantly, in the multi-copy mode, the cluster's scalability is limited due to the limitations of cluster metadata management [13–15].

The reliability enhancement technology based on multiple backups is intuitive and straightforward, easy to implement, and is the simplest and most widely used type of data redundancy mode in distributed storage systems. This strategy requires the sharing of multiple copies of the same electronic record to different storage nodes. Apparently, this strategy has a significant storage space overhead. Redundant electronic records are multiple copies of the original record. With the explosive growth of electronic records and the ever-increasing scale of storage devices, the management and operation of hardware devices will bring enormous costs. The choice of electronic record reliability preservation strategy needs to consider the record redundancy problem of the backup strategy, load balancing issues, and additional energy consumption issues.

3 Problem Statement

For the data reliability problem of storage systems, in recent years, scholars at home and abroad have conducted exploration and research and opened up a new storage path based on encoding redundancy strategy. Erasure codes are widely used in storage clusters, such as archiving systems, data centers, cloud storage, and so on. Among them, Solomon coding has become a typical data organization solution in fault-tolerant clusters because of its smooth operation and increased fault tolerance. Solomon coding guarantees data availability with extremely low storage overhead. Compared to data copies, erasure codes can provide equivalent fault tolerance with less storage overhead [14]. Most of the data is accessed for a short period throughout its life cycle. For example, more than 90% of data access in the Yahoo M45 Hadoop cluster occurs on the first day of data creation [15]. Therefore, it is economical to use erasure code to archive data copies. Today, some practical storage systems (for example, WAS [16], GFSII) adopt a mixed redundancy strategy, use a copy strategy for newly created data, and use an erasure code for archiving when the data access frequency is reduced.

Archiving improves storage utilization by reducing the storage overhead of infrequently accessed data. Existing distributed storage systems such as HDFS, and GFS. To ensure data availability, improve the degree of parallelism of operations, and use more copies to store data. The size of the default data block is 64 MB or 128 MB. With the exponential growth of data, the storage pressure of existing data centers is increasing. In the application scenario where multiple reads are written at once, after the data is generated, the frequency of use is negatively related to the time. The data with low access frequency is archived from multiple copies into an erasure code storage format, which can ensure data availability, improve storage space utilization, and relieve data center pressure.

4 Preservation Mechanism of Electronic Record Based on Erasure Code and Multi Copies

4.1 Erasure Code

RS-type erasure codes are the primary erasure coding techniques applied in distributed storage systems [17]. Its earliest application in distributed record system dates back to 1989. Rabin proposed an information splitting algorithm based on Rabin code for network server faults and bandwidth problems. Its core is the RS type erasure code. Reed-Solomon Coding [18] uses Galois Field operations for encoding/decoding, where the Galois Field addition operation is an XOR operation, and the multiplication operation is usually performed by searching for a corresponding Galois Field table.

RS code is a block-based MDS error correction coding, which is widely used in the field of communications and storage. In general, the $(k+r,k)$-type RS code indicates that each band of the code is composed of k data blocks and r check blocks. It uses data and a generation matrix to generate redundant data. The generation matrix consists of a $k \times k$ identity matrix and a $k \times r$ redundancy matrix. The RS coding process is actually about the linear operation of the data block, and the redundant block is calculated by the multiplication of the k data blocks and the $k \times r$ generating matrix. The encoding process of the Andermonde-RS $(k+r,k)$ algorithm is as follows.

$$\begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,k} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,k} \\ \vdots & \vdots & & \vdots \\ r_{n,1} & r_{n,2} & \cdots & r_{nk} \end{bmatrix} \otimes \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & \vdots & \vdots \\ 1 & 2^{r-1} & \cdots & k^{r-1} \end{bmatrix} \otimes \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \end{bmatrix} = \underbrace{\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \\ p_1 \\ \vdots \\ p_r \end{bmatrix}}_{\mathrm{E}} \tag{1}$$

If r-blocks in the E-matrix are lost, the corresponding rows of the r-blocks in the A-matrix and the E-matrix are deleted at the time of recovery, and a new $(n \times n)$-order matrix A$'$ and an $(n \times 1)$-order matrix E' are obtained. A' is non-singular, and inverts A' to get A'^{-1} recovery data: $D = A'^{-1} \cdot E'$. Extract the calculation part in which the redundant data $\mathrm{p}_1 \sim \mathrm{p}_\mathrm{r}$ is generated, that is, the process of generating redundancy check for the code, as shown below.

$$\begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,k} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,k} \\ \vdots & \vdots & & \vdots \\ f_{n,1} & f_{n,2} & \cdots & f_{nk} \end{bmatrix} \otimes \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & \vdots & \vdots \\ 1 & 2^{r-1} & \cdots & k^{r-1} \end{bmatrix} \otimes \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_r \end{bmatrix} \tag{2}$$

There are two drawbacks to using Disk Reduce directly: (1) Read performance in multi-copy mode is better, multiple pieces of data services at the same time, and load balancing is possible. (2) Under the erasure code storage mode, the degraded read of the failed data block and reconstruction of data blocks will bring about a large amount of disk IO and network data transmission, while only need transmit one data block in the multiple copy mode.

By using hybrid storage of copy and erasure codes, only data with a low frequency of access is encoded to improve storage space. In a real large-scale cluster, the data is used very shortly after it is generated. In this way, when the data heat is reduced, archival storage of the three-copy data using the erasure correction code can ensure data reliability and considerably save storage space without affecting the data access speed.

4.2 Preservation Mechanism of Electronic Record Based on Erasure Code and Multi Copies

Heterogeneous Storage Cluster

With the arrival of the era of big data, the scale of the system is getting bigger and bigger. Because the old system cannot meet the increasing demands of users on capacity and performance, the system must be upgraded. In this case, if a one-time hardware upgrade is performed on the system hardware, many resources will be wasted. With the system's multiple upgrades, and the masses will have a variety of different models, different performance hardware devices. The different performances of the nodes are specifically: the computing power of different CPUs, memory capacity, network bandwidth, and disk speed. Also, the scale of the system is expanded to increase the number of nodes, which are usually located in different racks, resulting in different bandwidths and delays between different nodes. In general, as the system and hardware upgrades, the increase in the size of the storage system makes the performance of different storage nodes heterogeneous.

On the other hand, User access requests are unbalanced, which also makes the heterogeneity between nodes more complicated. When a node is storing more hot data, a large number of users request access to the node over a period. Also, some nodes have better performance and may not have user's requests. This condition will timeout in an idle state, thereby reducing the quality of the service system, mainly when the nodes perform poorly. This situation not only does not share resources but also causes a waste of system resources due to too large pressures on another node.

The heterogeneity between large-scale storage system nodes is an unavoidable issue that must be taken into account. What needs to be emphasized is that due to the existing enterprise-class data centers, to improve the utilization efficiency of resources and the user experience, the clusters often provide 24-h services, so that the heterogeneity of clusters will become more and more prominent as time passes.

Competence-Aware Erasure Record Archiving

In the process of online storage degradation, the performance problems caused by uneven distribution of electronic records and uneven distribution of node user loads will occur, which cannot be solved by traditional data locality scheduling methods. This paper proposes a mechanism to balance the preservation of electronic records according to nodes' capabilities. According to the core metrics of the node bandwidth and its storage capacity, more coding tasks are allocated to nodes with robust capabilities, and fewer are assigned to nodes with weaker capabilities. So that when the heterogeneous hardware in the node, uneven distribution of electronic records, and different user load distribution occurs, people can more fully use the resources of the entire cluster, rather than that of a single node.

For the heterogeneity of node performance and the difference in I/O capacity on nodes, the bandwidth, and storage capacity. Ability value = I/O capability * time period/electronic record block size + remaining bandwidth. The capability value represents the maximum number of data blocks that can be transmitted by the network during a period. The size of the point capability value determines how much of the coded electronic record block is allocated to that node. The system allocates less coding tasks to the weaker nodes and allocates more coding tasks to the more capable nodes. This strategy can effectively prevent the nodes whose encoding speed is too slow from becoming the shortboard of the entire coding process. The node that encodes the user's heavy load allocates less coding tasks, and the nodes with lighter user loads allocate more coding tasks, which can effectively reduce the resources competition between the coding process and the user's access, thereby improving the work efficiency of the cluster.

Based on the capability values, the assignment of coding tasks takes place in a short period. The period divides a long massive job into small jobs within a plurality of time slices and predicts a load of a time window with the user load at the beginning of the small job, that is, the load is constant. The system also converts dynamic node loads to static encoding task schedules. Combine the completion of encoding tasks for each period on each node and correct the ability value of the node's current period. This capability value feedback method can continuously update the node's capabilities over the last period.

Ability Value Initialization

According to each node's current capability value as the primary factor in the selection of coding nodes. Each node's capability value is the remaining bandwidth of the node, and it is I/O capability. Individually, firstly, the number of processing tasks of the node *i* per unit time in the cluster is calculated. The value can be obtained by dividing the number of the code storage completed by the node in a certain time period by the consumed time. Second, calculate the remaining bandwidth of the nodes in the cluster.

Coding Task Assignment

The allocation of coding tasks mainly depends on the current capability value of each node and electronic record blocks distribution. Assume that W_i corresponds to the number of record blocks expected to be processed by node *i* within a unit of time. Let B_i denote the current remaining bandwidth of node *i*, sorting from the largest to the smallest according to the *W* and *B* values of each node, picking the node to which the encoding is to be assigned. If the remaining bandwidth of the node is insufficient or its capacity value does not meet the coding requirements, the node will not get a new coding task. This encoding task assignment process ensures that the sum of the encoding tasks and user loads for each node does not exceed the throughput of each node itself. In addition, in the case that the task of each node is not overloaded, the locality of the record can be combined, so that the resource consumption of the network bandwidth in the encoding process is as small as possible.

Update of Ability Value

Because the load of the task on each node changes dynamically, the value of the node's capabilities is obsolete after each task assignment. When a coding operation task is completed, the number of coded task records R_{ti} and the coding task processing time T_i in the time period are obtained. Therefore, the new I/O capability value is R_{ti}/T_i. In addition, the remaining bandwidth of the node during the calculation of the time period will also be evaluated. Such capability value updating process can correct and reflect the idle bandwidth and I/O capability of the node in real time when the user load of the node changes, so as to achieve the task of assigning codes accurately.

The total encoding time T of each stripe is determined by the longest one of all the nodes participating in the stripe encoding.

$$T = \overset{\mathrm{p}''}{\max}{}_{\mathrm{p}=1'',\mathrm{p}\neq \mathrm{En}}\{T_{PDisk} + T_{PNet}, T_{EnDisk} + T_{EnNet} + T_{Encode}\} \tag{3}$$

The p'' nodes participating in the encoding process of band *i* are $(SN1'', \ldots, SNp'')$, SN_{En} is the node for encoding nodes and receiving data blocks. T_{PDisk} is the time for node SN_p to read a strip of p_q data blocks on the storage medium, T_{PNet} is the time for the transmission of p_q data blocks in the node SN_p network. For non-coded nodes that participate in the coding of this band, reading p_q data blocks on the storage medium and sending p_q data blocks on the network card can be performed concurrently. The network transmission speed is slower than the storage medium reading speed. So, this time formula can be equivalent to:

$$T = \max_{p=1'',p\neq En}^{\mathrm{p}''}\{T_{PNet}, T_{EnNet} + T_{Encode}\} \tag{4}$$

Because the encoding calculation is not very time-consuming for network transmission, so this time formula (3) can be equivalent to:

$$T \approx \max_{p=1'',p\neq En}^{\mathrm{p}''}\{T_{PNet}, T_{EnNet} + T_{Encode}\} \tag{5}$$

For node p, measurement method according to the weight value W:

$$T_{PNet} \approx \frac{P_q}{W_i} \tag{6}$$

For coding nodes, the node reads En_q blocks of local storage media, while the network card accepts $\sum_{p=1^n,p\neq En}^{p^n} p_q\left(= k_{Enq}\right)$ blocks of data. In a rack-aware random distribution of three copies, when $k \geq 6$, k_{PEn}, this allows the reception of network data blocks to account for the primary time of the code compute node:

$$T_{EnDisk} + T_{EnNet} \approx T_{EnNet} \tag{7}$$

The measurement method according to the weight value W:

$$T_{EnNet} = \sum\nolimits_{p=1'',p\neq En}^{m} \frac{p_q}{W_{En}} = \frac{(K - P_{En})}{W_{en}} \tag{8}$$

Substituting the Formula mentioned above can be derived:

$$T \approx \max_{p=1'',p\neq En}^{\mathrm{p}''} \left\{ \frac{p_q}{W_p}, \sum\nolimits_{p=1'',p\neq En}^{m} \frac{p_q}{W_{En}} = \frac{(K - P_{En})}{W_{en}} \right\} \tag{9}$$

5 Conclusion

With the aim of ensuring the safety and reliability of the electronic record in the cloud storage servers, the preservation mechanism of electronic record based on erasure code and multi copies are proposed in the paper. This paper focuses on the erasure code archiving of electronic records and puts forward the ability aware erasure code archiving of electronic records. Moreover, the corresponding implementation algorithm and steps are described.

Acknowledgment. This work is supported by the NSFC (No. 61772280 and 61702236), Jiangsu Province Natural Science Research Program (BK20130809, BK2012461), the Project of six personnel in Jiangsu Province (2013-WLW-012), the research fund from Jiangsu Technology & Engineering Center of Meteorological Sensor Network in NUIST under Grant (No. KDXG1301), the Changzhou Sci&Tech Program (No. CJ20179027), and the PAPD fund from NUIST. Prof. Xiaorui Zhang is the corresponding author.

References

1. Xie, L., Wang, J., Ma, L.: Trusting records: findings of team Asia InterPARES. Arch. Sci. Study **2017**(4), 8–13 (2017)
2. Qian, Y.: Study on the long-term preservation standard of trusted electronic records in China. Arch. Sci. Bull. **2014**(3), 75–79 (2014)

3. Ren, Y., Shen, J., Wang, J., Han, J., Lee, S.: Mutual verifiable provable data auditing in public cloud storage. J. Internet Technol. **16**(2), 317–323 (2015)
4. He, D., Kumar, N., Wang, H., Wang, L., Choo, K.: Privacy-preserving certificateless provable data possession scheme for big data storage on cloud. Appl. Math. Comput. **314** (12), 31–43 (2017)
5. Jiang, Q., Zeadally, S., Ma, J., He, D.: Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. IEEE Access **5**, 3376–3392 (2017)
6. Shen, J., Zhou, T., Chen, X., Li, J., Susilo, W.: Anonymous and traceable group data sharing in cloud computing. IEEE Trans. Inf. Forensics Secur. (2017). https://doi.org/10.1109/TIFS.2017.2774439
7. Fu, Z., Huang, F., Ren, K., Weng, J., Wang, C.: Privacy-preserving smart semantic search based on conceptual graphs over encrypted outsourced data. IEEE Trans. Inf. Forensics Secur. **12**(8), 1874–1884 (2017)
8. Wei, F., Zhang, R., Ma, C.: A provably secure anonymous two-factor authenticated key exchange protocol for cloud computing. Fundam. Inf. **157**(1–2), 201–220 (2018)
9. Li, X., Niu, J., Kumari, S., Wu, F., Sangaiah, A., Kim, C.: A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments. Netw. Comput. Appl. **103**, 194–204 (2018)
10. Chao, L.: Research on electronic record migration model in cloud computing environment. Arch. Sci. Bull. **2013**(1), 53–56 (2013)
11. Ren, Y., Shen, J., Liu, D., Wang, J., Kim, J.: Evidential quality preserving of electronic record in cloud storage. J. Internet Technol. **17**(6), 1125–1132 (2016)
12. Fu, Z., Wu, X., Guan, C., Sun, X., Ren, K.: Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement. IEEE Trans. Inf. Forensics Secur. **11**(12), 2706–2716 (2016)
13. Jiang, H., Fan, M., Wang, X.: Low complexity array codes for random triple failures in distributed storage system. J. Converg. Inf. Technol. **7**(23), 247–250 (2012)
14. Xia, M., Saxena, M., Blaum, M.: A tale of two erasure codes in HDFS. FAST **2015**, 213–226 (2015)
15. Huang, J., Liang, X., Qin, X.: PUSH: a pipeline reconstruction I/O for erasure-coded storage clusters. IEEE Trans. Parallel Distrib. Syst. **26**(2), 516–526 (2015)
16. Huang, J., Zhang, F., Qin, X.: Exploiting redundancies and deferred writes to conserve energy in erasure-coded storage clusters. ACM Trans. Storage **9**(2), 1–29 (2013)
17. Wang, Y., Zhao, Y., Hou, F.: Minimum bandwidth regeneration code of distributed storage system. J. Chin. Comput. Syst. **33**(8), 1710–1714 (2012)
18. Hao, J., Lu, Y., Liu, X., Xia, S.: Survey for regenerating codes for distributed storage. J. Chongqing Univ. Posts Telecommun. (Nat. Sci. Ed.) **25**(1), 30–38 (2013)