Optimizing File Transfer in Distributed Storage System Based on Reconfiguration of Storage Structure

Kai Li, Shouming Chen*, Yingda Chen, Susheng Zhong, Kaitong Huang, Minnuo He, Lixin Yang, Bo Li, Yi Yan
Guangdong Power Grid Corporation Information Center, Guangdong 510080, China
*Corresponding Author email: chenshouming@gdxx.csg.cn

Abstract. Distributed storage is the storage part of distributed computing. It is an easy-to-expand and virtualized storage resource pool. In order to meet the needs of large-scale storage structure applications, reliable mass data storage services are provided. Redundant data storage is needed to deal with the data unreliability caused by node failure. The lower the cost of file transmission, the less traffic generated in the network. According to the time logic, the regeneration process is divided into stages: data distribution, data encoding and decoding, data transmission and data reconstruction. The feedback mechanism can reduce the overhead of acquiring the load information of the data server node while ensuring the real-time performance. The determination of the activity factor function of the data block access frequency statistics needs to be established in the performance adaptability of the data block activity factor and the specific access times. The single-point dependency of the centralized storage system on the primary node is changed, which reduces the waiting delay of concurrent users and the average memory usage of the metadata server.

1. Introduction
In distributed file system, in order to ensure the reliable storage of data, it is inevitable to introduce redundancy to ensure the reliability of data. The most commonly used method is to make a simple backup of each data [1]. Its generation efficiency and quality directly affect the operation effect of reconfigurable structure. Configuration information is a memory from the hardware point of view. Each processing unit in the reconfigurable structure reads configuration information from the memory [2]. Aiming at the data uploading, repairing and downloading stages of the code in distributed storage system, it needs to satisfy large-scale streaming reading and small-scale random reading [3]. The data placement strategy will not change its storage structure after storing the data on each node in the system, as well as large-scale additional writes and small-scale random writes [4]. Compare the file size, buffer size, and finite field size to the speed of the operation in the stage [5]. It is found that by properly configuring the relevant parameters, the commonly used optimization method is to package multiple small files in a data set for overall transmission, which has the advantage of improving the performance of file transmission [6]. It also has a significant disadvantage, that is, when any node fails and needs data repair, it needs to obtain and decode the complete original data from other active nodes. Providing reliable mass data storage services, the distributed storage system adopts an extensible system structure and locates files by controlling nodes [7].

Distributed storage system needs to ensure the reliability of the system, and ensure that when the application or client accesses its stored data, it can get complete data. Because distributed storage systems connect hundreds of storage nodes through the network [8]. The growth of data bus
bandwidth and disk speed can not meet the application's demand for data bandwidth. Storage subsystem has gradually become the bottleneck of the development of computer systems [9]. Metadata records the mapping relationship between logical data names and physical information, including all metadata needed for file access control [10]. Node management mainly studies how to select some nodes or disks in distributed storage to provide data services for upper application, and let other nodes enter low energy consumption mode or dormant state, so as to achieve the purpose of reducing energy consumption [11]. Redundant storage overhead required to maintain system reliability, bandwidth overhead and time overhead required to regenerate invalid data. The replication strategy has a good effect on both bandwidth overhead and time overhead [12]. Most of the file size is very large. Finally, in the process of using the file system, the vast majority of cases are adding new data instead of overwriting existing data. It is assumed that the case of randomly writing a file is almost non-existent. Parallel transfer of files is achieved by directing these file fragments to a specific storage server through the transport control module [13]. The selection of closed nodes in node management is closely related to the data management technology. Currently, the existing data management technologies are mainly divided into data management of static data placement, data management of dynamic data placement, and data management of cache prefetching [14].

In this paper, we propose a storage structure reconfiguration algorithm, which is an algorithm for file transfer optimization in distributed storage systems.

In summary, our contributions are as follows:

This algorithm is a new storage structure reconfiguration algorithm for file transfer optimization in distributed storage systems.

This algorithm is widely applicable in the environment of storage structure reconfiguration, and has high applicability for file transfer optimization in distributed storage system.

This algorithm has higher recognition, strong visualization and good operation results.

2. Materials And Methods

When configuration information is insufficient, configuration information is fetched from the configuration memory outside the processing unit. This structure greatly reduces the access frequency of configuration information. Nevertheless, it also brings a new problem of configuration space. The larger the index information entropy is, the larger the range of its value changes. The more important we judge the index.

| Table 1 Protocol Specific Step Parameters |
|------------------------------------------|
| **Parameter**                        | Transmission | Node |
|---------------------------------------|--------------|------|
| Signature identification             | 1.52         | 0.69 |
| Sending Node Messages                 | 0.36         | 0.41 |
| Accept data encoding                 | 0.35         | 0.38 |
On the server side, the service process continuously reads file information from the data channel. Without the need to change the data storage structure, there are usability requirements. The entire life cycle of a distributed storage system is divided into data insertion, data maintenance, and data reconstruction, respectively analyzing the storage, communication, and computational overhead in the phase. When the node where a block of code blocks is faulty, the coded block becomes inaccessible with the node failure. In order to ensure the reliability of the data, a new coded block needs to be generated on the new node.

Intra-block utilization rate, intra-block correlation rate and inter-block correlation rate between adjacent blocks can truly reflect the impact of small file distribution and blocking effect on network bandwidth utilization in data nodes. The effectiveness of the indicators is shown in Table 2 and Figure 2. The optimization methods and results of file block merging and file block splitting in the system are monitored, as shown in Table 3 and Figure 3.

| Table 2 Effectiveness of Indicators |
|-------------------------------------|
| **Average value** | **Effectiveness** |
| Intra-block utilization | 11.58 | 9.35 |
| Intra-block correlation | 9.17 | 8.24 |

On the server side, the service process continuously reads file information from the data channel. Without the need to change the data storage structure, there are usability requirements. The entire life cycle of a distributed storage system is divided into data insertion, data maintenance, and data reconstruction, respectively analyzing the storage, communication, and computational overhead in the phase. When the node where a block of code blocks is faulty, the coded block becomes inaccessible with the node failure. In order to ensure the reliability of the data, a new coded block needs to be generated on the new node.

Intra-block utilization rate, intra-block correlation rate and inter-block correlation rate between adjacent blocks can truly reflect the impact of small file distribution and blocking effect on network bandwidth utilization in data nodes. The effectiveness of the indicators is shown in Table 2 and Figure 2. The optimization methods and results of file block merging and file block splitting in the system are monitored, as shown in Table 3 and Figure 3.
Using data locality to save current data by caching is a good choice because there is a great possibility that the current data will be used in subsequent operations. In terms of advantages, the average response speed of continuous storage is faster, and the most important thing is to support sequential storage devices such as tapes. However, the system concentrates all the computational load on a special proxy node, which limits the actual performance of the whole system to the performance configuration of the proxy node. The retrieval and storage performance of metadata depends on the cache and the database. The cache can greatly improve the efficiency of the read operation, but it does not improve the write operation or even reduce the write performance, and its expansion is limited by the size of the main server memory. The operational parameters of the transmission process are shown in Table 4 and Figure 4.

### Table 3 Relation between File Block Size and Optimizing Method

| Optimization          | Average |
|-----------------------|---------|
| Number of block merges| 9.42    |
|                       | 3.50    |
| Number of block Splitting| 8.14  |
|                       | 2.68    |

### Table 4 Operational parameters of transmission process

| State                  | State | Storage |
|------------------------|-------|---------|
| Get the file location  | 21.05 | 16.92   |
| Creating transfer persons| 19.07 | 15.71   |
3. Result Analysis and Discussion
The effect of network bandwidth utilization on Optimization times is tested. For the same initial condition, different network bandwidth utilization targets are set, and various indicators are recorded to achieve this goal. The impact of network bandwidth utilization on optimization is shown in Table 5 and Figure 5.

| Disk Adjustment Data Volume | Total data volume ratio |
|-----------------------------|-------------------------|
| 95%                         | 23.7                    | 26.1                  |
| 90%                         | 17.5                    | 14.2                  |
| 85%                         | 13.8                    | 13.1                  |

4. Conclusion
In this paper, the optimization of file transfer in distributed storage system with reconfiguration of storage structure is studied. Because of the requirement of availability of data blocks, distributed storage systems can not simply solve the energy consumption problem by combining tasks with dormant idle nodes in traditional data centers. The system needs to be expanded according to the demand, that is, by adding new storage resources and migrating some data to new storage devices to
relieve the pressure. This operation is called storage system expansion. The data transmission of the regeneration process can not only bypass the link with low available bandwidth, but also bypass the active node with low reliability, and reduce the risk that the node data fails due to the failure of the forwarding node due to too many forwarding times.

References
[1] Lai W K, Chen Y U, Wu T Y, et al, Towards a framework for large-scale multimedia data storage and processing on Hadoop platform[J], Journal of Supercomputing, 2014, 68(1):488-507,
[2] González-Férez, Pilar, Bilas A, Mitigation of NUMA and synchronization effects in high-speed network storage over raw Ethernet[J], The Journal of Supercomputing, 2016, 72(11):4129-4159,
[3] Region-based fault-tolerant distributed file storage system design in networks[J], Networks, 2015, 66(4):380-395,
[4] Liao H, Liu D, Huang Y, et al, Load transfer capability analysis considering interconnection of distributed generation and energy storage system[J], International Transactions on Electrical Energy Systems, 2014, 24(2):166-177,
[5] Bowers A, Liao C, Steiert D, et al, Detecting Suspicious File Migration or Replication in the Cloud[J], IEEE Transactions on Dependable and Secure Computing, 2018, PP(99):1-1,
[6] Liang S, Yuan C, Kan H, Linear Exact-Repair Construction of Hybrid MSR Codes in Distributed Storage Systems[J], IEEE Communications Letters, 2014, 18(7):1095-1098,
[7] Duan H, Yu S, Mei M, et al, CSTORE: A desktop-oriented distributed public cloud storage system[J], Computers & Electrical Engineering, 2015, 42:60-73,
[8] Abawajy J H, Deris M M, Data Replication Approach with Consistency Guarantee for Data Grid[J], IEEE Transactions on Computers, 2014, 63(12):2975-2987,
[9] Ganesan A, Alagappan R, Arpaci-Dusseau A C, et al, Redundancy Does Not Imply Fault Tolerance[J], ACM Transactions on Storage, 2017, 13(3):1-33,
[10] Zhu B, Shum K W, Li H, Heterogeneity-Aware Codes With Uncoded Repair for Distributed Storage Systems[J], IEEE Communications Letters, 2015, 19(6):901-904,
[11] Zhu Y, Lin J, Lee P P C, et al, Boosting Degraded Reads in Heterogeneous Erasure-Coded Storage Systems[J], IEEE Transactions on Computers, 2015, 64(8):2145-2157,
[12] Huang W, Abu Qahouq J A, Energy Sharing Control Scheme for State-of-Charge Balancing of Distributed Battery Energy Storage System[J], IEEE Transactions on Industrial Electronics, 2015, 62(5):2764-2776,
[13] Pu Q, Ananthanarayanan G, Bodik P, et al, Low Latency Geo-distributed Data Analytics[J], ACM Sigcomm Computer Communication Review, 2015, 45(4):421-434,
[14] Zaheer A, Covic G A, Kacprzak D, A Bipolar Pad in a 10-kHz 300-W Distributed IPT System for AGV Applications[J], IEEE Transactions on Industrial Electronics, 2014, 61(7):3288-3301,