Transaction Data Management System based on Distributed Storage Architecture

Yuan Ji¹, Tian Xia¹, Honglue Zhang¹, Gang Chen²
¹Power dispatching control center of Guizhou Power Grid Co., Ltd., Guiyang 550007, Guizhou
²NARI-TECH Nanjing Control Systems Ltd., Nanjing, Jiangsu 211106

Abstract: With the continuous development of application requirements and technologies, the differences between distributed storage architectures and traditional storage architectures are becoming increasingly apparent. The distributed storage architecture not only has the advantages of small fault impact range, high scalability, and large throughput, but also is widely used in many application systems. In the transaction data management system based on the distributed storage architecture, a large amount of transaction data information is transmitted and stored using the distributed storage architecture, which can effectively ensure the accuracy and consistency of the data. At the same time, the stability of the transaction data management system will be enhanced.

1. Introduction
In recent years, data has become an indispensable form of information in daily production and life. With the rapid development of the Internet, the rapid increase in the amount of data and information has brought many benefits to people's lives. The hidden meaning and effective rules in the massive data add power to the development of science and technology. But the data explosion has made it an important issue how to obtain low-cost data storage and data transmission methods. At present, the amount of data generated by humans every day and the amount of data to be transmitted have reached unprecedented scale and volume. Only through fast and error-free transmission, storage and use of data can the data be used to its fullest extent. The supply space required for global storage in 2019 has reached 30,000 exabytes, and the storage space required in 2020 is expected to reach 42,000 exabytes. Therefore, in the case that the traditional storage structure cannot meet the security and availability of data, the distributed storage architecture is widely used in various systems. The comparison between traditional storage architecture and distributed storage architecture is shown in Table 1. Traditional storage architecture has advantages in terms of difficulty in architecture design, testing and errors, and deployment. In terms of isolation, system scalability, system performance, and technology used, a distributed storage architecture is still better.

Table 1: Comparison of traditional storage architecture and distributed storage architecture

|                     | Traditional storage architecture | Distributed storage architecture |
|---------------------|---------------------------------|----------------------------------|
| Architecture        | Less difficult                  | Difficult                        |
| Isolation           | Large range of impact           | Small impact range               |
| System performance  | Fast response time, Small throughput | Slow response time, High throughput |
| Technology          | Single and closed technology    | Technology is diverse and open    |
| System scalability | Poor scalability | Better scalability |
|-------------------|------------------|--------------------|
| Tests and errors  | Simple           | Complex            |
| Deploy            | Infrequently posted | Release frequently |
|                   | Easy to deploy   | Complex deployment |
| Management        | Focus on development costs | Focus on service governance |

The evolution from the traditional centralized storage architecture to the modern distributed storage architecture is not only the demand for the exponential growth of data volume, but also the specific requirements of each application system. In the transaction data management system, a distributed storage architecture is needed to store a large amount of transaction data. The distributed storage architecture can effectively guarantee the security and consistency of transaction data, and the system performance is relatively stable.

2. Distributed storage architecture
The client and the metadata server are indispensable parts of the distributed storage architecture, especially the latter is the core processing part of the entire framework. The main task of the storage architecture system is to achieve efficient and convenient management of metadata and to quickly and timely respond to client requests. So the quality and performance of the metadata server becomes critical. Secondly, another important component at the forefront of the storage architecture is the client. As the name implies, the client is mainly responsible for the transmission of various needs of the customer. Therefore, only the renewal request that has been effectively transmitted through the client can be received by the metadata server in the later stage, and then the next request processing is performed. A large amount of data processed by the core metadata processor needs to be stored on another important component, namely the data server. The reasonable use of the data server can effectively guarantee the availability and integrity of the data stored in it, which is an effective guarantee for the entire storage architecture. Therefore, having these three component structures enables the distributed storage structure to effectively complete the write-request, effective processing, and reasonable storage of the entire data.

At the same time, due to its distributed storage characteristics, the entire architecture is faster and more reliable in expanding storage capacity and storage performance. As shown in Figure 1, multiple users can access distributed storage data through multiple web servers. When data is stored, multiple file storage servers need to cooperate to complete it, and the storage servers can share and exchange information with each other. However, when one of the memories fails, the remaining memories can still continue to work in this mode, which greatly reduces the scope of the impact.

The distributed storage architecture is mainly divided into three types when applied: distributed storage of files, distributed storage of objects, and distributed storage of blocks. Depending on the storage object, clients in the three components of the storage architecture may change to users. However, the metadata server and data server are basically unchanged, and still maintain their original functions. In addition, the information interaction between these three components is also timely and effective. A series of request commands such as continuation writing can quickly establish a connection between the client and the metadata server. The data server mainly transmits information such as the basic usage of each storage device and the data storage status to the metadata server. The interaction from the client to the data server requires various media to complete.
2.1 Distributed Storage of Objects
The emergence of distributed storage of objects occurs when neither file-based distributed storage nor block distributed storage can meet the requirements. The distributed storage of objects can not only avoid the disadvantages of the two, but also effectively balance the advantages of both. Therefore, the distributed storage of objects is widely used. First of all, for file distributed storage, distributed storage of objects can not only inherit its advantages of sharing, but also avoid the disadvantage of slower data storage. For the distributed storage of blocks, the disadvantages that are not conducive to sharing are effectively avoided by the distributed storage of objects. And the distributed storage method of objects also has the advantage of faster reading speed. Therefore, the object storage method is a better distributed storage method.

The distributed storage of objects is mainly accomplished by multiple machines working simultaneously. The multiple server machines used to store the data first need to have a hard disk device.
with a large storage capacity, so as to ensure the effective operation of the storage operation. These servers for storing a large amount of data also need to install corresponding object storage software, so that they can be managed directly through the software. At the same time, for a large number of storage servers, another server device is needed to uniformly manage them. The server used for unified management can be called a management node server, and the management node server also needs to be managed by the same software equipment as the data storage server. The software used to service the management node server is called object storage management software. The effective coordination of the two can better serve the entire data distributed storage system.

2.2 Block Distributed Storage

Distributed storage of blocks, as the name implies, is to store a large amount of data in the form of blocks in the memory unit of the computer. The computer will provide all storable space on its disk for data use. Whether it is a physical hard disk or a logical hard disk, it is directly mapped to the host for block storage of data. After the host has properly partitioned and formatted all the hard disks, it will perform the next data storage work on the hard disks. In this way, the large capacity problem that the host can use during data storage can be effectively guaranteed. When the customer performs data storage, the system will provide the storage device directly to the customer in the form of "blocks". Because it is a storage space composed of multiple disks, parallel input of multiple disks can be relatively well implemented, which will greatly improve the overall read and write efficiency of the system and obtain higher storage effects.

The way block storage works makes disk arrays and hard disk facilities very important to the whole. However, it is also because of this working mode of block storage that the data information between the stored hosts cannot be shared at all. Because of this storage principle, the disk processed by the partition and format operation is considered by the host as its own local hard disk, and its own local disk is naturally impossible to be shared with other hosts. Therefore, the function of information sharing between hosts cannot be completed. The inability to share data has become a major drawback of the block storage method, but its advantages of higher read and write efficiency still make it widely used in many cases.

2.3 Distributed storage of files

Because the problem of data file sharing cannot be solved in the distributed storage of blocks, the distributed storage of files effectively solves this problem. In the process of distributed storage of files, a server such as FTP is required to provide support and protection. It is not necessary to use a too strict conditional environment, so a machine with a very low cost can meet the conditions. At the same time, the network's requirements for distributed storage of files are relatively low, and Ethernet can meet the environmental conditions required by the system. However, because Ethernet may have a lower network transmission rate when uploading and downloading data, it may affect the working rate of the entire distributed storage system. But the biggest advantage of distributed storage of files is that it can be very convenient to share files.

3. Transaction data management system

Transaction data management system refers to an application system that effectively manages a large amount of transaction data. Users can quickly and conveniently perform operations such as viewing transaction data on the client according to their own needs. As shown in Figure 2, users only need to log in to the application system, that is, they can connect to the transaction data management system with the help of application development tools. At the same time, the client's request instruction will also be transmitted to the data management system. According to different request instruction operations, the operating system can further perform corresponding operations on the transaction data in the database. At the same time, because transaction data is stored in a distributed architecture in the machine, when the corresponding transaction data operation is performed, the data throughput is large and the environment of the entire system is more stable.
The client's processing of transaction data mainly includes the following three aspects:

- **Entry of transaction data**: The entry of transaction data mainly includes the effective input of various information such as the transaction date, the parties to the transaction, the transaction amount, and the transaction status, so this process can also be called the process of establishing transaction data. As the first step in the establishment of the entire transaction database, the entry of transaction data must ensure its accuracy and consistency, otherwise it will continue to affect the subsequent operation of transaction data.

- **Modification of transaction data**: This is an important step in the reasonable application of transaction data. It is also an opportunity to correct errors made in the entry of transaction data, which can reduce the error rate in the database. However, you need to pay attention to the extraction of transaction data and the transmission of transaction data when making changes.

- **Query of transaction data**: The query of transaction data refers to the simplest task of viewing transaction information in the processing of transaction information, and it is also a method often used in the transaction process.

![Figure 2: Structure of transaction data management system](image)

### 4. Conclusions

In a transaction data management system based on a distributed storage architecture, the distributed storage architecture provides effective guarantees for application systems and provides stable and reliable data services for clients. Unlike traditional storage architectures, distributed storage architectures ensure data reliability and consistency to a great extent. At the same time, the distributed storage architecture has the advantages of strong scalability and small fault impact, which makes the transaction data management system more flexible and adaptable when facing massive data. Therefore, the system is more efficient in managing and processing transaction data, and it also saves a lot of labor...
and material costs. However, many devices of the management system still need continuous improvement in the details of interaction.

References
[1] Liu Yansen. Research on Fully Distributed Storage Architecture Based on Hadoop [D]. Xidian University, 2013.
[2] Zhu Huafang. Design and implementation of lightweight distributed college entrance examination performance management system based on SSM framework [D]. Shandong Normal University, 2016.
[3] Zhu Minxin. Design of Distributed Storage System Based on Zeromq [D]. Fudan University, 2013.
[4] Rao Wei, Zhou Aihua, Chang Tao, et al. Research on Multi-type Data Management Technology Based on Distributed Storage [J]. Electric Power Information and Communication Technology, 2016 (5): 76-81.
[5] Zhong Xiaojun, Lu Yuan. Construction of distributed data storage surveying and mapping results management system [J]. Geospatial Information (4): 11 + 135-137.
[6] Zhong Yunqin, Fang Jinyun, Institute of Computing Technology, Chinese Academy of Sciences, Beijing. A High-Performance and Massive Spatial Data Management Method Based on Distributed Storage Computing Architecture [J]. 2011 National High-Performance Computing Academic Conference (HPC china2011), 2012.
[7] Fan Juanjuan, Haiyang, Gao Mingxing, et al. Historical database analysis solution based on distributed storage and parallel computing architecture [J]. China Railway, 2014 (7): 60-63.
[8] Su Dong. Design and implementation of metadata management module for distributed structured storage system [D]. University of Electronic Science and Technology of China.