Scholar Publishing Group
Distributed Processing System
https://doi.org/10.38007/DPS.2021.020202
ISSN 2790-0916 Vol. 2, Issue 2: 9-17

A Distributed System Design Based on Cloud Computing

Nardelli Matteo[*]

Tech Univ Sofia, Dept Elect Apparat, Sofia 1797, Bulgaria

[*]*corresponding author*

Keywords: Distributed Systems, Cloud Computing, Data Encryption, Server Clustering

Abstract: With the development of the era of big data, it is becoming more and more important to store efficient and secure massive data. Distributed secure storage technology, combined with distributed storage technology and data encryption technology, has the characteristics of security and mass storage, and has become a hot spot in the field of current information security research. In distributed storage technology, the selection of storage nodes is a key problem. Whether the selected node is reasonable will affect the system performance and the effective utilization of storage capacity. Cloud computing is a form of information storage. The essence of "cloud" is the network. On the one hand, cloud computing is a network that can store data. Users can call data on the "cloud" when using it, according to the amount required by users, and at some level, the storage amount is unlimited. This paper designs and implements a distributed encrypted file storage system based on WebDAV protocol. After using the cloud computing mode, the server side encrypts and divides the file content, and the proposed node selection algorithm is used to store the data on the selected nodes in the server cluster. The function and performance tests of the system show that the system is functioning correctly functional, efficient, safe and easy to use, and has good scalability.

1. Introduction

With the rapid development of the Internet of Things technology, electronic products such as computers, mobile phones and wearable devices are closely related to our lives and work, generating large amounts of data every day. These data may contain personal privacy, company business secrets and other important information. If not handled properly, data damage and leakage, which will bring irreparable consequences. Therefore, the secure storage and processing of massive amounts of data is becoming more and more important [1].

The traditional storage mode is centralized single point storage, which stores all data in one storage server. The storage operation in this way is simple and easy to implement, but there are hardware capacity limitations [2]. At the same time, when a large number of users access concurrently, the data reading and writing operations are more frequent, and the single node server device cannot handle all requests will become the performance bottleneck of the system and cannot meet the requirements of mass data storage [3]. In order to break through the storage limitation of a single device and improve the overall storage efficiency, distributed storage technology came into being. It uses the network to connect a large number of storage servers, divide the data into multiple copies, save each data to different physical devices, and provide a unified interface for external access [4]. This structure can not only meet the capacity needs of large-scale storage, but also enable the storage pressure to be shared among multiple server devices, reducing the pressure of single point devices, greatly improving the scalability, availability and access efficiency of the system, and gradually being widely used [5]. As a distributed data processing platform, cloud computing can integrate a large number of computer resources and achieve a huge increase in technical capacity. It is more suitable for processing massive data than ordinary algorithms. In addition, the cloud computing model and cloud computing platform do not have high requirements for network nodes. Ordinary computers can also participate in cloud computing, which reduces the complexity and cost of cloud platform construction to a certain extent. The application extension is definitely far beyond the scope of the Internet. The combination of cloud computing theory and data mining technology is a trend. Although distributed storage technology has many advantages, it needs a large number of physical devices to participate, which makes its system implementation more complex and contains more uncertainties. For example, because the data is stored in different locations, the data transmission distance may be increased and more network consumption may be brought, and the file segmentation and aggregation processing need to be considered; How to ensure the stable operation of each storage device under multi-user concurrent access; When storing, how to select nodes to ensure the balance of its capacity and full performance; How to manage the storage server cluster to ensure that the system can respond quickly when the nodes change; How to ensure the security and reliability of data storage. These are the problems that need to be solved by distributed storage technology [6-7].

Compared with distributed storage technology, distributed encrypted storage technology expands on the basis of distributed storage technology and introduces encryption technology, which can improve the security and privacy of data storage. Accordingly, a system designed based on distributed encryption storage technology needs to add two modules: centralized key management and data encryption and decryption. Data stores are usually divided into metadata and file contents to speed up system access. Due to its advantages in data security, scalability and reliability, secure storage technology in distributed environment has become a hot research topic in the field of information security. Through the combination of cloud computing and distributed system, this paper designs and implements a distributed encrypted file storage system for individual users, and focuses on the selection of storage nodes and the control of file data access when storing files.

2. Review of Related Concepts

2.1. Cloud Computing

Cloud computing is a distributed parallel management system that relies on network technology. It relies on virtual resources to provide users with various services in the way of dynamic use of resources. Users can negotiate a charging agreement with cloud platform service providers according to their actual needs, and can pay and collect fees according to the year or any cycle [8]. Cloud computing packages all the resources needed in the computing process and provides them to enterprises in the form of services. In this way, the enterprise only needs to pay attention to the data needed by the enterprise itself, and no longer needs to pay attention to all the hardware deployment and other information technology details [9]. To put it simply, cloud computing refers to providing the storage space, computing mode, application software, database and other development platforms of a large-scale resource pool to users based on the current Internet mode, so as to realize low-cost, automatic, rapid provision and flexible expansion of IT services. In other words, cloud computing is actually the commercialization of computing resources required by enterprises. It takes computing capacity as a public service through distributed computing network, and provides efficient information processing services to traditional manufacturing enterprises and even the whole industry. Enterprises can increase or reduce computing capacity at any time as needed without paying attention to the expenses of hardware purchase, maintenance, data center, power consumption and cooling [10].

"Cloud" is not a single service, but a collection of many services, including the following five categories: SOA architecture layer, management middleware, virtual resource pool layer, and physical resource layer [11].

SOA architecture layer: encapsulate various computing service programs of cloud computing into standard web services and deploy them to the SOA framework for management and use. Its interface is independent of application services, which enables the services built in the system to interact and access uniformly on the Internet using standardized standard interfaces [12].

Management Middleware: responsible for the resource management of the cloud platform, monitoring its resource load or whether there is a fault; Conduct identity authentication, access authorization, account management and other security management for users; Manage tasks such as task scheduling and execution in the cloud platform. The effective operation of the service is ensured through the above management [13].

Virtual resource pool layer: integrate software, data resources, network communication, storage space and computing resources into a large virtual resource pool. The purpose of integrating more resources into a resource pool is to better manage and invoke resources [14].

Physical resource layer: it mainly provides some physical devices, such as large-scale storage devices, servers and other network devices, database systems and other infrastructure [15].

2.2. Node Selection Algorithm

In a distributed storage system, because data is stored in multiple nodes dispersedly, the disk utilization and performance load of each node will vary greatly over time, affecting the overall performance of the system [16]. Therefore, how to choose a reasonable node selection algorithm has become the key part of distributed storage. It is of great significance for distributed storage system to deeply study the reliable, efficient and adaptive node selection algorithm. A good node selection algorithm should fully consider the real-time use of nodes and make corresponding adjustments to ensure the availability, efficiency and scalability of the system [17].

According to whether it has the real-time information of considering nodes, some common node selection algorithms in distributed systems are divided into two categories, including direct calculation based on fixed algorithms and considering the real-time situation of nodes [18].

(1) Fixed algorithm direct calculation

This method usually selects nodes for storage according to a fixed rule and based on simple mathematical calculation. It has the advantages of simple implementation, fast calculation speed and low cost. However, it has poor fault tolerance and expansibility and cannot dynamically adapt to the changes of storage nodes. It is an optimistic strategy. It mainly includes polling method, random access method, hash modulus taking method, consistent hash algorithm and its extension.

(2) Consider the real-time situation of nodes

The algorithm can reflect the running status of the server in real time and select more reasonable nodes for data storage. However, it is usually necessary to use information feedback technology to obtain node information in real time based on additional monitoring services, which will cause large system overhead. It mainly includes the minimum load priority method, the fastest response priority method, the selection according to the storage capacity, the minimum connection priority method, and the combination of the storage capacity and the load.

For the traditional single node storage system, all data is stored in one server. With the passage of time, there are more and more users, and the available storage space of the system is also decreasing. At the same time, due to the impact of hardware performance, the single node storage also limits the processing capacity of the system. For this reason, distributed storage technology has emerged, which extends data storage to server clusters. This method can well solve the limitation of disk capacity, the pressure of distributed network broadband and hardware reading and writing, and meet the high concurrent ability of user access. However, there is also an inevitable problem. If a piece of data is stored in multiple servers dispersedly, if the storage nodes cannot be reasonably selected, the data distribution will inevitably be uneven. If the amount of data stored between the servers is too large, some servers will have problems such as insufficient space, frequent user access, large amount of network bandwidth occupied, high I / O utilization rate, and reduced performance load; On the contrary, for the server with a small amount of stored data, the user access is not much, the network bandwidth is idle, the disk utilization rate is low, the performance load is small, and the resources are not fully utilized. When reading data, the data aggregation operation is only carried out after the complete data is read from each node, which results in the reading and writing speed of a single node and affects the overall time consumption of the system. When the network broadband resources of a node are small, if a large number of read and write operations are carried out to the node, the data transmission will be slow, even the server will be down, and finally the system will crash. Therefore, how to share the pressure of the nodes, reduce the difference between the relative operation time of each node, and ensure a more balanced load has become an important and urgent issue in the distributed system. Reasonable selection of storage nodes can effectively use system resources, disperse operating pressure, stabilize the performance of each server, and improve the reading and writing efficiency.

2.3. Relevant Algorithms

When a new server is added, its total capacity QT and used space Qu can be obtained. After that, the amount of data QF sent to a node can be obtained every time it is stored, and the space utilization rate P of a node can be calculated in real time by accumulation

The calculation formula for the space utilization rate of each node is:

$$P=（Q_u+Q_f）/Q_t \tag{1}$$

When deleting a file, the real-time space utilization can be calculated by subtracting the size of the file. The calculation formula is:

$$P=（Q_u-Q_f）/Q_t \tag{2}$$

In a storage process, the main factors that affect the time consumption are file size and server performance. The larger the file, the longer the storage time; The higher the server load, the longer the storage time.

In order to remove the influence of file size on storage time, the algorithm uses unit storage time for subsequent calculation. When accessing a new node, a blank file test will be carried out, and a blank file will be uploaded to each storage node several times to obtain the average time consumption TW, which can make the storage time of unit size data more accurate. For the unit storage time of one operation, the calculation formula is::

$$T_{\mathrm{v}} = (T_f - T_{\mathrm{w}}) / Q_f \tag{3}$$

$$T_{\mathrm{B}} = (T_f + T_{\mathrm{w}}) / Q_f \tag{4}$$

Network fluctuation is inevitable, but a certain range of fluctuation is a normal phenomenon. If the decision attribute is completely dependent on the last operation time, the server performance load cannot be well reflected. Therefore, a certain number of time-consuming situations in the past can be used for calculation, which can weaken the impact of network fluctuations and reflect the node load performance. The operation time-consuming data has a certain timing, and the closer the time-consuming data is to the current time, the more it can reflect the real-time performance of the server. Introduction learning rate α, It can reflect the effect of the current operation time on the whole. Process the last N storage times and obtain the weighted unit storage time as the final calculated attribute value. The calculation formula is:

$$T^{i} = \begin{cases} 0 \\ T^{i-1} \cdot (1-\alpha) + T_v \cdot \alpha \end{cases} \tag{5}$$

3. System Design and Implementation

3.1. Overall Design

Compared with the centralized storage system, the distributed encrypted file storage system has greatly improved in security and performance, but its own architecture has become more complex. It not only expands the storage equipment, but also consists of a large number of software, including network services, storage systems, application software, storage clients, etc. at the same time, it needs to provide data encryption storage, user authentication and other services through a unified interface. In order to design an efficient and reliable distributed encrypted file storage system, this paper divides it into five modules. The user authentication module provides the user login interface and the relevant credentials requested by the user. The key management module provides operations such as production, query and deletion of file keys. The file processing module implements WebDAV protocol and provides file operation interface for users. The distributed storage module can store data, mainly including metadata management and data storage based on multi-attribute decision algorithm. The audit management module provides functions such as operation flow audit, display of user list, and configuration of system parameters.

Considering that the distributed encrypted file storage system is composed of multiple modules and the system complexity is high, if the single service architecture is adopted, many problems will be caused, such as: the code is bloated and the system startup time is increased; It is difficult to develop and test, and it is not easy to locate system abnormalities; The system has high coupling degree and poor fault tolerance. A small problem may cause the whole system to be paralyzed. Therefore, the system adopts a service-oriented architecture. In this mode, the application program is divided into multiple services, and each service provides an interface for data interaction. The communication is usually completed based on the HTTP protocol, which avoids the related problems in the single service architecture and has become the mainstream design architecture mode. At the same time, in order to better express the relationship between services, hierarchical architecture design is adopted to simplify the system design, reduce the dependency between services and increase the expansibility of the system. Each module in the architecture belongs to a certain level and provides services for the upper layer.

3.2. Hardware Mounting

This paper uses four cloud servers of Linux operating system as the test platform, and the software and hardware information are shown in Table 1.

Table 1. Software and hardware environment of test platform

Equipment item	Edition
Operating system	Ubuntu 18.04 server 64bit with ARM
Linux kernel	Linux version 4.15.0-70-generic
CPU	Huawei Kunpeng 920 2.6GHz
Memory	4GB
Hard disk	40GB
Network	1Mbit/s
MySQL	5.7.31-0ubuntu0.18.04.1
Redis	2.6.10

Package the developed program to generate executable program jar package, upload it to the Linux server, and start relevant services. In order to test the performance load of the system under concurrent requests, the Apache JMeter tool will be used for stress testing. This software is a desktop software developed based on Java language. The test objects can be static files, service instances, scripts, databases, etc. its principle is to simulate a large number of users' concurrent requests for test objects, and then analyze and display the response results. It has become a mainstream stress test tool. Based on this tool, we took the file display interface in the system with the most frequent requests as an example for testing. The test parameters are shown in Table 2, which represents the scenario of simulating 100 users accessing the service concurrently, and 100 tests were conducted.

Table 2. JMeter test parameters

Parameter name	Parameter value
Agreement	http
Route	WebDAV/74748EE139E5A432FF89313FB6D338D7A8B6D182AC935BB69BA40CC7B49A8EF69B7D6E7CB932DC42DEEB8CCA5B4F54
Type	GET
Number of threads	100

Number of cycles	100

4. System Performance Test and Analysis

4.1. Analysis of Pressure Measurement Results

Table 3. Test statistics

Number of samples	Average time	Minimum time	Maximum time	Abnormal rate	Throughput
10000	1584	254	3354	0%	59.3

Table 3 shows the test JMeter summary report results. The pressure test results show that the throughput (TPS) is 59.3, indicating that 59.3 requests can be processed per second. And the average time of each request is 15.8 milliseconds. The response speed of the system is basically met, and the data storage task can be well completed.

4.2. Summary and Analysis of Time Consumption

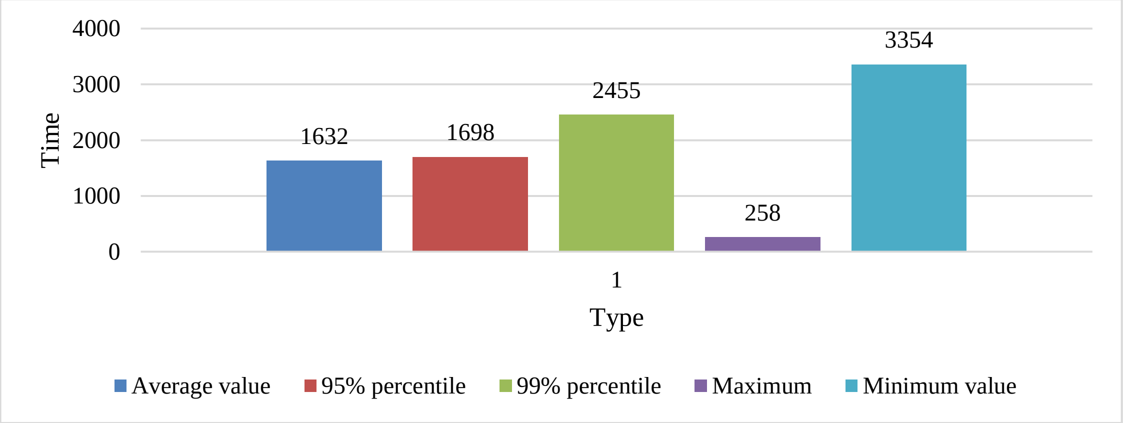

Figure 1. Time consumption summary

Figure 1 shows the summary of request time consumption during the whole test process. It can be seen from the figure that the average time-consuming value is approximately equal to the maximum time-consuming value of 95% of requests, indicating that the system processing speed does not change much for most requests.

4.3. Analysis of CPU Changes

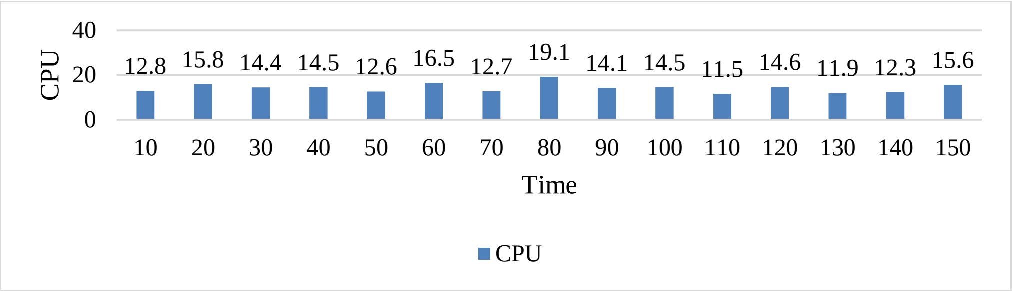

Figure 2. CPU change chart of file processing server

In the test, the average value of server CPU was 14.5%. Figure 5-8 shows the change of the server CPU. The average value of the CPU is calculated every 10 seconds. It can be seen from the figure that the CPU changes smoothly and the server runs stably.

5. Conclusion

This paper introduces the background and significance of distributed encrypted file storage system. On the one hand, the traditional data storage technology can not meet the needs of the current mass data storage, which is manifested in its inability to solve the limitations of the space capacity and performance load of hardware storage devices. On the other hand, the storage of private data usually needs to be encrypted for storage, and allowing the user to directly operate the key not only increases the risk of key loss and disclosure, but also makes the operation more cumbersome. We analyze the principle and related technologies of distributed encrypted storage technology and introduce several existing systems. Function and performance test of the system based on JMeter tool. The results show that wdcfs functions correctly, runs efficiently, is easy to use, and has good scalability.

Funding

This article is not supported by any foundation.

Data Availability

Data sharing is not applicable to this article as no new data were created or analysed in this study.

Conflict of Interest

The author states that this article has no conflict of interest.

References

[1] Parsons M A, Kara M Y, Robinson K M, et al. Early-Stage Naval Ship Distributed System Design Using Architecture Flow Optimization. Journal of Ship Production and Design, 2020:1-19.

[2] Klishin A A, Singer D J, Anders G V. Interplay of Logical and Physical Architecture in Distributed System Design. 2020,3(1):3-4.

[3] Zhang T, Song W S, Wang S, et al. Design of the Multi-Energy Complementary Distributed Energy System for Towns. 2021, 009(004):P.53-60. https://doi.org/10.4236/jpee.2021.94004

[4] Hou J, Wang J, Ji C, et al. A review of regional distributed energy system planning and design. International Journal of Embedded Systems, 2021, 14(1):89-89. https://doi.org/10.1504/IJES.2021.111984

[5] Chumnanvanichkul S, Suwanasri C, Wangdee W. Distributed Generator's Fault Ride-Through Capability Design with System Relay Coordination 2020 8th International Electrical Engineering Congress (iEECON). 2020,14(3):13-14.

[6] Ha A, Zo B, Maa B. A Long Short-Term Memory (LSTM)-Based Distributed Denial of Service (DDoS) Detection and Defense System Design in Public Cloud Network Environment. Computers & Security, 2021,3(2):1-2.

[7] Siritoglou P, Oriti G, Bossuyt D, et al. Distributed Energy-Resource Design Method to Improve Energy Security in Critical Facilities. Energies, 2021, 14(1):1-2. https://doi.org/10.3390/en14102773

[8] Makowski N, Campean A, Lambrecht J M, et al. Design and Testing of Stimulation and Myoelectric Recording Modules in an Implanted Distributed Neuroprosthetic System. IEEE Transactions on Biomedical Circuits and Systems, 2021, PP(99):1-1.

[9] Bednarski U, R Sieńko, Grygierek M, et al. New Distributed Fibre Optic 3DSensor with Thermal Self-Compensation System: Design, Research and Field Proof Application Inside Geotechnical Structure. Sensors, 2021, 21(15):50-51. https://doi.org/10.3390/s21155089

[10] Sallow A B. Design And Implementation Distributed System Using Java-RMI Middleware. Academic Journal of Nawroz University, 2020, 9(1):113. https://doi.org/10.25007/ajnu.v9n1a550

[11] Subiyanto S, Prakasa M A, Wicaksono P, et al. Intelligence Technique Based Design and Assessment of Photovoltaic-Battery-Diesel for Distributed Generation System in Campus Area. International Review on Modelling and Simulations, 2020, 13(1):63. https://doi.org/10.15866/iremos.v13i1.18147

[12] Ramdani F, Wirasatriya A, Jalil A R. Monitoring The Sea Surface Temperature and Total Suspended Matter Based on Cloud-Computing Platform of Google Earth Engine and Open-Source Software. IOP Conference Series: Earth and Environmental Science, 2021, 750(1):7-8.

[13] Ashammakhi N, Unluturk B D, Kaarela O, et al. The Cells and the Implant Interact With the Biological System Via the Internet and Cloud Computing as the New Mediator. Journal of Craniofacial Surgery, 2021, 32(5):1655-1657.

[14] Samriya J K, Patel S C, Khurana M, et al. Intelligent SLA-Aware VM Allocation and Energy Minimization Approach with EPO Algorithm for Cloud Computing Environment. Mathematical Problems in Engineering, 2021, 2021(6):1-13.

[15] Zheng P, Wu Z, Sun J, et al. A Parallel Unmixing-Based Content Retrieval System for Distributed Hyperspectral Imagery Repository on Cloud Computing Platforms. Remote Sensing, 2021, 13(2):176. https://doi.org/10.3390/rs13020176

[16] Amer D A, Attiya G, Ziedan I, et al. A New Task Scheduling Algorithm based on Water Wave Optimization for Cloud Computing. International Journal of Computer Applications, 2021(3),1-2. https://doi.org/10.5120/ijca2021921320

[17] Dubey A K, Mishra V. Analysis of Performance and Trust in Load Balancing Algorithm on Cloud Computing Environment. International Journal of Advanced Intelligence Paradigms, 2021, 1(1):1. https://doi.org/10.1504/IJAIP.2021.10030101

[18] Abdullayeva F J. Cloud Computing Virtual Machine Workload Prediction Method Based on Variational Autoencoder. International Journal of Systems and Software Security and Protection, 2021, 12(2):33-45. https://doi.org/10.4018/IJSSSP.2021070103