Key technologies of distributed file system for big data analysis

Wei Wan[1]

Abstract. With the continuous development of Internet technology, people have higher requirements for the ability of using computer to analyze and process data. In order to further clarify the way of effectively optimizing the distributed file system, the research status of the distributed file system at home and abroad was introduced first. The construction and application of the algorithm model were expounded. Finally, the data obtained from the system test was analyzed and the conclusions were drawn. The results show that the performance of distributed file system based on big data analysis is high and the time consuming is short. The results of the study were summarized by the author, and the relevant suggestions for optimizing the key technologies were proposed, so as to provide a strong basis for the effective application of distributed file system technology.

Key words. Big data technology, key technology.

1. Introduction

In recent years, the rapid development of big data technology has promoted the development of computer storage technology to a certain extent, and has led to the development of China's economy and society. The distributed file system is a data management system with better data storage and analysis ability. The stable operation of the distributed file system is inseparable from the application of big data technology. With the advent of the Internet era, people's demand for data diversity and structured applications arises at the historic moment. At the same time, the need for data storage technology is becoming more and more urgent. The distributed file system built by big data technology meets the demand of data storage management to a great extent. At present, more and more people use big data technology to manage large amount of files, and apply distributed system to store data and information. Distributed file system has been widely recognized by the majority of users, but with the rapid increase in the number of data storage, the bearing capacity of the distributed file system has become a concern. Therefore,

[1] School of Electronic Commerce, Jiangxi Institute of Economic Adminstrator, NanChang, Jiangxi, 330088, China

the study of distributed file system technology for big data analysis is of great significance.

Based on this, the key technologies of distributed file system for big data analysis were analyzed and studied by constructing algorithm model and system test method in this paper.

2. State of the art

The research on big data technology and distributed file system was originated in foreign countries. At present, foreign academia has made some achievements in the research of distributed systems. Our scholars' research on big data technology and distributed file system mainly focuses on the development of big data technology and the application of distributed file system.

The above researcher are the introduction and discussion of the development of distributed file system, although these studies have discussed the distributed file system in detail, the existing research results are mostly theoretical research of distributed file system, which lacks the research of the influence of big data technology on distributed file system. Therefore, in view of the shortcomings of the existing research, in this paper, the algorithm model and system testing methods were proposed, and the key technologies of distributed file system for big data analysis were analyzed and studied. In addition, in the third part, the specific content of the research object and algorithm model were expounded; in the fourth part, the algorithm model and system test data and results of data analysis were concluded; and finally, in the last part, the relevant conclusions were summarized.

3. Methodology

In this paper, the key technologies of distributed file system for big data analysis were mainly analyzed and studied by constructing Paxos algorithm model and system testing method. The distributed file system is a typical distributed system, and the architecture of the distributed system represents or determines the service capability of the system [1]. The emergence of large data technology effectively optimizes the design of distributed file system, and the system's difficult architecture design brings its huge service capability. The distributed file system refers to the physical storage resources managed by the file system, not necessarily directly connected to the local nodes, but connected to the nodes through the computer network [2]. Table 1 shows the distributed file system related issues and the main content of the main solution technology. The design of a distributed file system is based on the client server model. A typical network may include multiple servers for multi-user access. Specifically, a variety of distributed file systems can adapt to a certain application environment, and play a superior performance, which can well meet the requirements of the computing system on the storage system in the storage technology development at all stages. Song has believed that the design and implementation of distributed file system are based on a certain storage structure, which is divided according to the organization of storage media. Design and implementation of a distributed file system is a storage structure based on the storage medium according to the division of the organization, storage structure of the file system is divided into virtual storage structure and storage structure of object [3].

Table 1. Issues and techniques related to distributed file systems

Question	Relevant techniques adopted
Data equilibrium distribution	Improved consistent hashing algorithm
Data conflict processing	Vector clock
Temporary troubleshooting	Data return mechanism
Permanent fault recovery	Hash tree applications
Error monitoring	Error investigation

The Paxos algorithm is a consistency algorithm based on message passing. This algorithm is considered to be the most efficient of similar algorithms. In order to describe the Paxos algorithm, a virtual Greek city called Paxos is created. The island enacts laws in accordance with the political model of parliamentary democracy, but no one is willing to put all his time and energy on it. So neither the speaker nor the speaker or the clerk passing the note can promise to appear when someone else needs it, nor can promise to approve the resolution or pass the message. It is assumed that there is no Byzantine General problem, that is, although it is possible that a message is passed twice, but this will never be the wrong message. As long as waiting for enough time, the message will be passed to. Members correspond to each node, and the laws enacted correspond to the state of the system. Each node needs to enter a consistent state, and the consistency requirement corresponds to only one version of the legal [4]. Parliament corresponds to a distributed system, parliamentarians correspond to each node, and the laws enacted correspond to the state of the system. Each node needs to enter a consistent state, and the consistency requirement corresponds to only one version of the legal provision. The uncertainty of the MP and the attendant corresponds to the unreliability of the node and message passing channel [5] is

$$\lambda = \frac{1}{\mathrm{e}}\,. \tag{1}$$

The construction of Paxos algorithm model mainly includes metadata replication consistency and algorithm optimization and fault tolerance analysis and so on. The test of consistency of metadata replication should first define the failure rate according to formula (1), the time for data replication tasks should be set, and finally the reliability of the data should be calculated and estimated based on formula (2). Data reliability is the primary measure of metadata consistency testing. Consistency of metadata replication is the basis for the construction of the algorithmic model.

$$R(t) = \mathrm{e}^{-\lambda t}\,. \tag{2}$$

Other election of Axos algorithm with distributed computing as well as in the message passing algorithm based on two-phase commit process, due to the large amount of communication, the maintenance of data consistency process cost is too large, so that the big data platform using the consistency model, reducing the reliability and availability of the upgrade the space [6]. For example, due to its large amount of traffic, the two-stage commit process makes the process of maintaining data consistency too costly. As a result, the large data platform adopts a simple consistency model, which reduces the space for upgrading reliability and availability. Therefore, the optimization goal of the Paxos algorithm model is to reduce the traffic of the algorithm and improve the performance. The Paxos algorithm is optimized as follows:

Firstly, the core idea of the algorithm model is the basis of the operation, and the parameters of the distributed system are assumed and the research nodes of the server are found. Equation (3) is applied to the calculation and analysis of the damaged nodes of the distributed system. The calculation and analysis of the damaged nodes is mainly to clarify the basic data of the distributed system. At the same time, the analysis result of the damaged node can be used as a reference for the distributed system to reduce the communication loss, which is beneficial to the distributed system to improve its operation efficiency. At the same time, the operation of damaged nodes is the basic premise of system performance testing

$$R(t) = p\left\{T \geq t\right\} = 1 - \prod_{t=1}^{m} \left[1 - R_i\left(t\right)\right] . \tag{3}$$

Secondly, formula (3) is applied to analyze the frequency of fault occurrence and the probability of the specific situation of the algorithm model, and the results obtained can be used as a reference for the analysis of the running efficiency of the distributed file system. The problem of the distributed file system is caused by the number of nodes sending and receiving and the wrong path. And the failure problem is the key point of algorithm model optimization and improvement [7]. The fault problem is the key content of the optimization and improvement of the algorithm model. At the same time, the delay in the sending and receiving of node information is also a measure of the main indicators of the distributed file system. Formula (4) is used to calculate the delay time of the node information of the distributed file system, and determine the operating efficiency of the distributed file system through the obtained delay time. If the delay time is too long, then the distributed file system is less efficient. Once this happens, the algorithm model can be optimized and improved by modifying the algorithm model and repeating the calculation of the delay time [8].

$$R(t) = \mathrm{e}^{-\lambda t} \sum_{k=0}^{m-1} \frac{\left(\lambda t\right)^k}{k!} \tag{4}$$

Thirdly, the fault tolerance analysis of the algorithm model is mainly based on the formula (5). The fault-tolerant analysis of the algorithm model is based on the optimization of the algorithm model. The main process includes the following aspects: Firstly, the metadata is numbered and the data information is sent. Then, according to the situation of the delivery and feedback of the data information, the transmission and reception of data information is counted and the specific fault tolerance is obtained. The calculation of fault tolerance is mainly to measure the accuracy of the application of the algorithm model, so as to ensure the effectiveness of the applied algorithm model. In the process of calculating the fault tolerance of the algorithm model, the algorithm parameters may appear in the case of infinite loop competition, because any one of the larger parameter values will always repeat the same motion with the larger number. Therefore, the fault tolerance analysis of the algorithm model must be based on the reasonable setting of the algorithm parameters.

$$R(t) = \sum_{j=k}^{m} \left(\begin{array}{c} m \\ j \end{array} \right) \mathrm{e}^{-\mathrm{j}\lambda t} \left(1 - \mathrm{e}^{\lambda t}\right)^{m-j} . \tag{5}$$

Many large data analysis algorithms will eventually boil down to basic machine learning and data mining algorithms [9]. However, for dealing with large-scale data sets, many existing serialized machine learning and data mining algorithms are difficult to complete model training and data processing within an acceptable time frame. The Paxos algorithm proposed in this paper takes full advantage of big data to analyze the advantages of application algorithms, and the model constructed by this algorithm largely reflects the ability of big data analysis technology to deal with massive complex data. The design goal of big data distributed file system for large distributed data storage system design, and provide a general scalable distributed cache scheduling framework, through the framework of the big data used different data access patterns and caching strategies and upper integration, and provides a set of efficient scheduling strategy covering many different data access mode for the user, thus speeding up the application of the upper big data read and write data access performance [10]. Through this framework, different data access patterns and cache policies and large data applications on the upper layer are integrated, which can provide users with a set of cache scheduling policies that cover a wide range of different data access patterns, thus speeding up the read and write access performance of big data applications in the upper layer. Figure 1 shows the overall structure of the big data distributed file system. The distributed file system cache scheduling framework has very good versatility and independence, its architecture, access mode and cache strategy can be applied to any distributed storage system.

The operation and algorithm testing of the Paxos algorithm model requires an asynchronous reliable communication system. The transmitted data may be lost, delayed or repeated, but will not be tampered with. In this paper, the ordinary computer was chosen as the server, and the corresponding node was selected to analyze and study the key technologies of the distributed file system. In the course of the experiment, the data of the node was tested and the delay and loss were tested to ensure the credibility of the system test, and then the consistency of the communication node was checked. After the experiment, the number of nodes was controlled through the error mechanism, so as to effectively shorten the communication time and improve the communication efficiency. The algorithm model was

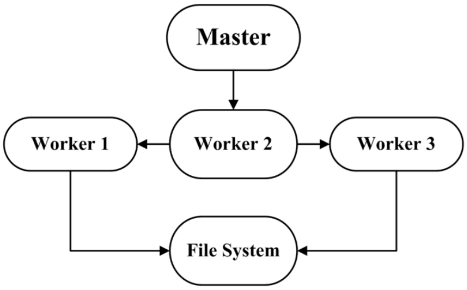

Fig. 1. Framework for big data distributed systems

applied to analyze the distributed file system of big data analysis, so as to effectively utilize the computer resources and improve the system availability on the basis of comprehensive understanding of node loss.

4. Result analysis and discussion

Different distributed file systems were set up in the computer A and B inside, and simulation experiments and tests were carried on for reading and writing large numbers of files. Computer A was set up the distributed file system with big data analysis, computer B was set up in general distributed file system. It was assumed that each node configuration of the simulation experiment was memory 4 G and hard disk 500 G. The simulation results obtained by running the file system are as follows:

Table 2. Comparison and analysis of system operation time

Number of read and write files	Time consuming of A system	Time consuming of B system
1	375	1190
3	1239	3465
5	2004	5987
7	2689	7982
9	3578	9861
11	4538	12445

The time-consuming situation of the operation of the distributed file system directly determined the operating efficiency of the system. As shown in Table 2, the time-consuming index obtained by the A-computer system test proposed in this paper was significantly lower than the time-consuming index obtained by the B-computer system test. In other words, the time-consuming index of big data distributed systems was only about one-third of the time consumed by other general distributed file systems. Thus, the operation time of big data analysis distributed file system was smaller, which reflected the good operation efficiency of big data analysis distributed file system to a certain extent.

Based on the above analysis of the operational advantages of the big data analysis distributed system, the performance test of the system operation of the computer A which had the big data analysis distributed file system was carried on. The results obtained by performing performance tests on different system operations under different numbers of servers are shown in Table 3. As can be seen from the above data, the performance of the new, access to information, delete, and rename the directory to create the metadata system operation were increased with the increase of the server. Among them, the highest increase in performance was the directory creation and renaming of two system operations, which increased by 159.32 % and 156.45 %, respectively. The performance gains of other system operations were slightly lower than those of the two system operations. Overall, the increasing number of servers has greatly improved the performance of metadata.

Table 3. Metadata performance improvement test analysis

Operate	Newly built	Get information	Delete	Directory creation	Rename
1	159.32	156.45	–3.92	–9.23	–7.37
3	112.12	147.25	5.13	6.33	5.16
5	103.56	136.92	6.51	8.30	8.29
7	100.73	116.56	7.7	9.52	9.05

Distributed file system is mainly used to store files. Therefore, the system needs to meet the needs of users to read and write files. Fig. 2 shows the changes in the reading (up) and writing (bottom) performance indexes of distributed file system before and after the application of big data technology.

As shown in Fig. 2, A1 and B1 is the application of big data technology optimization computer A and B system to read and write performance curve, A2 and B2 after the application of big data technology to computer A and B system to read and write performance curve, System is a typical distributed file system data read and write performance curve. From Fig. 2 it can be seen in the data, before the application of big data technology, the B1 performance index was less than the A1 performance index, the number of visible, reads the metadata server computer system B is significantly lower than that of A computer system, and the value of A2 and B2 two performance curves are almost equal, visible, of B computer system to optimize the application of big data technology with the computer A system to read and write approximately the same performance level, distributed file system of the computer built-in read metadata server capability has been significantly improved. Thus, the application of big data technology greatly promotes the performance of metadata reading and writing.

Distributed file systems need to complete different commands and tasks. The performance changes of the main system operations under different commands and tasks are shown in Table 4 and Fig. 3. The performance improvement ratios of three operations of deleting, obtaining information and renaming under different metadata processing tasks were 2.17 %, 4.55 % and 1.09 % respectively. The dynamic

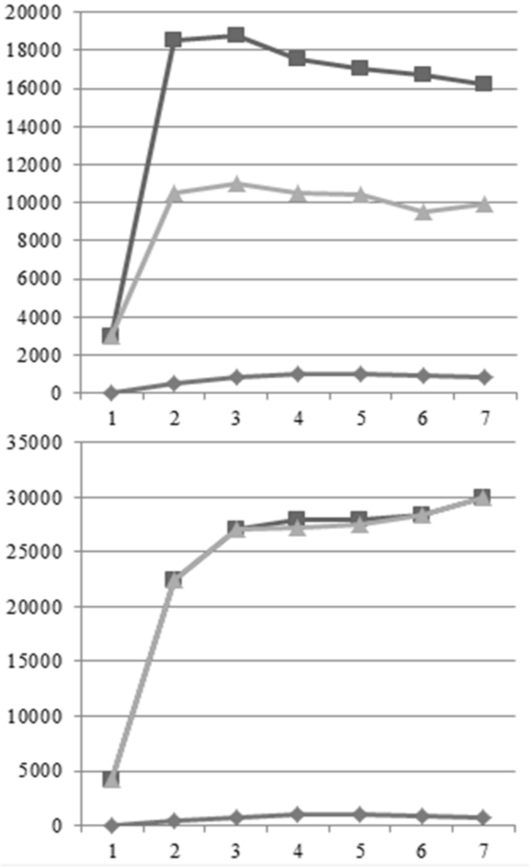

Fig. 2. Comprehensive test analysis of metadata read and write performance

trend of other performance was not stable, but overall, most of the performance index was slightly improved, the performance index declined, and the decrease in each operation was smaller, which was not more than 7 %. In summary, the main operational performance index of the distributed file system optimized by big data technology was higher, the overall performance of the distributed file system with big data analysis was better. Therefore, it was believed that on the basis of ensuring the normal operation of metadata processing, the application of big data technology can further enhance the scalability and usability of distributed file system. However, big data technology plays a less active role in other random operations of distributed systems. The improvement of mixed performance of distributed systems for big data analysis can be carried out in two aspects: enhancing system metadata throughput and reducing metadata consumption, thereby improving the overall performance of distributed file systems and promoting the continuous development of distributed file systems for big data analysis.

5. Conclusion

In order to find out the way to optimize the distributed file system by using the big data technology, the method of Paxos algorithm model was constructed, the metadata performance of distributed file system with large data analysis was systematically tested, the optimization of distributed file system was studied. Finally, the main conclusions were obtained as follows: the time-consuming of each operation of the distributed file system for big data analysis proposed in this paper is relatively short; at the same time, the overall performance level of the distributed file system optimized with big data technology is high.

Table 4. Analysis of metadata operations under different tasks

Task	Newly built	Get information	Delete	Directory creation	Rename	Build page	Fault feedback
Count	0.99	0.99	0.99	0.49	57.14	32.51	0.99
Text search	1.70	2.11	1.69	0.85	50.00	32.20	1.70
Random input	2.17	2.17	2.17	1.09	0.00	7.61	1.09
Sort	2.21	1.77	1.77	0.88	14.60	15.93	1.33

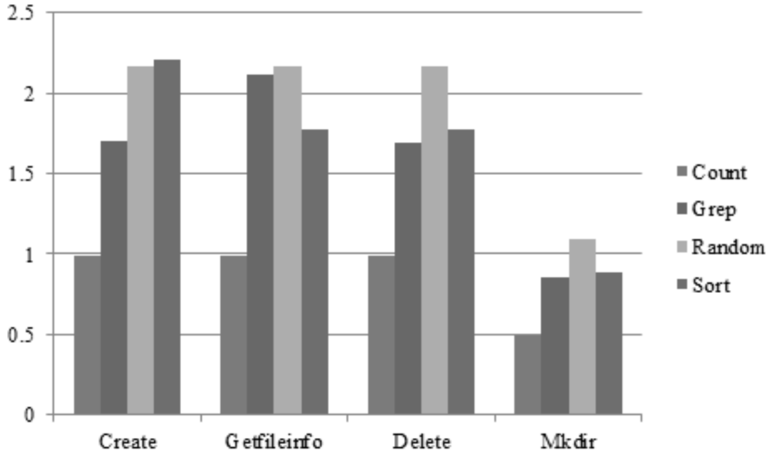

Fig. 3. Metadata performance comparison of different tasks

In summary, the Paxos algorithm model proposed in this paper is simple and reliable, and the application of the model is helpful to further analyze the effectiveness of the distributed file system. However, the research on big data analysis of distributed file system through the application of Paxos algorithm model lacks an analysis of the system user's visit. It can be seen that the algorithm model has some shortcomings in the application of comprehensiveness. Therefore, in the future research, it is possible to add the operation and system test of user access performance index in the process of model analysis, so as to provide a more favorable basis for the application feasibility of Paxos algorithm model.

References

[1] R. C. Ionescu: *A scalable system for primal-dual optimization.* Journal of Computer Research and Development arXiv:1507.01456v1, 2015.

[2] J. Wang, X. Zhang, J. Zhang, J. Yin, D. Han, R. Wang, D. Huang: *Deister: A light-weight autonomous block management in data-intensive file systems using deterministic declustering distribution.* Journal of Parallel and Distributed Computing *108* (2017), 3–13.

[3] J. Song, C. Guo, Z. Wang: *HaoLap: A Hadoop based OLAP system for big data.* Journal of Systems & Software *102* (2015), 167–181.

[4] E. A. Murray: *Method of securing files under the semi-trusted user threat model using symmetric keys and per-block key encryption.* US Patent US 9363247 B2, 2016

[5] J. Li, C. Pan, M. Lu: *A seismic data processing system based on fast distributed file system.* International Journal of Computers & Technology *14* (2015), No. 5, 5779–5788.

[6] M. Alam, S. Sethi: *Covert Channel Detection framework for cloud using distributed machine learning.* CoRR - Computing Research Repository - arXiv (2015), abs/1504.03539.

[7] Y. Cao, D. Sun: *Migrating large-scale air traffic modeling to the cloud.* Journal of Aerospace Information Systems *12* (2015), No. 2, 257–266.

[8] Z. Wang, Q. Ding, F. Gao, D. Shen, G. Yu: *iHDFS: A distributed file system supporting incremental computing.* IFIP Advances in Information & Communication Technology *503* (2015), 151–158.

[9] S. Saranya, M. Sarumathi, B. Swathi: *Dynamic preclusion of encroachment in Hadoop distributed file system.* Procedia Computer Science *50* (2015), 531–536.

[10] M. Mirakhorli, J. Cleland-Huang: *Detecting, tracing, and monitoring architectural tactics in code.* IEEE Transactions on Software Engineering *42* (2016), No. 3, 205–220.

Received July 12, 2017