Design and development of All-in-one computer for distributed file system

Lin Qian[1,a], Yan Chen[2,b], Jun Yu[1], Guangxin Zhu[1], Hengmao Pang[1], Xigao Li[1]

[1]Information system integration company, Nari Group Cooperation, Nanjing 210000, China

[2]State grid Shanghai municipal electric power company, Shanghai 200122, China

[a]qianlin@sgepri.sgcc.com.cn

[b]chen-yan@sh.sgcc.com.cn

Keywords: distribution, single failure, all-in-one computer, DFS

Abstract. Application requirements and the business volume increased significantly, considering the performance bottleneck and single point failure and other problems, distributed file system gradually will replace centralized storage. This paper discussed the independent designation and development of all in one machine with DFS, deeply customized software and hardware resources, it not only provides huge storage space, can also provide efficient and easy to use safe and reliable service for large scale data and high load business system, and also support high performance and flexibility to expand the file sharing storage platform.

Introduction

With the increase of company size and business volume, the associated development environment and business data also grow exponentially. In this situation, an independent server storage space's shortage has become a more prominent issue at present. Although there are a number of large-capacity internal servers can be used to save data, but the servers are usually located in different rooms, so there are difficulties in the management and organization and it can only meet the storage needs for short time. Therefore, it requires the free space on multiple machines to organizations to use to provide more disk space to break out single storage.

At present, a large number of enterprises adopt NFS solutions that use shared storage server to store network data to cope with explosive growth. NFS solutions in small scale and small traffic can meet the demand, but with the increasing amount of data and access amount, this method appeared to be inadequate and led to a sharp decline in performance and user experience. In addition, centralized storage need to ensure that no stopping computer completely, or its data-dependent applications will not be available.

Taking into account the performance bottleneck and single point of failure and other problems, distributed file storage system host is gradually replacing centralized storage position. In a distributed file system, the relationship of data between servers is from one-to-multiple to multiple-to-multiple. It can significantly improve performance and reliability. All in one machine with DFS has amount of storage space, and support for file shared storage platform with flexible expansion, high-performance access, and provide efficient, easy to use, safe, reliable service for large data traffic and high concurrent access applications.

Research Status

Status analysis of distributed systems. With the increasing of the storage capacity of the explosive growth of data and hardware technology, distributed file system has become a more active research field. Many domestic and foreign universities and research institutions as well as companies have been working to develop their own distributed file system to take advantage of the cluster system. Meanwhile, some in the open source community has also developed a number of distributed file system can be run on Linux operating system. These open source projects have greatly contributed to the development and application of distributed file system [1].

Currently, using MapReduce framework to build large-scale parallel data-intensive applications is the main technique of store mass data to distributed storage cloud. It also needs distributed file systems such as GFS, HDFS, etc. to store and manage vast amounts of data distributed and provide higher polymerization I / O bandwidth to support a large number of clients and the data of PB-class [4-6]. The rapid development of the cloud computing era and the explosive growth of data, distributed parallel file system with its large storage capacity, high polymerization I / O performance, high data fault tolerance and scalability comply with the information explosion of growth in demand. It is now become effective solution to solve the high-performance computing systems massive data storage and program I / O bottlenecks problem.

There are dozens of distributed file system solutions to choose, such as MFS, lustre, Hadoop, TFS and so on, some of which have been more widely used. The Inspur's Nehalem-based server platforms verify the function of MooseFS distributed file system, and evaluate the performance reliability and stability of the file system. At present, the solution which takes the InspurNF5280 as the core applied in Jilin cultural information resources sharing project successfully; Furthermore, Hadoop has also been a concern of many domestic and foreign software vendors, and it has been applied in e-commerce (eBay is one of the greatest practitioners), energy extraction (Chevron, second-largest oil company in US has been adopted) and mobile data (in US,70% of service support behind the smart phone data) and so on.

Comparison of mainstream Distributed Systems. Business system will generate a lot of large files of unstructured data such as images, GIS location information files, audio, video streaming and so on. Dealing with these data effectively will greatly enhance the performance of our information systems, improve the quality of service and reduce the costs of information systems operation and maintenance.

For distributed file system, we test different DFS mainly from two aspects include the functional and non-functional, including ongoing high concurrent data write performance, simulating various fault to test high availability, increasing the number of data nodes to test the degree of improved performance to obtain performance data for each distributed system and verify its fit to the existing business requirements.

A. HDFS distributed file system

HDFS has a feature of high fault tolerance, and it is designed to be deployed in low-cost hardware. And it provides high throughput to access to application data for those applications with a large data set. HDFS relaxes the POSIX requirements, which allows for streaming access to the data in the file system.

The reliability of HDFS, after a long period of uninterrupted high concurrency test, the file system is still very stable, and it can provideexternal7*24 hours service. System have no single point of failure. In case of any node fails, HDFS can switch automatically to restore service to achieve rapid results.

B. TFS Distributed File System

Taobao File System is internal distributed file system used in Taobao. It is a high scalable, high availability, high performance, Internet-service-oriented distributed file system, which is designed to support the mass of unstructured data". TFS makes a special optimization for access performance of read and write randomly of massive small files. It carries Taobao's master photos, product description and other data storage [2]. The system uses the flattened organizational structure of data completely, abandoning the traditional file system directory structure; and it establish its own file system based on block device, reducing performance loss caused by fragmentation of file systems such as EXT3; monolithic single-process management disk mode abandoned RAID5 mechanism; the central control node with HA mechanism strike a balance between security, stability and performance complexity; try to reduce the size of the metadata, and make the metadata loaded into memory to improve access speed; cross-rack and IDC load balancing and redundancy security policy; smooth expansion completely.

TFS has an advantage in its high availability, but its versatility and user interface has some deficiencies. Currently, TFS can only support the application of small files, and because of the algorithm mechanism, writing files for Client are synced, whose return needs all the data servers writing successfully, which greatly reduces the performance of the system. In addition, TFS has its own

file naming convention. If users use their file name, he needs to maintain their own mapping between the file names and TFS file names, which greatly increased the burden on the business staff.

C. MFS Distributed File System

MooseFS (MFS) is a network distributed file systems. It spreads data across multiple servers, but for users, they can see only one source [3]. MFS is also like other unix file systems, which including the hierarchy structure(tree), stores the file attributes (authority, last access and modified time), establishes a special file (block device, character device, pipes, sockets), symbolic links, hard links.

MFS provides a common file system without modifying the upper application; online expansion, highly scalable architecture; simple deployment; high-availability architecture, with no single point of failure for all components; highly available file objects(it can be arbitrarily set the degree of redundancy of the file and will not affect the performance of the read or write); providing windows Recycle Bin; providing garbage collection (GC)similar as java; providing commercial storage snapshot including netapp, emc, ibm and so on; providing web gui monitoring interface; improving the efficiency of read or write randomly; improving the efficiency of massive small files writing, but the efficiency of large files for reading and writing is not very good. In the current test, the read and write speeds for large files are usually at 180Mb / s .

For these distributed file system as above, we carried tests was under different circumstances of file size and the number of concurrent. The test results are shown in Table 1.

Table 1 file system performance test data tables

File system type / Concurrent	MFS	MFS	HDFS	HDFS	TFS	TFS
1000	70M/s	66M/s	55M/s	125M/s	103M/s	NULL
2000	167M/s	117M/s	110M/s	260M/s	217M/s	NULL
3000	250M/s	180M/s	200Mb/s	400M/s	360M/s	NULL
File type	Small files (16k)	Large files (1G)	Small files(16k)	Large files (1G)	Small files (16k)	Large files (1G)

Algorithm and Design

System Design. All in one computer of distributed file system should meet business needs. System should meet secure storage of different business file and trouble-free operation system based on the functional requirements, and the system must provide a friendly user interface. Specific features are as follows:

1. Distributed file system based network-coding.

The client use unified standard interfaces to provide services. To improve the performance of fuse, we need converted the size of file block to the coding matrix, that is, the system can automatically generate the appropriate (n; k) erasure code matrix depending on the file size, the user's policy selection to applied in coding operation. In order to make block Size dozens MB of magnitude, the system takes the form of large blocks of data file organization. It cut files with the transfer unit to improve system performance.

Meanwhile, for some accidentally deleted files, all in one computer can be recovered from the metadata, ensuring the maximum secure storage of user data.

2. High Availability Framework

Highly available distributed file management system designed mainly has three parts, which are the file system design of the basic functions, control system design and management platform design. Among them, the control system functions account for the major part of the design, it is divided into Agent subsystem, MON subsystem and Collector subsystem. The three sub-systems achieve integrated management of the entire machine through mutual surveillance and message communication mechanism to ensure all in one computer Real-time normal operating condition.

When some managed nodes shoot down their service, re-access the new machine, we can use high-availability network communications to complete the replacement of the machine, without stopping service of the whole of all in one machine to ensure availability of services and applications.

3. Monitoring and Management Platform of All-in-one Machine

Self-developed all in one computer management and monitoring platform comprises six functional test modules, which are home module, real-time monitoring module, resource management module, application management module, operation and maintenance management module and system management module. Each module displays different business model, it can clearly grasp the relevant information user-friendly and operate easily.

4. Disaster Recovery

The basic solution for disaster recovery is to build a remote disaster recovery redundant node, ensuring the continuity of business system. Distributed File System Design Framework Disaster Recovery solutions include local primary and secondary servers, local data servers, DNS servers, remote backup servers and remote data server. We syncing the local master server's log and remote data server backup data disaster recovery mechanism by remote backup servers to realize the disaster recovery.

Typically, disaster recovery system can withstand most events that may damage or destroy the current active data center. When a node paralyzed or destroyed accidentally, the system will switch to the disaster recovery side, so that the entire system can run normally.

Software and Architecture Design

1. Resource Management Subsystem Design

All in one computer use a hierarchical design for resource management architecture. The overall architecture composed with the interface layer, application service layer, business logic layer and the base framework layer. Among them, the interface layer will display various devices, servers, command and the relevant threshold. Application service layer shows the device application components, services, application components, command application components and the threshold application components, under which is the business logic layer. It shows the various parts of the components and their relationships. The bottom layer is the basic framework layer, including database operating components and XML file operations components.

2. File Operations Process design based on network-coding

In order to make the distributed file system use RScode or MBRcode in different situations, we designed a basic coding informational class CodingMatrix, erasure informational class RSCodingMatrixand regeneration code information class(MBRCodingMatrix and MSRCodingMatrix), shown as Figure 1. The coding scheme inherit all the basic coding informational class, so that to achieve a common, easily- expand target.

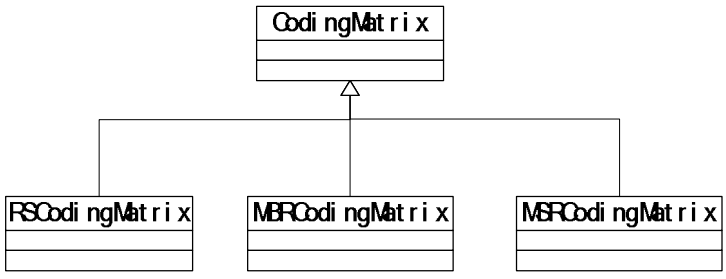

Figure 1 Generic coding CodingMatrix

When apply for writing a file in the system, you need to generate a coded matrix according to the file. The system will initiate a CodingMatrix class. CodingMatrixas a general class, it can refer to corresponding encoded information class according to different encoding scheme.

A. Writing Process Design

The client apply for file write operation request by launching a RPC remote call to node name, the node name combined with the corresponding permission checks whether the file to be created already exists. Name node determines file tile size based on file size, generating the appropriate coding matrix

coding Matrix, and then record in the INode index file. Then, INode index file has no corresponding list with the corresponding Block. File encoding process traverse the entire document in accordance with the transmission unit (hereinafter referred to as Packet).Based on the relationship in coding matrix, we put the Packet into corresponding coding cached domain forwaiting or coding. Every k times of traversal, we put the corresponding code cache into network transmission queue. Wake up the waiting file block transfer process when the queue is not empty.

B. Read Process Design

File reading process is shown in Figure 2, the client initiates RPC file read requests to remote name node; then the name node returned to the client located Blocks objects, the client according to certain policies (such as the recent network distance, IO, etc.) to choose k most recent data nodes to initiate simultaneously a connection request and to maintain k connections. K connections are managed via k data prefetching module, and the decoding module obtained data by the prefetching module without interacting with the underlying decoding function.

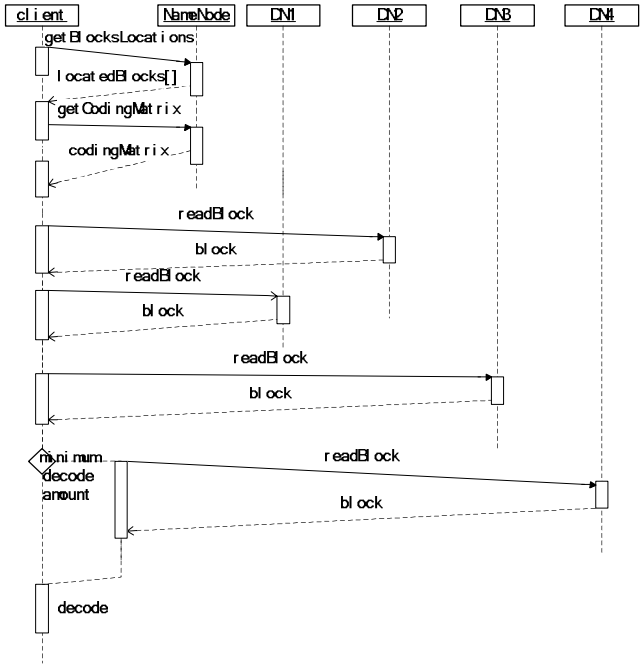

Figure 2 file read process

C. file repair design

When the data node monitor block failure, it will report to the name node by calling name node's block Report function via RPC; When the name node receives the block failure, it can select a data node as a new storage node new Comer according to the load balancing policy, and send commands to those data nodes participated in the repair function. After the data node receive the command, it will judge the coding scheme involved in repairing Block and then transfer data to new Comer according to the appropriate strategy. New Comer recover the block failure according to the data it received, and then add the block to received Block List waiting for reporting to the name node Name Node the next heartbeat.

3. Disaster Recovery Design

Distributed File System Disaster Recovery solutions include three scenes include working scenario, standby node failure scenario and standby node repair scenario. Public function is to carry out basic functions to achieve part of the disaster recovery.

Distributed Disaster Recovery of all in one computer includes data-level disaster recovery and application-level disaster recovery. On the data level, all in one computer increases remote Metalogger

server, ChunkServer and DNS server. And the remote Metalogger synchronize the metadata information of local Master. At least one off-site data in Data distributed file system backup in ChunkServer, so that it can ensure that data will not be lost when a local server fails.

At the same time, users get some functional services of Master server by access to a DNS server, and the address of the Master server is transparently for the user, that is, when the local server is down, remote Metalogger will start server function of Master, and the user is unaware of the changes of system architecture, so that it ensure uninterrupted service to achieve application-level disaster recovery.

Summary

Distributed file management machine has many features like efficient, fault tolerance, easy to use in distributed file management platform, integers soft and hard resources to provide integrated solutions for non structured data storage. This all in one machine realized effective method accessing to large amounts of non structured data, but also realized high redundant and remote disaster recovery system, and finally make information system run stably and efficiently.

References

[1] Z. Wenjing. Design and implementation of storage management of unstructured data management system MyBUD[D].Beijing: Renmin University of China, (2011).

[2] Information on http://tfs.taobao.org/

[3] X.Yun, Ai, Y.S.Tan, and J.Y. Wang. "Chunkserver load balancing selection algorithm on MooseFS." Microcomputer & Its Applications 5 (2013): 003.

[4] C, M.Shi, and S. W. Schlosser. "Map-reduce meets wider varieties of applications." Intel Research Pittsburgh, Tech. Rep. IRP-TR-08 5 (2008).

[5] Z, Ning, et al. "Towards cost-effective storage provisioning for DBMSs." Proceedings of the VLDB Endowment 5.4 (2011): 274-285.

[6] K, I, and S. D. Viglas. "Flashing up the storage layer." Proceedings of the VLDB Endowment 1.1 (2008): 514-525.