Cloud Data Storage Technology and Its Architecture Implementation

Shubham Kumar, Nilesh Kumar, Anjani Kumar, Anikesh Kumar
School Of Computer Science & Engineering Lovely Professional University
Phagwara, Punjab, India

***Abstract* —** The concept of cloud computing has become more and more popular in recent years. Data storage is a very important and valuable research field in cloud computing. This paper introduces the concept of cloud computing and cloud storage as well as the architecture of cloud storage firstly. Then we analyze the cloud data storage technology--GFS(Google File System)/HDFS(Hadoop Distributed File System) towards concrete enterprise examples. In the last part, we illustrate how to improve the traditional file storage method based on eyeOS Web operating system which realizes file distributed storage and fault-tolerant control through HDFS technology of Hadoop.

Keywords: Cloud Computing; Cloud Storage; Web Operating System; Distributed File System;

1. INTRODUCTION

In recent years, the concept of cloud computing has become more and more popular. Cloud computing as a a new business model is developed from distributed processing, parallel processing and grid computing. At present, **Google, Amazon, IBM, Microsoft, Sun** and other IT giants are all seeking to develop cloud computing technologies and products. For example, Google has been dedicated to promoting application engines based on the techniques of GFS [1] (Google File System), MapReduce [2], BigTable[3] and so on, which provide users methods and means to process massive data. In this paper, we introduce the concept of cloud computing and cloud storage as well as the architecture of cloud storage firstly, analyze the cloud data storage technology—GFS and HDFS (Hadoop Distributed File System) under the specific cases of enterprises, and build the cloud storage architecture through eyeOS Web operating system in our Computer.

2. *CLOUD COMPUTING* AND CLOUD STORAGE

2.1. CLOUD COMPUTING DEFINITION

Cloud computing arises from the combination of the traditional computer technology and network technology, such as grid computing, distributed computing, parallel computing, utility computing, virtualization. One of the core concept of cloud computing is reducing the processing burden on user's terminals through continuously enhancing the clouds' handling capacity. Eventually user's terminals are simplified into simple input and output devices. Users can use the powerful computing and processing function on clouds and they can order their service from the cloud according to their own needs.

2.2. CLOUD STORAGE DEFINITION AND IT'S ARCHITECTURE

Cloud storage is a system that provides functions such as data storage and business access. It assembles a large number of different types of storage devices through the application software which are based on the functions of the cluster applications, grid techniques, distributed file systems, etc. Cloud storage can be simply understood as the storage in cloud computing, and also can be considered to be a cloud computing system equipped with large capacity storage. Cloud storage system architecture mainly includes storage layer, basic management layer, application interface layer and access layer.

3. *CLOUD STORAGE* TECHNOLOGY OF ENTERPRISES

3.1. GFS [1]

1)* SYSTEM *ARCHITECTURE

A GFS cluster consists of a single master, multiple chunk servers and multiple clients, as shown in Figure 1(a). Each of these is typically a commodity Linux [1].

- **GFS Master:** Master manages all file system metadata and the files directory structure. GFS uses a single master policy which means in the same time only one master provides services so that it can avoid extra costs for coordinating between multiple masters synchronously. A client interacts with the master only for metadata, and interacts with the chunk servers directly for all other data.
- **CHUNK SERVER**: GFS files are divided into fixed-size chunks stored on each chunkserver and the default block size is 64M. Each chunk is identified by an immutable and globally unique 64 bit chunk handle assigned by the master as soon as the chunk is created. Each block is replicated on three chunk servers.

Users can set different replication levels for each region of the file namespace. As shown in Figure 1(a), there are four chunk servers and five chunks as C_0-C_4. Each chunk is saved on three chunk servers. • Client: GFS client code linked into each application implements the file system API and communicates with the master and chunk servers to read or write the master for metadata operations, but all data-bearing communication goes directly to the chunk servers[1].

2) WORKFLOW

As shown in Figure 1(a), thin solid lines represent the control information between clients and master or between master and chunk servers, thick solid lines represent the data communication between chunk servers and client, dashed lines indicate the control information between clients and chunk servers. Firstly, clients compute chunk index from files structure and chunk size, then send file name and chunk index to master (mark①). Secondly, master sends chunk handle and chunk locations to clients (mark②). Thirdly, clients send chunk handle and byte range to the nearest chunkserver (mark⑤). Finally chunkserver sends data to client (mark⑥).Once clients get chunk locations from master, clients do not interact with the master any more. Master does not permanently save the mapping from chunkserver to chunk.

Instead, it asks each chunkserver about its chunks at master startup or whenever a chunkserver joins the cluster (③④). The master periodically communicates with each chunkserver in HeartBeat message to give it instructions and collect its state (③④).

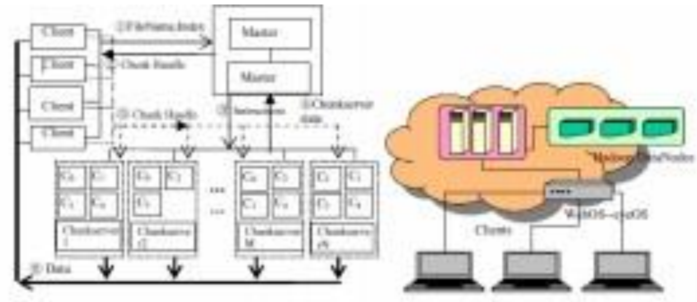

Fig.1.(a) GFS Architecture; (b) System Architecture

3.2. HDFS

Hadoop is hosted by the Apache Software Foundation, which provides support for a community of open source software projects. Although Hadoop is best known for MapReduce and its distributed file system (HDFS), the other subprojects provide complementary services, or build on the core to add higher level abstractions. The detailed contents refer to document [4].

The full name of HDFS [5] is Hadoop Distributed File System. HDFS is run on large clusters of commodity hardware and is like GFS of Google. The architecture of HDFS is master/slave and a HDFS cluster has one namenode and multiple datanodes. Namenode is the central server, equivalent to master in GFS. It is responsible for the namespace operation of file systems. Datanode is similar to chunkserver of GFS which is responsible for managing storage on datanodes, creating block, deleting block, copying block and etc. The files in HDFS are divided into one or multiple blocks which are stored in datanodes. Namenode and datanodes can be run on the low-cost Linux computer. HDFS is developed by the Java language.

4. CLOUD STORAGE *ARCHITECTURE* BASED ON HADOOP

4.1. EyeOS

EyeOS is a web desktop environment with office software and personal information management systems, and it enables the online storage, mobile office. Document management in eyeOS is simply stored in a single server, without fault-tolerant backup features and reliability is poor. Accessing files is a single thread and access performance is not high. In this paper, we improve the traditional file storage method and achieve file distributed storage as well as fault-tolerant control using HDFS technology.

4.2. SYSTEM IMPLEMENTATION

1) **ARCHITECTURE**

The storage system we designed is shown in Figure 1(b) which includes clients, web operating system eyeOS, cloud server (NameNode), cloud storage center (DataNode).

• **CLIENTS:** Each client is only pre-installed with a web browser and users log in this cloud storage through the web browser. Clients are the interface between users and cloud storage systems.

• **WEB OPERATING SYSTEM:** Web operating system receives users' access requests, verifies the users' validity, and interacts directly with the clients. It is based on eyeOS which offers a large number of applications to users. Users can download their required applications and achieve a personalized system. EysOS is also the file access interface for users and files can be saved in the cloud storage clusters by this interface.

• **CLOUD SERVER (Cloud NameNode):** Cloud storage cluster based on Hadoop includes cloud server (NameNode) and cloud storage center (DataNode). Cloud server is the namenode in Hadoop which manages file system namespace, computes the mapping from files to datanodes, allocates datanodes to save file blocks, and control external clients' access.

• **CLOUD STORAGE CENTER (Cloud DataNode):** Cloud storage center is datanode in Hadoop. It is in charge of saving files, realizing file distributed storage, ensuring load balancing, files fault-tolerant and etc.

2) **OPERATION PROCESS**

Users' operations based on eyeOS are writing files and reading files. When reading a file, we download the file to the local computer, then handle or display the file using the application software in the web operating system. When the files are modified and saved, the web operating system uploads them to the cloud storage system from the local computer. • Reading files process: ①Users log in the web OS from client through clients' browser and doubleclick a file icon on the web OS. Then eyeOS requests the file from the Hadoop namenode.② Namenode finds the related information of files, and computes the file's location. Datanodes which saved the blocks of the file send the blocks to the clients.③Clients download the file blocks from the datanodes and merge these blocks into a file.④ Applications associated with the file in the web operating system auto start and display the file.

• **WRITING FILES PROCESS:** ①Users log in web OS from client's web browser modify and save files using the selected application. EyeOS requests uploading files to Hadoop namenode. ②Namenode allocates storage space to data nodes according to the file size and the datanodes' storage condition after it received the uploading request. ③Clients upload file. Namenode divides it into one or multiple blocks and saves it in the allocated datanodes.

5. EXPERIMENTS

These experiments use five computers. Three are used as client, eyeOS, namenode respectively and the other two are used as datanodes. We assume the datanodes are Da and Db. There are files named FileX and FileY in Da and Db.

The experiments are done when Da and Db are always normal. As shown in Table 1, when creating File1, this file is saved in Da and Db at the same time. When deleting FileY, this file in Da and Db are all deleted. These means datas in invalid datanode will be updated automatically when the datanode recovers normal and data in the datanode are always latest.

We do experiments when Db is always normal but Da is invalid and the results are shown in Line 3 of Table 1. Then when Da recovers normal and the results are shown in Line 4 of Table 1.

• **Creating Files:** When creating File1, we can find the file in Db but can't find it in Da. If Da recovers normal this moment, we can also find File1 in Da. •

Deleting Files: When deleting FileY, the FileY can't be found in Db but can be found in Da. If Da recovers normal now, FileY is deleted from Da immediately.

Table 1.Experiments' Results

Operation	Creating file 1	Creating file Y
Da(Normal) Db(Normal)	File 1 can be found in Da and Db	.File Y are deleted from Da and Db
Da(Invalid) Db(Normal)	File 1 is saved in Db, but isn't saved in Da	File Y is deleted from Db, but not deleted from da.
Da(recove rs normal after invalid) Db(Normal)	File1 is saved as a duplicate file in Da automatically.	File Y is deleted from Da

6. CONCLUSION

Cloud computing is the inevitable product with the development of the internet, and it also brings more rich applications to the internet. Cloud data storage technology is the core area in cloud computing and solves the data storage mode of cloud environment. In this paper, we introduce the related concepts of cloud computing and cloud storage. Then we pose a cloud storage architecture based on the eyeOS web operating system in our computers. Experiments verified the system is well.

REFERENCES

[1] Sanjay Ghemawat, Howard Gobioff,Shun-Tak Leung. The Google file system[C]. Proceedings of the 19th ACM Symposium on Operating Systems Principles. New York: ACM Press, 2003:29-43.

[2] Jeffrey Dean, Sanjay Ghemawat. MapReduce:Simplified data processing on large clusters[C]. Proceedings of the 6th Symposium on Operating System Design and Implementation. New York: ACM Press. 2004:137- 150.

[3] Fay Chang, Jeffrey Dean,et al. Bigtable:A Distributed Storage System for Structured Data[J]. ACM Transactions on Computer Systems. 2008,26(2):1-26.

[4] Tom White. Hadoop:The Definitive Guide[M]. United States of America: O'Reilly Media, Inc. 2009. [5]Dhruba Borthakur. The Hadoop Distributed File System: Architecture and Design [EB/OL]. (2008-09- 02) [2010-08-25].

[6] Hbase Development Team. HBase: Bigtable-like structured storage for Hadoop HDFS[EB/OL]. (2010- 08-10) [2010-08-25].

[7] Mike Burrows. The chubby lock service for loosely coupled distributed systems[C]. Proceedings of the 7th Symposium on Operating Systems Design and Implementation, 2006.