GFS Master: Master manages all file system metadata and the files directory structure.

GFS uses single master policy which means in the same time only one master providing 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.

• 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 regions of the file namespace. There are four chunk servers and five chunks as C0-C4. 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[.

2) Workflow 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.

Secondly, master sends chunk handle and chunk locations to clients.

Thirdly, clients send chunk handle and byte range to the nearest chunk server

Finally chunkserver sends data to client.

Once clients get chunk locations from master, clients do not interact with master any more.

Master does not permanently save the mapping from chunkserver to chunk. Instead, it asks each chunkserver about its chunks at master start-up or whenever a chunkserver joins the cluster.

The master periodically communicates with each chunkserver in Heartbeat message to give it instructions and collect its state.

3.2.Hadoop Distributed File System**: 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... 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 name node and multiple data nodes.

Name node is the central server, equivalent to master in GFS. It is responsible for the namespace operation of file systems. Data node is similar to chunkserver of GFS which is responsible for managing storage on data nodes, creating block, deleting block, copying block and etc. The files in HDFS are divided into one or multiple blocks which are stored in data nodes. Name node and data nodes can be run on Linux Servers.

4. Cloud storage architecture based on Hadoop

4.1. It is a web desktop environment with office software and personal information management systems, and it enables the online storage, mobile office. Document management is simply stored in a single server, without fault-tolerant backup feature 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 includes clients, web operating system cloud server (Name Node), cloud storage center (Data Node).

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

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

• Cloud server (Cloud Name Node): Cloud storage cluster based on Hadoop includes cloud server (Name Node) and cloud storage center (Data Node). Cloud server is the name node in Hadoop which manages file system namespace, computes the mapping from files to data nodes, allocates data nodes to save file blocks, and controls external clients’ access.

• Cloud Storage center (Cloud Data Node): Cloud storage center is data node 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 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 web operating system. When the files are modified and saved, web operating system uploads them to cloud storage system from local computer.

• **Reading files process**: Users log in the web OS from client through clients’ browser and DoubleClick a file icon on the web OS. Then OS requests the file from the Hadoop name node.

② Name node finds the related information of files, and computes the file’s location. Data nodes which saved the blocks of the file send the blocks to the clients.

③Clients download the file blocks from the data nodes and merge these blocks into a file.

④Applications associated with the file in the web operating system auto start and display the file.

• **Writing files process**:

Users log in web OS from client’s web browser modify and save files using the selected application. OS requests uploading files to Hadoop name node.

Name node allocates storage space to data nodes according to the file size and the data nodes’ storage condition after it received the uploading request.

Clients upload file. Name node divides it into one or multiple blocks and saved in the allocated data nodes.