Design and Implementation of Rich Client Cloud Storage System

Meng Fanjun[1] Zhang Zhongwei[1] Lin min[1] and Jiong Xie[2]

1 Computer and Information Engineering College, Inner Mongolia Normal University, Hohhot, China
2 Inner Mongolia electric power information and communication center, China
mfj007.meng@gmail.com

Abstract

In tradition, we always simply purchase additional machines to solve the storage problems. However, this method only expands the storage capacity. It is un-efficient and high-cost, which cannot solve the storage problems due to the architecture issue. After future analyzing the reasons of storage issue, we bring the cloud computing and cloud storage technology (a rich client-based cloud storage model) into our storage architecture. In the present study, we design and implement a rich client-based cloud storage system. Experimental results show that our modified system can effectively manage huge amount of data with low developing costs and high processing speed. In addition, our design can be easily spread with high scalability and stability.

Keywords: *cloud storage, rich client, HDFS, Ajax*

1. Introduction

Recent IDC research shows that the amount of global information growth of more than 6 times during 2006 to 2010, increased from 161 EB 988 EB (1 EB = 1024 PB) [1]. Facing the storage requirements in PB level, the traditional SAN or NAS have bottlenecks in storage volume and capacity, which cannot fulfill the data storage requirements of high-performance, high capacity, and easy extensibility in the new era. In this situation, cloud storage is emerging. Cloud storage is an extension and developed concept of cloud computing. It refers to a system in which variable different types of storage devices work collaboratively and provide data storage and transaction access services, through clustered applications, grid technology and/or distributed file system.

Cloud computing [2, 3] is an Internet-based computing. According to the definition by NIST (National Institute of Standards and Technology), cloud computing can be divided into three service models: SaaS (Software as a Service), PasS (Platform as a Service), IaaS (Infrastructure as a Service). When a client needs some certain service, he does not need local software installation and hardware management to get the service, instead, he can receive the Web-based service by using a browser connecting to a cloud-computing server. Cloud storage, as one kind of services provided by cloud computing platform, provides service on the base of the Web-client.

The traditional web architecture model uses the "request - wait - response" mode. The disadvantages of this model include the presence of wait time for users, and that the efficiency of Web applications is low, which become the obstacle for the promotion of cloud computing perform. Ajax is a rich client Web-development technology, the most important feature of which is the synchronization requests taking place of asynchronous requests. It means that the client and server do not have to wait for each other, but can make some concurrent operations. This makes Web applications have the ability to respond immediately.

In the present study, the Web-client of cloud storage system is based on Ajax to achieve asynchronous data transfer.

2. Related Work

2.1. Cloud Computing and Cloud Storage

Cloud computing is a newly proposed computing model. The definitions given by Wikipedia: Cloud computing is a formula provided dynamic, scalable, virtualized computing resources through the Internet. Users do not need to understand the details of "cloud" infrastructure, do not have to possess specialty knowledge, and do not need direct control.

Cloud storage is derived from the cloud. Typically, cloud storage includes two aspects: one hand cloud storage is the storage portion of cloud computing, the portion stores resources and information required in the process of cloud computing; on the other hand cloud storage refers to a form of service, providing storage equipment or space. Users by using the browser or other clients use the storage service, avoiding of the local storage space occupation [4]. In the present study, cloud storage system mainly refers to the second aspect.

The architecture model of cloud storage system consists of four layers. From bottom to up, they are data storage layer, data management layer, application interface layer and data access layer, as shown in Figure 1 [5, 10, 11].

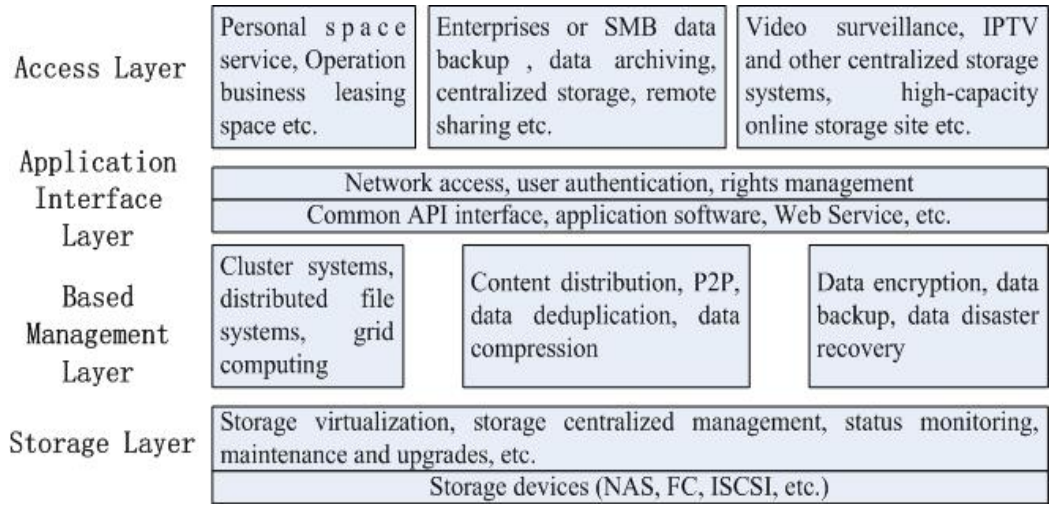


Figure 1. Cloud Storage System Architecture Model

(1) Storage layer. The storage layer, also known as the physical layer, is a fundamental part of cloud storage. On this layer, a large number of geographically distributed storage nodes are connected through a variety of networks, to accomplish unified data storage, centralized equipment management, and centralized state estimation. And storage service is then provided to users in different geographic regions.

(2) Basic management layer. The basic management layer provides a unified view of public administration for the upper cloud storage system. This layer guarantees the seamless joint between the storage layer and the application interface layer through a unified public data management interface. It implements the cooperation of multiple storage devices in cloud storage system through the clustered system, distributed file systems and grid computing technology, to provide powerful storage services.

(3) Application interface layer. The application interface layer, also known as data service layer, is the flexibly expanded, directly user-oriented portion of cloud storage platform. Based on the different types of actual businesses of different cloud storage operating units and user requirements, different application interfaces are developed to provide different services, such as data storage, video surveillance, network drives and multi-user sharing.

(4) Access layer. Through the access layer, any authorized user being anywhere can log in the cloud storage platform from networked terminal devices, using the standard application interface, and can enjoy the storage service. Different cloud storage operating units provide different access types to cloud storage systems, among which the major one is web-access.

2.2. Rich Client and Ajax

Rich Client, based on Rich Internet Application (RIA) using highly interactive rich-client technology, provides users with good and all-angle services. RIA adopts the C/S-like architecture. It has the advantages of high interactivity in C/S architecture and deployment flexibility in B/S architecture. Therefore, it provides better service to users. The rich client technology is the core of RIA. It is considered that RIA makes the programs directly run in the browser [6], and cloud storage is network-based data storage, so the collaboration of them two implements the high-speed parallel cloud storage services.

Ajax stands for Asynchronous JavaScript And XML, which is a collection of many mature technologies [7], including the JavaScript, CSS (Cascading Style Sheets), DOM (Document Object Model), and XMLHttpRequest objects. DOM is mainly used to creat the frame elements of pages. CSS mainly contributes to page layout and styling. XMLHttpRequest implements the asynchronous communications between Web client and server. JavaScript is a scripting language that can operate DOM, CSS, and use the XMLHttpRequest objects [12].

The core of Ajax is JavaScript object XMLHttpRequest [8]. The object was first introduced in Internet Explorer 5, which is a technology supporting asynchronous request. JavaScript both submits requests to the server and handles the responses through XMLHttpRequest, without blocking the users. When a Web site is created, it is more flexible for users to execuate screen updates on the client. Different from the traditional Web applications, Ajax accomplishes asynchronous communication with the server using XMLHttpRequest. When users trigger the XMLHttpRequest object, the client sends an asynchronous request to the server. Once the server handles the client request, the the response data is asynchronously sent to client. Client processes the response from the server by setting a callback function.

The advantages of Ajax application:

(1) Improvement of the user experience by the asynchronous model;

(2) Optimization of the transfer between browser and server, reducing unnecessary data back and forth as well as the bandwidth consumption;

(3) Ajax engine running on the client, takes the part of the original work undertaken by the server, thereby reducing the server load under large amount of users.

3. System Analysis and Design

3.1. System Design

Cloud storage system aims at providing users with transparent, efficient and local disk-like storage space, where users can upload their own files, thus saving local storage space. When a user needs a particular file, he only needs to search for the file in the storage system and download it to the local disk for normal use.

Cloud storage system consists of four parts, including browser-based rich clients, the primary server, monitoring manager, storage node clusters, as shown in Figure 2.

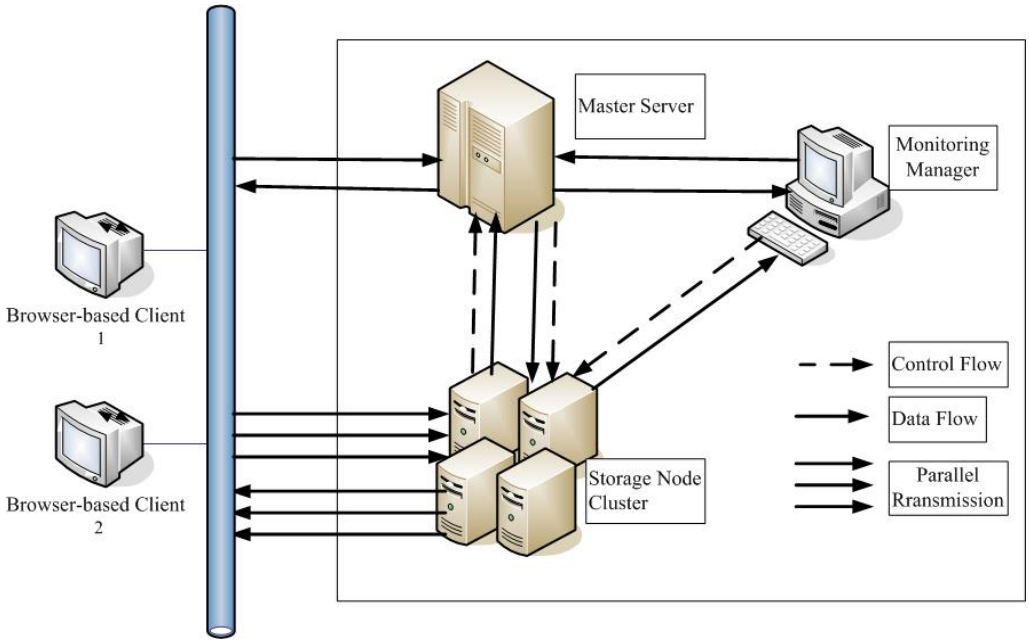


Figure 2. Cloud Storage System Architecture

(1) The browser-based rich client is mainly responsible for the interaction between cloud storage system and users, using rich client technology. Users need to log in the cloud storage system first, and then are able to operate file storage management. When users need to upload or download files, users need to send a request to the primary server, and then parallel upload or download files in blocks according to the information returned by primary server.

(2) The primary server is mainly responsible for the management of metadata, including management of file system name space and operations of client file access. When a user requests to upload a file, the primary server first processes cut calculation based on the file information, and then reads the storage node cluster monitoring information from monitoring manager. The file blocks can be assigned to the available nodes and file information and block information are written into the database. Subsequently, the primary server first finds the block information of this file from the database, and then reads information of storage node clusters from the monitoring manager. Based on the monitoring information, the file block information (number of blocks, block size, *etc.*) and the corresponding node address are returned to the client.

(3) Monitoring manager is primarily responsible for the regular collection and record of information about storage node clusters and the main server for monitoring administrator and the primary server. All machines in the whole cloud storage system (primary server and storage nodes) should regularly send system information to monitoring manager, and the latter then saves the data package into database. Both the monitoring administrator and the primary server can obtain the system information by accessing the database.

(4) Storage node clusters, which consist of a large number of low-end computers, are mainly responsible for the storage of file blocks. As its primary job is to provide massive storage space rather than computing power, the use of cheap low-end computers can set up such a fleet with the advantage of low price, high efficiency and easy extensibility.

Mysql database is used to record file information in cloud storage system, which mainly consists of four tables: storage table, folder table, file table, and block table. The main role of storage table is to record the usage information of users' cloud storage space, for rational use of resources by users and cloud storage system. The main purpose of folder table is to record the information of folders created by users and the affiliations of these folders, to establish an intact, clearly logical folder directory for users. The file table aims at recording the users' file index and file information as well as the correspondences between files and file folders, for users to flexibly arrange file directory and manage their folders. The main role of block table is to record block index and block information of files stored at different nodes in blocks, for users to organize file storage structure.

Mysql database is also used to record monitoring information of cloud storage system, in which the most important table is monitor table. Monitoring data come from each node, and the data are refreshed periodically. Based on the information recorded in monitor table, the primary server adjusts load balancing and node allocation. And the administrator extends and maintains nodes according to the information from the table.

3.2. System Sub-Module Design

Cloud storage system has two main functions: parallel file upload and parallel file download.The workflow of parallel upload module is shown in Figure 3.

(1) Rich Client Sends a Request

Users log in cloud storage system first, and select the local file to upload through the browser. Rich client obtains the information of the file and users' identity information, and then sends an upload request to the primary server.

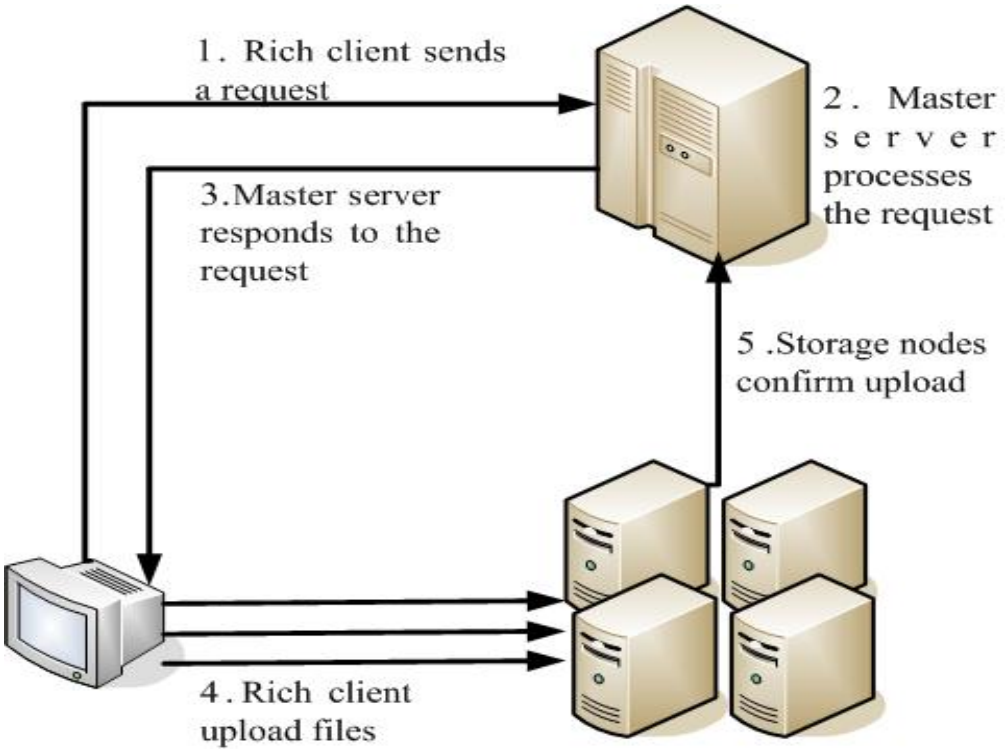


Figure 3. The Workflow of Parallel Upload Module

(2) Primary Server Processes the Request

The primary server receives and processes the request, including file segmentation and node allocation. File segmentation is based on the received information about file size, to 8MB/block. For node allocation, the status of storage nodes is determined first, and then each file block is assigned to 3 available nodes to complete redundancy backup. At last, the file information and block information are written into database. The upload status field is set to 0 indicating that upload has not begun yet.

(3) Primary server responds to the request

The primary server formats the information of file segmentation and node allocation, and then returns it to rich client.

(4) Rich client uploads files

The rich client receives the information returned from the primary server and uploads files. First, the rich client applies for a list for each node to save the file blocks to upload at this node, and then connects with each storage node to start parallel upload.

(5) Storage nodes confirm upload

When a storage node receives a file block, it sends a confirmation message to primary server. The primary server modifies upload status field of the file block in database to 1, suggesting successful upload of the file block. When the primary server monitors that all the file blocks of one file are successfully uploaded, the primary server will modify the status field of that file to 1.

The workflow of parallel download module is shown in Figure 4.

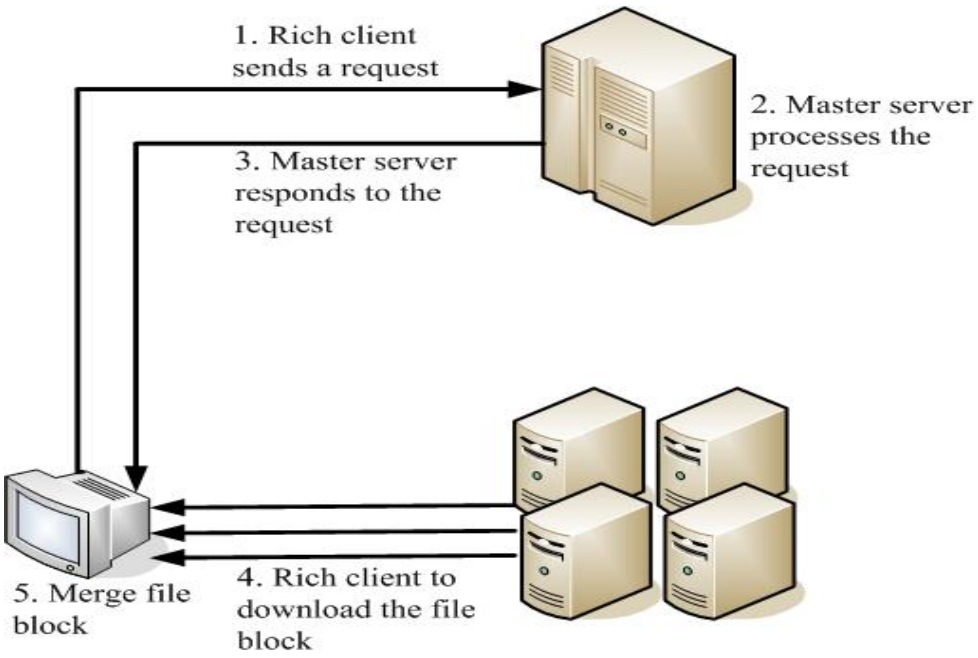


Figure 4. The Workflow of Parallel Download Module

(1) Rich Client Sends a Request

Users log in cloud storage system first, select the file to download through the browser, and send a download request to the primary server.

(2) Primary Server Processes the Request

The primary server receives and processes the download request, including file block lookup and node allocation. The primary server queries database table to find out all the corresponding file block records. For node allocation, the status of storage nodes is determined, and each file block is assigned to an available storage node, for users to connect with and to download.

(3) Primary Server Responds to the Request

The primary server formats the information of node allocation and returns it to rich client.

(4) Rich Client Downloads File Blocks

The rich client receives the information returned from the primary server and begins to download file blocks. First, the rich client creates a thread for each storage node, and then connects with the storage nodes parallel download all file blocks to local disks.

(5) File Blocks Merger

When the rich client downloads all the file blocks of one file, these file blocks merge in order to form the desired file. Finally, all the temporary file blocks are deleted to achieve download.

4. System Implementation

4.1. Deploy Cloud Storage Environment

Cloud storage system is built on five PC sets installed Ubuntu10.04 operating system. The fully open source Tomcat6.0 is selected as Web server. JDK 1.6.0 version is applied, and the master server and monitor manager use MySQL5.0 database. The specific information of each PC is shown in Table 1:

Table 1. Information of each PC

Name	IP	Role	Installed software
master	192.168.128.2	Master Server	JDK1.6.0、Tomcat6.0、MySQL5.0
slave1	192.168.128.3	Storage Node1	JDK1.6.0、Tomcat6.0
slave2	192.168.128.4	Storage Node2	JDK1.6.0、Tomcat6.0
slave3	192.168.128.5	Storage Node3	JDK1.6.0、Tomcat6.0
observer	192.168.128.6	Monitoring Manager	JDK1.6.0、Tomcat6.0、MySQL5.0

4.2. Main Function Modules

Cloud storage system is comprised of four main modules: user management module, parallel upload module, parallel download module, and monitoring module.

(1) User Management Module

The system provides registration page for new users, as shown in figure. Users fill in personal information, and the system saves the information into a database. When users complete registration, they obtain a user name. When users log into the system, the system identifies based on user information. Only have been approved users can apply other functions.

(2) Parallel Upload and Download Module

When users login in system to perform a parallel upload operation, they enter the upload interface, as shown in Figure 5. Users click the "Browse" button to pop up the dialog box, and then users select a file to upload. By clicking the "Upload" button, the file is uploaded, and the upload progress is displayed. After the file is successfully uploaded, the dialog box pops up and then closes the upload interface.

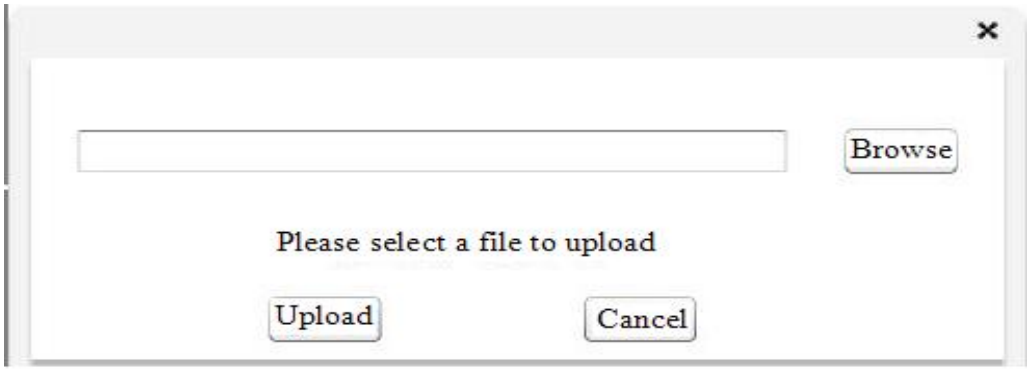


Figure 5. File Upload

Upload module code list of files in parallel as shown in Table 2.

Table 2. Code Files of Parallel Upload Module

Filename	Note
upload.jsp	Implement the interface of upload module. It needs to call upload.js to implement upload file block in parallel in this file.
upload.js	Used for processing files of client and uploading file blocks, etc. When click the “Browse” button, upload.jsp calls the browser() method of upload.js to select file, and display the file's information in upload interface. When upload.jsp received the returned information from the master server, it calls the sendInit() method of upload.js, the method create a queue for each node which store the node's id, ip, and block number corresponding to the node. File blocks upload sequentially in each queue whereas in parallel among queues.
upload.java	The servlet file of master server to receive and respond the upload requests, it calls methods of uploadHandle.java for processing. In master server, upload.java accepts the file information and user information send from client, first determine whether the file exists. If the file does not exist, insert the file information into the database, and then call the methods of uploadHandle.java for the computation of file cutting, node allocation, and database-related operations, furthermore, return the generated block information to the client in XML format.
uploadHandle.java	The java back-end code for the master server to conduct the computation of file cutting, node allocation, and database-related operations. In this class, chenkNode() method is used to detect node status and return lists of available nodes; checkBlock() method is used to check whether the block information of a file in the database exists, if it exists, return the list of block information, otherwise call blockToNode() method, and return the list of block information; blockToNode() method is used for the computation of file cutting and node allocation, first obtain the total number of blocks through "file size / cut size", then assigned the blocks to nodes in avNode by turns according to the serial number, for each allocated block, pre-allocate a storage space equal to the block size on the corresponding node; fStateUpdate() method is used to change the file status in database and value of prehd, usedhd after flies upload successfully.
uploadAccept.java	Store the servlet of nodes receiving and saving the file blocks. uploadAccept.java receive the binary stream send from client and save it as a file block.

Download module is the reverse process of upload module. When users login in and perform parallel download operation, they enter the download interface, as shown in Figure 6. Users single click the "Browse" button to pop up the o dialog box, and then users choose where to save the downloaded file on client side and enter the file name. When the "Download" button is single-clicked, the client side sends a download request to the server. After the client side receives the block information returned from server, the file blocks are parallel-downloaded from the corresponding node to a local temporary folder. After all the file blocks are downloaded, the file blocks are integrated locally, and the file blocks are deleted.

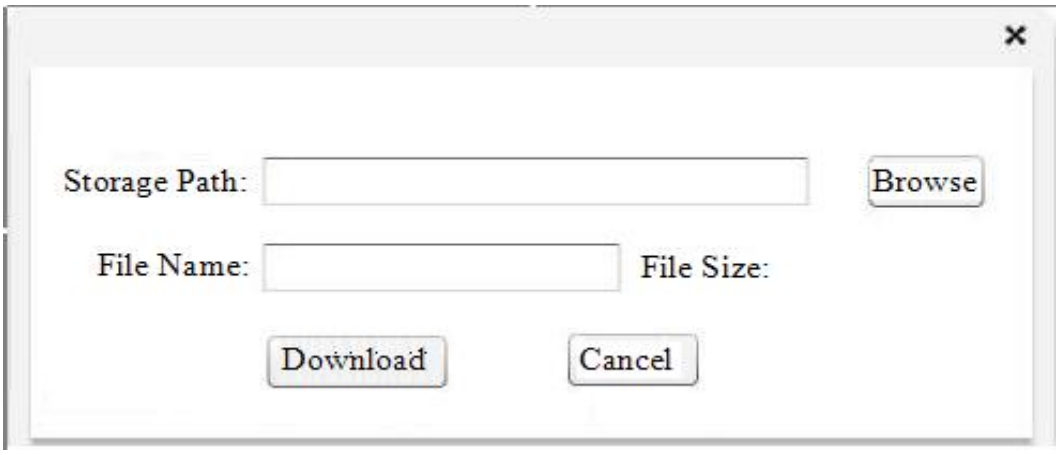


Figure 6. File Download

Download module code list of files in parallel as shown in Table 3.

Table 3. Code Files of Parallel Download Module

Filename	Note
download.jsp	Implement the interface of download module. In this file, It needs to call download.js to implement operations like file block download and file integration.
download.js	According to block information, parallelly downloading the file blocks and integrating files.When you click the "Browse" button, download.jsp calls the seldir () method in download.js to select the files which are downloaded into the local path.When you click the "Download" button ,download.jsp calls the sendrequest method in download.js , sending download request to the primary server .Upon receiving the block information which returns from the master server , parsing XML with the xml_response method .According to analytical results, parallelly downloading file blocks.
download.java	Master server receives and responds to the Servlet files from download requests .Download.java in master server receives the file information and user information which the client sends, and then looks for block information of the file in the database , and sends the block information in XML format to the client.
downHandle.java	Storage nodes process the servlet downloaded from the file blocks. DownHandle.java searchs the block files according to request and sends it back to the client in binary stream format.

(3) Monitoring Module

The basic tasks of the monitoring module in cloud storage system include monitoring the status information of storage nodes and the primary server (covering CPU information, memory, and disk space) and saving it into the database, for the application by monitoring personnel and the primary server. The monitoring module is divided into three parts: data acquisition, data transmission, data storage and display.

Because the storage node clusters are under Linux, a reading / proc virtual file system method is applied to collect data. Then socket communication is applied for data transfer. The monitoring manager creates ServerSocket object to monitor a particular port. If there is a computer connected to the clusters, then a new thread is started to receive data for the node. Monitoring Manager receives data, inputs data to a database table. The display side reads the information in the database and display monitoring information. Monitor display shown in Figure 8.

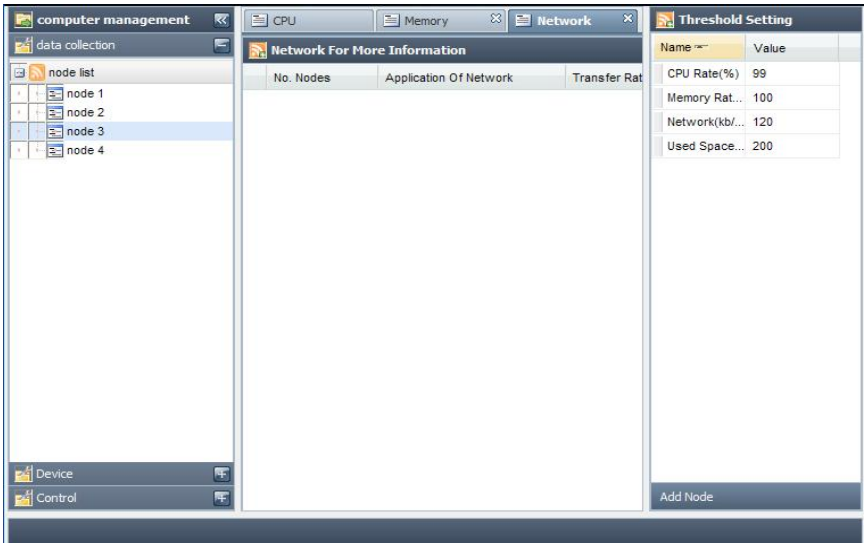


Figure 8. The View of Monitor

5. Conclusion

According to the study of cloud computing background as well as GFS and HDFS performs, a rich-client-based cloud storage system is implemented. The system can provide data to upload and download services for users, with the advantage of low cost, ease of implementation and simple extension.

Currently, the system function is still relatively single, and there are still many problems for further improvement. The next main work includes: to study the redundant backup knowledge of file blocks for more reasonable scheme of partitioning and file block allocation; to study security of cloud storage; and to improve cloud storage security in use of appropriate technology.

Acknowledgements

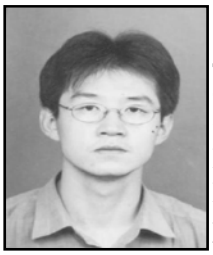
This research was supported by Scientific Research Project of Higher Education of Inner Mongolia Autonomous Region, China (NJZY13052).

References

[1] T. Lindhorst, G. Lukas and E. Nett, "Data-mining-based link failure detection for wireless mesh networks", Proceedings of the 29th IEEE International Symposium on Reliable Distributed Systems, New Delhi, India, **(2010)** October.
[2] Cloud Computing, http://en.wikipedia.org/wiki/Cloud_computing, **(2013)** January 25.
[3] M. Armbrust, A. Fox and R. Griffith, "Above the Clouds: A Berkeley View of Cloud Computing", Technical Report No.UCB/EECS-2009-28.2009.2.
[4] H. Chen, "The Optimization and Implementation of Cloud Storage System based on HDFS", South China University of Technology, GuangZhou, **(2012)**.
[5] S. Tuo, "Cloud Computing and Data Storage Technology, computer development application", vol. 9, no. 23, **(2010)**.
[6] S. Yong-ge, X. Jian-lin and S. Feng, "Research of rich client technology application", Computer Engineering and Design, vol. 3, no. 29, **(2008)**.
[7] A New Approach to Web Application. http://www.adaptivepath.com/publications/essays/archives/000385.php, **(2013)** February 18.
[8] C.-Bifeng, "Technology and Appl ication of Rich Client Based on AJAX", Computer Science, vol. 10A, no. 38, **(2011)**.
[9] W. Peng, "About Cloud Computing", People's Posts and Telecommunications Press, China, **(2009)**.
[10] Z. Di, Z. Li-gu and H. Zhen-yu, "Research of cloud storage technology for mobile terminal based on WEB", Computer Engineering and Applications, vol. 46, no. 36, **(2010)**, pp. 66-69.
[11] H. Bin, "The Research for the Application of the Cloud Storage Technology in Design Institute", Information Security, vol. 9, **(2012)**.
[12] W. Qi-Zhi and F. Li, "Research of Memory Leak Based on AJAX Rich Internet Application", Journal of Anqing Teachers College (Natural Science Edition), vol. 2, no. 18, **(2012)**.
[13] Y. Kwak, B. Goo, D. Ko, I. Oh, J. Y. Hwang and S. Cheong, "An Analysis and Design of the Storage Management System Based on SMI 1.1.0", IJUNESST, vol. 3, no. 2, **(2010)** June, pp. 51-60.
[14] D. Shin, D. Shin and J. Jeong, "An Efficient Storage Mapping Method for Semi-Structured Microarray Data Based on Structural Similarity", IJSEIA, vol. 6, no. 2, **(2012)** April, pp. 179-184.
[15] L. Ran and H. Jin, "Real-time and Flexible Management of Storage Service Provider in Distributed Storage", IJHIT, vol. 5, no. 2, **(2012)** April, pp. 219-224.
[16] P. Xiao and Y. Zhang, "CS-Mobile: A Cloud-based Distributed Storage Middleware for Mobile Devices", IJSH, vol. 7, no.1, **(2013)** January, pp. 87-98.

Authors

Meng Fanjun received the BS and MS degrees in computer science from Inner Mongolia Normal University, China, in 1999 and 2007. Currently, his research interests include scheduling techniques and parallel algorithms for clusters, and also multi-core processors and software techniques for I/O-intensive applications.



Hailong Wang received the BS in computer science from North Jiaotong University, China, in 1998 and received the MS in computer science from Lanzhou Jiaotong University, China, in 2007. Currently, he is an assistant professor in Computer & Information Engineering College at Inner Mongolia Normal University, China. His research interests include embedded system and muti-core processors and also fault tolerance and real-time database.