

Typical Cloud Computing System Design

HaiYan Liu

Computer science and engineering department,
North china institute of aerospace engineering
LangFang, HeBei 065000, China
zy-lhy@163.com

Zhiyuan An

Computer science and engineering department,
North china institute of aerospace engineering
LangFang, HeBei 065000, China
Azy01@263.net

Abstract—Cloud computing more widely applied to study the structure of the cloud computing system, and in-depth understanding of cloud computing technology is necessary. Is very meaningful for the development of cloud computing systems. Designed in a typical cloud computing system framework, including the central server, storage sub-node and client, and typical applications of cloud computing. I believe a great help to the people began to learn of cloud computing.

Keywords—cloud computing; cloud storage; cloud client;

I. INTRODUCTION

Cloud computing has been proposed in the middle of the last century, it is the product of the development of computer technology. From the technical point of view, cloud computing is the external provision of services through the visualization's technology and distributed technologies. The suppliers are responsible for all hardware resources such as computers, storage devices, and network.

Cloud computing is one of the hottest technologies of the IT sector. But for now, many people have the understanding of cloud computing is not very thorough, most of the technical staff to use cloud computing applications, but do not understand the cloud. The paper presents the design of a typical cloud computing systems thinking. Intended to allow more people understand the work pattern of the cloud computing system.

II. SYSTEM DESIGN

This system is not a perfect system, intended to constitute the basic framework to show the cloud computing platform, it only has the basic functions of cloud computing: mobile computing tasks to the storage node, and distributed file storage, and parallel computing, and so on.

Consider the problem of system design: File cloud storage; cloud computing; mobile computing tasks to the storage node; to achieve the management of the storage node and client; to provide the necessary interface functions, the user can use the system through the interface, and users do not care to calculate and storage problems; multi-threaded approach to improve system efficiency.

Given the above considerations, the system is divided into three functional modules: a central server; storage sub-node and client.

The central server is mainly responsible for the day-to-day management, and systems needed configuration file.

Storage sub-node is mainly responsible for user file storage and computing capabilities.

The client is responsible for the file blocks, and distributed to each storage sub-node, as well as through the system interface to send commands and receive data. Specific functional structure shown in Figure 1:

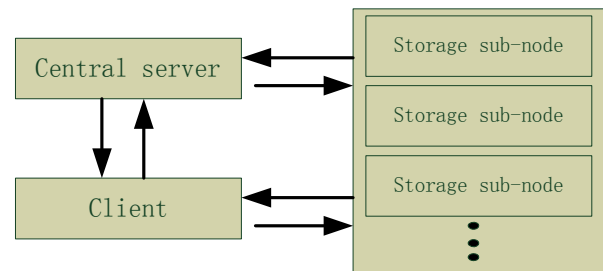


Figure 1. System structure

III. FUNCTIONAL DESIGN OF A CENTRAL SERVER

Central server as a strategist role, it is necessary to manage the storage sub-nodes of the host information, but also responsible for directing the client to complete cloud computing operating, so the module can not be responsible for cloud computing operations, or the central server where the host will be very busy can easily lead to the bottleneck of the entire cloud computing system. Therefore, he need only be responsible for the command, as front-line army commander, and do not have to charge forward.

Center server features include: user registration, user login, delete users, access to user information, access to the node information, the management of storage sub-nodes.

He's not complex, but very busy, all systems in the cloud computing operations, such as storage, computing, we need him to be coordinated. Order to be able to remember each user's information, the storage sub-node's information, the

system requires two files to be stored. One file is UserInfo.doc (can also be other names), specialized store user information; another file for node.doc (can also be other names), specialized for storage sub-node's information.

A. Register

Center server requires a port to listen for client registration request, to receive the user name information, and assigned the storage sub-node's IP to the user. Finally, the information is stored in the configuration file UserInfo.doc.

B. User login

That part is relatively simple; the central server receives the user's username, and then finds the configuration file UserInfo.doc. If found, then continue to receive follow-up instructions of the user. If not, then return failure and disconnect.

C. Delete user

Client proposed to delete the user information, that is, cancellation of the account. This delete is the same idea of the hard drive deleted file, that is, when users send delete command, you first need to reconfirm guarantee not misuse. The system does not really delete the user's files, but all the content related to the user set to delete sign, and set an expiration mechanism, and only the extended data will be completely removed.

D. Get user information

When the user needs to know the circumstances of his stored files, he can achieve their goals through this function. The central server receives the user's command to get user information, then to find UserInfo.doc configuration file based on user name, if found, return to the user.

E. Access to node information

When users need to store files, the center server according to storage conditions, and the file size of the user assigned to the user the storage sub-nodes, and inform the user of the storage sub-nodes group list.

F. Manage the storage sub-node

The host system used to store user files called the storage sub-node. Users to store files, or access to his files, he can not know the file storage location, the central server must store the information. So, central server needs to know which hosts can be used in the current system. This is the management of the child nodes.

The central server involved in the work tedious, and often, each transaction requires a separate thread for processing, here divided into four types of thread. The main function of the central server workflow shown in Figure 2-6:

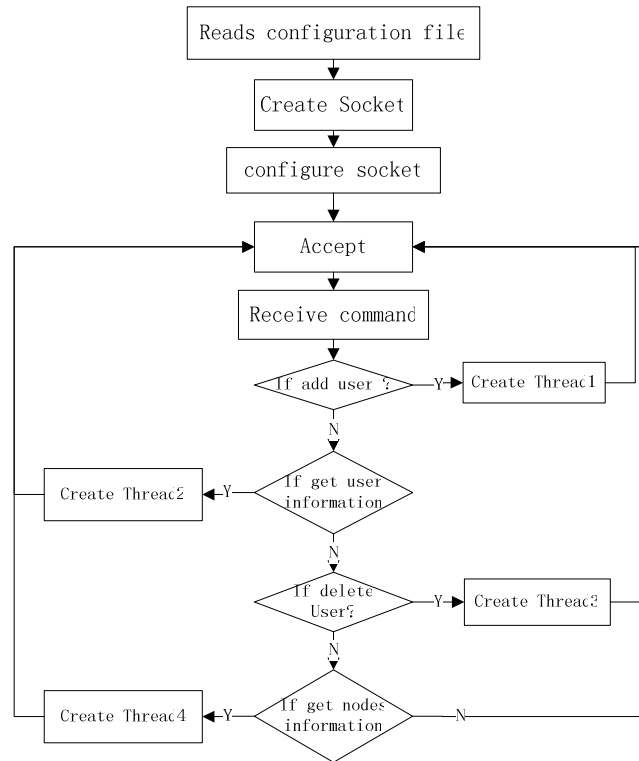


Figure 2. Central server workflow

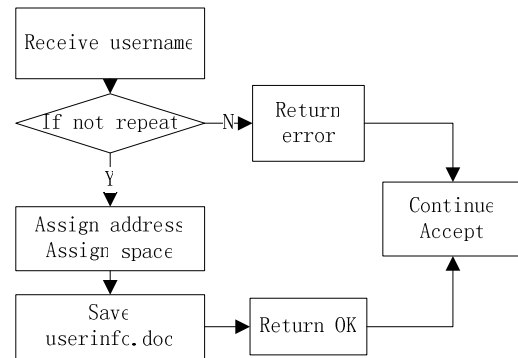


Figure 3. Thread1--add user workflow

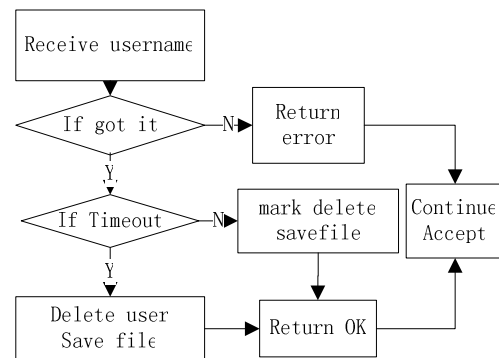


Figure 4. Thread2--delete user workflow

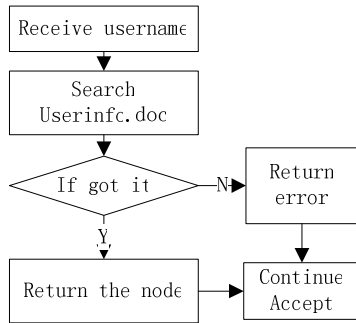


Figure 5. Thread3—get user info

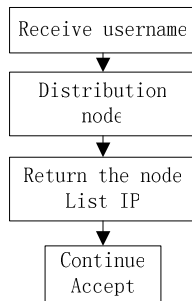


Figure 6. Thread4—get node info

Associated data structure is defined as:

1) The user-related data structures

Struct userInfo

{

String username; // user name

String IP // host address where the user configuration file

Int port // service program of the host listening port number where the user configuration file

};

The structure is mainly used to store user information, and each user's username is unique. All of the information on a linked list, linked list node data type is userInfo, each user's information is one node of the linked list

2) Related data structures to storage sub-node

Struct nodeInfo

{

String IP // host address

Int Port // host service program listening port number

Long int Tsize // total disk capacity

Long int lastsize // remaining space

};

The structure variable used to store the the host information of storage sub-nodes, including the address of the host, monitor port, storage space, and other information. Each of the storage sub-node address is unique, and storage sub-nodes in a linked list, linked list node's data type is nodeInfo. Each storage sub-node's information is one node of the linked list.

IV. FUNCTIONAL DESIGN OF STORAGE SUB-NODE

The main function of the storage sub-node is responsible for monitoring the command of the client, receive, parse and execute.

Each storage sub-node running the server side of TCP communication program, and has been in a listening state, when receiving the client's information, parse information instruction processing in accordance with the instructions to create a different thread, here the design to four threads: Get file storage, file chunking information is added, a block of data storage, data block calculation, as shown in Figure 7:

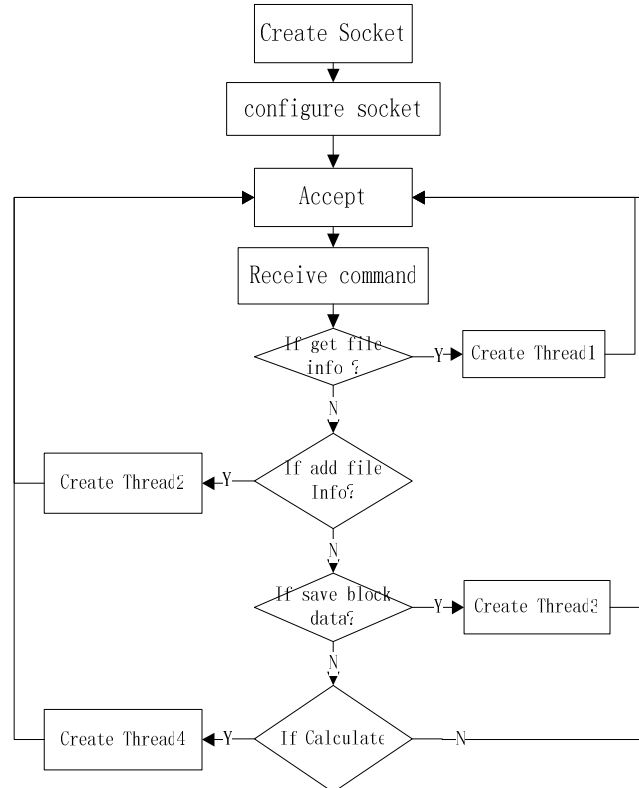


Figure 7. storage sub-node workflow

Among them, the thread 1 is responsible for file storage information; Thread 2 responsible for handling the add file information; thread 3 is responsible for storing user data block; thread 4 is responsible for the user data block calculation.

The relevant data is structured as follows:

Struct userfileInfo

{

String filename // file name

Int blocknum // block number

String IP // address of the host which store the data block

Int Port // port of the host which store the data block

};

The structure used to store user data files described. Each user file is a linked list of nodes of the type described. Each file is divided into how many pieces, the linked list has how many nodes.

V. FUNCTIONAL DESIGN OF THE CLIENT

In order to reduce the burden of storage sub-nodes, to avoid the drawbacks of the storage sub-nodes of the bottleneck, the client function of the system compared to other systems in the client slightly fat bigger, mainly reflected in the file the division of work bear down.

The client's main function is divided into two categories:

One is a server-related feature, including: registration, login, access the storage sub-node's IP, and access the address of the host where has its configuration file.

The other is related to the storage sub-node features, including: the division of documents, to obtain its own configuration file, adds the profile information, store files, read file block, and calculate the file.

The client's functional block diagram shown in Figure 8:

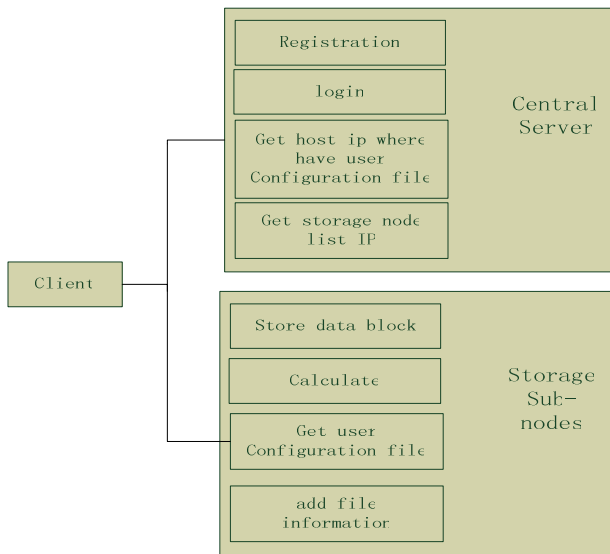


Figure 8. Client Functional modules

Client registration, sign-on feature is relatively simple and so is not discussed here. Analyses file storage and computing two functions.

A. File storage

First, according to the file block size of the system design, client divides the file into small pieces. Named as "username_filename_blocknum.dat"(for easy future file to find). And then create a distributed thread. Next, start the thread distribution data block. And finally, to circulate the document information is sent to a central server.

Each distributed thread is responsible for a block of data sent to a storage sub-node process: First, create a stream socket client, and then connect the storage sub-node of a specific port, and then send the instructions of the distribution of data, followed by sending a username file

name, block number, data length, block data, and finally close the socket.

B. File calculation

Before the calculation, first connect to a central server to obtain its own configuration file, then create a thread to send commands to each storage sub-node, and finally, the calculation results for each thread waits for the storage sub-node, the main thread is responsible for summary of the results.

The client data structure definition:

```

Struct data
{
  Int Bnum;//block number
  Int len;//block length
  Char buf[Bsize]);//block data
};
  
```

The user's files are divided into data blocks of a predetermined size (Bsize), each data has its own number, all data will be stored in a sequence table, each node in the linked list are above the type of structure. Schematic diagram of the sequence table shown in Figure 9:

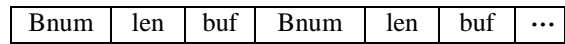


Figure 9. Client data block link list

VI. SUMMARY

The system is intended to demonstrate the basic functions of the cloud computing platform, to ensure that the redundant part of the system robustness without making consideration, to ensure system security mechanisms are also not designed. Readers who want to design the complete system, can do extension in the basis of this system.

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their valuable comments. This research is supported by the Langfang research and development projects for scientific and technological (2012011009&&2011011008).

REFERENCES

- [1] CHEN Kang, ZHENG Wei-Min . Cloud Computing: System Instances and Current Research[J], JOURNAL OF SOFTWARE, 2009, 20(5)
- [2] LUO Jun-zhou, JIN Jia-hui. Cloud computing: architecture and key technologies[J], JOURNAL ON COMMUNICATIONS, 2011, 38(11)
- [3] ZHAO Junhua, WEN Fushuan. Cloud Computing: Implementing an Essential Computing Platform for Future Power Systems[J], AUTOMATION OF ELECTRIC POWER SYSTEMS. 2010, 34(15)
- [4] Liu Zhengwe, Wen Zhongling, Zhang Haitao. Cloud Computing and Cloud Data Management Technology[J], Journal of Computer Research and Development, 2012, 49(z1)
- [5] DING Ze-liu, GUO De-ke. Researching Data Center Networking Topology for Cloud Computing[J]. JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY. 2011, 33(6)