Design and Implementation of Network Storage System Based on P2P

Zengyu Cai [1,3], Wenqian Wang [1,3], Jianwei Zhang [2,3] + and Zuodong Wu [2,3]

[1] School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, 450002, China

[2] Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou, 450002, China

[3] Henan Key Laboratory of Food Safety Data Intelligence, Zhengzhou, 450002, China

Abstract. File storage system based on P2P. Firstly, the design requirement, function design, system design and the other module design is described. Secondly, the key technologies such as algorithm design, performance design, security design and database design of implementation of file storage system were investigated. Finally, the results of constructing the file storage system were discussed. Then the results show that the file system has good performance, and the user can complete the login, file sharing, download files and other functions, etc.

Keywords: file storage; file sharing.

1. Introduction

The file storage system based on P2P can download and upload shared files in LAN, and ensure the security of data during data transmission [1~4]. With the progress of science and technology and the rapid development of society, people's demand for network resources has increased, and network resources have been widely used in all aspects. In the traditional C/S mode, when a large number of users request the server, the pressure of the server will increase, and then there will be network congestion and insufficient disk. The demand for real life drives the development of technology, Therefore, there are numerous P2P file sharing software in the world [5~7]. among which Napster, Skype, emalu, etc. are more famous. Many colleges and universities have begun to study the P2P network, such as Popular Power of Tsinghua University and Groove of HUST. and the most representative of which is the Maze system developed by Peking University [8~11]. This paper takes the centralized file storage system based on P2P as an example to study the general method for the design and implementation of the p2p-based file storage system.

2. System Design of P2P File Storage System

2.1.System Flow Design

Firstly, after the client connects to the server, registration of connection information and Shared file information with the server, and the client sends the search file request to the index server. then the index server returns the queried file information to the client, Finally, the data is transmitted between the peer client and the peer client. The system flow chart is shown in Figure 1.

+ Corresponding author. Tel.: + 136 0382 9696.
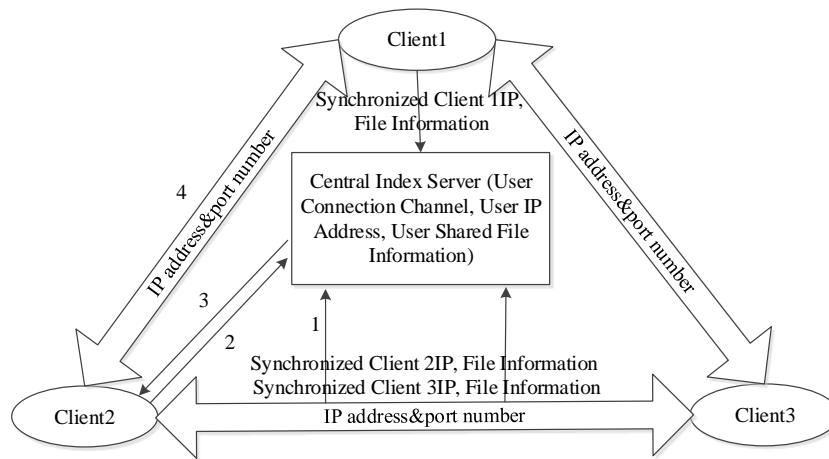*E-mail address*: mailzjw@163.com.

Fig. 1: System flow

2.2.Function Design

The main functions of the file sharing system based on P2P are file sharing, file download, file search and permission management. As shown in Figure 2.
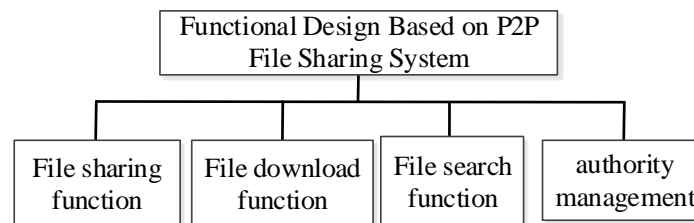

Fig. 2: P2P file sharing system function design

File Sharing Function:(1) the client synchronizes its own shared file information to the server, (2) after the server receives the share information to check whether the database already exists, (3) and if there is, determines that the previous information expires and deletes, and then adds new file information to the data table, (4) if not, add a new file information directly to the data tables.

File Download Functions: (1) The client initiates request according to the resource address sent by the server; (2) the resource owner queries whether the requester has download authority; (3) if there is no authority, sends an error message to reminder; (4) if you have authority to download, establish a connection to start data transmission, after data transmission, check whether the file is completely downloaded. If complete, close the data stream; if it is not complete, download again.

File Search Functions:(1) the client initiates a resource search request to the server; (2) the server receives and parses data to match in database according to conditions; (3) if the file is matched, the user who is not online is filtered out, and send online user information to the client; (4) if no file is matched, the query result will be prompted to be empty.

Privilege Management Functions: (1) user login; (2) the central server receives information sent by the client and verifies it in the database; (3) if the verification fails, an error message will be sent to the client; (4) If the verification passes, the login succeeds and the client automatically sends the shared file information to the server.

3. Implementation of P2P-Based Network Storage System

3.1.Message Generation

Considering its own system, because the data transmitted by each message is relatively small, the string definition is used to encapsulate the delivery of the message, and the socket encapsulated by the .net framework is used to send the message. When the receiver receives the data, it disassembles the message according to the protocol defined by the string and parses the data. and the message generation process of the system:(1) message monitoring. Regardless of the server or the client, it is necessary to constantly monitor a certain port

to receive the request sent to itself. in the system, the server mainly needs to click a button to enable port monitoring. (2) Message processing. Each message is handled differently. in this system, different processing methods are used for each type of message. When the message monitoring module listens to a message, it will transmit the message to the message processing module. After the message processing module parses the message, according to the cmd field, to match the corresponding message processing module, makes the corresponding processing, and return. (3) Message encapsulation. When we need to send a message, we will encapsulate the message according to the type of message we send. In addition to the message body, we add the protocol that we have formulated and encapsulate it with the required fields. (4) Message transmission, after the message is encapsulated, the encapsulated message is transmitted to the message sending module, which sends the encapsulated message to the corresponding IP address.

3.2. Communication Module

3.2.1. Central Server Communicates with Peer Client

The system consists of a central server and an edge client. When the client connects to the central server, the file resource and location information will be shared by each client is saved. The information stored by the server to the database is the size of the file, the IP address where the file is located, the file name and the hash value, and information of the user and other information are managed, the client will only connect to the central node server when searching for resource files, and does not need to request the central node server during the download process, This separates resource search from file download, and reduces the pressure on the central node server and greatly increases the download speed. The process of interacting between the server and the client is shown in Figure 3.
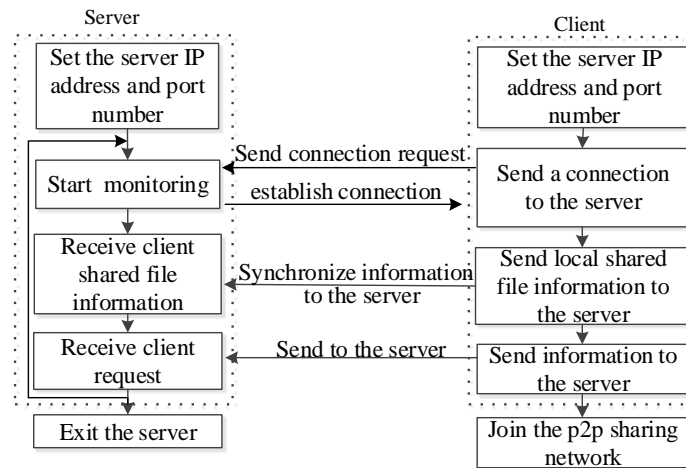

Fig. 3: Server and Client communication process

3.2.2. Peer-to-Peer Communication Between Clients

After the client connects to the designated server, the client will open the monitoring module and wait for the download request from other clients. When other clients send the request, the two client nodes will establish a connection and then request by the file. The data request information is initiated, and the related data is transmitted. After the transmission is completed, the file requester will initiate the disconnection request. At this point, the notification process between a complete client is over, and the communication process between peers is shown in Figure 4.
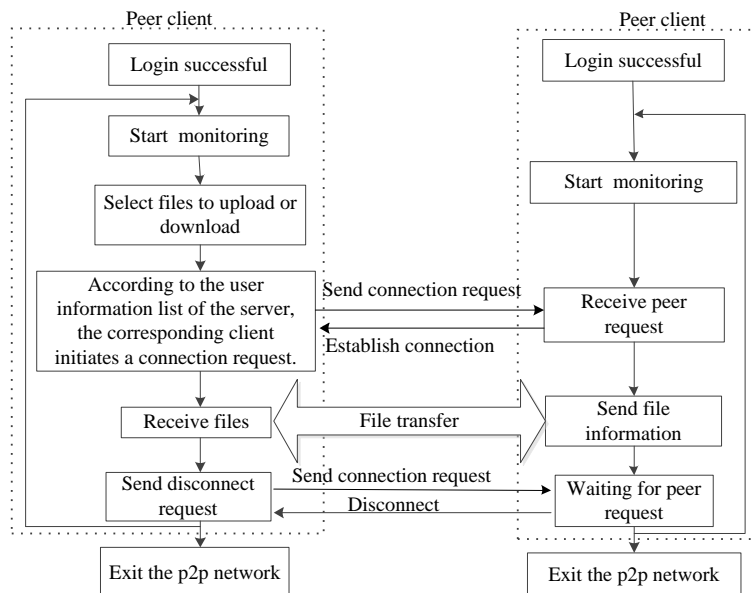
Fig. 4: Peer-to-Peer client and peer client communication process

4. Experimental Results and Discussion

4.1.Test Environment

This system uses .NET +MSSQL +Socket to develop this system, the development language is C#, and the development environment is Visual Studio 2013, the TCP protocol is the transmission protocol, the database is MSSQL 2012, the operating system is Windows 10, and the processor is Intel (R) Core(TM) i7-4790CPU@3.60GHz 3.60GHz, RAM is 8.00G, and the running platform is .net 4.0.

4.2.Functional Test Results and Analysis

In terms of functional testing, various functions based on the P2P file sharing system were tested according to the functional design. and the test results are shown in Table 1. The test results show that the functions involved in the system can be realized, the development function design fulfil the demanded requirements, and can realize the login to the user and the sharing, uploading and downloading of the files.

4.3.Performance Test Results and Analysis

In this test case, the performance of the system, including the system's disaster tolerance and download speed, which was tested. Fifty users were simulated on-line at the same time, and some nodes were randomly selected to exit and join the network for multiple times. The system can work well without being affected. Through the following download speed, we can see that with the increase of users and the increase of download demand, the bandwidth of the system has not been affected, and the download speed is maintained a good download at all times. The performance test results are shown in Figure 5.
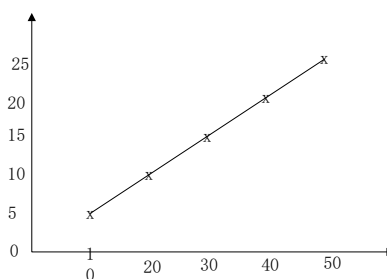

Fig. 5: Performance test results

5. Conclusion

In this paper, a file storage system based on P2P is designed and developed by using C# programming language. The user interface is simple and easy to operate. The shared and download of user files was realized,

462

it has a certain application prospect and lays a foundation for the next improvement. Compared with the traditional C/S network model, the P2P file storage system can make people's life convenient and fast. people will download resources on the network more and more faster, making the transmission of information more convenient and satisfying, and can meet people's needs very well.

6. Acknowledgement

7. References

[1]   Bok K, Kim J, and Yoo J, "Cooperative Caching for Efficient Data Search in Mobile P2P Networks," *Wireless Personal Communications*, vol. 97, no. 3, pp.1-23, 2017

[2]   Hua K L, Chiu G M, and Pao H K, "An Efficient Scheduling Algorithm for Scalable Video Streaming over P2P Networks," *Computer Networks*, vol. 57, no. 14, pp. 2856-2868,2013

[3]   Tu, Chao-fan, Zhang, Qi-fei, Chen and Ying-zhuang, "A Hierarchical Storage Strategy for Hybrid Clusters on a P2P Based Distributed System," *International Conference on Computer Science and Technology*, pp. 514-524, January 2016

[4]   Das, Sajal K., Naor, Zohar, Raj and Mayank, "Popularity-Based Caching for IPTV Services over P2P Networks," *Peer-to-Peer Networking and Applications*, vol. 10, no. 1, pp.156-169, 2017

[5]   Bandara, H. M. N. Dilum, Anura P, and Jayasumana, "Community-Based Caching for Enhanced Lookup Performance in P2P Systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp.1752-1762, 2013

[6]   Jang H C and Hsu T Y, "Infrastructure Based Chord Structure for P2P File Sharing over Vehicular Network,"*Journal of Networks*, vol.8, no. 3, pp.588-597, 2013

[7]   Keshvadi S, Rahmani A M, and Rostami H, "Recommend Top-k Most Downloaded Files in the Chord-Based P2P File-Sharing system," *Peer-to-Peer Networking and Applications*, vol. 10, no. 2, pp.208-215, 2017

[8]   Sun X, Chao Z, and Liu Y, "An Efficient Semantic-Based Search Schema in Unstructured P2P Network," *International Journal of Distributed Sensor Networks*, vol. 5, no. 1, pp.78-78, 2009

[9]   Andreas Binzenhöfer, Kurt Tutschku, Björn, Fiedler, Auf Dem Graben, Markus Fiedler and Patrik Arlos, "A P2P-based Framework for Distributed Network Management," *Lecture Notes in Computer Science*, vol. 3883, no. 1, pp.198-210, 2006

[10] Yang Xinfeng, Zhao Huanping, and Liu Kecheng, "The Research of P2P Network Architecture Based on QoS," *Energy Procedia*, vol. 6, no. 13, pp.101-102, 2011

[11] Martalò, M, Amoretti M, and Picone M, "Sporadic Decentralized Resource Maintenance for P2P Distributed Storage Networks," *Journal of Parallel and Distributed Computing*, vol. 74, no. 2, pp.2029-2038, 2014