Journal of Intelligent Systems and Internet of Things
Vol. 15, No. 02, PP. 14-28, 2025
DOI: https://doi.org/10.54216/JISIoT.150202

Design and implementation of intelligent home data cloud storage system with large system and big data

Yangxia Shu[1,2,*] , Hai Liu [1,3]

[1]College of Big Date Science, Jiangxi Institute of Fashion Technology, Nanchang 330201, Jiangxi, China

[2]Information Technology Integration Innovation Center, Intelligent Research and Innovation Team for Clothing (Jiangxi Institute of Fashion Technology), Nanchang, 330201, Jiangxi, China

[3]Clothing big data research center, Intelligent Research and Innovation Team for Clothing (Jiangxi Institute of Fashion Technology), Nanchang 330201, Jiangxi, China

Emails: yangxiashu@outlook.com; liuhai@jift.edu.cn

Abstract

The increasing maturity of 5G technology and Internet of Things technology makes people feel the convenience brought by high-tech in their daily lives, and smart homes gradually penetrate into people's lives. Aiming at the disadvantages of traditional data storage such as low flexibility and slow speed, an effective cloud storage system for data storage and management is designed. Through the design of the data cloud storage system structure and database, and the hardware design of the smart home data cloud storage system, this paper provides users with various functions, verifies the practicability of the cloud storage system through system testing and analysis, and improves the functions of the smart home data cloud storage system.

Received: September 17, 2024 Revised: November 14, 2024 Accepted: January 04, 2025

Keywords: Big data; Smart home; Data cloud storage; Design; Realize

1. Introduction

In the context of the information age, smart home has been put forward a higher level of requirements, smart home data cloud storage is an important aspect that cannot be ignored. Green, environmental protection and energy saving are the important concepts of the current society. The intelligent control scheme provided by smart home for home equipment not only provides users with safe and fast home services, but also meets people's needs for safe, comfortable and convenient life. At the same time, it is more efficient, energy saving and environmental protection [1]. The main function of smart home is data storage and device management. In order to effectively meet the specific requirements of home intelligent control, this paper gives the overall structure of the storage system based on the characteristics of actual smart home data storage, analyzes the hardware structure of each section of the system in detail, and promotes the efficient operation of the designed smart home cloud storage system. Improve the economic benefits of smart home design and application [2-3].

2. Structure of the data cloud storage management system

This time, cloud service technology is used to design and implement a data cloud storage system in the data cloud server, which is responsible for the centralized and unified management of data in the smart home gateway [4-5]. The data cloud storage and management system provides the following functions: centralized storage and remote management of smart home gateway data, and backup and recovery of smart home data [6].

The data cloud storage management system consists of three modules: Web service module, data cloud storage module and background management module. Among them, the Web service module provides the function of interacting with the administrator is responsible for visually displaying the data in the smart home gateway on the Web page, and receives the remote management operations of the manager, and the data cloud storage module is responsible for centralized storage and management of the data in the smart home gateway. The background management module is responsible for data synchronization with the smart home gateway, as well as data backup and recovery of the smart home gateway [7]. The structure of the data cloud storage management system is shown in Figure1.

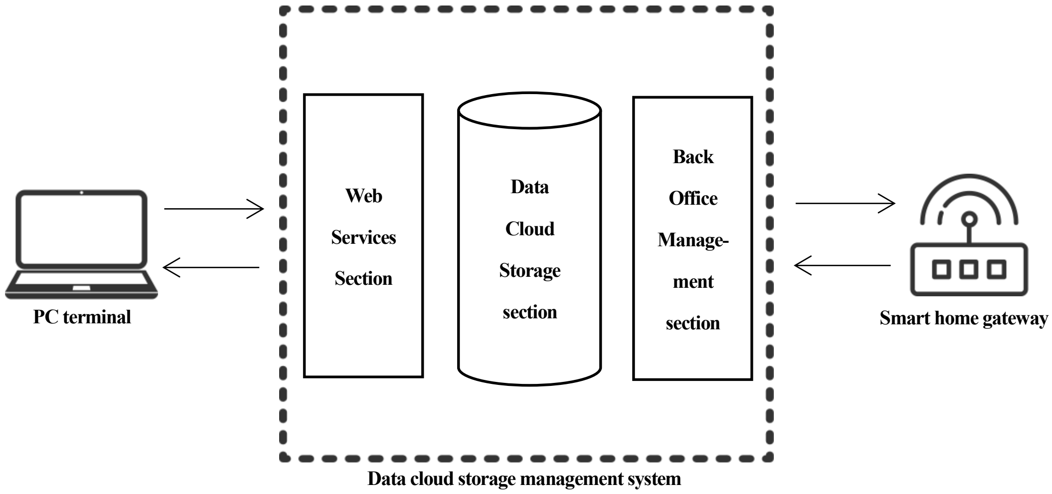

Figure 1. Structure of data cloud storage management system

2.1 Web service module

The Web service module is an interface that interacts with the administrator, who manages the data stored in the smart home gateway through the Web service module. To ensure data security, administrators need to perform dual authentication to log in to the cloud system. First, verify the administrator's user name and password. After the authentication is successful, they need to obtain user authorization. The administrator must obtain the mobile phone number bound to the user and the authorization code of the user before obtaining authorization from the user. When a user provides the mobile phone number and authorization code bound to the smart home gateway to the administrator, the administrator is considered to be authorized to manage the data of the home. The administrator can successfully log in to the cloud management system after the dual verification, and then can manage and maintain the data of the smart home gateway. It mainly includes viewing home device information, viewing home situation pattern information, editing home device information, editing home situation pattern information and data recovery operations.

The specific business flow of the Web service module is shown in Figure 2.

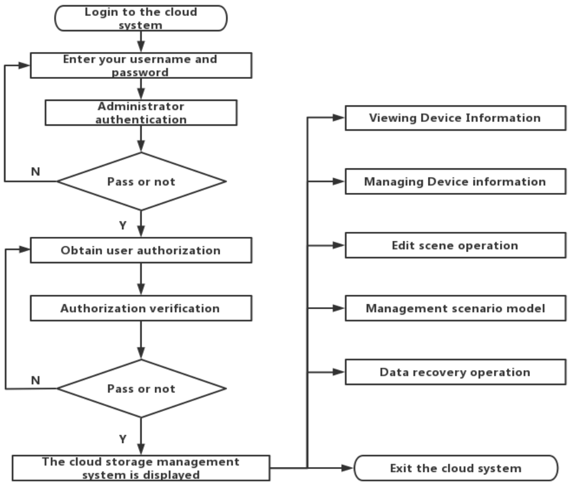

Figure 2. Shows the specific business flow diagram of the Web service module

2.2 Data cloud Storage plate

2.2.1 Cloud storage account information

There are two kinds of account information stored in the cloud server, one is the user information of the smart home, and the other is the administrator information of the data cloud storage management system [8]. The data cloud storage management system is provided for the after-sales management personnel of the smart home environment. The administrator's task is to regularly maintain the normal operation of the smart home, and remotely assist users to manage the smart home when they need assistance.

The design of the administrator information table is shown in Table 1.

Table 1: Design of administrator information table

Fields	Field meaning	Type	Instructions
Id	Job number	int (32)	Major key
UserName	Administrator name	varchar (32)	Sole
Password	Login password	varchar (32)	Non-null

At the same time, in order to ensure the security of the cloud storage management system, the cloud storage system only communicates with the smart home gateway that is successfully registered in the system. When receiving the communication message sent by the unknown smart home gateway, the cloud storage system directly discards it and does not communicate with it [9]. Therefore, when the user installs the smart home gateway for the first time, the manager will register the user information and gateway Mac in the cloud management system. When the smart home gateway communicates with the cloud management system, the gateway Mac is carried in the beginning line of the packet. The cloud server verifies the gateway Mac and communicates with it only after the verification is successful. The design of the user information table is shown in Table 2.

Table 2: Design of user information table

Fields	Field meaning	Type	Instructions
Id	Id number	int (32)	Major key
UserName	Username	varchar (32)	Non-null
PhoneNumber	Bound mobile phone number	varchar (16)	Non-null
Mac	Gateway Mac	varchar (48)	Non-null
Code	Authorization code	varchar (8)	
RegisterTime	Registration time	date	Non-null
Address	User address	varchar (48)	

2.2.2 Cloud storage device information

The cloud storage management system is responsible for centrally storing the data in the smart home gateway. The data stored in the smart home gateway mainly includes device information, scenario mode information, and device operations in scenario mode. Therefore, three types of data tables are mainly designed for storage: device information table, scenario pattern table and scenario relationship table. In the cloud storage management system, due to the increasing number of users in the use of smart home environment, in order to prevent the data expansion of a single table, the data is stored horizontally. The data table is divided horizontally according to the year of registration. The name of the data table after segmentation is named by the year and the specific name.

The specific design of the data table is shown below.

Table 3: Design of equipment information table

Fields	Field meaning	Type	Instructions
Id	Virtual primary key	int (32)	Major key
Home	Family	varchar (48)	Foreign key
DeviceMac	Device Mac	varchar (48)	Sole
DeviceName	Device name	varchar (64)	Non-null
Position	Installation position	varchar (32)	
Type	Device type	varchar (16)	

Table 4: Design of scenario information table

Fields	Field meaning	Type	Instructions
Id	Virtual primary key	int (32)	Major key
Home	Family	varchar (48)	Foreign key
SceneNumber	Scenario number	int (32)	Sole
SceneName	Scenario name	varchar (48)	Non-null

Table 5: Design of scenario relationship table

Fields	Field meaning	Type	Instructions
Id	Virtual primary key	int (32)	Major key
Home	Family	varchar (48)	Foreign key
SceneNumber	Scenario number	int (32)	Foreign key
DeviceMac	Device Mac	varchar (48)	Foreign key
DeviceState	Equipment Operation	varchar (16)	

2.3 Background Data Management module

The home device data operated by the administrator after logging in to the cloud management system is only the device data corresponding to the home gateway stored on the cloud server. When the administrator operations are completed, the background management program controls how to send these operations to the smart home gateway. The main function of the background management program is to synchronize the data modification operation of the cloud server and the data modification operation of the smart home gateway to ensure the data consistency and integrity of the two ends.

2.3.1 Data synchronization

In the smart home system, in order to facilitate the centralized and efficient management of smart home data, the cloud system is responsible for centralized storage of family data. When the data in the cloud or smart home gateway changes operations, the data at both ends need to be synchronized in time [10].

During synchronous initialization, the client information is registered on the server. The Mac address of the client is registered. After successful registration, the client encapsulates all initial data in a message starting with the keyword SendData and sends it to the server. The server writes data to the data table to complete the initial synchronization between the server and the client. After the wait period is complete, the client queries the operation record table. If the data modification operation is not synchronized to the server, the client encapsulates the operation in a request message starting with SendData and sends it to the server. The server receives the SendData message from the client and parses the data field of the message. Then the corresponding data modification operations are carried out on the server data [11]. If the operation is successful, the SynOk message is returned. If the operation fails, the SynError message is returned and the error code is encapsulated in the ErrorCode in the header field. After receiving the response message, the client parses the keyword of the message. If SynOk, clear the corresponding records in the operation log table. If it is SynError, the error message in the header field is parsed and written to the log file. No operation is performed on the operation record table. Then, the client sends a detection message starting with the keyword Detect to detect whether the server has data modification operations. After receiving the Detect detection message from the client, the server queries the operation record table. If there is a data modification operation that is not synchronized to the client, the operation is encapsulated in a response message starting with the keyword SendData and returned to the client.

After receiving the SendData message, the client parses the data field of the message, and then carries out corresponding data modification operations on the client data [12]. If the operation is successful, the SynOk message is sent to the server. If the operation fails, a SynError message is sent to the server with the error code encapsulated in the ErrorCode header field of the message. After receiving a message from the client, the server parses the keyword of the message. If SynOk is used, the corresponding records in the operation record table are cleared. If it is SynError, the error code in the header field is parsed and written to the log file without modification to the operation record table. A response message beginning with the

EndConnection keyword is then returned to the client. After receiving the response message, the client disconnects from the server. The data synchronization is complete and the waiting period is displayed.

The specific data synchronization process is shown in Figure 3.

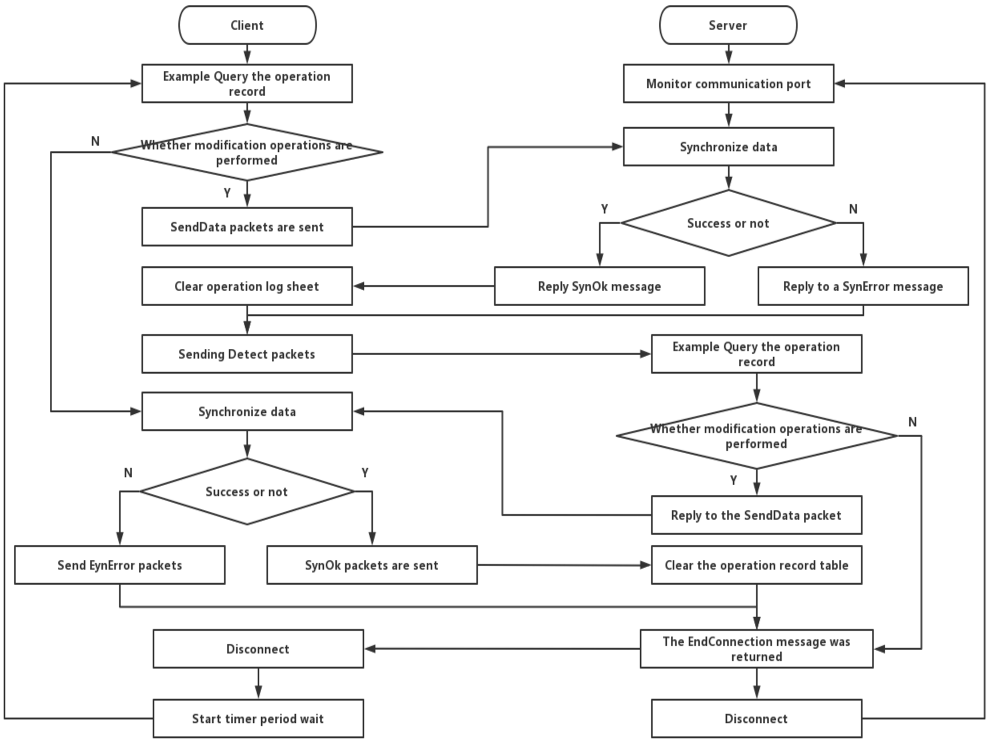

Figure 3. Data synchronization process between client and server

2.3.2 Data backup and restoration

Data backup is the basis of disaster recovery. In a smart home environment, the smart home gateway needs to work continuously for hours. To prevent data loss in the event of a system fault, the cloud management system provides data backup and recovery for the smart home gateway. There are two common data backup modes: full backup and incremental backup. Full back up is a complete backup of all data every day. This backup mode is simple and fast, but requires a large amount of storage space for data backup. Incremental backup is a complete backup of data at the beginning, and then only the new data and modified data on the same day are backed up. This backup method has the advantage of saving disk space, but the disadvantage is that the data recovery process is complicated when a fault occurs [13-14].

In a smart home environment, assume that a single family has 10 devices, set 10 scenario modes, and set an average of 1 scenario device in each scenario mode. According to the design of the data table in the preceding section, one piece of device information occupies about 30B, one piece of scenario information occupies about 22B, and one piece of scenario relationship information occupies about 22B. Therefore, the amount of data in a family is about$10 * 30\mathrm{B} + 1 * 22\mathrm{B} + 1 * 1 * 22\mathrm{B} \approx 0.5\mathrm{KB}$。 This amount of data is not large for a cloud server. In addition, when the cloud server manages multiple home gateways at the same time, the processing speed of home gateway data is the main bottleneck. Therefore, the advantage of using full backup is that the processing speed is faster during data backup and data recovery, thus improving the processing efficiency. To sum up, the cloud management system uses the full backup strategy to improve the data processing speed [15]. According to statistics, the frequency of smart home use is lowest from 1 a.m. to 4 a.m., so the cloud system carries out full data backup during this period every day. To prevent data expansion, backup data will be stored in the cloud system for 30 days, and the overdue backup data will be deleted by the background system in time. If you need to restore data, you can select a point in time within 30 days. The background management program reads the backup data of the family now and then performs data restoration. After the data recovery operation is performed by the cloud system, the data recovery operation is synchronized to the smart home gateway using the data synchronization protocol.

3. Database design and implementation

3.1 Architecture implementation of database server

In the information age, the traditional storage method cannot store massive data easily and quickly. Especially for Internet companies, all kinds of log information show explosive growth with the increase of users and website traffic. HDFS is a distributed file system that can be run on a common server. With elastic resource expansion and high concurrency, HDFS can reliably store large data files [16]. HDFS is different from other distributed file systems in that it has a high fault tolerance mechanism and backup mechanism to ensure disaster redundancy. HDFS has features such as dynamic expansion capability of seamless expansion, automatic data verification mechanism, reliable backup data storage mechanism, and high throughput access [17]. The HDFS architecture is shown in Figure 4.

The HDFS architecture adopts the master-slave structure. NameNode is a single primary server of HDFS and is used to manage the entire file system tree, directory, and metadata information. The actual file data is stored on multiple Datanodes [18]. The NameNode and DataNode can be separated to maintain and manage distributed files. The NameNode stores metadata information such as the file system tree, file segmentation, and storage location. The NameNode performs all the creation, deletion, opening, closing, naming, and directory operations of files. To read and write files, any client must first request access from the NameNode and construct a communication channel. The client and the specific DataNode [19] do the actual work.

The NameNode has the following functions: (1) Manages metadata and data blocks. NameNode manages file metadata, including namespace, mapping between files and data blocks, and mapping between data blocks and Datanodes [20]. Because data blocks may move between Datanodes and the location of data blocks changes frequently, Datanodes periodically send block reports to the NameNode. The NameNode has the location information of all data blocks. The NameNode periodically scans the metadata information to determine the data blocks that need to be moved to maintain the system load platform, the number of duplicates that need to be replicated, and the data blocks that need to be reclaimed. Then, the Namenode takes the next step as required.

(2) Persistent metadata. In order to speed up metadata response to access requests, NameNode generally stores metadata of files in memory for fast access [21]. To ensure metadata security, the NameNode records metadata modification operations in transaction logs. The transaction logs are usually stored on the NameNode’s hard disk. Persistent data includes file attributes of metadata and mapping between files and data blocks. Datanodes feed the mapping between data blocks and Datanodes back to NameNode when files are started by the system. Such information does not need to be persistent, because the information may change every time the system is started [22].

(3) Processing requests. The NameNode processes client and DataNode requests. After the NameNode starts, it listens to client DataNode requests, responds to requests in a timely manner, processes requests, and returns results. The client requests for creating files and directories, reading and writing files, renaming files, and deleting files cause metadata changes. DataNode requests are data block status reports and heartbeat responses sent by NameNode. The NameNode responds to the request and returns the processing result. The HDFS provides data interaction and data maintenance functions. Data interaction includes the HDFS process of reading and writing data. Figure 5 shows the HDFS file reading process.

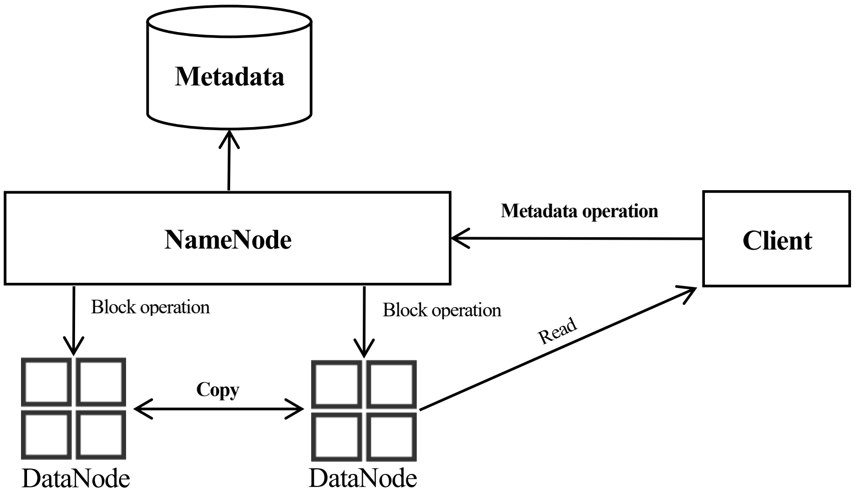

Figure 4. HDFS architecture diagram

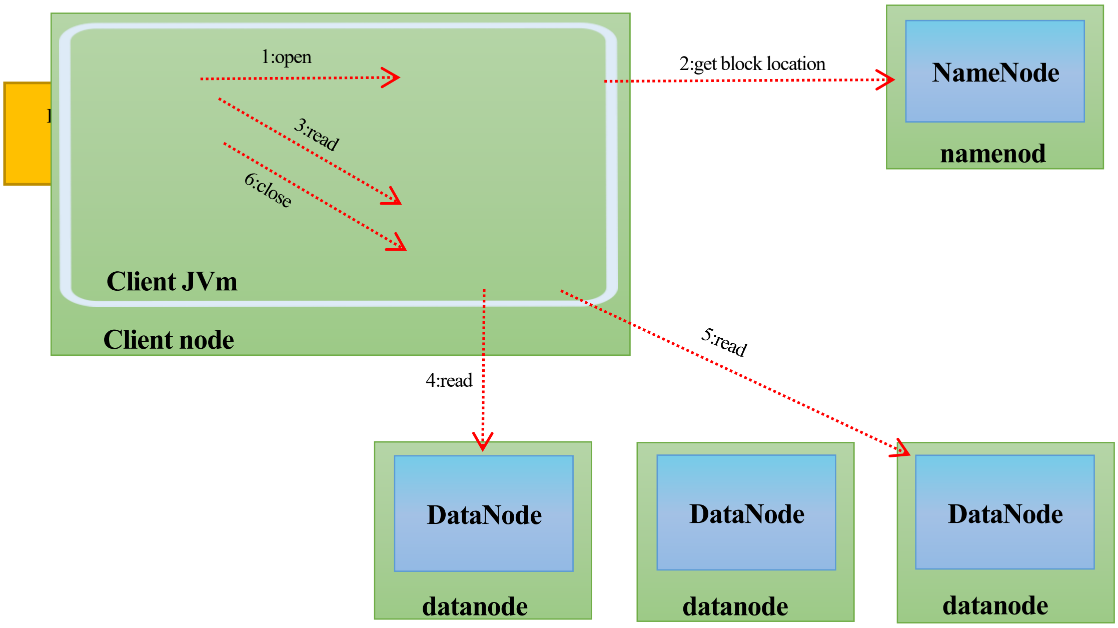

Figure 5. HDFS data reading process

The client accesses the open method to obtain an instance of HDFS DistributedFileSysterm. After the distributedFilesysterm instance communicates with NameNode through RPC, the location of file data blocks is obtained. These data blocks are sorted according to the Hadoop topology, with those closest to the client coming first. The DistributedFileSysterm instance will return to the client an instance of FSDataInputStream, which will be encapsulated as a DFSInputStream instance. The DFSInputStream instance facilitates the management of IO (data flow) that processes NameNode information of Datanodes. When the client calls the read method, the DFSInputStream instance finds the DataNode of the nearest data block to the client, reads the information of the data block, and returns it to the client. If the data of the first data block is read, the DataNode connection to the first data block is closed, and then the next data block is read and the above reading process is repeated. If the communication between the client and a DataNode fails during the reading process, the client automatically connects to the DataNode where the copy resides. At the same time, remember the faulty DataNode and notify the NameNode to ensure that subsequent data blocks of the DataNode are not read again. This operation ensures the high concurrency feature of HDFS. From the perspective of the client, the process of reading is the process of reading a continuous stream of data.

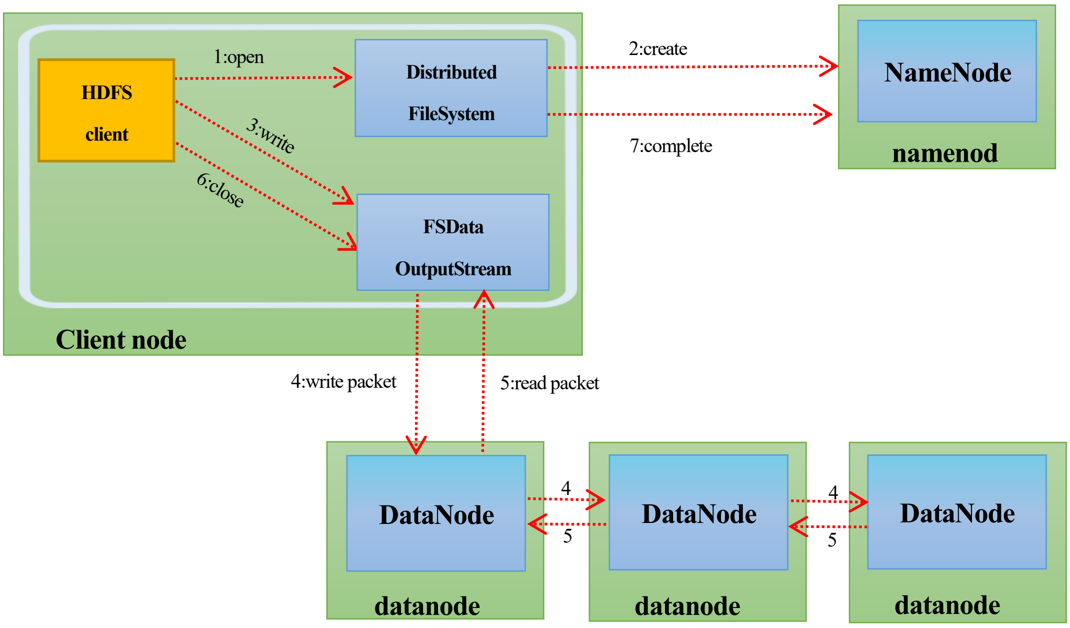

Figure 6. HDFS data writing process

Figure 6 shows the HDFS data writing process. The HDFS data write process is more complicated than the HDFS data read process. The client uses the create DistributedFileSystem method to create an instance, and the DistributedFileSystem communicates with NameNode through RPC. NameNode creates a new file metadata with no data block associated.

3.2 Process of distributed storage of data files

Dat file to check whether the user exists. If the user exists, the system obtains the IP address of the node where the user data block file resides. Read the Node.dat file from the management node to read the IP address list of the child node, according to the above information to complete the data segmentation, start the multi-threaded function to connect each child node at the same time to save the data in each child node, and finally update the Username table to find the file distribution. The Username file will be stored on a certain node, and the management node will assign the IP address of a node to the user according to the distribution of the existing Username file. The file name is the user name of the user, because the user name is unique in the system. Therefore, the Username of each user is also unique, which does not cause confusion. The distributed storage process of the file is shown in Figure 7.

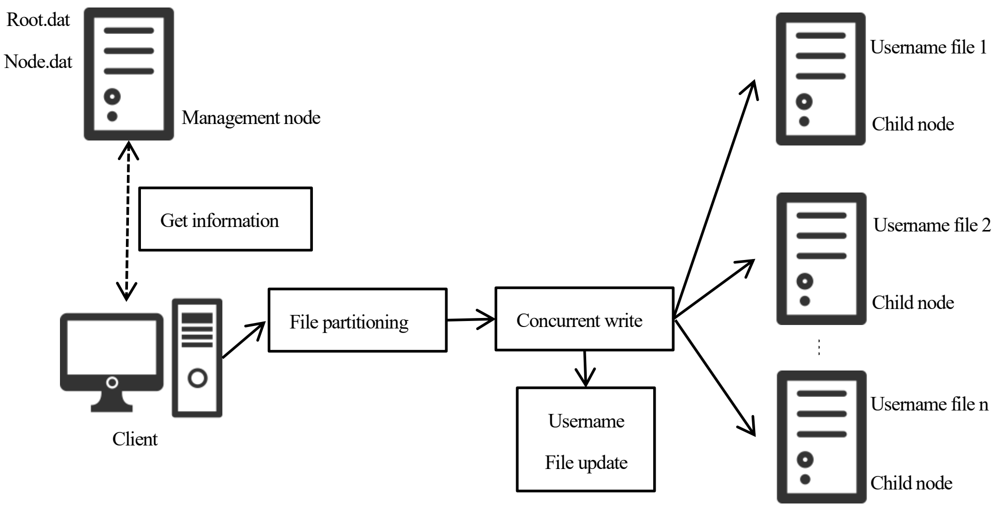

Figure 7. Distributed file storage process

3.3 Database file storage and calculation integration

In the cloud service system, using the obtained user name, file name, data block number and IP address of the file block information file, the location and file name of a data block can be accurately determined. At the same time, the client sends the command to start the calculation to the sub-node, and each sub-node will read the data block of the local file and calculate it. After the calculation is complete, the result of the summary is sent to the client. In this process, any data does not need to be moved, and the node storing the data block will automatically complete the data processing function. This way of migrating the computation to the nodes in a distributed and parallel manner according to the client instructions avoids the transmission of data in the network, which greatly improves the efficiency of data processing. For the relatively large data in the smart home, the advantages of this method are obvious. For different data, cloud application developers can define different data processing methods, the application field of cloud service system will be expanded, and the ability to deal with problems will be improved accordingly. The flow of system integration in this way is shown in Figure 8.

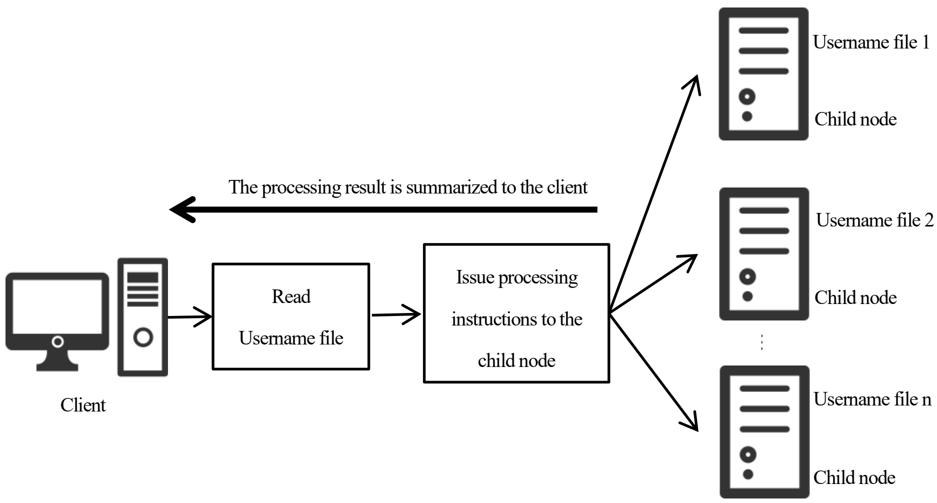

Figure 8. Process of database file storage and calculation integration

4. Hardware design of smart home cloud storage system

4.1 User access module

The user access module is the link between the user and the cloud storage system in the smart home. The big data oriented cloud storage system of the new generation of smart home design can use this module to provide users with various functions of the underlying realization. The hardware structure of the module is shown in Figure 9.

The user access module mainly consists of Web front-end access module and SDK access module. Web front-end access module provides users with registration, login, upload and download files and other functions in the form of a browser. The SDK access module provides users with access to basic file upload, download, delete and other operations. The biggest feature of the SDK access module is the ability to upload large files that are larger than the browser limit.

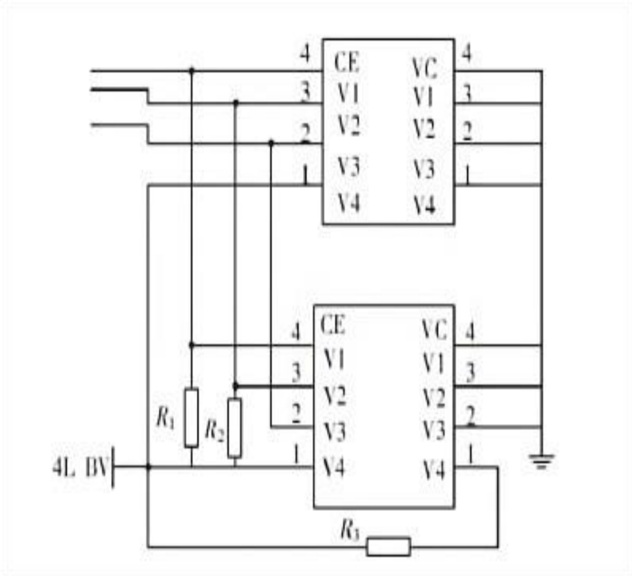

Figure 9. Hardware structure of user access module

4.2 Logical control module

Logic control module is the core of the whole smart home big data cloud storage system, is the hub of logic processing, all business requests are processed in this module, because the fuzzy processing features are added, it is also known as fuzzy processing module. The fuzzy processing module mainly includes agent module, Jetty Server module and metadata storage module. The detailed hardware structure is shown in Figure 10.

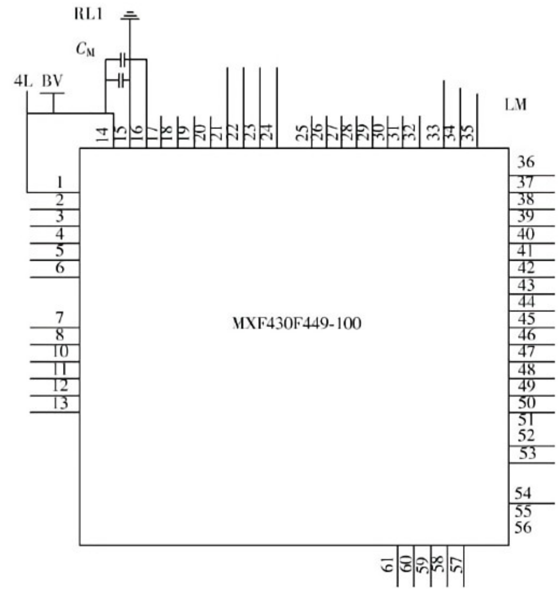

Figure 10. Hardware structure of logic control plate

The agent module receives the user's request to access the block, and then processes the requested data and home-related data types respectively. For the requests of each business class, the proxy module will send the request to the Jetty Server module, receive the processing results, and send the received results to the front-end browser to realize the processing of user requests. Both the agent module and the Jetty Server module are related to the access of user metadata during the processing of user requests, but only the Jetty Server module communicates with the storage module. In order to meet the requirements of concurrent performance of the system, Jetty Server can be regarded as a Server engine, and the processing logic of the whole system can be established on Jetty Server. Jetty Server mainly includes Connector components, Handler components, and thread pools. The metadata storage module stores the metadata information of the entire system, including user information, bucket information, object information, and is a key part of the system.

4.3 File read/write module

The file read/write module is mainly responsible for isolating the upper logical processing and the bottom storage to make the system designed in this paper more robust. The hardware interface structure of the module is shown in Figure 11.

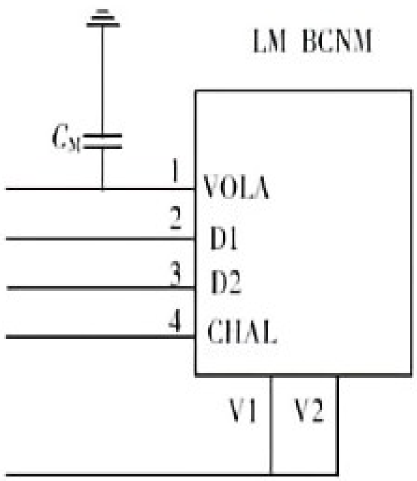

Figure 11. Hardware structure of file read and write module

In order to smooth the application of upper-layer modules, the new-generation smart home cloud storage system uses HDFS for storage access and designs a file read/write module. This module encapsulates the read/write interface of the HDFS of the underlying storage system and obtains the HDFS Operate interface. The upper-layer module can use this interface to access the cloud storage module. The access interface is unified, storage details are shielded, and the system structure is greatly enhanced.

5. Cloud storage system testing and analysis

In order to verify the effectiveness of the smart home big data cloud storage system designed in this paper, relevant experimental analysis is needed. The experiment takes the smart home big data cloud storage system based on ARM processor as a comparison analysis.

5.1 Cloud storage system performance test

This article uses average transactions per second, average transaction response time, CPU usage, and memory usage as metrics to measure system performance. Under the condition that each concurrent user sends 200 requests, the test results of the system in this paper and the ARM system are shown in Table 6.

By analyzing Table 6, it can be seen that the average number of transactions per second, average transaction response time, CPU usage and memory usage of the system in this paper are superior to the ARM system, indicating that the performance of the system in this paper is significantly higher than that of the ARM system and can meet the requirements of users.

Table 6: Performance comparison results

Index	System	Number of experiments		
		1	2	3
Average	Text system	324.51	331.56	322.67
Transactions persecond	ARM system	301.23	295.48	292.34
Average transaction Response time /ms	Text system	309.72	311.54	310.29
	ARM system	334.29	329.58	331.47
CPU Usage rate /%	Text system	98	97	95
	ARM system	82	76	83
Memory averaging Rate /%	Text system	57	59	63
	ARM system	40	38	35

5.2 Scalability test

Scalability is an important feature of big data cloud storage. Based on the performance of the system in this paper and the ARM system under different number of virtual machines, the scalability of the two systems is compared, and the results are shown in Figure 12.

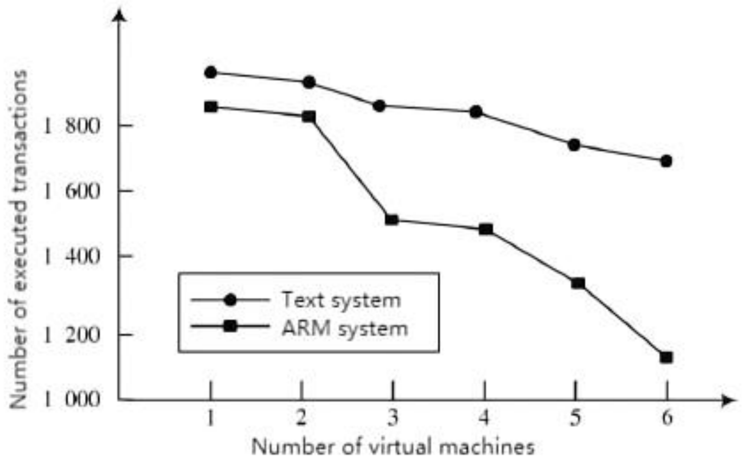

Figure 12. Scalability comparison results of the two systems

As can be seen from the analysis of FIG. 8, when the number of virtual machines gradually increases, the number of transactions that can be executed by both the system in this paper and the ARM system gradually decreases, but the decline rate of the system in this paper is significantly lower than that of the ARM system, indicating that the scalability of the system in this paper is better than that of the ARM system.

6. Closing remarks

With the continuous development of intelligent technology, the smart home industry is also showing new vitality and vitality. More and more new technologies are introduced into the home environment to provide people with a more comfortable living environment. This section provides technical support for smart home data management by analyzing HDFS data management. Based on the features of system data files, HDFS is used to manage data for the system to realize the combined storage of massive small files, ensuring that the crash of a single server does not affect data requests of the system. Through the design and implementation of the data cloud storage management system and database, this paper designs and implements a cloud storage system for smart home big data. Through the user access module, it provides users with various functions of the underlying realization transparently, and uses the logic control module to process the business requests of the system. By using the file read/write module, the upper logical processing and the bottom storage are isolated, which makes the designed system more robust, and realizes the incremental synchronization of cloud storage and heterogeneous database between the system and smart home. The results of system test and analysis show that the data cloud storage system is highly practical and feasible.

Acknowledgement

University level projects: Research on Smart Home System Based on Big Data Environment (JF-LX-202001)

Reference

[1] J. Zhang, "Design and Application of Smart Home Control System," *Automation Technology and Application*, vol. 40, no. 3, pp. 141-143, 2021.

[2] L. Guohao, C. Xiaoping, W. Botao, *et al.*, "Smart home system design based on Internet of Things," *Journal of Information Science and Technology*, 2019. [Online]. Available: https://doi.org/CNKI:SUN:ZXLJ.0.2019-22-015.

[3] B. Wang, "The smart home system based on Internet of Things technology," *Journal of Digital Technology and Applications*, no. 4, p. 2, 2018. [Online]. Available: https://doi.org/CNKI:SUN:SZJT.0.2018-04-037.

[4] Y. Li, "Simulation research of data information encryption and secure storage based on cloud computing," *Electronic Design Engineering*. [Online]. Available: https://doi.org/10.14022/j.issn1674-6236.2021.11.028.

[5] A. Ma, "Design of multifunctional smart home system based on Internet of Things and cloud platform," *Electronic World*, no. 12, pp. 6-7, 2020. [Online]. Available: https://doi.org/10.19353/j.carolcarrollnkiDZSJ.2020.12.002.

[6] H. Sun, "Design and implementation of massive big data storage system based on cloud computing," *Information and Computer*, vol. 34, no. 23, pp. 147-149, 2022.

[7] T. Zhao and J. Deng, "Cloud storage gateway for mass data backup study," *Computer CD Software and Applications*, vol. 16, no. 12, 2013. [Online]. Available: https://doi.org/CNKI:SUN:GPRJ.0.2013-12-041.

[8] T. Jiao, *Smart Home Design and Innovation in the Internet Era*, 2018.

[9] W. Shou, "Application of optical fiber communication technology in IoT analysis," *Journal of Digital Communication World*, vol. 160, no. 4, p. 192, 2018.

[10] S. B. Wang, Z. H. Wang, Z. T. Gao, *et al.*, "A data access method for Internet of Things devices," *Internet of Things Technology*, vol. 13, no. 3, pp. 64-66, 2023.

[11] L. Wu, P. Yuan, and W. Ding, "Smart home gateway and cloud server data synchronization protocol research," *Computer Technology and Development*, vol. 28, no. 9, pp. 5, 2018.

[12] N. Hrle, D. Martin, C. Mohan, *et al.*, "Data replication in a database management system," 2018.

[13] H. Deng, L. Liu, Z. Zhang, *et al.*, "Intelligent household design based on wireless communication," *Journal of Electronic Technology Applications*, vol. 47, no. 9, 2021.

[14] Z. Ma, Z. Wang, and Y. Yang, "Intelligent household monitoring system based on cloud platform design," *Journal of Electronic World*, no. 16, 2018.

[15] C. Yu and Z. Wang, "Unstructured data storage for cloud storage research," *Computer CD Software and Applications*, vol. 17, no. 19, 2014.

[16] Z. Liu, Z. Li, and X. Chang, "Design and implementation of the smart home system based on Arduino," *Journal of Electronic Design Engineering*. [Online]. Available: https://doi.org/10.14022/j.issn1674-6236.2021.06.007.

[17] C. Dong, X. Zhang, W. Cheng, and J. Shi, "Performance optimization of new storage device based on distributed file system," *Computer Applications*, vol. 40, no. 12, 2020.

[18] J. Wu, J. Xiong, H. Dai, *et al.*, "MIX-RS: Multi-index system for remote sensing data storage based on HDFS," *Journal of Tsinghua University: Natural Science Edition*, vol. 27, no. 6, 2022.

[19] T. Yang, M. Wang, and Y. Zhang, "Cloud computing data center HDFS differences stored energy saving optimization algorithm," *Journal of Computers*, vol. 42, no. 4, pp. 47-61, 2019.

[20] X. Chen, Z. Wang, and P. Zheng, "Implementation of power big data storage based on distributed file system," *Bonding*, vol. 2022, no. 6, 2022.

[21] J. Jiang, Y. Liang, and Z. Peng, "Data integration of data management platform design for large power systems," *Journal of Electronic Design Engineering*, vol. 28, no. 14, 2020.

[22] T. Yang, M. Wang, and Y. Zhang, "Cloud computing data center HDFS differences stored energy saving optimization algorithm," *Journal of Computers*, vol. 42, no. 4, pp. 47-61, 2019.