Journal of Software Engineering

ISSN 1819-4311

Academic Journals Inc.

www.academicjournals.com

Journal of Software Engineering 9 (4): 797-807, 2015
ISSN 1819-4311 / DOI: 10.3923/jse.2015.797.807

Application of Cloud Storage Technology in the Management of Massive Digital Teaching Resources Based on HDFS

Hui Cao
Henan Radio and Television University, Zhengzhou, 450008, China

ABSTRACT

In recent years, with the wide use of information technology in teaching resources construction, the number of digital teaching resources presents a massive growth. This study researches that applying cloud storage technology based on HDFS to the management of massive digital teaching resources so as to store them scientifically with information tools, solve all kinds of problems occurred in the process of the storage management, give full play to the role of teaching resources and serve the greater education group.

Key words: HDFS, cloud storage, massive digital teaching resources

INTRODUCTION

Cloud computing is the development of Distributed Computing, Parallel Computing and Grid Computing. Cloud storage is a new concept on the extension and development of cloud computing (Cao, 2014a). Cloud storage is receiving high interest in both academia and industry. As a new storage model, it provides many attractive features, such as high availability, resilience and cost efficiency (Chen *et al*., 2014b). Small businesses and startups have taken advantage, using low-cost cloud services during their first few years. Cloud storage services are also a boon for individual users, most of whom do not back up their computers and mobile devices regularly or at all (Neumann, 2014).

The management of massive digital teaching resources based on HDFS is a cloud system, that is used to solve the integration, integration and sharing of the lifelong learning resources which is distributed, heterogeneous, massive and multi mode and that offers the public services of digital teaching resources. The management of massive digital teaching resources based on HDFS is the specific application of storage management of the digital teaching resources, through the cluster applications, grid technology and distributed file system and other functions. This system adapts the application software to organize the devices, that are distributed in different regions of the network and in a large variety of different types, to load balancing and work together to offer data storage and business visits for the users of the digital teaching resources library (Cao, 2014b).

Hadoop is widely used in the cloud (Guo *et al*., 2014). It has emerged as a successful framework for large-scale data-intensive computing applications (Dong *et al*., 2014). Hadoop is an open source cloud computing platform of the Apache Foundation that provides a software programming framework called MapReduce and distributed file system, HDFS (Singh and Kaur, 2014). HDFS is the core part of the open-source cloud-computing platform Hadoop frame, used to realize storage of a large amount of data files in the cluster composed of many computers. HDFS supports high throughput data access, is suitable for applications that require massive data storage and is not high on the requirements of the hardware facilities, can be a very good management of cheap servers and using old equipment, reduce the equipment cost.

Now-a-days, there are many free cloud storage services available on the Internet and these services can be more convenient, when the following problems are solved. The first problem is the limitation of free service such as space limit, file size limit and file type limit. Secondly, different cloud storage services has different functions, thus some users may need to use more than two services simultaneously in order to fulfill their needs. The process of logging on and using multiple services will be very complicated. In addition, incidents such as server down or other internet problems often make users unable to access their data or even worse-to lose all their data (Su and Chang, 2014). So, we propose our demands and related technologies for cloud storage system here.

The digital teaching resources are storied in distributed servers, these servers maybe distributes in different data centers or areas and the distributed character leads that there are probably many differences in server type, operating system type and version. How to deal with heterogeneous of hardware and operating system is the technology issue that must be considered. HDFS was developed on the basis of Java, Java has the characteristic of platform independent and packaged with different platform interfaces. Therefore, HDFS has the same ability to cross platforms, can shield the differences of software and hardware equipment and design the unified client and server programs.

The digital teaching resources are widely used in distance education. Distance education involves many disciplines, education resources are very rich and the counts are large. When the amount of data is very large and it cannot be handled by the conventional database management system, then it is called big data. Big data is creating new challenges for the data analyst. There can be three forms of data, structured form, unstructured form and semi structured form (Pal and Agrawal, 2014). The construction goal of massive digital teaching resources management is to regulate huge resources accumulated in a long term, form a large-scale resources pool and realize the effective storage management. After the system of distance education resources management runs, resource storage work should be periodic, high-quality resources produced in a stage will be audited and storied unified. In daily operation, the system takes more work is to read resources. HDFS read and write mechanism is that only one client is allowed to write file and the file will on longer allow modification or writing once it was created, which is called write-one-read-many access model. HDFS performs badly in storing and managing a great number of small files as a result of the great memory occupation of the single Namenode and massive seeks and hopping from datanode to datanode (Zhang *et al*., 2014). HDFS usually deals with big file (M to T), the file changes are less in demand and the requirements of sequential read and treatment are common. Simple data read-write mechanism simplifies the problem of data consistency and improves the data access throughout. This design idea of HDFS fits the management demands of massive digital teaching resources very well.

Many vendors of commercial cloud storage services do not provide adequate means to secure the cloud from within the cloud infrastructure (Slamanig and Hanser, 2012). The management of massive digital teaching resources must story correctly, read accurately and ensure data integrity. Therefore, the technology supports the system should tolerate faults strongly and handle errors perfectly may appear on soft or hardware. HDFS frame a big cluster, any part in the cluster may be error. HDFS consider hardware error is normal, its core architecture aim is that it can check errors automatically and return to normal. HDFS provide several fault-tolerant mechanisms: The fault-tolerant method of NameNode is to add the number of core data structure copies; heartbeat detection between DataNode and NameNode can find failure data node in time and create new node; ensure the integrity of the text block by detecting the checksum; set time limits to delay the

complete of file deletion operation, improve file delete threshold and prevent misoperation that cannot be undone; when load balance is destroyed, data migration among DataNodes can make the cluster reaches new balance. These measures can ensure that when errors occur, the client and server can still provide resource servers normally.

The DataNodes of HDFS may distribute in or not in the same rack, even not in the same data center. When the DataNode failure occurs, the system adopts a serial of measures including data move, copy operation and rebalancing the load etc, when the volume of business changes, the system has the scalability. All of these technical movements are transparent for users, the client and the client programs are relatively fixed. The users do not need to care how HDFS runs, need only to familiar with the methods of the client applications, without feeling any difference because of file remote storage. This character provides a good user experience both for the client users of digital teaching resources or the users who develop the upper application.

MATERIALS AND METHODS

Compared with the traditional way of storage, cloud storage has the scalability and high performance, low cost and other advantages. Different cloud storage systems have different technical architecture and characteristics (Liang, 2014). The technology of HDFS is discussed as follows:

Block, datanode and namenode: Block, DataNode and NameNode is the basic logical unit of HDFS.

Block is the smallest unit of HDFS storage file. A file needed to storage is divided into several blocks, the block size is same except the last one. HDFS default block size is 64 MB but this value is not must and can be set according to the actual needs. The files storied in HDFS are all big, using block mode can avoid the deficiency of single node disk space. If the block damages, only a small part of a file will be affected, HDFS fault-tolerant mechanism will detect and recover the file immediately. Therefore, using block storage mode is advantageous both for storage and fault-tolerant.

DataNode, as the name suggests, is a node stored data. The physical location stored file is the so-called DataNode. Usually, a DataNode instance is deployed in a server of the cluster but there is also case that multiple DataNodes run on a single server.

There is usually only one metadata node in a HDFS (Hadoop already exists version to support deploying multiple metadata nodes). The metadata node is deployed on a server which has better performance in the cluster. It is responsible for the management of file metadata information, including creating file and directory, to delete, to rename and storage location etc.

Core framework: It is shown as Fig. 1 that HDFS is a Master/Slave distributed file system. The metadata is deployed on a single server which can also run a DataNode. Other servers in the clusters run a DataNode themselves. A typical HDFS cluster is grouped with one metadata node and a large number of DataNodes. The metadata is manager that is responsible for arbitration and storage of all metadatas in the file system, dealing with read-and-write requesting from the client, managing file operation including opening, closing and renaming and data block mapping. The DataNode is used to story file. When the file is storied, it will be firstly divided into fixed size block. File storage is actual to store these file blocks. The file block is located on the DataNodes, the DataNodes do the operation of creation, deletion and copy according to the orders of the metadata node.

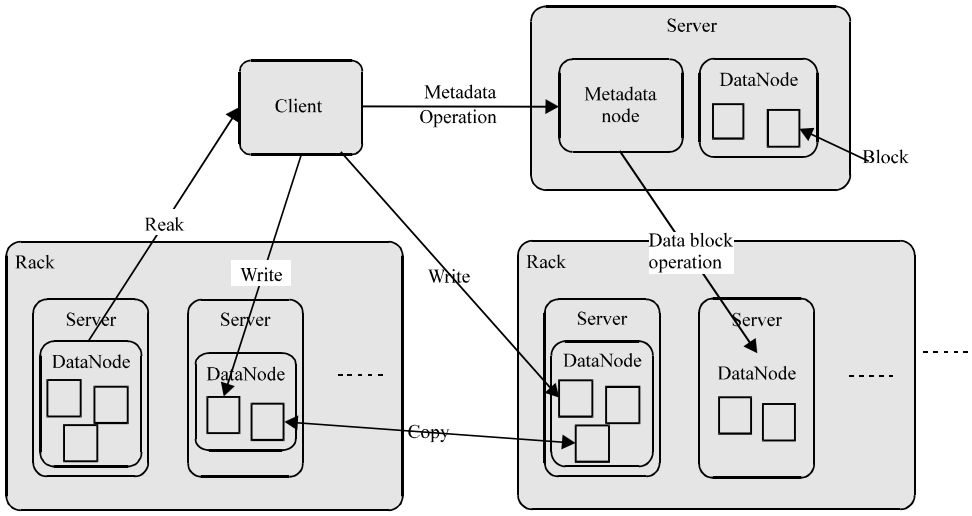

Fig. 1: Core framework

Metadata management strategy: The metadata node is responsible for the management of all metadata in HDFS. A metadata management mechanism was discussed in this part, which is applied for cloud storage (Chen *et al*., 2014a). The hard disk, with the system which the metadata node belongs to, story tow documents that respectively is the Editlog files and FsImage. The metadata node manages the metadata through the two documents.

The Editlog stories the changes of all the metadata such as creating file, changing the replication factor etc. The FsImage is responsible for metadata persistent storage including namespace, text block mapping and file attribute.

It is shown as Fig. 2, when the metadata node starts, the system will read the newest Editlog and FsImage from the disk. If the version of the two documents is different, the management process will be prematurely terminated, in contrast, the metadata operation recorded in the Editlog will be acted on the FsImage. After the update is complete, the FsImage will include all previous metadata operations that the system acts on. Because these operations need to be saved as important information for long time, the FsImage in memory will be refreshed into disk next. The records in the Editlog have already been saved, therefore, the Editlog would be cleared.

Data storage strategy

Data storage method: HDFS is used to provide storage service for big files. Instead of being directly stored on disk, these files are divided into a series of data blocks of equal size (except for the last one) that are the smallest units to store files. HDFS allocates these blocks to the DataNodes that are responsible for saving blocks.

Redundant data storage method: Cloud storage is an ideal way of data storage, so cloud storage has been widely used. But the data security in cloud storage must be solved. Cloud storage must ensure that data is not lost or damaged. Most cloud storage systems use replica redundancy technique to solve the problem of the loss of data (Mao *et al*., 2014). In order to guarantee data security, each data block of HDFS will be stored redundantly. The system provides a parameter called as replication factor which value is settled to determine the number of copies of redundant data. When storing the data block, the system will take into account the copy data to support

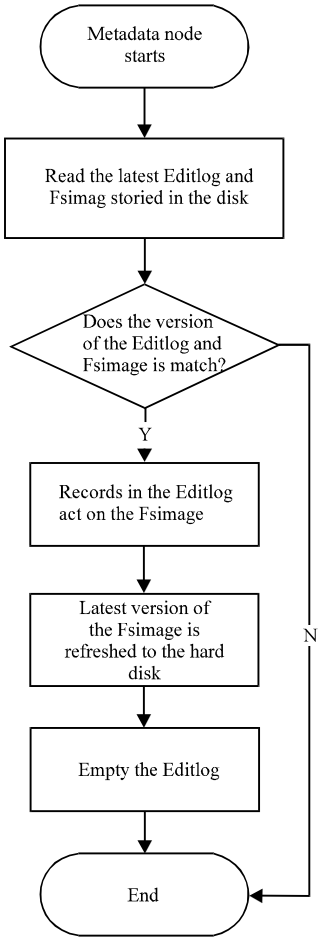

Fig. 2: Metadata management strategy

storing redundantly. When the DataNode fails, the system will think that the original copy has expired and will also start the copy program to recreate redundant data.

The system adopts heartbeat mechanism to check whether the DataNode is effective. Figure 3 shows that the DataNodes periodically send heartbeat information and data block report to the metadata and the metadata node judge the data validity according to whether it can receive heartbeat information from the DataNode. If the heartbeat information from some DataNode is not received, the metadata will think that the DataNode is failed. The number of data block copies will be less than the minimum value of the copy factor because of fail DataNode. Once the number of data block copies is not satisfied with the copy factor setting, the metadata node will start data copy so as to return the copy factor to normal state.

Copy factor can be flexibly arranged according to the demand of application, the default value is 3. When the copy factor is 3, one data needs to be stored in three copies. Considering the data access efficiency, two copies will be storied in the DataNode in the same rack, one copy will be storied in the DataNode in the other rack.

The size of HDFS' metadata is limited to under 100 GB in production, as garbage collection events in bigger clusters result in heartbeats timing out to the metadata server (NameNode) (Hakimzadeh *et al*., 2014).

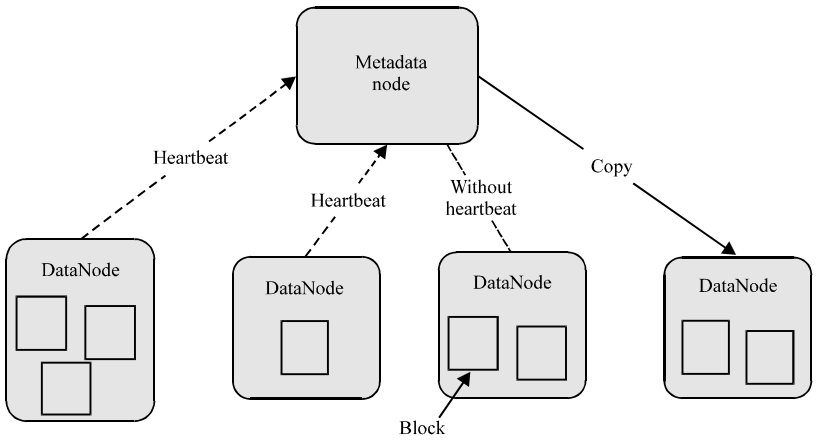

Fig. 3: Heartbeat mechanism

Effective data management is an important issue for a large-scale distributed environment such as data cloud. This can be achieved by using file replication, which efficiently reduces file service time and access latency, increases file availability and improves system load balancing (Long *et al*., 2014).

Data fault tolerance mechanism: The task of HDFS is to store files, so ensuring the file data safe and accuracy is the target that the system must complete. There are multiple factors to affect data validity in the process of practical application, such as storage device failure, software defect and abnormal network transmission etc. In some extent, these factors are inevitable, therefore, the system is required full fault-tolerant mechanism, can deal with various data errors and not affect the normal operation of the system.

The common storage errors of file data can be classified into three, data redundancy, abnormal data, the Editlog and the FsImage error.

DataNode server failure, abnormal DataNode, increscent replication factor and the lost data block will all cause data redundancy problem. The system finds redundancy problem through the heartbeat mechanism, so as to take measures to increase redundancy.

Data exception refers to the errors or loss of the data storied in the disk. During storing the file, the system collects relevant information of each data block, calculates the checksum and creates the checksum file. The checksum file is storied hidden in the system. When reading the file, the system finds whether the data is abnormal through data block checksum. If the checksum does not match, the system will retrieve the other data block to read the file.

The Editlog and the FsImage is the core data structure of the file system, manages all metadata information of the system. The lost or errors of the two documents is disaster for the system. HDFS has set up a number of backup files and the secondary server in the metadata node server stores the two documents. The system modifications of the Editlog and the FsImage will all be synchronized to their backup files and select the latest version in the need to the records.

RESULTS AND DISCUSSION

The construction of the massive digital teaching resources is a systematic project. According to the construction needs of resource management, combining with the research

findings of HDFS file storage technology and fully integrating cloud storage design idea, this study provides the cloud storage model based on HDFS.

Overall architecture: Figure 4 shows that the massive digital teaching resources center is located in the center node and a large number of resource storage nodes distribute in the architecture. The center node is subordinate with the other resource storage nodes and the resource storage nodes are equivalence with each other. The resource storage nodes provide resource storage service and send the resource catalogue information to the center node. The resource catalogue is registered in the center node. Besides, the center node can also bear part of resource storage task. The overall architecture runs based on HDFS technology.

Basic architecture: The basic architecture of the systematic project of massive digital teaching storage manage is shown as Fig. 5. The basic architecture is divided into four levels from bottom to top. These levels are hardware layer, cloud storage software based on HDFS layer, application interface layer and business application layer. The hardware layer mainly manage a variety of

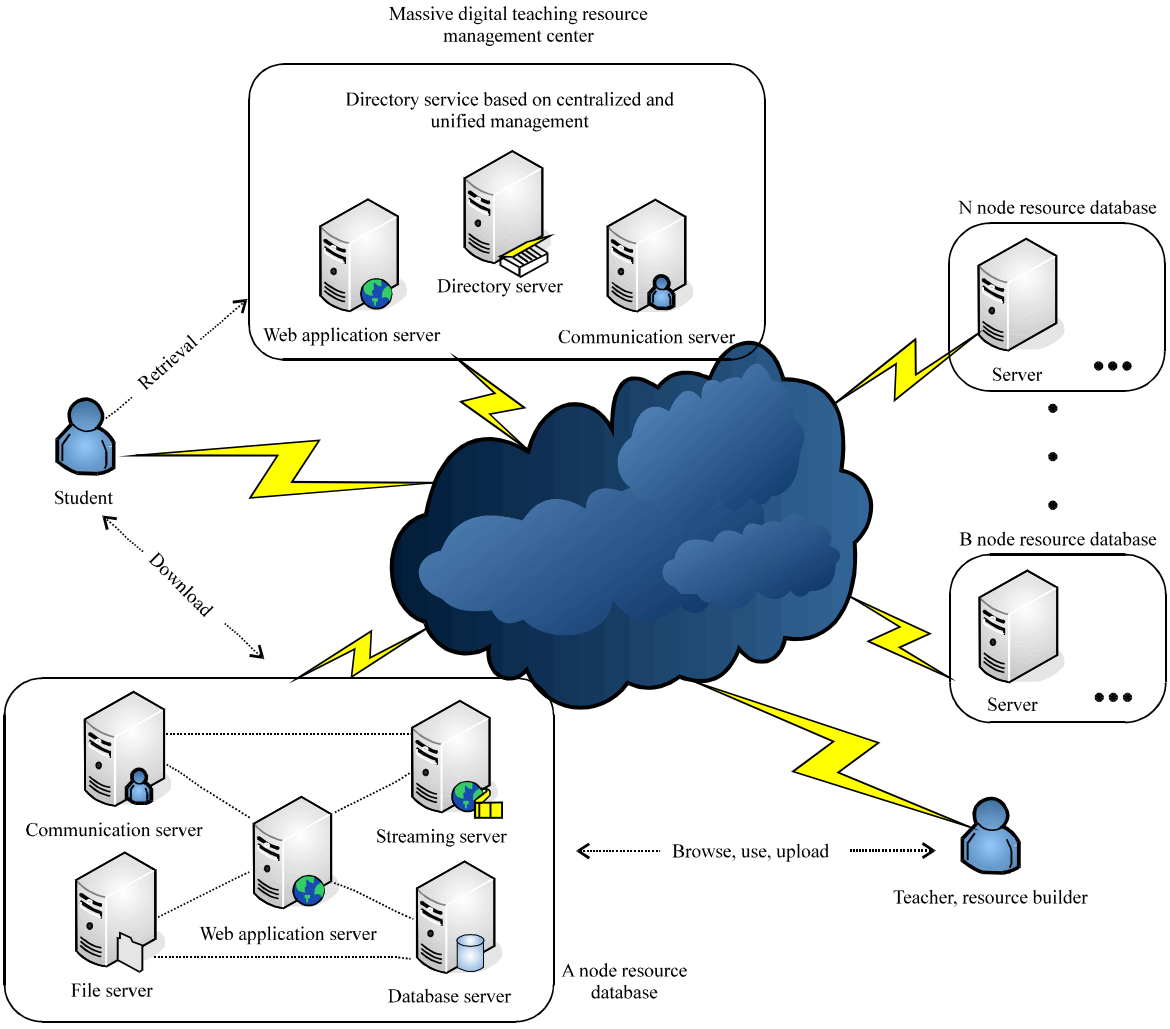

Fig. 4: Overall architecture

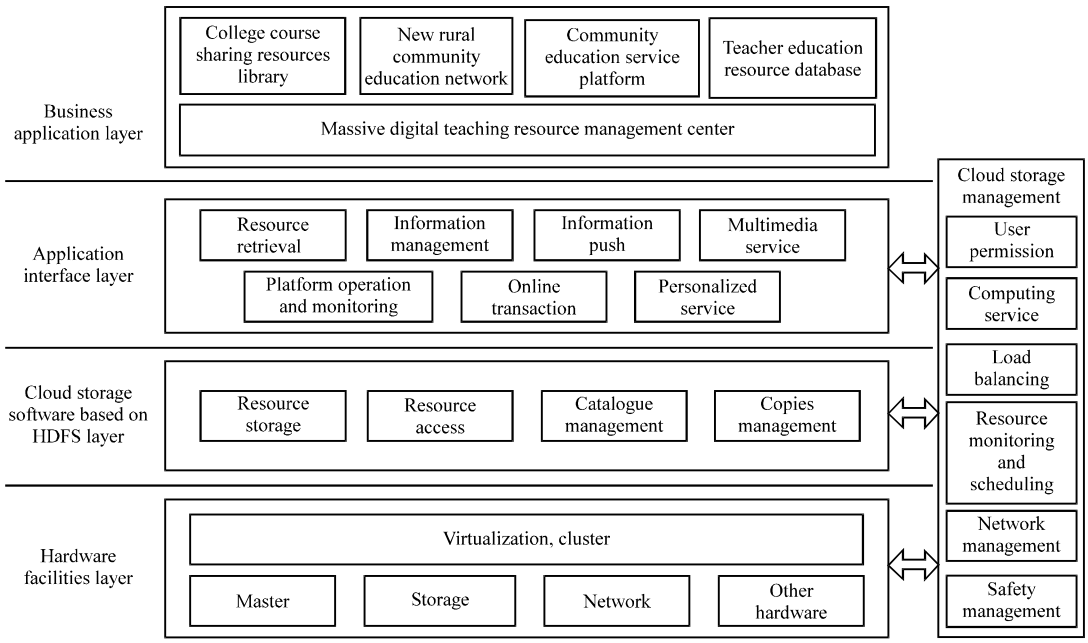

Fig. 5: Basic architecture

hardware facilities, including storage servers, network servers etc. The cloud storage software based on HDFS layer adopts distributed file storage technology to realize storing resource, visiting resource, managing catalogue and copies. The application interface layer mainly refines the commonness of various business applications, designs and realizes common interfaces, so as to reduce the code burden of business application layer, realize code reuse and improves software development efficiency. According to the specific work and the user needs, the business application layer can build all kinds of application platform based on the massive digital teaching resource management. When the user submits resource storage requirement in business system, the business application layer will call the corresponding interface from the application interface layer. After the interface finishes data operations, the requirement is passed down to the cloud storage software layer. The cloud storage software layer completes data storage action according to the orders and passes data to the hardware facilities layer. The data will be located in the storage server for persistent storage.

The hardware facilities layer realizes the management of several of hardware including storage servers, network servers etc, through virtualization technology. Virtual technology can shield the heterogeneity among the hardware and can realize unified management of the cluster hardware facilities distributed in different locations, so as to provide storage equipments for the cloud storage software layer. The technical details of the underlying hardware are completely transparent to the users. The users do not need to care about where the resources are storied, which server is using and which operating system the server uses, etc.

The cloud storage software based on HDFS layer adopts distributed file storage technology to realize resource storing, resource visiting, catalogue management and copies management, etc. The storage layer supports digital resource discovery and location services. It maps the data access service to underlying storage access operation of heterogeneous distributed storage environment, get digital resources from storage nodes according to certain scheduling strategy and realize unified

resources access and management mechanism. In order to improve the system performance, the cloud storage management designs the resource scheduling strategy according to resource access hit rate level combined with the storage state of the node and network load, so as to reduce data access latency, reduce network transmission load and improve data communications efficiency. At the same time, the storage layer create and manage the data copies according to the resource access rate in order to get better resource access efficiency fault-tolerant performance. The technology details and functions of the storage layer have been described, so here will not cover those again.

The application interface layer provides common interfaces for the business application layer. For the users can find and use information more accurate from massive information, the interface layer studies retrieval technology suitable for public service of digital resources, including Full-text retrieval technology, concept retrieval technology, query classification technology, knowledge based information retrieval technology and ontology based information retrieval. Resource retrieval provides full-text retrieval service function operated on massive digital resources of multiple formats and types, uses intelligent full-text search engine to help readers to effectively locate the required information. The retrieval system to support multimodal retrieval, mainly include the integrated multi-level information retrieval for the massive heterogeneous information. From the point of view of resource distribution, multi mode retrieval is divided into three kinds. They are unified retrieval of cross-database, cross-type and cross-system. From the point of view of storage resources types, multimodal retrieval mainly include full text retrieval, metadata retrieval, graphics and image retrieval, audio and video retrieval and video on demand, comprehensive literature retrieval and so on. Information management is with the workflow as a link, with the task list as drive, with service as target, reflects the work mechanism of division and cooperation, achieves a complete set of tasks of production, publication, management, analysis of information. Information management mainly includes, edit information, review and maintain the editing information, publish reviewed information and so on. Information push using advanced push technology to provide timely and accurate resource information for users. Multimedia service adopts streaming transmission technology to realize on-demand network function so as to use multimedia book for users. Platform operation and monitoring mainly monitor, count and manage the key data of kinds of application system. It mainly realizes management of statistics, monitor and edit log. The statistics management counts kinds of key data, including comprehensive statistics, the click statistics by the time, he click statistics by the area, he click statistics by the channel, the client software statistics, workload statistics and so on. The monitor management mainly realizes to monitor the current running state, user access and so on of the servers in real time. The edit log mainly collect, summarize and manage varieties of application system, database system, operating system and server system. Online transaction is business circulation means of digital resources. It ensures the safety of online transaction data and realizes safe network transaction environment through identity authentication and data encryption technology. Personalized service supports users setting individual special topics, establishing personalized online resource database and realizing online learning and studying.

The business application layer can establish kinds of branch library and research platform. It is the user client realized business function. The massive digital resources management center has all catalogues and entities of resources. The other business applications are branch database or research platform, such as the college course sharing resources library, new rural community education network, community education service platform, teacher education resource database and so on. The resource center provides two kinds of data supports for the branch database. One is directory mapping; the other is to provide resource entities.

CONCLUSION

This study analysis the technology needs of massive digital teaching resources management and study the core technical architecture, metadata management strategy, data storage strategy and data fault-tolerant mechanism of cloud storage technology based on HDFS. The technical characters of cloud storage based on HDFS can solve the technical problems of the management of massive digital teaching resources. So, this study designed the model of the management of massive digital teaching resources based on cloud storage, specifically introduces the overall architecture and the basic architecture of this model. This model has already been applied in the technical realization of the public service platform of lifelong education resource in Henan Province, China.

ACKNOWLEDGMENTS

This study is supported by the projects of Research and explore the cloud storage in the construction of distance education resources (project number: 14A520044), Research on distance education resource database scheme based on the cloud storage (project number: 20140672) and the model study of the public service platform of massive digital resources of lifelong education in Henan Province based on cloud storage.

REFERENCES

Cao, H., 2014a. Research on the optimization of the storage strategy of HDFS. Adv. Mater. Res., 989-994: 2450-2453.

Cao, H., 2014b. Research on the security of the digital learning resources library based on cloud storage. Adv. Mater. Res., 989: 5007-5009.

Chen, F., M.P. Mesnier and S. Hahn, 2014a. Client-aware cloud storage. Proceedings of the 30th International Conference on Massive Storage Systems and Technology, June 2-6, 2014, Santa Clara, CA., USA., pp: 1-12.

Chen, X.F., Y.S. Lou and D.M. Hu, 2014b. A metadata management mechanism based on HDFS. Applied Mech. Mater., 577: 1026-1029.

Dong, B., Q. Zheng, F. Tian, K.M. Chao, N. Godwin, T. Ma and H. Xu, 2014. Performance models and dynamic characteristics analysis for HDFS write and read operations: A systematic view. J. Syst. Software, 93: 132-151.

Guo, D., W. Shi and M. Hou, 2014. Improvement and implementation of hadoop HDFS model in private cloud. Proceedings of the 9th International Symposium on Linear Drives for Industry Applications, Volume 4, July 7-10, 2014, Hangzhou, pp: 663-670.

Hakimzadeh, K., H.P. Sajjad and J. Dowling, 2014. Scaling HDFS with a strongly consistent relational model for metadata. Proceedings of the 14th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems, DAIS 2014-held as part of the 9th International Federated Conference on Distributed Computing Techniques, June 3-5, 2014, Berlin, Germany, pp: 38-51.

Liang, X.Y., 2014. Analysis on non-center cloud storage architecture of gluster. Applied Mech. Mater., 587: 2346-2349.

Long, S.Q., Y.L. Zhao and W. Chen, 2014. MORM: A multi-objective optimized replication management strategy for cloud storage cluster. J. Syst. Archit., 60: 234-244.

Mao, H.X., K. Huang and X.L. Shu, 2014. Research of cloud storage and data consistency strategies based on replica redundant technology. Proceedings of the International Conference on Computer, Intelligent Computing and Education Technology, March 27-28, 2014, Hong Kong, pp: 1053-1056.

Neumann, P.G., 2014. Risks and myths of cloud computing and cloud storage. Commun. ACM, 57: 25-27.

Pal, A. and S. Agrawal, 2014. An experimental approach towards big data for analyzing memory utilization on a hadoop cluster using HDFS and MapReduce. Proceedings of the 1st International Conference on Networks and Soft Computing, August 19-20, 2014, Guntur, pp: 442-447.

Singh, K. and R. Kaur, 2014. Hadoop: Addressing challenges of big data. Proceedings of the IEEE International Advance Computing Conference, February 21-22, 2014, Gurgaon, pp: 686-689.

Slamanig, D. and C. Hanser, 2012. On cloud storage and the cloud of clouds approach. Proceedings of the International Conference for Internet Technology and Secured Transactions, December 10-12, 2012, London, pp: 649-655.

Su, W.C. and S.E. Chang, 2014. Integrated cloud storage architecture for enhancing service reliability, availability and scalability. Proceedings of the International Conference on Information Science, Electronics and Electrical Engineering, Volume 2, April 26-28, 2014, Sapporo, pp: 764-768.

Zhang, S., L. Miao, D. Zhang and Y. Wang, 2014. A strategy to deal with mass small files in HDFS. Proceedings of the 6th International Conference on Intelligent Human-Machine Systems and Cybernetics, Volume 1, August 26-27, 2014, Hangzhou, pp: 331-334.