Internet of Things and Cloud Computing
2016; 4(3): 28-33
http://www.sciencepublishinggroup.com/j/iotcc
doi: 10.11648/j.iotcc.20160403.13
ISSN: 2376-7715 (Print); ISSN: 2376-7731 (Online)

Survey of Big Data Storage Technology

Wang Weichen, Gao Jing, Cao Rui

College of Computer and Information Engineering, Inner Mongolia Agricultural University, Hohhot, China

Email address:
ye891080965@163.com (Wang Weichen), gaojing@imau.edu.cn (Gao Jing)

To cite this article:
Wang Weichen, Gao Jing, Cao Rui. Survey of Big Data Storage Technology. *Internet of Things and Cloud Computing*. Vol. 4, No. 3, 2016, pp. 28-33. doi: 10.11648/j.iotcc.20160403.13

Received: April 27, 2016; **Accepted**: June 4, 2016; **Published**: June 21, 2016

Abstract: Big data storage is the foundation of big data processing and analysis. By researching and summarizing main processing technology of data storage, this paper respectively investigates and analyzes the following four aspects: distributed file system, NoSQL database, database appliance and new-type data storage technology of MPP architecture. In addition, this paper gives some recommendations applicable to different environments in favor of grasping the development states of data storage technology from different angles. This paper summarizes file segmentation, appropriate scenarios and merits and faults of distributed file system, and mainly analyzes and summarizes the theories and appropriate scenarios of four data storage models of NoSql database. Furthermore, this paper investigates and concludes the developments and features of database appliance minutely. At the same time, outline MPP (Massively Parallel Processing) architecture, a new data storage technology. At last, the research trends of storage technology are prospected, providing references to the research of big data storage technology.

Keywords: Big Data Storage, NoSql, Distributed File System, Database All-in-One Machine, MPP Architecture

1. Introduction

In the past few decades, with the expansion of [1]application scale, web services evolved from a single form into multimedia form, leading to diverse data structures and forms, and exponential data growth. International Data Corporation (IDC) predicts that in future data size will double in every two years [1]. The pioneer of big data research—McKinsey & Company, a consulting company in the United States, defines big data as: the data set whose scale is beyond the ability of acquisition, storage, management, and analysis of conventional database tools [2]. The traditional data storage system has reached a bottleneck, and cannot finish data processing in time. Big data has such features as high capacity, various data types, low value density, high processing speed, complex dynamic relation between data. And the requirement of high availability, scalability and reliability [3], poses challenges to traditional data storage technology.

This paper researches and summarizes new kinds of data storage technologies, for the problems of application scale expansion, rapid data growth, multiple data types. First, investigate and analyze the research status of distributed file system. Besides, analyze and compare the characteristics of main distributed file systems, such as GFS [4], HDFS [5], GlusterFS [6], GridFS [7], TFS [8], Lustre [9], FastDFS [10]. In addition, by researching four data models of NoSql database, respectively contrast the storage technology of key value model, column type model, document model and graphic model. And research the applicability and architectural feature of current big data all-in-one machine. At last, introduce the definition and scenarios of new kinds of database cluster using MPP (Massively Parallel Processing) [11] framework, and forecast the probable research trends of storage technology in the future.

Fund Project: National Natural Science Foundation of China (61462070); Regional science and technology planning project (20130364).

2. Big Data Storage Technology

This paper investigates and analyzes big data storage technology from the following four aspects: distributed file system, NoSQL database, new-type data storage technology of MPP architecture and database all-in-one machine. In addition, this paper gives some recommendations applicable to different environments in favor of grasping the development states of data storage technology from different angles.

2.1. Distributed File System

File system is the basic of application program. However, with the development of network application, data grows rapidly. So big data storage technology has become the main task of enterprises and research institutions. Because the storage capacity is limited, traditional storage systems can hardly solve the problem of big data storage. So we use distributed file system to transfer system load to multiple nodes. Distributed file system provides polymeric storage capacity and I/O bandwidth, so that the scale of the system can be easily extended [12].

Generally, determining whether a distributed file system is successful depends on the following three factors: data storage mode, reading rate, security mechanism. However, there are still improvements to be made in distributed file system. For example, GFS and HDFS designed for large files cannot meet the storage requirements of many small files. Small file's access frequency is high, leading to high frequency to access the hard disk. So the performance of I/O has been reduced [13]. Small files can also lead to the production of a large number of metadatas, which will affect the metadata server management and restorability, and then result in the decline of total performance. What's more, since the file is relatively small, it is prone to creating file fragmentation, wasting disk space. Creating links for each file will cause network delay [14].

Table 1 summarizes and contrasts several common distributed file systems broadly, such as GFS, HDFS, TFS, Lustre, etc.

Table 1. *Analysis list of common distributed file systems [15-20].*

Name	File Segmentation	System Backup	Merit	Demerit	Application scenarios
GFS	Files stored in GFS are divided into fixed-size blocks	Each block is copied to multiple chunk servers, saving 3 copies in default.	GFS will not regard hardware failures as abnormal. Usually an update is processed by adding new data rather than changing the existing data.	Do not apply to the small file storage. Extra small files will degrade performance.	Large distributed massive data set. Data size is generally among 4G~40G.
HDFS	Large files are divided into some blocks whose default size is 64MB. Each block will store some copies over more than one data node.	Support data replication. Store multiple copies over different nodes.	Expansibility is very strong. A single HDFS instance can support tens of millions of documents. And has high real time capability.	Cannot be used to the scenarios that requests low latency in data access. Cannot store large-scale small files.	Very large data sets whose size is at GB to TB level.
GlusterFS	Do not support file segmentation	Support data replication and provide the global namespace. Multiple copies of multiple files can be stored in different hosts. While reading a copy, the system will choose the closest copy acquiescently.	Support CIFS, NFS and native machine using GlusterFS clients. Multi-file system can be deployed on virtual distributed file system. Use the admin console, managing the central update easily. All the nodes can be used to get data.	Can manage system only on one server, with no redundancy. Cannot add a new node halfway. Don't know how to add multiple disks to each node; There is no security policy in GlusterFS.	Support massive large file storage of PB level.
TFS	Do not support file segmentation. A large number of small files will be merged into one large file.	TFS stores data files in blocks, and store multiple copies in case of data security.	The operation is simple, with smooth expansion and load balancing. Support linear scaling. Can easily extend to PB level.	When concurrency is high and file size is more than 5 MB, severe bugs arise in TFS. In rare cases, support large file storage. Do not support catalog and user permission.	The storage and processing of vast amounts of unstructured data, and massive images on taobao website.
GridFS	Support dividing a large file into multiple small document files.	GridFS stores file data and file metadata in MongoDB. Copy files to cope with failover, and data integration. And can also be used to read extension, hot backup or be used as data sources of offline batch processing.	It is based on the structural pattern of object storage, reducing the access overhead furtherly during the runtime. GridFS is designed to regulate access mechanism, to adapt to faster application performance put forward by I/O mode. Can provide the fastest I/O performance that application cluster needs. GridFS ensures the load balance in each storage devices, rather than filling a single node.	The speed of reading a file from the GridFS is slower than reading from the file system directly. If the file is large and is stored as multiple file, can't lock all the file blocks, when modifying this large file. When changing the documents stored in the GridFS, it can only remove the old documents first and then re-save the documents.	Suitable for large files that seldom need to get changed.

Name	File Segmentation	System Backup	Merit	Demerit	Application scenarios
Lustre	Divide data into a fixed number of objects. Each object contains several data blocks. When one data block written to the object exceeds its capacity, next writing will be stored in next object. Lustre can distribute file into 160 objects at most to store in.	Lustre provides two backup tools. One is used to scan file system, and another one is used to package backup and pressure recovery.	Can provide the ability of data sharing and parallel processing. The scalability is very strong. Can provide failover technology for metadata and target data under lustre management, achieving access with high reliability. The distributed management mechanism can achieve concurrency control. Provide access in multiple networking protocols.	It is difficult to implement data mirroring. The failover between nodes relies on third-party heartbeat technology. There are only two metadata management nodes. If the system size has achieved certain scale, the management node will reach overload. Lustre kernel can only be deployed on Linux, with some limitations.	Support massive large file storage of PB level, and is suitable for large computer cluster or supercomputer.
Ceph	Adopt RAIDO pattern to across multiple hard disks. Disperse continuous data on multiple disks to access, to adapt to the load balancing.	Support data replication. There are multiple metadata servers.	Store data and metadata separately. Manage matadata using dynamic distribution. Has the reliable automatic distributed object storage.	Technology is not mature, may not be applied to the production environment.	Support massive large file storage of PB level.
FastDFS	FastDFS does not store files in blocks. Files uploaded by clients are corresponding to the files stored on the server.	FastDFS adopts the storage mode of grouping. The storage servers within the same group backup each other.	FastDFS server has only two characters, tracker and storage nodes. So it has the feature of lightweight. FastDFS adopts the storage mode of grouping, flexible and strongly controlled. In FastDFS, each node is primary node, with peer-to-peer structure. Can change the number of trackers at any moment, according to the pressure of the server.	FastDFS does not store files in blocks. So it is not suitable for distributed computing scenario. Storage capacity is limited by a single storage server.	Is suitable for the high traffic service, with file as the carrier, such as photo album website, video website, etc. And vast amounts of small files, 4K~500M.

2.2. NOSQL Database

With the rapid growth of data size of enterprise users and enhancement of user demand for service level, the traditional relational database has some limitations. Traditional database use flat file based on structured record to store all application data, leading to mismatching between application and database. This happens when application is coded in a declarative language. Its structure is completely different with these databases [21]. At past, in order to improve the performance of the system, the components and resources are extended vertically. However, since the storage and application are no longer separate, each expansion of the resources will be service disruption and applications reset.

Most data we create is heterogeneous data. The existence of a large number of structured and unstructured data makes it difficult to determine the perfect and unified relational data model in advance, and the horizontal expansion ability of relational database is bad [22]. Most relational databases do not support large-scale distributed storage. At the same time, it is hard to meet the real-time requirement of high concurrency and large amount of data. So the underlying storage technology should not only be flexible to allow the data to be stored in its natural form, but also meet the demand of the frontier.

Compared with relational database, NoSQL database storage system supports the storage and dynamic management of mass data. It avoids the unnecessary complexity, has high throughput, and can handle horizontal scaling well. And high fault-tolerant can store structured, semi-structured, and unstructured data to avoid the object-relational mapping. The design idea of NoSQL database is to extract the indexing mechanism of relational database, combine distributed storage strategy, and delete those needless on some problems in the SQL system. Therefore it achieved relative good efficiency, expansibility and flexibility [23].

Nosql database is mainly divided into: key-value storage, column-based storage, document storage, the graphics storage.

Key-value database is designed to support simple query operations, leaving complex operations to application layer. Data set will map the key to one or a set of values. That is, the key is the only keyword to find each data address, which also means it is indispensable. The value is the content that data actually store. Key-value storage provides a hash table with key-value pairs on remote servers of a distributed cluster, to implement the mapping from key to value. The hash value based on the key locates the address of data directly, achieving rapid high concurrency query, and also supports the operation of mass data. Key-value storage are divided into key-value type, key-document type and key-column type [24]. Key-column type is the typical expansion of key-value pairs of key-value type. Because of its simpleness and flexible extensibility, it is also the mainstream of data model.

Generalized column-based storage replace columns with column family. The idea of relational database is to store all tables with a line on the disk. That is, a list of entries associated with the same specific row id will be stored together [25]. Since banks or financial institutions need to maintain a large number of related records, do not guarantee that all values are always stored in a continuous way. In the database of column data, a whole column of table is stored together, mapped to a key. Because all listed items have indexes, we can only search part of the table. A column can also have nested columns of hierarchical structure, and one of them is super column [26]. This provides simple query and quick access, at the same time avoids unnecessary overhead of looking for the single key of a record.

Document database is another kind of relational database, used to store semi-structured data. There is XML (extensible markup language), JSON (JavaScript object notation), or other similar formats [27]. A document can be seen as a line in the relational database, which contains all relevant information of documents. A group includes multiple documents, and each document can have different patterns and different data storage quantity and types [28]. Storing text messages is special optimization. Due to related data sets are stored intensively, the expense of SQL JOIN operation is saved. Although database is schema-free design, it stores semi-structured records and is hierarchical structure.

Graph database fits traversal and application search best, such as finding related links on LinkedIn, finding friends on Facebook [29], etc. It pays more attention to the relationship between data items rather than the data itself. They highly optimize rapid traversal and use graph algorithm efficiently. For example, the shortest path is first in order to find the relevance between information, etc.

Table 2 analyzes and concludes main storage types of NoSQL database:

Table 2. *Analysis list of four storage models in NoSql database [30-34].*

Database type	merit	demerit	data model	application scenarios	instances
Key-Value Storage	Has very high concurrent reading and writing performance. Data is indexed and segmented according to the key value. Search is rapid, and data model is simple.	Data has no structure, and do not support complex logic data operation.	key-value mapping between key and value	Content cache. Mainly used for the log system.	Dynamo, Redis, Voldemort
column-based storage	Search is rapid, expansibility is good, and save a lot of I/O operation. It is easier for distributed extension.	The function is relatively limited.	Column-based storage, where data in the same column is stored on the same page	Distributed file system	Bigtable, Cassandra, HBase HyperTable
document storage	Don't need to define the data structure in advance. Use document of specific format instead of tuple as the unit of data storage.	The query efficiency is not high, and lack of unified query syntax.	The value points to the structured data.	Web application	CouchDB, MongoDB XML Database ThruDB
graphics storage	Use graph theory and associated algorithm to improve the storage performance, management and operational data.	The function is relatively limited.	Graph structure	Social networking, relationship graph	Neo4j, GraphDB InfoGrdi

2.3. Database All-in-One Machine

In recent years, facing mass data processing and storage, many traditional hardware manufacturers propose the integrated solution---database all-in-one machine, which has become a hotspot. By the product form of all-in-one machine, it simplifies the complexity of deploying and managing the infrastructure of data center, solving the problem of continuous expanding of basic hardware resources at the age of big data, the requirement of all-in-one machine, and the storage cost of mass data. International manufacturers, such as IBM, Oracle, EMC, launch integration products and solutions for big data [35]. Following them, Chinese manufacturers also develop its own database all-in-one machine. For example, database all-in-one machine of Huawei makes use of its hardware architecture advantage of computing, storage and network convergence, and the feature of high throughput and high IOPS, integrates excellent characteristics of intelligent network card, SSD and other hardware, solving the performance bottleneck between computing and storage. XData big data processor of Shuguang separates the data storage unit and processing unit. By building efficient services middleware, polymerize the underlying data storage node adopting shared-nothing structure into a single data processing system image. Langchao clouds big data all-in-one machine covers technical sessions such as data storage, data processing, data presentation, etc. And there is also Yunchuang Storage data cube cloud computing all-in-one machine, Zhongzhiheda big data all-in-one machine, Zhiyitu Hadoop-Based big data all-in-one machine [36].

Database all-in-one machine is generally suitable for data model of complex storage relations. At the same time, computing needs high transactionality and consistency. Generally speaking, the database engine server configuration depends on the concurrency demands, and the database storage nodes server configuration depends on the data size demands [37]. Database all-in-one machine adopt fully distributed big data processing architecture, integrating the hardware and software in a system. With the growth of user data and the expansion of business, it can be improved by extending hardware lengthways, and can also achieve linear scaling by adding nodes breadthwise, guaranteeing the

performance of low latency, high throughput and the continuity of the business [38]. All-in-one machine is a combination of software and hardware, wholly designed for mass data storage processing. And it is made up of a set of integrated servers, storage devices, operating systems, database management system and pre-installed software for data management. It provides big data storage solution, mainly for large data warehouse market. And its high throughput capability facilitates solving I/O bottleneck problem. The user can choose different series of products according to the requirements, customizing on-demand.

However, database all-in-one machine also faces challenges. In the era of big data, the amount of data is increasing shockingly. Therefore if the users need to expand all-in-one machine, they can only add a equipment cabinet, leading to inflexible expansion. And because the all-in-one software is highly integrated, it is hard to be deployed in other environments.

In some industries, demand changes more quickly. So business model will change very quickly with it. Using all-in-one machine will limit the action of enterprises on the contrary. But in some relatively mature and stable application, all-in-one machine embodies the value of simplifying IT.

2.4. New Database Cluster of MPP Architecture

MPP is large-scale parallel processing system, which is a kind of method for system resource extension, mainly of parallel processing. This means that a single computer has multiple network processors [39].

Horizontal expansion is the primary design goals of MPP architecture database. It is linked by multiple SMP servers via internet of fixed nodes, collaborating for common tasks. From users' level, it is a server system, supporting strict data relation model. The biggest characteristic is that each node can only access their own local resources, with no sharing.

Database cluster using MPP architecture can effectively support mass structured data storage of PB level. It is based on the Shared Nothing architecture. By big data processing technology of column storage, coarse-grained indexes, etc., and combining its distributed computing model with high performance, it completes technical support of the storage application of analysis class. Operating environment is mostly low-cost PC, and it has the advantages of high performance and high scalability [40]. It can improve the performance of data processing, improve the data process load, improve the efficiency of mass data processing and reduce the overall cost of processing each TB. Therefore it has been widely used in enterprise data warehouse of new generation and structured data analysis field.

3. The Prospect of Bigdata Storage Technology

Due to the large amount of unstructured and semi-structured data, traditional relational database has been powerless. However, new storage technology, such as the NoSQL database and distributed file system, is superior to traditional storage, no matter in fault tolerance, scalability and mobility of data. And it is suitable for persistent data storage and mass data storage management [41]. But for the real-time performance of data processing, there is a certain gap between the new storage technology and relational database. So each has its good side. At present, the combination of relational database and distributed parallel processing system can improve storage efficiency, processing speed and analysis speed [42]. This approach is also the hot trend in the future. The core problem of big data storage technology is performance. A single technique and platform can no longer meet the demand of data explosive growth and the requirement of data analysis and storage from operators. In subsequent development, the new-type database will gradually be mixed with Hadoop ecosystem or Spark ecosystem [43], providing SQL and transaction support for application. Use Hadoop or Spark to achieve semi-structured, unstructured data processing. So in the future, storage will also be developed towards the combination of MPP parallel database cluster and Hadoop/Spark cluster. In addition, with the explosive growth of enterprise data, big data all-in-one machine will certainly become a hot technology, and be widely used.

By researching new data storage technology, this paper minutely summarizes and contrasts distributed file system, NoSQL database, database all-in-one machine and new-type database cluster of MPP architecture from different angles. And the future research tendency was put forward. Big data storage is still in the stage of rapid development. The development space is very big, and needs researchers to explore constantly.

References

[1] Zhang X, Xu F. Survey of Research on Big Data Storage [C] // International Symposium on Distributed Computing and Applications To Business, Engineering & Science. IEEE Computer Society, 2013: 76-80.

[2] Biesdorf S, Court D, Willmott P. Big data: What's your plan? [J]. Mckinsey Quarterly, 2013 (2): 40-51.

[3] TU Xinli, Liu Bo, Lin Weiwei. Survey of Big Data [J]. Application Research of Computers, 2014, 31 (6): 1612-1616.

[4] Garcia H, Ludu A. The Google file system [C] // Acm Sigops Operating Systems Review. ACM, 2003: 29-43.

[5] Tong Ming. Research and application of distributed storage based on HDFS [D]. Huazhong University of Science and Technology, 2012.

[6] Davies A, Orsaria A. Scale out with GlusterFS [J]. Linux Journal, 2013, 2013 (235): 1.

[7] Hows D, Membrey P, Plugge E, et al. GridFS [M]. Apress, 2013.

[8] Zhao Yang. Depth Profiles of TaoBao TFS [J]. Digital Users, 2013 (3).

[9] Wang Bo, Li Xianguo, Zhang Xiao. Research on performance optimization of Lustre file system [J]. Microcomputer Applications, 2011, 27 (5): 31-33.

[10] Yu Qing. Analyses of Distributed File System FastDFS Architecture [J]. Programmer, 2010 (11): 63-65.

[11] Golov N, Rönnbäck L. Big Data Normalization for Massively Parallel Processing Databases [C] // International Workshop on Modeling & Management of Big Data. 2015.

[12] Thereska E, Gunawardena D S, Scott J W, et al. Distributed File System: US, US20120254116 [P]. 2012.

[13] Li Hongqi, Zhu Liping, Sun Guoyu, et al. Design and Implementation of Distributed Storage System Facing Vast Small Files [J]. Computer Engineering and Design, 2016 (1): 86-92.

[14] Qi Ying. Research on Low Latency Access Technology of Distributed File System with Vast Small Files [D]. University of Chinese Academy of Sciences, 2013.

[15] Weil S A, Brandt S A, Miller E L, et al. Ceph: A Scalable, High-Performance Distributed File System [C] // 7th Symposium on Operating Systems Design and Implementation (OSDI '06), November 6-8, Seattle, WA, USA. 2006: 307--320.

[16] Gpfs B. A Shared-Disk File System for Large Computing Clusters [C] // of the First Conference on File and Storage Technologies. 2010.

[17] Xu Chunling, Zhang Guangquan. Comparison and Analysis of Distributed File System Hadoop HDFS and Traditional File System Linux FS [J]. Journal of Soochow University (Engineering Science Edition), 2010, 30 (4): 5-9.

[18] Xiong Wen, Yu Zhibin, Xu Chengzhong. Feature Analysis and Performance Comparison of Several Common Distributed File System [J]. Journal of Integration Technology, 2012, 1 (4): 58-63.

[19] Sawicki A, Nowak T. NETWORK DISTRIBUTED FILE SYSTEM:, US20080320097[P]. 2008.

[20] Shi Xiaodong. Research on High Availability of Distributed File System [D]. Institute of Computing Technology, Chinese Academy of Sciences, 2002.

[21] Qin Xiongpai, Wang Huiju, Du Xiaoyong, et al. big data analytics --Competition and Coexistence of RDBMS and MapReduce [J]. Journal of Software, 2012, 23 (1): 32-45.

[22] Shen Derong, Yu Ge, Wang Xite, et al. Survey of Research on NoSQL System Supporting Big Data Management [J]. Journal of Software, 2013 (8): 1786-1803.

[23] Curé O, Kerdjoudj F, Faye D, et al. On The Potential Integration of an Ontology-Based Data Access Approach in NoSQL Stores [J]. International Journal of Distributed Systems & Technologies, 2012, 4 (3): 166-173.

[24] Wang Jieping, Li Haibo, Song Jie, et al. Research on Cloud Data Storage and Management Standardization [J]. Information Technology and Standardization, 2011 (9): 28-31.

[25] Liu Y, Zhu L, Jiang W. Column caching mechanism for column based database:, EP2743839 [P]. 2014.

[26] Bhogal J, Choksi I. Handling Big Data Using NoSQL [C]// IEEE International Conference on Advanced Information NETWORKING and Applications Workshops. IEEE, 2015: 393-398.

[27] Amirian P, Basiri A, Winstanley A. Efficient Online Sharing of Geospatial Big Data Using NoSQL XML Databases [C] // Fourth International Conference on Computing for Geospatial Research and Application. IEEE, 2013: 152-152.

[28] Deka G C. A Survey of Cloud Database Systems [J]. It Professional, 2014, 16 (2): 50-57.

[29] Castelltort A, Laurent A. Fuzzy Historical Graph Pattern Matching A NoSQL Graph Database Approach for Fraud Ring Resolution [M] // Artificial Intelligence Applications and Innovations. Springer International Publishing, 2015.

[30] Dong-Hai L U, Xian-Bo H E. The Analysis of NoSQL Database [J]. Science & Technology of West China, 2011.

[31] Srivastava P P, Goyal S, Kumar A. Analysis of various NoSql database [C] // International Conference on Green Computing and Internet of Things. IEEE, 2015: 539-544.

[32] Chandra D G. BASE analysis of NoSQL database [J]. Future Generation Computer Systems, 2015, 52: 13–21.

[33] Gu Y, Wang X, Shen S, et al. Analysis of data replication mechanism in NoSQL database MongoDB [C] // IEEE International Conference on Consumer Electronics - Taiwan. IEEE, 2015.

[34] Han J, Haihong E, Le G, et al. Survey on NoSQL database [C] // Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on. IEEE, 2011: 363-366.

[35] Hinshaw F D, Meyers D L, Zane B M. Programmable streaming data processor for database appliance having multiple processing unit groups: US, US7577667[P]. 2009.

[36] Zhang Dong, Qi Kaiyuan, Wu Nan, et al. Architecture and Key Technology of Yunhai Big Data All-in-one Machine [J]. Computer Research and Development, 2016, 53 (2): 374-389.

[37] Yue Junfeng, Zhao Junfeng, Zhao Wei, et al. Analysis of Database All-in-one Machine Technical Architecture [J]. Electric Power Information and Communication Technology, 2013, 11 (4): 60-64.

[38] Pu Siyu, Bai qionghua. A Kind of New-type Cloud Storage All-in-one Machine Backing Up Data Information Automatically:, CN204305087U[P]. 2015.

[39] Li C, Yang J, Han J, et al. The Distributed Storage System Based on MPP for Mass Data [C] // Proceedings of the 2012 IEEE Asia-Pacific Services Computing Conference). IEEE Computer Society, 2012: 384-387.

[40] Chen Z, Song L. A solution based on the MPP'S to storage mass data [C] // Mechatronic Science, Electric Engineering and Computer (MEC), 2011 International Conference on. IEEE, 2011: 868-871.

[41] Cheng Lianjuan. Application Practice and Beneficial Reference of Big Data Promotion in the U. S.: An Analysis from the Perspective of Library [J]. Information & Documentation Services, 2013, 34 (5): 110-112.

[42] Li G. Research Status and Scientific Thinking of Big Data [J]. Bulletin of Chinese Academy of Sciences, 2012.

[43] Chen Jirong, Yue Jiajin. Survey of Big Data Solution Based on Hadoop Ecosystem [J]. Computer Engineering and Science, 2013, 35 (10): 25-35.