An Assessment of Large Data Storage on MongoDB

Fan LinXiu

College of Mathematics and Computer Science, Gannan Normal University,
Ganzhou, China
fanlinxiu@126.com

***Abstract* — Mongo DB is a popular non-relational database due to its support of distributed storage for big data applications. However, due to the limitations of Mongo DB self-slicing technology, the data distribution in each node is not uniform, which will cause late data relocation and costly in time and resources. Load balancing technology based on consistent hash algorithm can make Mongo DB distribute data evenly at different nodes, and ensure the normal operation of the system. Our experimental comparisons show the load balancing technology based on consistent hash algorithm is superior to Mongo DB's own partitioning technology. When inserting new data to the database, the data will exist but its partition is uncertain. If it appear in the load partition and the partition has little space, this will cause the uneven distribution of data. Due to the limitations of Mongo DB itself, the latter will inevitably lead to a large number of data migration, resulting in memory problems.**

Keywords - MongoDB database; large data storage method; automatic slicing technology; load balance; research and Application

I. INTRODUCTION

For half a century, with the full integration of computer technology into social life, information explosion has accumulated to a degree of initiation of change. This not only fills the world with more information than ever before, but it also speeds up. In 2006, personal user data has just moved into the TB era, the world produced a total of about 180EB of new data; in 2011, this figure reached 1.8ZB. And market research institutions predict: by 2020, the world's total data will grow 44 times, reaching 35.2ZB (1ZB=10 billion TB)! So in this data volume soared era, big data concept came into being, and this concept is almost applied to all human intelligence and development in the field. Taobao website every day more than tens of millions of transactions, single day data generated more than 50TB (1TB equals 1000GB), storage volume 40PB (1PB = 1000TB). Baidu Inc currently nearly 6 billion of the total amount of data, the number of pages stored close to 1 trillion pages, about every day to deal with the search request for a few times, dozens of PB data 1000PB. The emergence of a large number of new data sources leads to the explosive growth of unstructured and semi-structured data. Unit of information data by TB-PB-EB-ZB level. These information created by us behind the data already far beyond the current human can handle the scope of it[1]. How to manage and use these data gradually become a new field, so the concept of big data came into being. Today, big data (big data) more people are mentioned, people use it to refer to the era of big data generated massive data. Data is growing rapidly, using efficient, stable storage and management technology can not only store large amounts of data, but also to yield efficient data analysis. Data storage is one of the decisive factors in the future development of the enterprise, although the enterprise may not realize growth data explosive issue, but as time goes on, people will be more aware of the importance of enterprise data. Big data era of human data management capabilities put forward new challenges, but also for people to get a deeper, comprehensive data analysis provides opportunities never had. Is unable to meet the needs of users in relational database and under the condition of No SQL (Not Only SQL) database, Mongo DB No SQL in the database may be the most famous, most similar to relational database, so quickly gained popularity. The face of massive data generation and storage, traditional storage technology has many shortcomings. High concurrency access - the rapid development of the information age brings rapid increase in the amount of visits, the same time the site visits are quite large. The traditional database server can support a large number of users at the same time the access links, the database is stable, low efficiency and even the collapse of many problems such as scalability; storage to increase hardware equipment capacity, cannot meet the endless data demand, pay a huge hardware cost, cannot bring the corresponding performance improvement. The data will become the choice of horizontal expansion, data organization dispersed in the computer group or a group of servers, data processing, using multiple computers to work together, it solves the problem of data storage, improve performance; access efficiency -- from below that relational database query is according to the data page traversal query, to spread. Overflow problem. Moreover, the complex data structure is saved in cascade mode, occupying a lot of data space, and using the cascade query data, the query speed is very slow[2]. Mongo DB the key to data model, model free, fast inquiry. In recent years, distributed storage technology has developed greatly. With the expansion of the scale of enterprises, tables, forms, historical data and many other information to be saved in the computer. The enterprise adopts the distributed enterprise data storage server has become increasingly common, each branch enterprises distributed in each city home, each branch also needs in the work arrangement,

application process and other aspects are consistent, the best data stored in one place, not in each branch has a copy of the data, such data is not easy to cause consistent security. Therefore, the enterprise will be located in a specific location of several servers for specialized storage enterprise generated data, to better service enterprise management. Due to the large amount of data, high concurrency, storage and scalability and usability requirements, the traditional relational structure database cannot meet the needs of such high-end. Therefore, Mongo DB provides a distributed storage model for the document for the majority of users, and soon by the users, the rapid popularization, but also provides the actual demand for the research of mass data storage technology. Produce huge amounts of data makes the traditional storage model has been unable to meet the growing demand for data, using document storage mode of cluster database has Mongo DB, the distributed storage not only solves the problem of data storage, query efficiency, data type flexible, easy to expand etc. also has a good effect. So Mongo DB in China in recent years has been rapid development. Data increase, the previous single storage mode cannot meet people's desire for high quality, high security data. Faced with large data storage, people use a lot of new technologies with distributed storage theory to meet the needs of massive data processing. First of all, the relational data model to Key_value model, method of data file associated with the storage structure, and using the method of distributed storage hardware technology and memory optimization that combines the underlying technology to meet the big data era of external equipment of high capacity and expand the memory requirements for. Mongo DB's own partition technique using range partitioning, which according to the key size, the key data partition a range to a piece, a range of key data stored in the next two, and so on. But this method, for the situation is not balanced, fragmentation effect is poor, simply cannot meet the daily needs of the chip load balancing requirements[3]. Therefore, I apply the consistency hash load balancing algorithm to Mongo DB, try to use each server as hash node to achieve load distribution equalization. From the following experiments can be seen, in fact, this method of fragmentation effect is excellent. Large data storage technology schematic shown in Figure 1, distributed storage technology schematic shown in Figure 2. Through the example of Mongo, DB, document collection and slice storage structure; BSON file format, using the file encoding a specific binary conversion of data files stored in the database for data storage; large use of large data file Grid FS file storage system, put into multiple modules separately stored, so can effectively improve the efficiency of memory usage, reduce memory fragmentation and technology application analysis. P2P distributed system is a distributed system of common and typical, through the application of distributed P2P system, especially based on DHT (Distributed Hash Table, distributed hash table) analysis and application of distributed storage technology, can help us to further study the automatic segmentation algorithm. In the process of using Mongo DB, many people will find that Mongo DB for memory requirements are too high, unless there is a very large hard disk (if any, don't forget the memory optimization, otherwise) must be a good memory management can. Even if it will still be a lot of memory problems, or even downtime. Mongo DB is a memory database, the operation of the data are done in memory, therefore, we should avoid unnecessary data migration. When new data is generated, the data will be in which the data partition is also uncertain, if appear in the load partition, and the partition is not little load data, this will cause the uneven distribution of data, because the Mongo DB slice mechanism, later will inevitably lead to a large amount of data transfer, memory problems caused by growth[4].

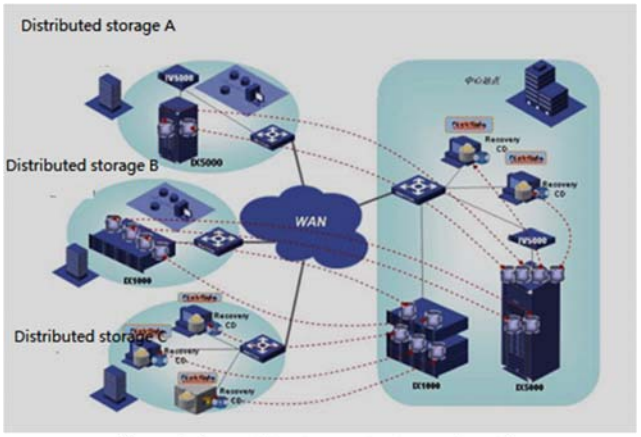

Figure 1. Large data storage structure schematic

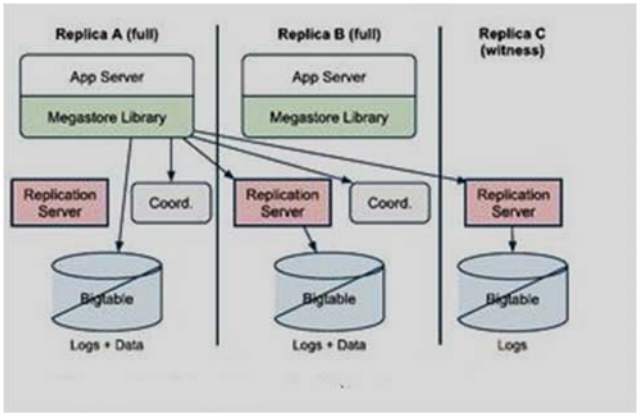

Figure 2. Schematic diagram of distributed storage structure

Index in the database status is not shaken, according to different amount of data, it uses special data structure can shorten the query time to ten times to several hundred times. Choose a simple data type as a key index key to save about 5% of the memory space as the ratio of complex data types, which is according to the size of the database to the results, if the database is large, it saves the memory space is certainly more than that. Use Mongo DB internal function can simplify query complexity, but the error will also delay the use of internal functions query speed. Skip function is a frequently used internal function, using it can be paged query. In this article, it is efficient when there is no index, but the effect is not satisfactory when there is an index. Therefore, we should be aware of the frequently used functions,

understand the query principle, to help us better operate the database.

II. KEY TECHNOLOGIES INVOLVED IN THE SYSTEM

A. *MongoDB database*

MongoDB is a database based on distributed file storage. Written by C++ language. Designed to provide scalable high-performance data storage solutions for WEB applications. MongoDB is a product between the relational database and the non-relational database, is the most abundant in the non-relational database, most like the relational database. He supported the data structure is very loose, similar to the JSON bson format, so you can store more complex data types. Mongo is the biggest characteristic of his support of the query language is very powerful, its syntax is similar to object-oriented query language, almost like a relational database can be achieved most of the functions of a single table query, but also support for data indexing. The so-called "set oriented" (Collection-Oriented), meaning that data is stored in a data set grouped, known as a collection (Collection). Each collection has a unique identification name in the database and can contain an infinite number of documents[5]. The concept of a collection is similar to the table (table) in a relational database (RDBMS), unlike the fact that it does not need to define any schema (schema). Nytro MegaRAID technology in the flash memory cache algorithm, can quickly identify large data sets within the database data, providing consistent performance improvements. Schema free (schema-free) means that for a file stored in a mongodb database, we do not need to know any of its structural definitions. If necessary, you can store files of different structures in the same database. Document stored in a collection is stored as a form of key value pairs. Keys are used to uniquely identify a document as a string type, while a value can be a variety of complex file types. We call this storage form BSON (Binary Serialized Document Format). Distributed file system (Distributed File System) refers to the file system management of physical storage resources are not necessarily directly connected to the local node, but through the computer network connected with the node. Distributed file system design based on client / server model. A typical network may include multiple servers for multi-user access. In addition, peer-to-peer features allow some systems to act as dual roles for clients and servers. HBase is a distributed, column oriented open source database, which comes from the Google thesis written by Fay Chang: Bigtable: a structured data distributed storage system. Yonghong Data Mart is based on its own technology developed a data storage, data processing software. Yonghong Data Mart distributed file storage system (ZDFS) is based on the Hadoop HDFS transformation and expansion, the server cluster all nodes in the file storage management and storage. MongoDB server can run on Linux, Windows or Mac X platform to support 32 bit and 64 bit applications, the default port is 27017 os. Recommended to run on the 64 bit platform, because the maximum file size MongoDB supports in the 32 bit mode runtime is 2GB. A collection is a set of documents, similar to tables in a relational database. Collections are modeless and documents in a collection can be various. For example, {"Hello, word": "Mike"} and {"foo": 3}, their bond is different, different types of values, but they can be stored in the same set, i.e. the different mode of the document can be placed in the same set. Since collections can store any type of document, why use multiple collections? This is because all documents are placed in the same set, both for developers or administrators, it is difficult to manage the collection, and in this case, the query collection operations are not efficient. So in actual use, often will be stored in a different set of document classification, for example, for Web log records can be stored according to the log level, Info level log in the Info collection, Debug level log in the Debug collection, it is convenient for management, but also provides query performance. But it should be noted that this document is divided to separate storage is not mandatory requirements of MongoDB, users can choose flexibly. You can use "." to partition a collection into subsets by namespace[6]. For example, for a blog system, blog.user and blog.article may include two sub sets, this division just let the organization structure better, blog set and blog.user, blog.article does not have any relationship. Although the subset does not have any special place, but the use of sub sets of clear data structure, which is recommended by the MongoDB method. MongoDB multiple documents in the collection, a collection of databases. A MongoDB instance can load multiple databases. They can be regarded as mutually independent, each database has independent permission control. On disk, different databases are stored in different files. The following system databases exist in MongoDB. MongoDB database structure schematic shown in Figure 3.

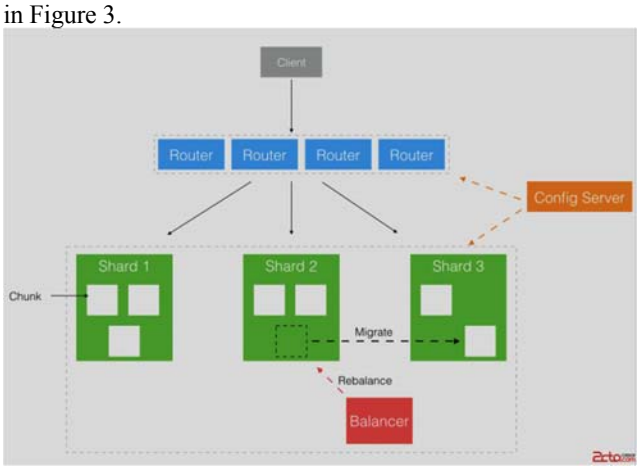

Figure 3. Schematic diagram of MongoDB structure

B. *Automatic slice technology*

Before you decide how to slice, you'd better figure out from the beginning whether you really need a piece. Admittedly, fragmentation is the only way in the case of large scale database requirements. Not just for MySQL, for most of the same technology. However, due to the emergence of many emerging technologies, more and more applications support the need to run a database without fragmentation. Now, we can easily run TB level data on each

MySQL instance, and support tens of thousands of queries in many OLTP environments. Shows that we can build very large applications without slicing. We should remember: all is the last resort approach to all environment. Even if you use the database to support out of the box slice function, it will be due to the introduction of more components and complexity and trouble. Constructing a good distributed query execution plan is a very complex task, which needs to consider the topology and load of the network. Before deciding whether to slice, you should first consider whether there is an alternative to extending your application. In the MySQL world, there are usually some options to consider.If you have an application, with the business is getting better, the amount of data involved in the system is also growing, you want to involve the issue of system scalability (Scale). A typical extension method is called "upward Up (Scale)", which means improving the system performance parameters by using better hardware[7]. The other method is called Out (Scale), which means that by adding additional hardware (such as server) to achieve the same effect. From the "cost" or "limit" point of view, the "outward telescopic" generally better than "upward expansion", so most of the size of the system will be considered "out" to a certain extent. Because many systems are in the bottleneck of data storage, so called "a data slice (Database Sharding) as the data architecture, this paper will discuss a typical implementation of the data structure of the. None of these data partitioning strategies have an absolute advantage over which strategy is based solely on the business or data characteristics of the system. It is worth emphasizing that: data fragmentation is not silver bullet, it brings some benefits to the system performance and scalability (Scalability), but also will bring many complexity to system development. For example, there are two records respectively in different servers, so if there is a business is to create a "relationship" for them, then it is likely that the "related" recorded in the two partition must be placed in the a. In addition, if you value the integrity of the data, the transaction across the data partition immediately becomes a performance killer. Finally, if there are some services that need global search, it is very difficult to have the advantage of data slicing strategy. MongoDB database automatic slicing technology is based on certain rules are divided into several small collection of original database, these small pieces by mongos routing management, when a query or write, routing will be based on the piecewise shard key rules to find corresponding slice. Slicing solves write intensive operations for dispersing single write server loads. Or the original storage space is not enough, this time may be through the slice operation will later data written to other storage space. It can be seen that the partition of the collection is similar to the partition table of the database, and each slice supports the write operation. Due to fragmentation, cause data to be distributed to different storage server, when a server problems could lead to the loss of data, then mongos routing will be a problem, another storage slice information to configure the server may also occur. Of course we can use master salve/Replica sets mechanism to backup each partition, Mongos, configs. The following official website configuration icon, even if we use the server cross backup also requires a lot of server resources, so fragmentation is a very resource consuming things. The shard key analysis is simple, when not writing intensive operations, but simply because the storage space is not enough, we can use some of the shard key without the upper limit of key, such as the creation time, so that the new record will write a new patch on the server. When you need to make every piece of uniform distribution of data, or write intensive, the best selection of a certain range of key values, of course, this range is not too small, like sex, and so, this will lead to two points only automatically, so be sure to choose the appropriate shard key to achieve the ideal effect. MongoDB database automatic slicing technology schematic shown in Figure 4, the realization of the core code listed as follows.

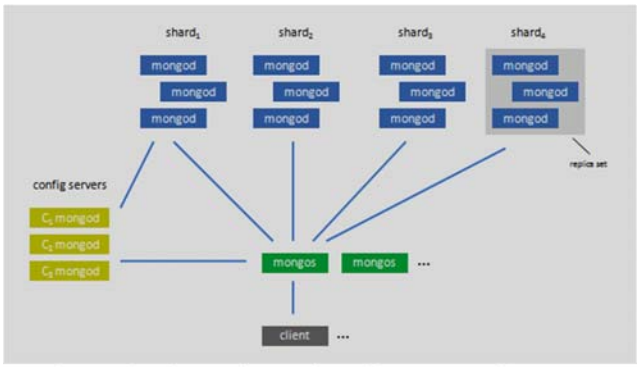

Figure 4. Sketch map of automatic partition structure of MongoDB database

```
class {
private:
 MatchType _matchType;
 NamespaceString _ns;
};
enum MatchType {
 matchNever = 0,            /// Matches no resource.
 matchClusterResource = 1,  /// Matches if the resource is the cluster resource.
 matchDatabaseName = 2,     /// Matches if the resource's database name is _ns.db().
 matchCollectionName = 3,   /// Matches if the resource's collection name is _ns.coll().
 matchExactNamespace = 4,   /// Matches if the resource's namespace name is _ns.
 matchAnyNormalResource = 5, /// Matches all databases and non-system collections.
 matchAnyResource = 6       /// Matches absolutely anything.
};
void Privilege::addPrivilegeToPrivilegeVector(PrivilegeVector* privileges,
                                              const Privilege&amp; privilegeToAdd) {
 for (PrivilegeVector::iterator it = privileges-&gt;begin(); it != privileges-&gt;end(); ++it) {
```

```
        if (it-&gt;getResourcePattern() ==
privilegeToAdd.getResourcePattern()) {
            it-&gt;addActions(privilegeToAdd.getActions());
            return;
        }
    }
    // No privilege exists yet for this resource
    privileges-&gt;push_back(privilegeToAdd);
}
```

C. *Load balancing technology*

Load balancing based on the existing network structure, it provides a cheap and effective method of transparent extension of network equipment and server bandwidth, increase throughput, strengthen network data processing ability, increase network flexibility and availability. Load balancing, English name for Load Balance, it is allocated to perform multiple operations unit, such as Web server, FTP server, application server, key enterprises and other mission critical servers, which work together to complete the task. Supports the establishment of VPN connections on multiple lines simultaneously and loads multiple VPN lines. Not only improve the enterprise headquarters and branches VPN access speed, but also solve the problem caused by a ISP line disconnection inaccessible. When VPN load balancing VPN access data will be simultaneously on multiple VPN lines upward transmission[8]. When a VPN line fails, all traffic will automatically switch to the normal VPN line for transmission. Personal bandwidth management: everyone can achieve network bandwidth allocation, management, can be set to ensure bandwidth to protect personal applications from the overall environmental impact. Daily: bandwidth quotas for individuals, groups or departments were set up bandwidth quotas, so you can reasonable use of bandwidth resources, eliminate the waste of resources, but also put an end to the staff and work related things, such as watching movies online, download large files and so on. Software load balancing solution refers to one or more server operating system installed on the corresponding one or more additional software to achieve load balancing, such as DNS Load Balance, CheckPoint Firewall-1 ConnectControl, its advantage is based on the specific environment, simple configuration, flexible use, low cost, can satisfy the load balancing general requirements. Software solutions are more disadvantages, because each server to install additional software operation consumes system resources is not quantitative, more powerful function, more consumption, so when the connection request when particularly large, the software itself will become a key server success; software scalability is not very well, by operating system restrictions; because the operating system itself Bug, often lead to security problems. Hardware load balancing solution is to install the load balancing device directly on the server and the external network, this equipment is usually called the load balancer, due to special equipment to perform specific tasks, independent of the operating system, the overall performance is much improved, with various load balancing strategy, intelligent traffic management, can achieve the optimal load balancing needs. The load balancer has a variety of forms, in addition to the load balancer as independent sense, some of the load balancer integrated in exchanging equipment, placed between the server and the Internet link, some with two network adapters to integrate this capability to the PC, a connection to the Internet, a connection to the backend server group the internal network. Load balancing from the geographical structure of its application is divided into local load balancing (Local Load Balance) and the global load balancing (Global Load Balance, also called regional load balancing), local load balancing refers to balance the load on the local server group, global load balancing refers to were placed in different geographic locations, different network structure between servers for load balancing. Local load balancing can effectively solve the data flow is too large, the network overload problem, and do not need to spend money to purchase expensive high-performance servers, make full use of the existing equipment, server to avoid single point failure caused by data flow loss. It has flexible and balanced strategy to allocate the data flow to the server within the server group. Even to extend the existing server upgrade, simply add a new server to the service group, without changing the existing network structure, stop existing services. Global load balancing is mainly used to have their own servers in more than one area of the site, in order to make global users with only one IP address or domain name can access to the nearest server, so as to obtain the fastest access, can also be used for subsidiaries scattered widely distributed sites of large companies through Intranet (intranet) to achieve the unification of the rational allocation of resources. Load balancing technology schematic shown in Figure 5. Load balancing technology core code cited as follows.

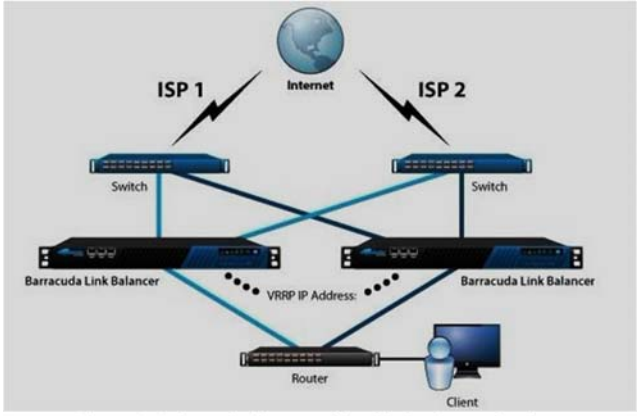

Figure 5. Schematic diagram of load balancing technology

```
public class TestRoundRobin {
    static Map<String,Integer> serverWeigthMap =
new HashMap<String,Integer>();
    static{
        serverWeigthMap.put("192.168.1.12", 1);
        serverWeigthMap.put("192.168.1.13", 1);
        serverWeigthMap.put("192.168.1.14", 2);
        serverWeigthMap.put("192.168.1.15", 2);
```

```
23        serverWeigthMap.put("192.168.1.16", 3);
24        serverWeigthMap.put("192.168.1.17", 3);
25        serverWeigthMap.put("192.168.1.18", 1);
26        serverWeigthMap.put("192.168.1.19", 2);
27    }
28    Integer  pos = 0;
29    public  String roundRobin()
30        {
32                Map<String,Integer> serverMap  = new
HashMap<String,Integer>();
33            serverMap.putAll(serverWeigthMap);
35            Set<String> keySet = serverMap.keySet();
36                        ArrayList<String>  keyList  =  new
ArrayList<String>();
37            keyList.addAll(keySet);
39            String server = null;
41            synchronized (pos) {
42              if(pos >=keySet.size()){
43                  pos = 0;
44              }
45              server = keyList.get(pos);
46              pos ++;
47          }
48          return server;
49      }
```

III. APPLICATION ANALYSIS OF MONGO DB DISTRIBUTED STORAGE TECHNOLOGY

A. DB Mongo Data Type Analysis

Distributed storage time is not long, in the society, many enterprises are still using the traditional relational SQL database, Oracle database, while the database in order to keep up with the pace of the times, in their respective distributed storage method of database are integrated of all kinds, but due to their own conditions, these methods cannot perfectly distributed storage to solve the massive data bring long operation time, small storage space, and many problems such as redundant actions. Mongo DB is a non-relational database, support complex data structure, data type flexible, no matter what kind of data structure that can be easily represented in the database, and the data operation, the use of key value pairs, can quickly and effectively retrieve the desired data. Also, this database has good data scalability and flexibility, Mongo DB is its support for distributed storage technology, it can effectively store data in different node system, and maintain an equal relationship between each node, if there is new data, only need to add a node to the system, and a server system this server will automatically be added to the distributed storage system, as an optional partition follow-up data distribution. Therefore, in the future, the application of distributed storage technology based on Mongo DB will be more and more widely. Mongo DB its own storage systems, the data is in the form of a document stored in the database, and translating documents into a string of binary code using BSON encoding, using Grid FS binary file storage system the binary code is split into a plurality of data stored in the database module. And Mongo DB is to support the distributed system, so the use of distributed storage technology can store these data modules in different nodes, maintaining node data balance, laterally improve storage space. Mongo database belongs to non-relational distributed database, which emphasizes flexible data type and complex structure. In today's massive data generation, combined with distributed storage technology, in theory, Mongo DB can store unlimited amounts of data. When a new data increases, you only need to add a node in the original system equally can, its mechanism of Mongo DB system will bring new nodes as a part of the whole storage system storage device. Therefore, Mongo DB has good scalability. And Mongo DB query efficiency, Mongo DB using non-relational data types, data structure more flexible, query, there is no longer a lot of unnecessary query operations, saving a lot of time, improve query efficiency. The document is the core concept of Mongo DB, the document is the smallest unit of data storage, and the document contains the key with the key, because the Mongo BD uses the BSON format data, the key can be embedded in the document. Computer storage expansion is divided into vertical extension and horizontal expansion, vertical expansion refers to the continuous upgrade of computer hardware equipment, high cost and slow data processing; horizontal expansion, namely slices, refers to the data are distributed to different servers, each server form a distributed topology structure, easy expansion, high data processing speed. Mongo DB uses range partitioning method, according to the film keys, move the data to the server where the value range of tablets [9].

B. P2P Distributed Storage System

P2P non central distributed set, the system is divided into: distributed structured system, distributed unstructured system, semi distributed system. Unstructured system is relatively loose and random, so flood (flooding) method is used to search data. According to the DHT (Distributed Hash structured system Table, distributed hash table) Rules node, and insert data, according to the data object key calculation of the hash value of the query, hash search data according to query key calculation. Gnuntella is the representative of distributed unstructured systems, nodes scattered throughout the distributed system, there is no fixed connection between nodes, and data storage is random, irregular. When adding a new node to the system, the node will only have contact with neighboring nodes, when you want to search for a new node with data, will be the first to find the neighboring nodes, the routing information will be above the new node, finally find a new node data. In flooding mode, each node contains a neighbor table, which contains the routing information of neighboring nodes. Between nodes is to form a huge distributed system with each other through such neighbor tables[10]. The specific route is this: the user sends a query command Ping information, information will be routed to the nearest node, if the node has to find the data node will return PONG information, the routing process is finished. If the node was not found for the data, the node will according to the temporary table itself, to send commands to all nodes,

near if there will return the results, if not then routing down, until the route to find the data[11]. Search instruction has a TTL (time_to_live) domain, TTL is a counter, whenever the next level routing once, TTL will be reduced, when TTL reduced to zero, the route will stop. DHT (Distributed Hash Table) distributed hash table can make the load evenly distributed to all nodes in the system, regardless of whether the system nodes or nodes increase hang, DHT can again load balance in the consumption of system resources is very small under the condition. Each node is a form, but this form of storage node information in the distance is half of the number of nodes node distance, so find the complexity from O (N) is now down to O (N/2). The following describes the specific data distribution methods and query methods. DHT storage schematic diagram shown in Figure 6.

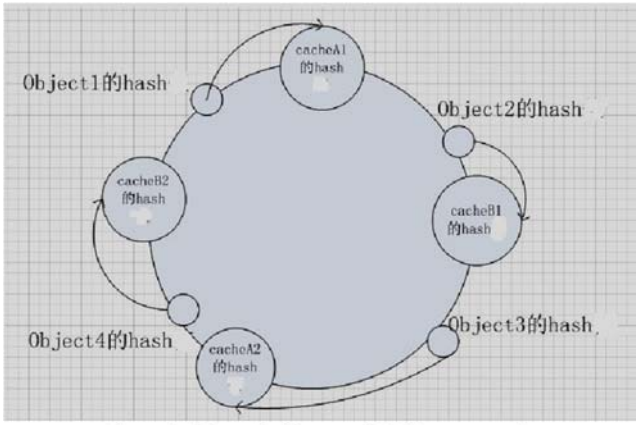

Figure 6. Schematic diagram of DHT storage mode

C. *Automatic slicing technology based on Mongo DB*

Slice Mongo DB technology itself is the scope of the original database partition, set according to certain rules and cut into several pieces, these small tablets by mongos routing management, when a query or write, routing will be based on the piecewise shard key rule to find the slice corresponding to the choice of key distribution as the division standard, which is of great help to the balance of load. Mongo DB partition functions are as follows: configuration server, Mongos routing server, and Mongod client. Each client access through the Mongos routing server area, Mongos needs all the fragments of information is stored in the server configuration, here, the configuration of the server and Mongos routing server can not only serving as a computer, can also be a group of servers as such, but also can improve the safety, and can improve the overall performance of the computer. First of all, the range of Mongo DB the method of partitioning, a chapter of the experiment proved that this kind of slice technology in key complex data type or value distribution, fragmentation effect is very poor. In this chapter, the load balancing technology based on the consistent hash algorithm is adopted for automatic slicing technology, which proves that this method can effectively maintain the load balance of each node, and the results are satisfactory. Load balancing technology based on consistent hash algorithm is a load balancing technique based on hash (Hashi) computing. In fact, in the era of relational database, some people use hash algorithm to load the load evenly on different servers, help database managers save a lot of manual maintenance work. Not only that, but also to ensure that the load balance will not have a lot of meaningless data migration, which will reduce the risk of system collapse. Then, the "appropriate" data type is used to reduce the memory loss of the index, which effectively alleviates the severe state of memory usage tension. Indexes are created in almost all database operations. Because of the big data era, almost all of the enterprises in the amount of data are very large, if you want to find, or operating a record, do not use the index case, a query with a simple record of time or even a few hours. And if you create an index, it may take only a few minutes to find the desired record. The index is like a book to us, and the catalog can quickly locate what we need. Finally, analysis of the Mongo DB comes with some flip function, Skip function, and prove that the improper use will cause the query efficiency is low, therefore, suggestions according to the specific use of the environment to select the correct query. After the previous section of Mongo DB own chip technology, it cannot meet the load balancing requirements. Here I will collect data through experiments to prove consistency hash load balancing can help us improve this shortcoming. The Hash algorithm can be transformed into the input signal for infinite binary strings of fixed format, whether you enter the letters and numbers, Chinese characters or other special format data types, according to the hash formula can be effectively transformed into a binary string unified, late in the process, will keep the signal consistency. The thought and consistency of hash algorithm DHT distributed hash table has great relevance[12], but the consistency of the hash distribution does not need to use broadband and saving resources, that is not the same as that of DHT each node to maintain only a few nodes of the current node, and then use the hash table to find the range of node of the next increment. But the node mapping all throughout the 1 - 2^{32} numerical circle, when the query, according to the hash value directly find the corresponding node.

```
db.TestcollectionServer1.find()
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bb97"), "Name" : "zhanglulu1" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bb98"), "Name" : "zhanglulu3" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bb99"), "Name" : "zhanglulu4" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bb9a"), "Name" : "zhanglulu5" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bb9b"), "Name" : "zhanglulu6" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bb9c"), "Name" : "zhanglulu7" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bb9d"), "Name" : "zhanglulu10" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bb9e"), "Name" : "zhanglulu11" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bb9f"), "Name" : "zhanglulu12" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bba0"), "Name" : "zhanglulu14" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bba1"), "Name" : "zhanglulu15" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bba2"), "Name" : "zhanglulu18" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bba3"), "Name" : "zhanglulu19" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bba4"), "Name" : "zhanglulu20" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bba5"), "Name" : "zhanglulu23" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bba6"), "Name" : "zhanglulu27" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bba7"), "Name" : "zhanglulu28" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bba8"), "Name" : "zhanglulu30" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bba9"), "Name" : "zhanglulu31" }
{ "_id" : ObjectId("55137eb8e4f096ef6fd8bbaa"), "Name" : "zhanglulu37" }
```

Figure 7. Schematic diagram of the data storage of the partition 1

```
public class TestWeightRandom {
18      static Map<String,Integer> serverWeigthMap =
new HashMap<String,Integer>();
20      static{
21          serverWeigthMap.put("192.168.1.12", 1);
22          serverWeigthMap.put("192.168.1.13", 1);
```

```
23        serverWeigthMap.put("192.168.1.14", 2);
24        serverWeigthMap.put("192.168.1.15", 2);
25        serverWeigthMap.put("192.168.1.16", 3);
26        serverWeigthMap.put("192.168.1.17", 3);
27        serverWeigthMap.put("192.168.1.18", 1);
28        serverWeigthMap.put("192.168.1.19", 2);
29    }
31    public static String weightRandom()
32    {
34                Map<String,Integer> serverMap = new
HashMap<String,Integer>();
35        serverMap.putAll(serverWeigthMap);
37        Set<String> keySet = serverMap.keySet();
38        Iterator<String> it = keySet.iterator();
40                        List<String> serverList = new
ArrayList<String>();
42        while (it.hasNext()) {
43            String server = it.next();
44            Integer weight = serverMap.get(server);
45            for (int i = 0; i < weight; i++) {
46                serverList.add(server);
47            }
48        }
49        Random random = new Random();
50                                int randomPos =
random.nextInt(serverList.size());
52        String server = serverList.get(randomPos);
53        return server;
54    }
```

IV. SIMPLE DATA TYPE INDEXES REDUCE MEMORY USAGE AND QUERY OPTIMIZATION

Indexing is the most common means of data manipulation in the database. Indexing can transform complex, time-consuming operations into simple, fast operations. The index data structure of the core technology of the index, what kind of data structures the index determines the application of the index, and the efficiency of the index. Although in the traditional SQL, Oracle and other relational databases are detailed, but in Mongo DB and other large non-relational database is also a great potential for use. This paper makes a detailed study on how Mongo DB can use index to reduce database memory loss[13]. Whether it is in the traditional database system or in today's popular distributed storage system, it is vital to quickly query the data you want. Whether your data storage space used again superior, if in the late data operation, unable to meet the user's rapid access to information needs, this data technology is also difficult to be accepted. Create a database in addition to saving data, another important role is to support business needs, we can query the specified data. Query optimization is an accompanying database development problem, numerous technicians from the code, data structure, index creation and other angles to improve the query time efficiency. Some sacrifice space costs, some sacrifice code complexity. In the hundreds of billions of data, we may need only a few hundred, or dozens of. When we know the approximate range of data, we can direct the query to the approximate range, and then query, so that the speed will be much faster. In Mongo DB, the Skip function can be paging query, Skip allows us to improve the query efficiency of nearly 10 times, of course, this is in the experimental case, if in other environment, or even reduce the query efficiency. Surprisingly, this query does not use 1ms, the results are surprising, just use the index to the original 2S reduced to less than 1ms, but it is not a little help to the skip query. Therefore, for the use of Skip, in the absence of index, it does save time than the general way, but if there is an index, then skip will be much slower than the index search method. Index and optimization query core code listed below, SEQ key index after the index state diagram as shown in Figure 8. This section is the core of the paper, the most important is the use of a consistent hash algorithm, in terms of load balancing experiments proved that the effect is very satisfactory, the best case is the data distribution ratio close to 1:1. Secondly, database indexing is a frequently used function, choose a simple data type as the key, but also can meet the experiment with 5% memory reduction requirements, of course, this is just a simple experiment in practical application process, the effect will be better. Finally, the operation of the database query is the most widely used, a measure of flexibility in the database, the database operation, we use the function similar to Skip, we must fully consider the use of the environment, conditions of use, and otherwise it will not bring much benefits [14].

```
mongos> db.stats()
{
        "raw" : {
                "127.0.0.1:5555" : {
                        "db" : "mytest",
                        "collections" : 3,
                        "objects" : 100005,
                        "avgObjSize" : 56.0004399780011,
                        "dataSize" : 5600324,
                        "storageSize" : 11268096,
                        "numExtents" : 9,
                        "indexes" : 2,
                        "indexSize" : 5780432,
                        "fileSize" : 201326592,
                        "nsSizeMB" : 16,
                        "dataFileVersion" : {
                                "major" : 4,
                                "minor" : 5
                        },
                        "ok" : 1
                }
```

Figure 8. Schematic diagram of index state after SEQ key creates index

```
mongos>                                db.Testcollection
Server2.find().skip(1999999).limit(20).explain()
{
"cursor" : "Basic Cursor",
"is Multi Key" : false,
"n" : 20,
"nscanned Objects" : 20,
"nscanned" : 2000019,
"nscanned Objects All Plans" : 20,
"nscanned All Plans" : 2000019,
"scan And Order" : false,
"index Only" : false,
```

```
"n Yields" : 2,
"n Chunk Skips" : 0,
"millis" : 291,
"index Bounds" : {},
"server" : "PC201411280943:5555",
"millis" : 291
}
mongos> db.Testcollection Server2.find({"name" : {"$gte" : "zhanglulu2000000", "$
lte" : "zhanglulu2000020"}}).explain()
{
"cursor" : "Basic Cursor",
"is Multi Key" : false,
"n" : 23,
"nscanned Objects" : 3708660,
"nscanned" : 3708660,
"nscanned Objects All Plans" : 3708660,
"nscanned All Plans" : 3708660,
"scan And Order" : false,
"index Only" : false,
"n Yields" : 2,
"n Chunk Skips" : 0,
"millis" : 2068,
"index Bounds" : {
},
"server" : "PC201411280943:5555",
"millis" : 2068
}
```

V. CONCLUSIONS

Now is the era of big data, a variety of distributed storage technology frequently appeared, the traditional centralized storage technology has been unable to meet people's data storage capacity requirements. Mongo DB is a popular database in recent years, it is completely different from the traditional relational database: different storage methods, different file systems, different data structures and organizational forms. Mongo DB to meet the urgent needs of the people on the big data storage, through the effective management of the process of distribution in different regions of the computer (Mongos) for unified management, as long as the client can run, can access to the database. Studied the relational data model and BSON data model, analyzed how the file BSON binary code generated, understand the nature of storage. Introduced the concept of data pages and data files, as well as the query process, help us understand the query principle, involving query efficiency, can help us understand. Finally, we introduce the concept of document, collection and partition in Mongo DB, which can help us understand the difference between non-relational database and relational database, and do theoretical basis for the following. Big data storage technology will be combined with cloud computing, both in terms of data scalability or data processing speed will be unprecedented strong. Moreover, the development of the Internet of things, will vigorously promote the rapid integration of these two technologies. Big data not only refers to the big data storage, large data computing, etc., so the future of big data is likely as a separate discipline more comprehensive into people's lives and learning. Big data storage security is a concern, people put a lot of data stored in a distributed system, if the data leakage occurs, and data is dishonest people, so it will cause huge loss. Therefore, the future of large data storage technology will certainly be a big effort in data storage security. For traditional data cheating means, in the storage of large data, should pay special attention to. And to improve the rights and data transmission means to ensure data security.

REFERENCES

[1] Zhang Lin, Tan Jun, white. MZ MongoDB based proteomics data storage system design based on computer application, 2016, 36 (S1): 232-236.

[2] Shi Yuliang, Wang Xiangwei, Liang Bo, et al. The communication platform of MongoDB data storage mechanism. Based on the grid technology, 2015, 39 (11): 3176-3181.

[3] Zhang Yanxia, Feng living, Hao Wei, et al. Study on NoSQL file type data storage. Based on the technology of manufacturing automation, 2014 (6): 27-30.

[4] Ceng Qiang, Miao Li, Qin Zheng. Big data processing Hadoop and MongoDB integration technology oriented research. Computer applications and software, 2016 (2): 21-24.

[5] He Jianying. Research on MongoDB database archive storage to big data. The modern electronic technology, 2015, 38 (16): 51-55.

[6] Hu Xiaodong, Zhang Xin, Qu Jingsheng. Remote sensing resource storage and management method of big data architecture. Journal of Earth Science, information science, 2016, 18 (5): 681-689.

[7] Hu Yinglong, Chen Jie. Application of NoSQL spatial data management in provincial water resources data sharing service platform. Bulletin of Surveying and mapping, 2015 (12): 88-92.

[8] Li Zhaokui, Yan Wenying, Yang Wu, et al. 3D city model data partition and distributed storage method. Journal of Earth Science, information science, 2015, 17 (12): 1442-1449.

[9] Ma Lei, Yang Hongxue, Liu Jianping. Study on the storage method of user privacy data under the big data environment. Computer simulation, 2016, 33 (2): 465-468.

[10] Liu Daoxin, Zhang Jian, Hu navigation, et al. Study on key issues and application of big data in the whole life cycle. Chinese ofthecsee, 2015, 35 (1): 23-28.

[11] Shen Xueyi, Mai Xiaoqin, and. Application of large data collection based on Internet platform in social cognitive research. Science Bulletin, 2015 (11): 986-993.

[12] Anderson, Liang Zhihao, Xu Shou Ren. Research on the security of smart city construction based on big data. Journal of Chinese Institute of Electronic Science, 2016, 11 (3): 229-232.

[13] Zhu Jiansheng, Wang Jianxiong, Zhang Junfeng. Research and application of the large NoSQL database query technology based on Chinese Railway Science, 2014, 35 (1): 135-141.

[14] Zhu Jiansheng, Wang Jianxiong, Zhang Junfeng. Research and application of the large NoSQL database query technology based on Chinese Railway Science, 2014, 35 (1): 135-141..

.