Key Technology Study on Data Management in Spectrum Monitoring System Based on Mongo DB

Xu Junying[1, a], Kong Deyang[1, b], Li Jing[1, c] and Chen Jingye[1, d]

[1]Beijing Aerospace Control Center, Beijing 102206, China;

[a]xujunying88@126.com, [b]leok126@126.com, [c]243726337@qq.com, [d]huifajiang@163.com

Keywords: spectrum monitoring, data management, MongoDB.

Abstract. The paper introduced the composition of a spectrum monitoring system based on MongoDB's distributed database system, analyzed the key issues of the data management system design which should be resolved with special focus and provided specific storage solution design based on MongoDB with high performance and large capacity. In the end, it was proved that the program can meet the requirements of high throughput in application services with high concurrency.

Introduction

Satellite communications spectrum monitoring system can rapidly detect whether satellite communication link is disturbed or not and eliminate the interference by analyzing the carrier features of the interference source. Spectrum acquisition, signal processing, data storage, information comparison and alarms are realized mainly through the coordination of software and hardware. In addition, user interface and B / S system are also provided to facilitate local and remote real-time and network-wide spectrum monitoring carried out by the user.

Developed in such background, satellite communication spectrum monitoring system for satellite communication network administrative departments uses hardware board to achieve the acquisition and high-speed transmission of satellite spectrum signal. Data storage is realized by using software. The monitoring system can conduct spectrum calculation, modulation mode recognition and characteristic parameter extraction. It can carry out automatic identification of alarms by comparing standard templates. Besides, in order to realize local and remote real-time and network-wide spectrum monitoring carried out by the user, user interface and B / S system are provided.

Introduction of spectrum monitoring system

Spectrum monitoring system consists of five major parts, including spectrum acquisition equipment, core software system of spectrum monitoring service, database management subsystem, data communication middleware and application terminal software of monitoring management. Among those parts, the database management subsystem is responsible for the storage of acquired spectrum data and provides data retrieval function. Data management system is related to the stability and scalability of the entire system. Framework of the software system is shown in Figure 1.

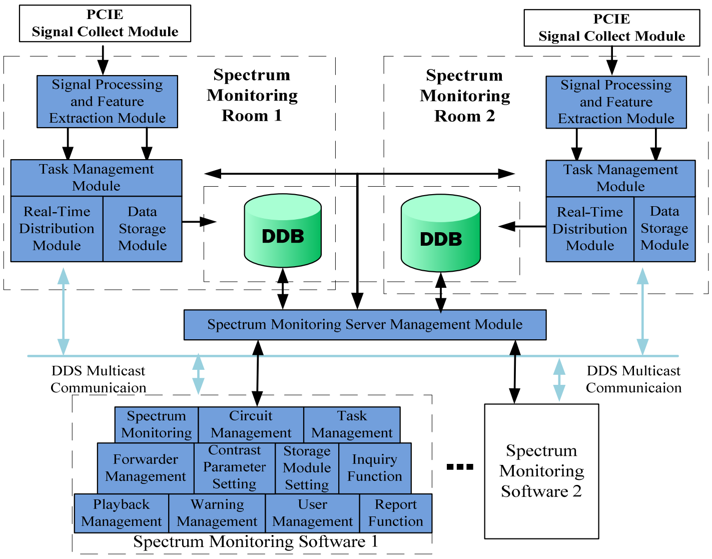
PCIE
Signal Collect Module
Signal Processing and Feature Extraction Module
Spectrum Monitoring Room 1
Task Management Module
Real-Time Distribution Module
Data Storage Module
DDB
PCIE
Signal Collect Module
Spectrum Monitoring Room 2
Signal Processing and Feature Extraction Module
Task Management Module
Real-Time Distribution Module
Data Storage Module
DDB
Spectrum Monitoring Server Management Module
DDS Multicast Communicaion
DDS Multicast Communicaion
Spectrum Monitoring
Circuit Management
Task Management
Forwarder Management
Contrast Parameter Setting
Storage Module Setting
Inquiry Function
Playback Management
Warning Management
User Management
Report Function
Spectrum Monitoring Software 1
...
Spectrum Monitoring Software 2

Fig.1 Organization chart of the software of spectrum monitoring system

Introduction of MongoDB

MongoDB [1] [2] is a product between relational database and non-relational database. It is a NoSQL database and is based on the open source project of distributed file storage. MongoDB is a scalable database with high performance. The data it supported has very loose structure, which is similar to the formats of JSON and BSON, and for this reason it can store more complex data types. The biggest feature of MongoDB is that it has powerful query language, of which the grammar is similar to object-oriented query language. Its query language can almost realize most functions of single-table query of similar relational database, and it supports data indexing.

Therefore, by using MongoDB as file-level storage in the key database of this system, the bottleneck of high-concurrent location data storage could be resolved.

Key Technology

Structure Design of Overall Storage. Storage structure of the system adopts distributed[3][4] system structure, using multiple storage servers to share storage load and location servers to locate stored information. Data management subsystem consists of two parts: one part is the data storage which uses MongoDB cluster’s data read-write interface and is operated on spectrum monitoring server, and this part is composed by several database servers; the other part is data read-write interface based on MongoDB which is deployed together with the core system of spectrum monitoring system and response to the query and write request of each monitoring terminal. The structure of overall storage is shown in Figure 2.

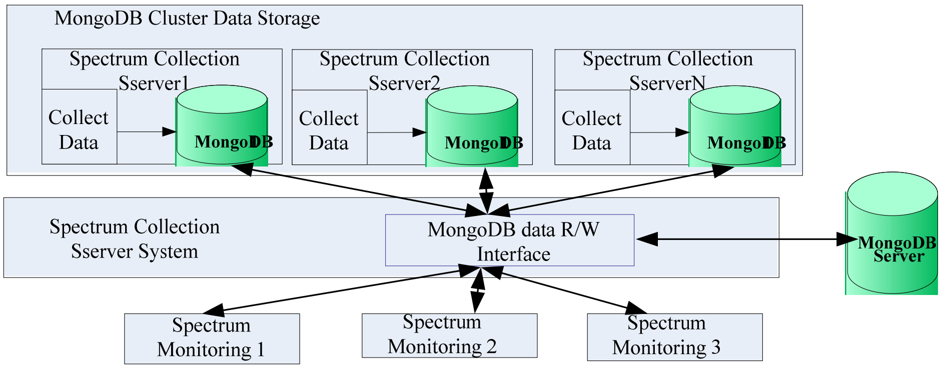
MongoDB Cluster Data Storage
Spectrum Collection Sserver1
Collect Data
MongoDB
Spectrum Collection Sserver2
Collect Data
MongoDB
Spectrum Collection SserverN
Collect Data
MongoDB
Spectrum Collection Sserver System
MongoDB data R/W Interface
MongoDB Server
Spectrum Monitoring 1
Spectrum Monitoring 2
Spectrum Monitoring 3

Fig.2 Structure of overall storage

Analysis on the Storage Requirements of Spectrum Data. Spectrum monitoring system supports two storage methods: continuous storage and interval storage. In continuous storage, storage needs to be carried out every time after a sample is taken, and the sampling is conducted every 0.1s. The system supports maximum 48-hour continuous storage, and after 48 hours existing data will be covered. Internal storage means the storage is carried out at regular intervals (1s, 10s, 100s).

The system needs to support multi-user data access at the same time, including: data playback function, data query function, alarms automatic analysis and comparison.

The number of channels supported by the system is scalable, and the increase of the number of channels should not influence the operation of the software of the system.

Thus, based on above analysis, the storage design of spectrum data should consider the following three aspects:

1. The user's read and write requests should be balanced to avoid overpressure in single database server, which negatively influence the performance.

2. Redundancy needs to be provided to avoid the system failure or data loss generated by the service invalidation of one database.

3. With the new acquisition servers being added, the data continue to expand, and it is necessary to consider the horizontal extension function of the entire storage system.

Storage Design of Spectrum Data. Therefore, database cluster and data sharding design should be given due consideration in the storage design of spectrum data. Data storage design mainly includes the following three points:

1. MongoDB cluster configuration[5]

The cluster configuration of MongoDB provides load balancing and redundancy design, which includes three services: 1) Slice Server: Slice server is used for the storage of actual data blocks. 2) Config Server: Config server is used to store the metadata information of the cluster, including each block information. 3) Route Server: Route server is for the operation of MongoDB instance. The design of spectrum database server cluster is shown in Figure 3.

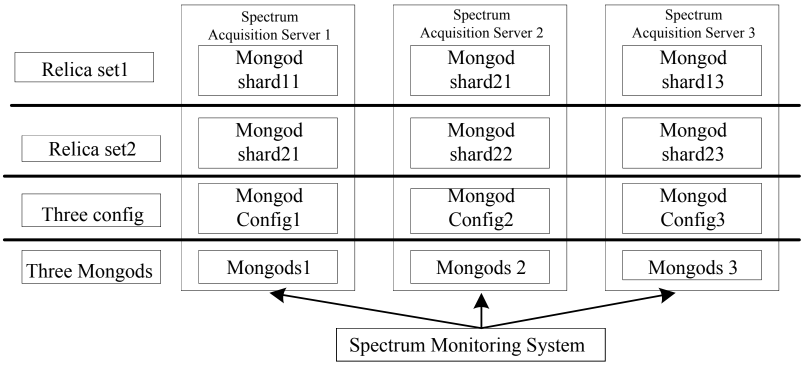

Fig.3 Design of server cluster

In the system, two Mongods (database service instance) are operated in each spectrum acquisition server to constitute two replica sets (Replica set 1 and Replica set 2) as the sharding of two clusters. Each machine runs one Mongod config as the config server. At the same time, each machine runs one Mongos process as a route server which is used for client-side connection. Application side is directly connected to the route server for database read-write operation. In data writing, the route server balances the load based on relevant configuration. And in data reading, the route server assigns the queries to each slice based on the scope of key words of the query requests and combines the returned results to return to each application side.

2. Balance design of cluster and load

The entire storage structure adopts multiple route servers. Based on the insertion of application sides, the route server obtains relevant metadata information from configuration server and automatically decides the distribution of storage to corresponding shards, avoiding the read-write overpressure of the single point. Each machine which compose the replica cluster of shard adopts master-slave configuration. The master machine can read and write, but the slave machine can only read. Master and slave host machines are synchronized at certain time intervals [5-6].

3. Data sharding design

When the spectrum data is stored, the cluster of record becomes very large over time. Reasonable design should be made so that the route server of the cluster can distribute different clusters to different shards and carry out segmentation processing to distribute to different shards, giving full play to the cluster's function of load balancing. Data sharding design mainly includes the following two aspects:

1) Design of spectrum data cluster.If all spectrum data is stored in one cluster, the data amount of one cluster will be to large, and this is not to the benefit of management and distribution. Therefore, in the design, the data should be divided to the nearest clusters and the storage distribution of each cluster is decided by the route server of MongoDB.

2) Cluster sharding design. If the data in one acquisition server keeps increasing, it is necessary to carry out segmentation process for the cluster to perform the functions of the cluster in a better way. To carry out sharding process for the clusters stored in the database of MongoDB, it is necessary to select keywords. The clusters will use the key words for the sharding of document to distribute to different database servers.

Conclusion

The author sets up the a test environment as follows: Windows7 64-bit operating system; RAM: 4GB / 1067MHz / DDR 3; CPU: 2.53GHz Intel Core 2 Duo. The performance of data management system established based on above ideas is tested. According to the test, in terms of data insertion performance, it takes 25 seconds to insert 100,000 pieces of data. After the recorded in inserted, the memory usage of MongoDB is as mush as 2G, and it can be presumed that this is caused by the pre-allocation of space carried out by MongoDB. The speed of obtaining records through primary key is 33 seconds per 100,000 times, and the update speed through primary key is 25 seconds per 100,000 times. The test results are shown in Table 1.

Table 1 Test results

Number of concurrent users	Number of records inserted (10,000 pieces)	Average throughput capacity (piece/second)
5	110	6600
10	220	6300
20	220	6300
50	550	4500
100	550	4100

References

[1] Dannis Ludy, Statellite Communication[M] Beijing: People's Posts and Telecomminication Press, 1997.

[2] MongoDB: http: / /www. mongodb. org/display/DOCS/Home.

[3] Dongmei Lu, Xianbo He,Brief Analysis of NoSQL Database[J]. Science and technology of Westen China,2011，(2):14～16.

[4] Xianli Huang . The development and application research of non-relational NoSQL Database[J］. Fujian's computer，2010，(7):30～32.

[5] Chodorow K.MongoDB:the definitive guide[M].O'Reilly,2013.