Memory Database Application in the Processing of Huge Amounts of Data

Daqiang Xiao¹, Qi Qian², Jianhua Yang³, Guang Chen⁴

¹State Grid Corporation of Central China, Wuhan 430077, China

²NARI Group Corporation (State Grid Electric Power Research Institute), Beijing 100192, China ³State Grid Corporation of Central China , Wuhan 430077, China

⁴NARI Group Corporation (State Grid Electric Power Research Institute), Beijing 100192, China

Keyword: mass data process, memory database

Abstract. In large electric power data acquisition system, huge meter, data collector, concentrator real-time message processing and timely rapid response to database system efficiently, presents a high technical requirements. At present, single disk database is very difficult to meet the demand of high efficiency and low delay. In view of the requirement of massive data acquisition system in real time processing of massive data, a new data processing framework based on massive data in memory database is proposed. We comprehensive compare the single disk database with the improvement method, and illustrate the advantage of new framework through the experiment.

1 Introduction

With the development of information technology, the traditional database disk (DRDB) plays a very important role in maintaining and managing data. However, in the current large electric power data acquisition system, with the explosion of data and information quantity, database systems are required to not only possess strong and complete functions and complex processing capacity but also to collect and process massive data within a specified period as well as react in a real-time, efficient and quick manner. Traditional disk databases need to process data by accessing disk very frequently, and thus, in terms of processing massive data, a large amount of I/O operation becomes the bottleneck of real-time massive data processing performance. Therefore, it is difficult to provide real-time data needed by applications is stored in the memory, while all the operation of the database is in the memory, which is, thus, effectively eliminates the bottleneck of disk I/O operation[1]. Of course, memory database also has limits in its own features and it is difficult to store a large amount of data to the memory database completely. Therefore, it is very necessary to study a new real-time data processing framework based on massive data in the memory database.

2 Memory Cache Technology

The data acquired by the system is getting larger and larger in scale and has reached massive data level along with the growth of number of users accessed to the electric power data acquisition system, the deepening of the carrying out of system service applications, the increase of the number of data items acquired by the system, and the increasing frequency of acquisition. The loading performance of the system's database is seriously affected when it is performing functional application, data statistical analysis, and exterior interface integration as well as other application and extension based on such a huge amount of massive data. In recent years, as the memory capacity is rising constantly with the price dropping down and down, 64-bit operating systems have been able to support larger address space, and thus, it has become a hot spot to take full advantages of memory cache technology to enhance system performance.

Memory cache technology refers to the technology of storing data in the memory, the essential feature of which is that, the read-write speed of its master-copy or "working version" resident memory is several orders of magnitude higher than a disk, and the application performance is

improved greatly when the data is saved in the memory rather than accessing data in the disk. Meanwhile, giving up the traditional way of disk data management, the memory cache redesigns the architecture based on the fact that all the data is in the memory, with improvements also being made in data cache, fast algorithms and parallel operation correspondingly, which greatly accelerates its data processing compared with that of traditional databases.

The data storage platform based on memory cache in the main electric power information acquisition station system has effectively eased the read-write and computing pressure for the database, and reduced time of data statistical analysis, which can provide references for making marketing and management decisions and timely identifying economic trends based on power consumption analysis. It will also provide more direct, objective, truthful and timely data for facilitating the healthy development of regional economy and the implementation of corporate energy saving.

3 Data Processing and Analysis

Data in an electric power data acquisition system, according to the use attributes, falls into 2 categories namely [2], real-time data and dynamic data. Real-time data is mainly made up of real-time messages which are very intensive at one same time and may even reach thousands of batches per second. The speed responding to them will directly affect the availability of the system. Dynamic data refers to the data changing very frequently, the quantity of which can be very large and some of which is even historical data. Therefore, such data is usually stored in the massive database of the disk database, and its highly frequent change may also require high instantaneity. In addition to processing on application layer, sometimes real-time data may also be required to be stored to massive database as historical data, and historical data in disk will also be imported to the memory database by massive database in a repeating and replaying mode and become real-time data. In a large electric power data acquisition system, because of the storage of massive database so as to fulfill the requirements of processing massive database, the main objective should be to ensure that at least the data sets required by one transaction can be found in the memory database.

4 Real-time data processing

As for real-time data, the processing efficiency is slightly lower than the efficiency of loading it, and the type of messages during the test is largely the intensive one, which will ensure the efficiency of transaction operation in the memory database. In an electric power data acquisition system, such real-time messages are not long-lasting data, which means that after being processed completed, the data will be eliminated from the memory database or transferred to the massive database as historical data as required so as to ensure the availability of the memory. Of course, such real-time message processing method may produce 2 results: 1. The memory database is empty without any data, indicating that all the data has been processed completed by the data processing needs to enters the standby condition; 2. The memory database is fully written, the main reason of which is the data processing program is not started up, and thus, an elimination algorithm is in need. For example, FIFO algorithm is used to eliminate data used least frequently lately, or tailored an elimination algorithm according to the specific need.

5 Dynamic data processing

As for dynamic data, the data quantity is very large, and therefore, it is necessary to put data in and out of the memory database and the massive database per specific data exchanging strategy and ensure no I/O operation being performed during the information processing transaction in process.

To achieve this goal, the following requirements should be met:

(1) Establish transaction form and prioritize transactions as per their levels;

(2) Acquire transaction data per the transaction form, and load data in the memory database with transaction data set as the basic loading unit;

(3) When new matters are incorporated, the priority should be redone, and the transaction set data swapped in should be adjusted;

(4) Start up timing data loading;

(5) Establish an increment list to record data newly added or updated. The consistency processing module ensures the consistency of massive database and memory database by reading the increment list.

6 Design of Memory Database

The balance between the memory database and the massive database of the disk database can be effectively achieved through improvements of memory database, so as to ensure that the real-time program can complete the processing of all kinds of transactions in the memory database.

7 Design principle

The large electric power data acquisition system is divided into many stations according to the test requirements, with each one processing the corresponding tasks, and thus, a distributed memory database system is established based on the functions of each station. Distributed database is a data set physically distributed on each node of the computer network but logistically belonging to the same system. In this system, it is necessary to satisfy that:

(1) The memory database of each node has its own autonomy;

(2) Data distributed to the memory database of each node is exclusive, which can lower the possibility of same data being loaded more than once;

(3) The memory databases of different nodes are cooperating with each other, the best way of which is to make each memory database serve as the service end of other nodes;

(4) Keep the data distributing transparently. Such distributed memory database system can properly meet the two major features, namely distribution of data and cooperation between every two databases. With the improvement of balance between the memory database and the massive database, the requirements for real-time massive data processing can be met well.

8 Applicable scheme

Redis possesses outstanding performance, including its throughput rate and average response time [3], which, however, does not support distributed type. Redis database not only is excellent in single-point performance and supportive to multiple data structures but also provide user-friendly API and operating performance.



Fig. 1 Structure Diagram of Redis-based Distributed Write Cache Subsystem

Support Redis-based distributed write cache subsystem by expanding Redis, so as to use it to cache data acquired by acquisition and communication system. When the quantity of the data reaches a certain level, the data should be written in the relational database D3BASE in batch. Batch-writing of data is helpful for raising the IO utilization rate of disk, which, thus, largely reduces the transaction processing load of database. When the entrance speed of the relational database is lower than the speed of data generation, the data can be stored in the memory temporarily, which will be then written in the database later when the database is capable to process, and by such means, the system can support the entrance of massive and highly concurrent data into the database for processing [4].

In the expansion of this technology, Redis is improved from two aspects: namely (1) realizing distributed cache mechanism so as to facilitate the dynamic expansion of the system; (2) revising the "Key Expiration Processing Logistic" of Redis into the procedure of writing acquired data into the relational database. The structure of its system is shown in the figure. It adopts a concentrated inquiry mode, with one distributed management center performing distributed information management, namely the partition routing node shown in the figure. In the process of data access, the cache client will acquire the storage nodes of the data from the partition routing node, and then the client will connect to the cache nodes to write the data.

Meanwhile, there is a round-robin queue in the database entrance node which is used to receive data triggered and pushed by the expiration of Key from the Redis cache node, and meanwhile, the connection of single relational database is maintained, with the batch writing into the relational database being performed! This kind of system structure can be achieved simply, with small load being caused to the management end. Thus, it is available to focus on the maintenance of data distribution cache node information, which, thus, can bring great dynamic expandability to the system.

Communication mechanism between two nodes:

(1) Registration, exit, and heartbeat protocol of the Redis cache node.

The Redis cache node will broadcast its own status (startup/shutdown/operation).

Partition routing node receives the broadcast and updates the status table for Redis cache node, and it will be responsible for responding the inquiry request initiated by Redis cache node to the database entrance node.

(2) Protocol for request of Redis cache node connection

Cache client asks the partition routing node for Redis cache node.

The partition routing node receives request, provides Key according to the cache client, and then selects corresponding Redis cache node per the two-phase partition scheme.

The Key expiration processing strategy in the cache system is as follows:

Measurement data acquired, based on acquisition, mainly consists of two major categories, namely daily acquisition and monthly acquisition, wherein the daily one includes data acquisition with a frequency of 15 minutes. Obviously, when the cache data period is longer than the

acquisition interval, it will have no significance, and therefore, it is necessary to ensure that the data in the cache can be stored in the database timely, or data acquired might be lost. Set the Key expiration times respectively based on the acquisition frequencies, and rewrite Key expiration processing of Redis. When the Key is expired, the measurement data that is not stored in the database currently is pushed to the "database entrance" node of the cache subsystem, and meanwhile, the database-entrance zone bit (True) of the current Key is changed, so as to avoid repeated database entrance.

9 Safety design of memory database

The potential safety hazards of memory database system are mainly from the permanent memory of data and exception handling such as outage [5]. In the embedded system, the system resources are compact, and the regular database cannot be used, resulting in limited optional data management schemes. Moreover, in order to guarantee the persistence of data, the operating information of the database is stored in the log buffer with the engine technology of memory database, and then such information is synchronized to the log file on the disk by the background process. Meanwhile, the data increment of memory is written to the data file on the disk at intervals, and the log files that have been synchronized are cleared. By doing this, automatic recovery can be conducted through such files after the database is restarted. In addition, to prevent the breakdown of the whole electric power data acquisition system arising from the loss of data in the case of abnormal outage, the HA mode with multi-mode disaster recovery is adopted, to conduct one-to-many or many-to-many backups for the memory database; in case of abnormal outage for the master station, other slave stations will be automatically upgraded to the master station, to maintain the normal operation of the system. Initial synchronous timer, heartbeat detection timer, transmission timer and reception timer are used to ensure the synchronization between the master station and the slave station.

10 Performance test

3 Redis nodes are adopted for the distributed memory cache database of the testing environment, and are deployed in three common PCs, with the memory capacity of 2G, and D3BASE is adopted for relational database.

For the test system, 112 substation terminals, 8,948 special transformers, 5,559 power distribution areas and over 560 thousand low voltage users are accessed. There are more than 15 thousand terminal devices and over 590 thousand meters, including station telemetering, special transformer users, distribution transformer areas and low voltage users.

The test system has two cases; one is that the acquisition nodes of the measurement acquisition subsystem directly establish the connection to the relational database, to conduct the operation of record storage; the other is that the cache subsystem proposed by the project is adopted (as shown in the figure), and the acquisition nodes submit it to the distributed cache system rather than directly inserting into the relational database; the database entrance operation of all measurement data information is realized by the database entrance subcomponents in the distributed cache system.

The response time of data storage operation and the average number of insert records per second are selected as the comparative indices, and comparison is conducted for the two cases of adopted cache subsystem and unadopted cache subsystem.



Fig. 2 Real-time change of insert operation response time

Fig. 2 shows the real-time insert operation response time that is randomly observed, and the response time refers to the delay in storage operation of the communication subsystem; the result shows that the insert operation response time when cache subsystem is not adopted has obvious fluctuations, with the delay range from a few tenths of a millisecond to around 1 millisecond; after the cache subsystem is adopted, the delay in storage record operation of the communication subsystem is almost unchanged and maintains at around a few hundredths of a millisecond.



Fig. 3 Change of average response time of insert operation with the access scale

Fig. 3 shows the average response time of insert operation when the access acquisition terminal scale is increased from 100 thousand to 600 thousand. It is shown in the figure that the response time just slightly increases when cache subsystem is adopted, but it is only a few hundredths of a millisecond, reflecting the stable expansion capacity. When the cache subsystem is not adopted, with increasing quantity of access acquisition terminal, the response time is increasing, and after the quantity of access acquisition terminal is increased to 400 thousand from 300 thousand, the average response time jumps obviously, and the increase tendency of the response time is more obvious thereafter; the reason is that before cache subsystem is adopted, each acquisition node is directly connected to the database server, and the database entrance performance is seriously limited by the database server, and the expansion of subsystem is also limited; a connection to the database server is added when an acquisition node is increased, which not only increases the load of the database server, but also further decreases the relational database performance with the effect of transaction mechanism as the same data sheet is operated by each connection.



Fig. 4 Comparison of average speed of storage records

Fig. 4 compares the average insert speed, and for the index, statistics are conducted by the "database entrance" subcomponents in the cache subsystem; results in the figure show that after cache subsystem is adopted, the record insert speed is increased from around 1,000 pieces per second averagely to above 30,000 pieces per second averagely, showing that the average write operation performance of the relational database is significantly enhanced.

11 Conclusions

In large electric power information acquisition system, based on the requirements for real-time processing of information such as huge meter data, data collector and concentrator management and electricity purchasing sheet as well as data from the disk database, the paper proposes the new scheme of real-time processing based on massive data in memory database, improves the memory database system and combines the cluster processing of memory database, to share the massive data to multiple memory databases for concurrency control[6]. Currently, the real-time processing model based on massive data in memory database is successfully applied to the electric power information acquisition system, and the efficiency and reliability are testified.

References

[1] Tang Zhijun, Real-time Message Storage Access Strategy Based on Massive Data in Oracle

[2] Hector Garcia Molina, Kenneth Salem. Main Memory Database System: An overview [J]. IE EE Transactions on Know Ledge and Data Engineering (1992), 4 (6): p.509 - 516.

[3] Liao Guoqiong, Embedded Mobile Real-Time Database Transaction Processing Technology [M]. Beijing: China Railway Publishing House (2010).

[4] Ouyang Jingwu, Introduction to Distributed Database System [M]. Beijing: Aviation Industry Press (1989).

[5] Xiao Yingyuan, Distributed Real-time Database Technology [M]. Beijing: Science Press (2009). ET

[6] Xia Huosong, Data Warehouse and Data Mining Technology [M]. Beijing: Science Press (2011).