Research on Performance Optimization Technology of Complex Equipment Software Database

Bo-Jiang Liu, Li-Jin Wu, Wei He, Xin-Yu Han and Long-Li Tang

China state shipbuilding corporation software Quality and Reliability Testing Center, China

E-mail:cimtec1bj@163.comg

Abstract. Complex equipment software usually uses a database to store data. In order to enable it to run stably and efficiently during use, it is necessary to ensure the performance of database access. How to design and manage the database reasonably and optimize the database query has attracted much attention. This article studies the database access performance problems of MySQL equipment used by complex equipment software, studies database optimization techniques, and provides a database optimization solution for the database access delay problems that occur during project operation. Reduced waste of resources, thereby improving the efficiency of system database access, and solving the problem of system access to the database delay.

1. Introduction
With the development of informatization, the equipment software in the marine field is becoming more and more complicated. In the case of a surge in information and data volume, data transmission between networks is busy, and the throughput of the system becomes large. These have put a lot of pressure on database performance, and database optimization technologies have received much attention. How to ensure the data processing ability of the complex equipment software database by optimizing the database performance has become an important research topic.

2. Overall system design

Figure 1. costs&SQL optimization relationship
2.1. Costs & SQL optimization relationship

It can be clearly seen from the figure 1 that the common methods of MySQL database optimization and the level of cost. The optimization of the SQL statement and the optimization of the index are the least costly but the most effective methods. About these two points, I have summarized the following optimization methods:

- Sub-queries are not used in SQL statements, such as delete from user where uid not in (select id from order), because using a sub-query database to create temporary tables in memory consumes resources. If the two tables are related and are often called together, It is best to create a foreign key in another table using a join statement query, such as: delete from user left join order on user.id = order.uid;

- It is best not to use "*" instead of known fields in the sql statement. Even if you want to query all fields, you must write out each field, because using * instead of the field database must first query which fields in the table before executing the sql statement Inquiry, meaningless one more inquisitive work;

- Reasonable indexing including: Indexes can greatly speed up database retrieval, especially when columns appearing in where and join or when sorted by Order by are much faster (when judgment or comparison or sorting is required); However, it is not better to build more indexes. Too many indexes will cause redundancy, because each delete, update, and add will refresh the index. Too many indexes cause other operations to consume too many resources, and it is not necessary to create too small tables. Index, no one has seen a two-page leaflet or a catalog.

- The second most important thing is the optimization of the table structure. There are also a few summary points in this regard: Select the most suitable field attribute and use the smallest data type that can have data, such as postal codes, mobile phone numbers, and other fixed-length numbers can be char (6), char (11); gender or whether this kind of judgment text can Use tinyint; the field attribute is not null as much as possible so as not to judge whether it is empty, reduce one step (in other ways to express the NULL you want to express, such as -1); if you must use this type of text, it is best to use a table storage; Store frequently used information and infrequently used information, such as the user table of a mall website, user nicknames, avatars, passwords, account numbers and other fields that users will use to log in, and the users 'hobbies and favorite colors will This field is stored in separate tables. I believe that the personal information in your Jingdong account may have been opened when you registered, and you haven't noticed it in the future.

2.2. Database optimization research status

The concept of database performance optimization was first proposed by the famous computer expert Tony Doherty 45 years ago[1], and its proposal greatly promoted the research and development of database access. Later, someone gave a definition of performance optimization, pointing out that the activity of changing system performance by modifying system parameters, system configuration (hardware adjustment), and optimizing component applications is performing performance optimization[2]. How to optimize database performance by unblocking network traffic, saving disk I/O overhead, and reducing CPU time is the focus of attention.

Minimize the response time of each access by optimizing and maximize the data transmission capacity of the database service. In response to the sharp increase in data volume, research on database optimization technology has been stepped up at home and abroad. The University of Michigan has done a lot of research on database query optimization. They have proposed techniques such as query detection, query sampling, fuzzy query, and attenuation cost assessment to optimize global queries. At present, in the general system design and development process, sometimes the blind pursuit of full-featured and development speed is ignored, and the performance optimization of the database is ignored, and problems are often resolved only when they occur. Therefore, our country's research on database technology and database optimization technology needs to be improved.
2.3. Error Report

Database optimization technology can also be processed from the perspective of designing a database. Generally, anti-standard processing is used to merge many tables with a small amount of data. This reduces many I/O operations and achieves database query optimization. In the case of multiple users, in order to reduce the number of concurrent connections to the database, the database connection pool is used; when the data records increase sharply, in order to increase the query speed, an index or row prefetch method is used., etc[3].

3. Database optimization technology introduction

The amount of data for complex equipment software has gradually increased. A large amount of data has caused irresistible obstacles to the development of complex equipment software. Database performance has become the focus of people's attention, and database optimization technology has been born. Database optimization is mainly to reduce the time and resource consumption during database access through some techniques and measures, and then achieve the purpose of improving system performance[4].

3.1. Database connection pool technology

The first step in database access is to establish a connection with the database, and the last step is to disconnect from the database. It can be seen that database connection plays a vital role in database access. In addition, each connection must complete tasks such as user authentication and security context configuration. This requires a certain amount of communication and memory resources, so it often becomes the most time-consuming operation. For all database connection problems, choosing an optimal connection method can greatly improve the database performance of the system.

The database connection pool is a better database connection technology than the JDBC direct connection technology. It establishes a buffer pool to store a certain number of database connection objects. Before the system accesses the database, it is necessary to first request a database from the database connection pool. Connection objects. If there are idle connection objects in the connection pool, the connection pool will allocate the idle objects to the system. After the system accesses the database, the database connection is released and the connection pool recycles it.

The core idea of database connection pool is connection reuse. In the connection pool, there is a set of own mechanisms to establish, manage, and close connection objects, and control the number of connection objects and the length of time that the connection exists. The connection objects that have been released after use are recycled to the connection pool and can be re-used. Once used, this can avoid frequent database connection establishment and closure during database access, saving access time and system overhead resources[5]. The connection pool works as shown below.

![Connection pool working mechanism](image)

**Figure 2.** Connection pool working mechanism
3.2. Database indexing technology
In database access, the main operation is still database query. In database access time, query time also accounts for the largest proportion, so query optimization is also a key step in database optimization. The best optimization method in query optimization is database indexing technology.

An index is a database structure that can help speed up queries on a field column. Appropriate indexes can be sorted in a specific way, which eliminates the need to perform a full table scan when scanning a data table, does not need to scan redundant data records, reduces the number of data records to be scanned during a query, and saves time required for a query.

Ascending or descending order of the field columns according to the value of a key field required by the query is the process of creating an index file. The ordering is not aimed at the physical order of the fields, but the logical order of establishing data records according to an index key. The reason why the index can speed up the query is that after the index is established, the data records are ordered in the index. At this time, various methods can be used to speed up the query[6], such as binary search and other query methods.

Indexes are not a panacea. Only appropriate indexes can achieve the corresponding optimization effect. Indexes need to follow the following principles:

1. Need to pay attention to the size of the database table to create the index, that is, the amount of data in the field column. The amount of data should be moderate, otherwise the performance of the application system will be affected.
2. Pay attention to the values in the field column. If there are many empty values, this field is not suitable for creating a field.
3. The number of created indexes does not need to be too much, and only the needed and most commonly used fields are indexed. It is best not to exceed 5 in each table, and more will only waste system overhead.
4. If the multi-table joint query is involved in the database query access, it is necessary to perform index design on the key fields of the connection.
5. The optimization effect of a composite index is often better than that of a single index. It is best that all fields of the query have an index design, so that there is no need to query the database table, and only the index file can be used to obtain the required data results.

3.3. SQL statement tuning techniques
SQL stands for Structured Query Language. It is a standard computer language for accessing and processing databases. Under the condition that the SQL statement is guaranteed to be correct, and then appropriately adjusted and optimized, the time required for the system to access the database can be greatly saved.

SQL statement tuning technology mainly refers to reducing the amount of data to be scanned during a table query, or reducing the number of required query tables, to find an optimal query path through the format of the SQL statement. There are many SQL statement tuning techniques based on SQL statement design. This article lists several commonly used optimization methods:

1. Filter first and then join. First, filter out a large amount of data in the tables to be connected according to the query judgment conditions, and then perform multi-table union. It is not necessary to filter all the data in these tables once in the joint query.
2. Avoid subqueries. A subquery is to query all the data in another table under one condition. The amount of data is the product of the two tables. This data is very large. If there is a next level of nesting, the data flow will increase dramatically. Inefficiency is even lower.
3. If subqueries are unavoidable, it is necessary to use filtering conditions to filter out as much data as possible. Optimize the order of the conditions in the Where statement, and the conditions are arranged to the left in detail[7].
4. Replace IN with EXISTS and NOT IN with NOT EXISTS. In many base table-based queries, in order to meet one condition, it is often necessary to join another table. In this case, using EXISTS (or NOT EXISTS) will usually increase the efficiency of the query.
(5) reduce the number of access to the database table. If possible, when multi-table joint query, try to determine the database table where the required data is located as much as possible, and do not perform a table scan on the unnecessary data tables.

(6) Temporary tables can be used appropriately in joint queries. A temporary table is a data table used to temporarily store query records. In the SQL statement, use AS to name the temporary table.

(7) In fact, indexing is also a method of SQL tuning. The statement that creates the index is an SQL statement.

4. Case analysis
This chapter takes the ship's driving control platform detection management system project as an example, through the investigation of database development scale and database access performance requirements, combined with database access technology and database optimization technology research, to improve its performance.

4.1. Demand analysis
4.1.1. Large amount of data
The amount of data is divided into two parts: one refers to the amount of data in the database tables. Because of the rapid development of the communications industry today, higher demands have been placed on the development of detection and perception, which has led to the size of the database and the need to manage the database. The amount of data in the table is increased by the primary key; the other part refers to the amount of data to be traversed during database operations, because in the implementation of system functions, it is often necessary to perform nested queries, causing query records to be accumulated in a productive manner and the amount of data accessed It can become bloated and affect system performance[8].

4.1.2. Complexity
Because the information is complex, in the design of the system database, you need to design a suitable information table, the data table fields are appropriate, and the table data information is related to each other. In some database access operations, the data table needs to be considered as a whole. This will cause the SQL statements of the database operation to be very complicated and the judgment conditions will be very chaotic, which will further affect the performance of the system.

4.1.3. Database tables changes frequently
Due to the needs of the task scenario, the information in the database table needs to be updated regularly. The data update of the system database is often irregular in a certain period of time, so there is a certain degree of uncertainty in the system access. The number of database access operations increased, consumes a lot of system resources and access time.
At present, a large amount of data has been accumulated in the system database, and many system function modules require joint table query, and the system access performance has experienced a certain delay. If the amount of data is further doubled, the efficiency of the system will also decrease. Database access is generally divided into three processes. The first process is to establish a database connection, the second process is to perform database operations, and the third process is to process the data result set.

4.2. Analysis of system database performance optimization requirements
4.2.1. Requirements analysis for database connection optimization
The most important functional modules of complex equipment software are data insertion and comprehensive query, and transaction operations will be more intensive periodically. Each time a data record is inserted, because there are many query conditions, the query content will accumulate more, which requires frequent database connection establishment and release. In database processing, the biggest resource overhead is to establish a database connection. For example, according to the previous processing mode, every time a database connection is established and released, it will
consume a lot of resources and time, and it will easily lead to memory leaks and system maintenance difficulties.

4.2.2. Optimization

SQL statements will affect the efficiency of database access. As the equipment software becomes more complex, the amount of data generated by the system continues to rise, and the data size becomes larger and larger. As the size of the database continues to expand, the time required for each query will become longer and longer, and the efficiency of database access will inevitably become lower.

4.2.3. Demand analysis of database stored procedure optimization

Business processing logic is performed in the application, which will involve multiple database connections. Each query needs to be performed separately, and the code reuse rate is low, causing unnecessary waste of resources. The dramatic increase in data volume has affected the query performance of complex equipment software, and an optimization measure is urgently needed to optimize the database access performance.

4.3. Summary

The optimization of the complex equipment software database must be performed during system design and deployment. After the system function modules are divided, the type and quantity of information used in the function modules are determined, and the number of information tables and the field names and data types used in the information tables are further determined.

Three steps for database access: (1) Establish a database connection. (2) Perform database operations. The main operations are adding, deleting, modifying, and checking. (3) In the process of accessing data records, the program needs to take out the data records obtained from the query in the database and submit them to the front-end page for processing, analysis, or display.

In database access, these three steps will affect access efficiency. This article researches and analyzes the three steps of database access to find out the key points that affect the efficiency of the database. To optimize the efficiency of database access from the database, the following technical measures can be taken:

1. Optimize the query statement: reduce the running time of the SQL statement as much as possible, try not to judge the field null in the where statement, use in and not in with caution, avoid performing this operation in the where statement.

2. Create an index: The index will sort the key field columns we want to query, which will speed up the query speed and improve the system access efficiency.

5. Conclusion

This paper studies the database access performance of complex equipment software as the research object, and discusses the optimization design of the database. At present, with the increase of the amount of data, the size of the database is getting larger and larger, which has greatly affected the access performance of the database. This paper studies the performance optimization of databases at home and abroad, optimizes the query process of complex equipment software databases by optimizing SQL statements and building indexes, and optimizes the process of database query results by using data transfer technology.

References

[1] Liman. 2015 Development of Fault Identification Method and System Based on Machine Learning (Beijing: Beijing Jiaotong University)

[2] Xie Weiwei. 2009 Application Research of Machine Learning in Analog Circuit Fault Diagnosis D. Qingdao: Ocean University of China 25-37.
[3] Poole, David L, Maclcworth and Alan K. 2010 Artificial Intelligence: Foundations of Computational Agents (Cambridge University Press) 33-46
[4] Alexander Gammerman. 1996 Machine Learning:University of London 76-98.
[5] Chen Xiuying 2011 Research on Network Fault Diagnosis Expert System Based on Self-learning Mechanism Command Information System and Technology 2(1) 41-6
[6] Fang Hao. 2017 Software Defect Prediction and System Development Based on Machine Learning Nanjing University of Posts and Telecommunications
[7] Du Lei. 2018 Research on Fault Diagnosis of Power Battery Based on Deep Learning. Harbin University of Science and Technology
[8] Huazhi Song. 2018 An Automatic Bug Triage Approach Based on Deep Learning. Wuhan University