Optimizing Database Management Systems: Techniques and Challenges in the Information Age

Chijin Yu[a]
School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China

Keywords: Database Management System, Query Technique, Technology Optimization.

Abstract: The massive emergence of data in the information age makes the development of the field of data management increasingly concerned, database management system as an important tool for data management is also widely used, but the database management system still exists in part of the problem, and it needs to be optimized to facilitate its use. In this context, this paper mainly explores the existing database management system optimization techniques and classifies the existing optimization techniques in order to understand the progress of database management system optimization more intuitively. This paper concludes that the existing optimization techniques and data management systems still exist unsolved problems, in order to provide ideas for subsequent research on database management system optimization. In conclusion, this paper is a classification of the existing database management system optimization techniques and interpretation, hope that the views in this paper can provide some ideas for subsequent research on database management system.

1 INTRODUCTION

In today's fast-developing information technology era, more and more data and information have to be stored, and the cost of information storage increases while the risk of information leakage is also gradually increasing. Therefore, people want to store information most efficiently, securely and cost-effectively. With the further development and popularization of distributed system architecture, how to realize the efficient application of data in a distributed environment becomes the direction to be optimized. Reasonable data sharding can improve the availability and fault tolerance of the database in a distributed environment, thus further improving the system performance and usability. Therefore, reasonable data sharding has also become a direction that needs to be optimized further.

With the further development of science and technology, big data and cloud computing technology are constantly updated. Big data puts forward the demand for massive data processing and storage, and cloud computing puts forward the demand for different paradigms, which need to be completed by the database. Therefore, the database structure and query efficiency must be optimized to meet the large-scale data processing and high concurrency requirements of big data and cloud computing. With the development of the Internet of Things and other fields, real-time analysis of data application technology has also been further developed, thus increasing the real-time data processing requirements of the database. Database update and query functions need to be more efficient to meet the demand for real-time processing. With the popularization of electronic payment, more and more important data to be stored, the risk of information leakage and harm also increased. Based on today's social environment, people also put forward a higher demand for data security. Adopting a more secure way of storing and accessing data is also an area that needs to be considered for optimization. With the increase in labor costs and the further development of artificial intelligence, the database will combine the database with artificial intelligence technology so as to realize that the database's self-maintenance has also become an optimized direction of development. This will further improve the degree of automation and intelligence of the database, thus helping the

[a] https://orcid.org/0009-0000-4566-1345

Yu, C.
Optimizing Database Management Systems: Techniques and Challenges in the Information Age.
DOI: 10.5220/0012925700004508

In *Proceedings of the 1st International Conference on Engineering Management, Information Technology and Intelligence (EMITI 2024)*, pages 264-268
ISBN: 978-989-758-713-9

database to better adapt to the development trend of the information age.

Nowadays, the research related to databases is more comprehensive and there are many optimization researches based on different development platforms and optimization methods related research is also more comprehensive.AB Ammar et al.(Ammar A. B. ,2016) have published a paper on query optimization techniques for graph databases.JM Hellerstein et al.(Hellerstein J. M. ,1998) have published a paper on query optimization using expensive methods.

This paper aims to analyze the current ways and means of database optimization and classify them according to different optimization aspects to facilitate subsequent database optimization operations. This paper will summarize the role of the database in simplifying data management and database efficiency to improve the relevant research and methodology and will be combined with specific cases and practical situations to analyze and explain so that readers can understand so as to facilitate the follow-up on the database optimization of learning and research. In terms of simplifying data management, this paper will analyze the current situation and problems of data management, explain in detail how to simplify the data management process of database technology and research and explain different scenarios. In terms of database efficiency improvement, this paper will summarize the techniques of concurrency optimization, query optimization, storage optimization, security optimization and will analyze the application scenarios based on actual cases to further explain the techniques. Finally, an outlook on the current status of database optimization is given.

2 OVERVIEW OF THE CURRENT STATUS OF DEVELOPMENT OF DATABASE MANAGEMENT SYSTEMS

In the 1950s, with the birth and maturity of computers, computers began to be used in data management, and at the same time, data management technology also developed rapidly. The traditional file system makes it difficult to cope with the challenge of data growth, but it is also unable to meet the needs of multi-user data sharing and rapid retrieval of data. Against this background, database management systems emerged. At present, databases can be divided into three main categories. The first category is the data model as the core of the database management system, such as the relational database management system. The second category, combined with other technologies database management system, such as distributed database management system. The third type of database management system combined with the utilization scenario, such as cloud database management system.

Early database management systems came from file systems, i.e., file-based databases. This system only supported the creation and description of file file directory structures and allowed for long-term storage of large data volumes with minimal functionality in the case of backups. It is also unable to handle concurrent access, which may result in data loss due to concurrency. Cannot use high-level language queries, resulting in low efficiency. However, because of the simple implementation, the file-based database is still a modern operating system internal profile.

In order to compensate for the limitations of file-based databases in reading data, hierarchical database management systems and mesh database management systems have emerged. Hierarchical database management systems have improved the efficiency of reading specific data to a certain extent, but it is still difficult to read complex data. Although it has its drawbacks, it is still used in Windows and Linux file systems. Mesh management systems are based on the mesh data model, and the emergence of its further development of hierarchical databases, to solve the problem of hierarchical databases, have and only have a parent node for developing database management systems to lay the foundation. However, it also brings higher complexity, and the query and update still need to traverse the chain table, so it is gradually replaced.

Although the centralized storage, management and sharing of data, mesh database management systems and hierarchical database management systems have given a better answer. However, there is still a big lack of data independence and abstraction level. In order to solve these problems, a relational database management system came into being. In 1970, Edgar Frank (Ted) Codd published a paper named A Relational Model of Data for Large Shared Data Banks(Codd E. F. ,1970), the relational data model was born, the paper laid the theoretical foundation of the relational data model which ushered in a major change in database management systems. The model proposes that the database management system should be presented to the user in the form of relational tables, and through the relationship so as to realize the rapid querying, adding, deleting and modifying of data, and better guarantee the security of data. In the relational data model, the theoretical support of the relational database management system was born, it is the existing database in the longest living, the most vitality, the most widely used database model. A relational database solves the many-to-many relationship representation, facilitates the direct reading of data in any position in the table, and introduces the concept of foreign keys to facilitate the connection of multiple tables. At the same time, relational databases support ACID characteristics, providing customers with the convenience of protecting the consistency of the data and solving the problems caused by concurrency. Relational database greatly improves the convenience, efficiency and security of database management system and gives the database more powerful flexibility and adaptability. By 1990, relational database management systems had become a standard.

In the mid-1970s, the related research of distributed database management system began to appear, and the world's first distributed database system, SDD-1, was realized by CCA in 1979 on DEC computers.Since the 1990s, distributed database systems have entered the commercialization and application stage, and the traditional relational database products have been developed into distributed database products centered on computer networks and multi-tasking operating systems. The traditional relational database products are developed into computer network and multi-tasking operating system as the core of the distributed database products, while the distributed database gradually develops to the client/server model. Distributed database management system has a more flexible architecture, fast response time for local applications, good scalability and easy-to-integrate existing systems, widely used.

3 SIMPLIFIED MANAGEMENT OF DATABASES

Currently, data management is widely used in the fields of data analysis, data modeling, data warehousing, data security, data mining, and strategic data architecture.

For all the fields involved in data management, in an era of information technology, the need to store more and more redundant information, more and more data need to be stored, how to store all of these data integrity is the current problem faced by the data management. For the field of data analysis, the timely realization of real-time processing of data is also a problem to be solved in the field of data management, with the development of the network, the data real-time update speed for part of the data, once the data is not processed in a timely manner, the data is likely to lose its value and utility. In the field of data security, further improving the security of the database to protect the accuracy of the data is also a problem in the field of data management.

At present, using the view technology and indexing technology can improve the efficiency of data management and simplify the management.

In order to improve data security further and simplify the user's operation of the data, the view technology was added to MySQL 5.0.1. A view is a virtual view that defines a special kind of relational table, which is not stored in the database but can still be queried. The view's rows and columns are dynamically generated when the view is referenced, and the rows and columns are free to define the table referenced by the view's query. The view only provides query operations, and can not modify the data itself, and the user is not allowed to access the underlying relational tables and do not need to know the underlying relational table relationship structure can be queried. This not only protects data security and further improves the security of data storage but also simplifies the use of user queries. Views also support nested use to simplify the query further. In short, the view not only improves data security and data query flexibility but also facilitates user operations, user-friendly combination of different ways to operate the data, and gives the data a certain degree of independence.

In addition, in the case of rapid query of specific data, under the commonly used indexing techniques. Indexing technology was born in the relational database. The index is a separate, physical number of database tables in one or more columns of the value of the sort of storage structure. It is a table in one or more columns of the value of the collection and the corresponding to point to the table in the physical identification of these values in the list of logical pointers to the data page. index according to the storage method of classification, can be divided into two types: focused index and non-focused index. From a logical point of view indexes are divided into five categories: general indexes, unique indexes, primary key indexes, spatial indexes and full-text indexes. In practice, indexes are divided into single-column indexes and multi-column indexes.

Indexing technology greatly improves the data query speed, accelerates the connection between tables, ensures the uniqueness of each row of data in the database table, improves the query efficiency, and facilitates the query. In addition, index searches will not traverse all the information, which further improves the security of the database.

For database designers, choosing the right index is critical. Suitable indexes can improve the performance of the database management system and greatly increase its efficiency. Indexes can be optimized based on selection, optimized for maintenance, etc., and they can also be optimized using a query optimizer.

4 SOPTIMIZATION STRATEGY

4.1 Concurrency Optimization

It is common in database systems, especially in networked environments, for multiple users or processes to read and write data simultaneously, including queries or updates. If left unchecked, they may affect each other. The concept of database transactions was born. Transactions in databases have the following properties, i.e., A (atomicity), C (consistency) I (isolation) D (durability). Databases set transaction isolation levels to ensure serializability, so that operations appear to be executed serially. Isolation levels are essentially locks placed on database elements to prevent two functions from accessing them at the same time. The four isolation levels are: read uncommitted, read committed, repeatable read, serializable, which can be serialized less efficient. In order to ensure data persistence type of successful commit transaction data will be stored, unsuccessful commit will be re-executed. The database implements these features through transaction logging. This feature protects data persistence and, at the same time, allows the database to handle concurrency.

A typical usage scenario is when an airline provides a customer with an interface to select a seat for the flight he is traveling on, and when the customer clicks on an available seat, he books the seat. However, it is possible that not only one customer has booked the same seat on the same flight. At the same time, another customer may also see the seat, and if the seat is still shown as empty, two customers may buy the same seat. If the two calls to the seat selection operation are executed serially, then the error mentioned above will not occur. One customer sees that the seat is empty and makes a reservation, then the other customer's call is executed, and the seat that was just reserved is not used as one of his choices.

4.2 Storage Optimization

Traditional database when the amount of data to be stored is too large, will choose to expand vertically. Both storage is not enough to add the disk, computing is not enough to add the CPU, memory is not enough to add memory, add to the back to add can not be on the mainframe of the small machine. A distributed database better solves this problem. Due to the large amount of data storage, in order to efficiently save all the data, distributed databases use the idea of horizontal scaling to spread the data to different machines according to a certain slicing strategy, separate data management. Therefore, storage optimization is mostly seen in distributed databases.

Such databases do store more data better and more efficiently, but they also have some problems. First of all, distributed databases need to plan ahead of time to slice the rules; once the rules are set, it is difficult to move, and there are expansion difficulties. Secondly, distributed database data migration is more difficult. Finally, distributed database operation and maintenance costs are high. In general, there is still a lot of room for the database to be optimized in terms of storage.

Distributed databases can be roughly categorized into two types according to different architectural styles. The first is from the traditional library and performed into, plus additional scheduling nodes to achieve slice routing global clock to achieve distributed transactions. It basically constitutes a distributed database, but each data node is still a single database, through the master-slave replication to achieve high-availability, we call this type of PG-XC style. The representative works of this category are ZTE's GoldenDB, Huawei's GaussDB, and Tencent's TDSQL. the second is that the whole system is completely reconfigured. Each component in the database adopts distributed design. The underlying storage mostly adopts the key-value (KV) system, and at the same time, introduces the majority election algorithm to realize multi-copy synchronization, and the storage, computation, and scheduling are completely separated, known as the NewSQL style. NewSQL style. The representative works of this category are foreign CockroachDB, YugabyteDB, PingCAP's TiDB, Ant's OceanBase.

4.3 Security Optimization

Databases are protected by setting different permissions and managing them to achieve data security, while allowing for read-write separation and reducing pressure on the database. Whenever a database object is created, it is assigned an owner, who usually executes the create statement. For most types of objects, the initial state is that only the owner (or superuser) can modify or delete the object. To allow another role or user to use it, permissions must be set for that user. There are twelve types of permissions in a database, SELECT, INSERT, UPDATE, DELETE, TRUNCATE, REFERENCES, TRIGGER, CREATE, CONNECT, TEMPORARY, EXECUTE, and USAGE. the highest privileged person assigns permissions to a user with the GRANT statement. The highest authority assigns privileges to a user with the GRANT statement and withdraws them with the REVOKE statement., and can create a role for a group of users with the same privileges to facilitate management.

Permission management is used in many scenarios, for example, each sales record can be seen only by the sales billed for this order, the customer and the manager of this order. No one else is authorized to access this data.

In addition to the view technology already described earlier can also be a good way to improve database security.

5 CONCLUSIONS

This paper mainly summarizes and explains the optimization direction of the database. This paper successively describes the database optimization techniques in terms of querying i.e. view indexing techniques, concurrency i.e., transaction isolation level, storage i.e. distributed database techniques, and security i.e. database privilege assignments and settings and view techniques. This paper concludes that many optimization techniques and ideas are available in the database and data management field. The research in this paper helps in facilitating the understanding of the optimizations that are available in databases today so that it can better assist the researchers to continue with the subsequent optimization of databases. Nowadays, more emerging technologies are flourishing, and combining them with database management systems may further improve database management systems. For example, with the development of artificial intelligence, the database may be able to combine with artificial intelligence to conduct more testing queries and improve the automation of the database management system. However, the existing optimization is still insufficient, and it is necessary to update and upgrade the existing optimization technology further. For example, although the distributed database to a certain extent to solve the problem of large amounts of data storage, it still with its own limitations, the need for further optimization of the technology to improve. In summary, the database has been gradually improved but still faces problems and needs to be further optimized.

REFERENCES

Ammar, A. B, 2016, Query optimization techniques in graph Databases. arXiv preprint arXiv:1609.01893.

Hellerstein, J. M, 1998, Optimization techniques for queries with expensive methods. ACM Transactions on Database Systems (TODS), 23(2), 113-157.

Codd, E. F, 1970, A relational model of data for large shared data banks. Communications of the ACM, 13(6), 377-387.

Ceri, S., Pernici, B., & Wiederhold, G, 1987, Distributed database design methodologies. Proceedings of the IEEE, 75(5), 533-546.

Özsu, M. T., & Valduriez, P, 1999, Principles of distributed database systems (Vol. 2), Englewood Cliffs: Prentice Hall.

Ullman, J. D, 2007, A first course in database systems. Pearson Education India.

Garcia-Molina, H., Ullman, J. D., & Widom, J, 2000, Database system implementation (Vol. 672). Upper Saddle River: Prentice Hall.

Hevner, A. R., & Yao, S. B, 1979, Query processing in distributed database system. IEEE Transactions on Software Engineering, (3), 177-187.

Warren, J., & Marz, N, 2015, Big Data: Principles and best practices of scalable realtime data systems. Simon and Schuster.

Widom, J., & Ceri, S. (Eds.), 1995, Active database systems: Triggers and rules for advanced database processing. Morgan Kaufmann.