Application of distributed database in management information system

Chenyi Liu

Abstract:

This article discusses the application methods of distributed databases in management information systems. First, the two main types of structured and isomorphic databases are analyzed. Then, it elaborates on the application technology of distributed databases from five aspects: structural design, data transmission, security strategy, function implementation, and data synchronization maintenance, and proposes the selection of network topology mode, data segmentation strategy, and optimized transmission algorithm according to business needs. , establish security protection mechanisms, and support OLTP and OLAP processing methods. Finally, distributed databases provide essential support for management information systems by horizontally expanding storage and computing capabilities but also bring technical challenges.

Keywords: distributed database, management information system, structural design, data transmission, security strategy

1. Distributed database type

Distributed databases can manage distributed databases in information systems, including "homogeneously distributed databases" and "heterogeneous distributed databases." Each node of this distributed database has independent CPU, memory, disk, and other resources, and the nodes communicate through the network. Typical heterogeneous distributed database structures include:

· Client-server structure.

· Multi-layer client-server structure.

· Peer-to-peer service structure.

In the client-server structure, the server node is used to store the database, and the client is used for interaction; in the multi-layer client-server structure, the middle-tier server is introduced for elastic expansion and load balancing; the peer-to-peer service structure is based on the centerless Organized in a centralized manner, each node can act as a client and a server.

Homogeneous distributed database: This type manages data by sharing disk storage among multiple processing nodes. The basic idea is to provide larger storage space and more robust data processing capabilities by integrating more processing capabilities into the back-end storage system. Typical isomorphic distributed database structures include centralized, master-slave, and peer-to-peer structures. In the centralized structure, all data is stored in the central data storage; the master-slave structure will designate a controller node for reading and writing and other agent nodes for backup and disaster recovery; the peer-to-peer structure will use multiple equally distributed data among nodes. Distributed databases have broad application prospects in information systems, can provide scalability and high availability, and are suitable for big data management.

2. Application methods of distributed databases in management information systems

2.1. Structural design

An essential task for management information systems to utilize distributed databases is to design their network topology. Structural design must comprehensively consider system processing performance, reliability, security, and scalability requirements. When creating the structure, choose a master-slave or peer-to-peer architecture based on business needs. For example, a core business system that requires high data consistency can choose a master-slave architecture. In contrast, a transaction processing system that requires high response speed is more suitable for a peer-to-peer architecture. A reasonable number of different nodes should be set, and geographical distribution strategies should be considered. Too many nodes will increase the system complexity, while too few nodes will limit the scalability. Proper geographical distribution can improve disaster recovery capabilities. It is necessary to design flexible middleware to achieve efficient data transmission and transaction processing between nodes. Commonly used middleware includes distributed file systems, distributed database middleware, etc. Data fragmentation is the key to structural design. A reasonable sharding strategy is the basis for ensuring system scalability. Common sharding strategies include range sharding, hash sharding, list sharding, etc. The optimal strategy must be selected based on system characteristics. Make full use of caching technology to improve system performance. Standard methods include setting up caches on database nodes with frequent business queries or building independent cache server clusters to implement distributed caching.

2.2.Data transmission

Choose an efficient data transfer protocol. Commonly used distributed database network communication protocols include TCP/IP, UDP, RDSP, etc. Depending on the type of business, you can choose TCP/IP with high transmission reliability or UDP with lower latency. The protocol can also be customized to optimize specific business scenarios. Optimize database read operations and reduce network overhead. You can use stored procedures to process on the database server side to reduce the amount of query data, use indexes and partitions to improve query efficiency, and set up the cache to reduce disk IO. These methods can effectively reduce data transmission. Optimize database write operations and improve efficiency. Standard optimization methods include Write-Ahead Logging, which only transmits logs to improve writing speed; group commit and merge write operations to reduce network interaction; and asynchronous writing to hide data transmission time. Write scalability can also be improved through shard replicas. Parallel data transfer enhances network utilization. The overall pass rate can be significantly improved for data transmission through multi-channel and multi-path transmission.

The use of RDMA technology in modern networks can achieve parallel communication that far exceeds the efficiency of TCP. Use data compression to reduce the amount of data transmitted—commonly used compression algorithms such as LZ77, LZ78, LZW, etc. The compression ratio can generally reach 2 to 5 times, but the CPU overhead needs to be weighed. Transmission techniques such as approximate query processing can also be used to avoid raw data transfer. Use incremental synchronization and differential synchronization algorithms. To avoid transmitting large amounts of duplicate data, this type of technology is often used for data replication in remote disaster recovery centers, which can significantly reduce network load.

2.3. Security policy

Establish strict access control mechanisms, including firewall filtering, user authentication, data permission control, etc. The key is to identify legitimate users and control their access scope. Commonly used technologies include RBAC role-based access control. Strengthen the encrypted transmission of network data and use encrypted transmission protocols such as SSL/TLS to ensure the confidentiality and integrity of data transmission between nodes. You can also use VPN, IPSec, etc., to encrypt at a lower network layer. Enhance the security protection of core database servers and key database nodes can be isolated using hardware firewalls. The operating system must be patched regularly for security use, and unnecessary software must be turned off to prevent the attack surface from expanding. Establish a complete database audit mechanism, record access logs, and issue early warnings for abnormal behaviors. Risky database query behaviors can be detected by combining AI and behavioral analysis technology for security use. Data backup and disaster recovery technology are important security measures to prevent data from being exposed to damage, deletion, and tampering risks. Develop a complete data backup plan and conduct regular drills. At the same time, an off-site disaster recovery center is established to avoid service interruption caused by local failures or attacks. Personnel security is also crucial. It is necessary to cultivate employees' data security awareness and strictly manage permissions and passwords. External service personnel must also review access and follow the principle of least privilege.

2.4.Function implementation

SQL and transaction functionality support. This is the cornerstone of building a management information system. It is necessary to ensure that SQL statements can be executed correctly in a distributed environment and support complete CRUD operations. It is also essential to ensure ACID transactions and high data consistency. Build online analytic processing (OLAP) analysis capabilities. This function can quickly extract aggregate indicators from extensive data sets through multi-dimensional data cubes and perform complex statistical analysis. To optimize multi-table join queries, support distributed aggregation, and data warehouse construction. Support online transaction processing (OLTP) transaction processing. Typical high-frequency transaction writing in management information systems requires low latency and high throughput. The key is ensuring performance through sharding, caching, and object storage technologies. Also, pay attention to distributed transactions and concurrency control. Big data processing capabilities are very important. The amount of data in modern management information systems is increasing, and it needs to be expanded horizontally to several petabytes to support complex machine learning and data mining algorithms and perform predictive analysis. The efficiency of small file access also needs to be optimized. The management system has many small logs, reports, temporary data, etc. Appropriate use of object storage can effectively reduce the read and write latency of this part of the data. For data visualization functions, distributed databases should provide rich BI tools to enable managers to understand various operational indicators through reports, charts, geographical information systems, and other means.

2.5. Data synchronization and system maintenance

Establish an efficient data replication mechanism. This is the basis for ensuring data consistency in distributed databases. Commonly used technologies include master-slave replication, statement-level replication, consensus replication, mirror replication, etc. Choosing the optimal mechanism that adapts to the business characteristics is necessary. Use incremental synchronization and differential synchronization algorithms. Minimize replication overhead by transferring only updated differential data. Over time, full resynchronization is performed periodically to avoid data drift. Synchronization and write concurrency issues also need to be dealt with. Build a disaster recovery system and backup database. Critical businesses must build a disaster recovery center in a remote location and maintain near-real-time data backup. Offsite centers can be quickly switched to respond to regional outages. The standby database also provides data access in maintenance scenarios such as database upgrades and system testing. It is necessary to plan a complete full backup and incremental backup plan and strictly implement it. At the same time, multiple media backups are required to prevent single points of failure. Data verification is also essential to ensure the correctness of recovery—real-time monitoring of database performance indicators, server operating status, network traffic, and other information. AODSIM issues abnormal warnings to help administrators quickly locate and resolve faults.

3. Conclusion

In the practice of management information systems, the design of the network topology of the database is the key. The appropriate model must be selected according to the business characteristics; the optimization of data transmission can significantly improve the system performance; the emphasis must be placed on data security, and strict access control and encrypted transmission must be established. Supporting complex query analysis and big data applications in functional implementation is also essential. In addition, real-time data synchronization and automatic disaster recovery switching are also necessary to ensure the regular operation of the system. Distributed databases provide critical support for large-scale management information systems but bring specific challenges. It is essential to carefully design the topology according to particular application scenarios, choose appropriate technical methods, and ensure high-standard security policies and maintenance mechanisms to maximize the advantages of distributed databases. As technology advances, distributed databases will play a key role in more management information system projects.

Reference

Zhao Sijia. Application of distributed databases in management information systems [J]. Wireless Internet Technology, 2021, 18(19):2.DOI:10.3969/j.issn.1672-6944.2021.19.015.

Liu Feifei. Application of distributed databases in management information systems [J]. Electronic Technology and Software Engineering, 2019(17):2.