

# The Distributed Database Design Solution for Internet Application Platform

Xiaoyan Wang, Guojian Cheng

School of Medicine, Xi'an Peihua University, Xi'an, China

1350529199@qq.com

**Keywords:** Distributed databases; Centralized databases; Shared database clusters

**Abstract.** Distributed database can provide the more powerful concurrent processing ability, the better resource extensible and architecture flexibility, so it has been widely used in commercial fields recently. According to the deep research on distributed database and comparative analysis of multiple deployment solutions, this paper presents a better distributed database design solution for Internet Application platform.

## 1 Introduction

Distributed database systems have been widely used in commercial fields in recent years. Compared with the centralized database, the advantage is that it provides more powerful concurrent processing capabilities, better resource scalability and architectural flexibility, but the data consistency is slightly weaker than the centralized database. There are four mainstream distributed database architectures: Multi-active shared database clusters, massively parallel processing databases (MPP), distributed data file processing systems (typical products Hadoop), and non-shared database clusters based on data middleware scheduling management[1]. Among them, multi-active shared database clusters and non-shared database clusters based on data middleware scheduling management are more suitable for online transaction applications. The other two architectures are mainly applicable to the field of big data analysis. This article through the comparative analysis of a variety of programs, designed a large-scale Internet application platform based on the combination of multiple live shared database cluster and non-shared database cluster distributed database architecture.

## 2 Technical Analysis

The multi-active shared database cluster is mainly implemented through Oracle RAC and DB2 pureScale, but it only implements distributed architecture at the level of computing resources, and does not really realize distributed design at the logical structure level of data. Non-shared database clusters based on data middleware scheduling management mainly use distributed clusters of MyCAT middleware and MySQL database. This architecture enables data to be vertically and horizontally segmented flexibly. It is a truly distributed database architecture.

In the case of domestic Internet application platforms, large-scale Internet companies tend to use MySQL distributed database solutions because of considerations such as reducing costs and autonomous and controllable technological development. Financial companies take into account the relatively large technical risks and limited technical capabilities of open source database products, and they prefer the distributed database solutions of enterprise-level commercial applications (Oracle, DB2, etc.). However, some financial institutions are gradually trying to use open source technologies and products, adopt a gradual and robust strategy, first used in the peripheral applications, and gradually promote to important applications as the open source technology and product maturity are improved and the capabilities of their own technical teams are improved.

DB2 pureScale technology has only matured and commercialized in recent years. There are few use cases and it is easy to encounter first-time failures in use. It needs to be tested by the market and

time. Oracle RAC technology has been introduced for more than ten years and has been widely recognized by the industry. There are many cases in the financial field and the technology maturity is high. The flexibility and scalability of the MySQL distributed database architecture are good, and the cost is economically plentiful. In the case of the ultra-large-scale Internet application platform, the actual test of complex transactions and ultra-high intensity concurrent transaction scenarios has been obtained.

### 3 Application Scenario Analysis

The application scenarios of the Internet application platform are mainly divided into three categories: front-end interactive application data, mid-end transaction application data, back-end accounting and platform management and control data. The front-end interactive application data mainly includes online banking, merchants, marketing campaigns, transaction push, etc. This type of data does not require high performance, and does not require the use of sub-database sub-table design. Centralized database design can reduce technical complexity; mid-end transaction applications, the types of data mainly include accounts, payments, products, customers, etc. These types of data have extremely high requirements for flexibility and responsiveness, and the design of sub-distribution tables can better meet the requirements for rapid deployment and iteration of application products and services. Can it greatly improve the flexibility and scalability of data splitting or integrating? The back-end accounting and management and control data mainly include accounting, platform management, and platform monitoring. Accounting data requires high data consistency, while platform management and monitoring data are small in size, and it is not necessary to use a separate library design. Therefore, a centralized database structure is more suitable.

### 4 Design Goals

In response to the relevant characteristics of the Internet application platform, the following goals are proposed for the database system design[2]:

(1) **High availability:** The Internet application platform database system should have a good high availability design at the logical and physical levels to ensure that the device hardware and data logical copies have multiple redundancy.

(2) **High concurrent processing capacity:** The Internet application platform database system must not only have the processing power of normal day trading, but also be able to smoothly cope with the impact of high concurrent transaction volume in special time or spike scenarios.

(3) **Scalability:** The Internet application platform database system can adapt well to the development trend of business transaction volume and has powerful scalability in various performance and capacity resources.

(4) **Flexibility:** For rapid iteration and change of application system, the Internet application platform database system should have strong deployment flexibility. Can better adapt to the changing needs of data splitting and integration.

(5) **Business continuity:** The Internet application platform database system should have strong disaster recovery capabilities, be able to restore system service status as soon as possible after human and natural disasters occur, reduce the amount of data loss and scope of impact as much as possible, and at the same time require data security. Key indicators.

### 5 Design Plan

In order to meet the design goals of the Internet application platform database system, a flexible and reliable distributed database technology solution is proposed, which is a distributed database system design scheme based on the hybrid architecture of Oracle RAC and MySQL. Its specific software and hardware design solutions include:

With two types of hardware-based server types, the servers deployed with Oracle RAC clusters use high-end 4-way X86 rack servers with a similar structure to that of the A solution; the server that

deploys the MySQL database uses midrange 2-way X86 rack servers. Using MySQL master-slave replication technology to achieve high-availability clusters[3], while each cluster is equipped with a quasi-synchronous replication read-only database, through the read-write separation mode to improve performance load.

Oracle RAC cluster A deploys corporate e-banking, merchants, marketing campaigns, transaction push, and other databases; RAC cluster B deploys databases such as accounting, platform management, and monitoring; and two sets of MySQL high-availability clusters are deployed to deploy databases of accounts, payments, products, and customers.

As business growth and performance requirements continue to increase, the database for back-end transaction banking applications can be split horizontally at multiple nodes to implement the split deployment of data modules and improve the overall performance of the database. The internal table of the database can improve the use performance and capacity of the data structure through fragmentation design.

There are two scenarios for storing a database. Oracle RAC database cluster data files are stored in an external shared storage array; MySQL database is a built-in hard disk (SSD hard disk or flash memory card to store data files in the database server, to meet the high read and write performance requirements of the database), and use RAID10 to carry out data Protection and performance increase. The storage solution with built-in hard disk is not subject to the sharing capacity and space constraints of shared storage arrays and SAN resources, and can more quickly and flexibly expand database server nodes.

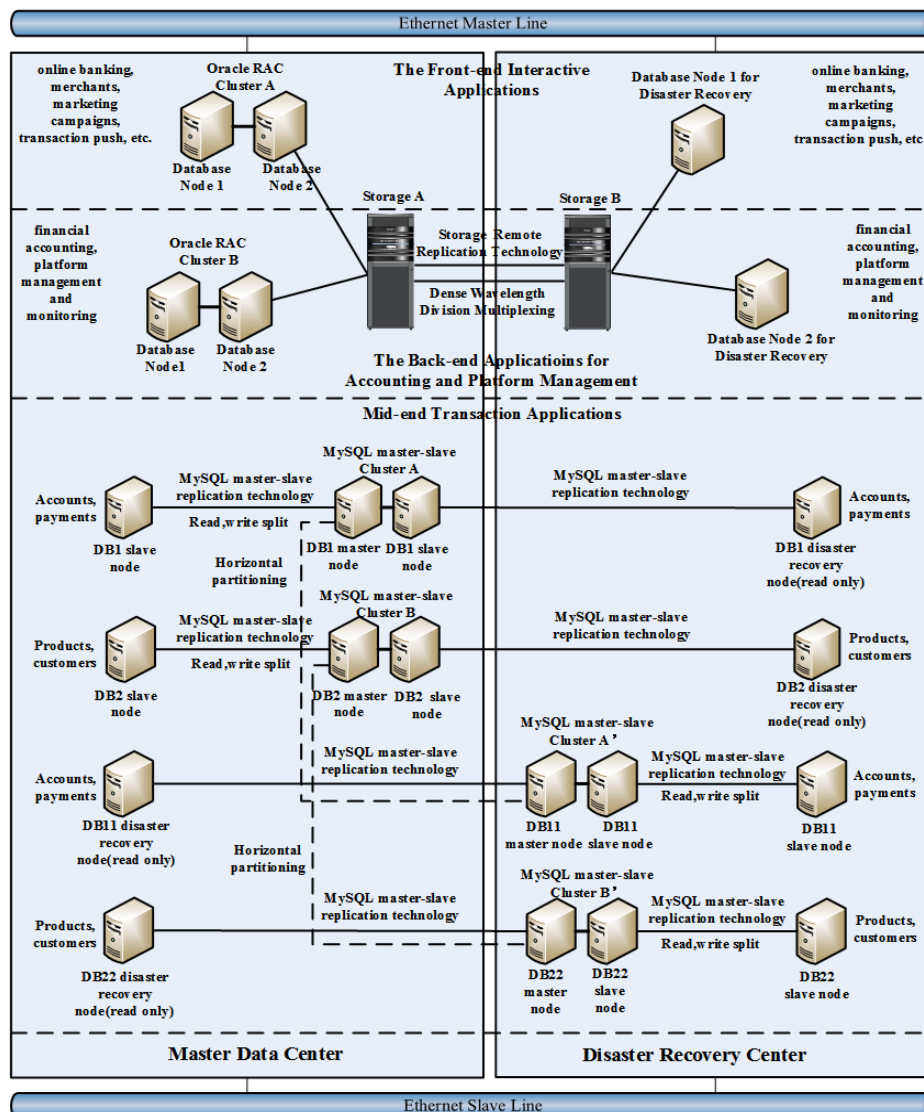


Figure 1. Design structure diagram

In terms of disaster recovery, Oracle RAC clusters use storage underlying remote replication technology to achieve disaster recovery. MySQL distributed database master-slave replication technology cannot be real-time synchronous replication, cannot guarantee the loss of zero data, so commonly used disaster recovery strategy is to minimize the data module, scattered, and choose multi-site deployment (three and more). This strategy can greatly reduce the dependence of the MySQL database on the disaster-tolerant environment. The disaster-tolerant architecture has become a distributed and multi-active architecture. It strictly controls the scope of impact of faults and disasters on data and reduces the amount of data loss.

Therefore, the design scheme is to horizontally split the transactional application database into two data units, each of which contains associated accounts, payments, products, and customer information. The two data units are deployed in the main production center and the disaster recovery center in the same city, and a disaster recovery database environment is established at the opposite end through the database master-slave asynchronous replication technology ( $RPO \gg 0$ ); if the network resources have limitations, the limitation is not enough to achieve cross-site Deployment can deploy all MySQL high availability clusters in the main production center. The city's disaster recovery center only stores asynchronous replication database disaster recovery environments. The disadvantage is that large amounts of data will be lost. The future needs to be improved through the computer room environment and network resources. Gradually realize a distributed deployment architecture that spans more sites. The system design scheme is shown in Figure 1.

## 6 Conclusions

In the face of the widespread adoption of open source database technology for Internet applications, the project construction of enterprise-level Internet application platforms must focus on the safe and stable operation of related technology products, and more importantly, objectively to recognize this technology trend and its own good development background. The rationality and inevitability reflected below. Therefore, comprehensive consideration of technical development, program feasibility, application adaptability, cost investment, operation and maintenance management and self-development and many other factors eventually led to the selection of an Internet application platform database based on Oracle RAC active-active cluster and MySQL distributed cluster. System design plan. The construction of the database system of the Internet application platform is a very complex and huge project. In addition to the most critical architectural issues, there are still many technical details that require in-depth consideration. Such as MySQL database fragmentation program, the preparation of distributed SQL statement specification, distributed database system data migration and operation and maintenance management system. The purpose of this paper is to discuss the design and analysis of the database system architecture of the Internet application platform. Other technical issues need to be further studied and considered in the next step.

## References

- [1] Zhou Yingxi. Research and Implementation of Common Shared-nothing Database Cluster[J]. Computer Engineering and Applications, 2007 43(32).
- [2] Chen Zhenghang. Research and Application of Distributed Database System[D]. Southwest Jiaotong University, 2014.
- [3] Chen Guanhua. Research on high availability MySQL cluster based on message middleware[D]. Huazhong University of Science and Technology, 2007.
- [4] Hu Jinzhu, Tan Zhipeng. Object Management Strategies in Distributed Object Database System[J]. Mini-micro Systems, 2001(01).
- [5] Dai Yirong. Research and application of data synchronization mechanism based on distributed database[D]. Wuhan University of Technology, 2010.
- [6] Su Yanqiang. Research on Oracle Distributed Database and Its Application[J]. Computer Applications and Software, 2004.
- [7] Chen Yuyu etc. Research on Data Consistency Maintenance Method in Distributed Database System[J]. Journal of National University of Defense Technology, 2002.