

## 摘 要

数据库、操作系统和编译器并称为三大系统，可以说是整个计算机软件的基石。其中数据库更靠近应用层，是很多业务的支撑。这一领域经过了几十年的发展，不断的有新的进展。从最开始的层次数据库和关系数据库，到近几年火热的Nosql数据库，再到最近以Google Spanner和F1为代表的NewSql数据库。

在互联网时代，海量数据的存储与访问成为系统设计与使用的瓶颈问题，对于海量数据处理，按照使用场景，主要分为两种类型：联机事务处理（OLTP）和联机分析处理（OLAP）。

关系型数据库，是建立在关系模型基础上的数据库，其借助于集合代数等数学概念和方法来处理数据库中的数据。其中以MySQL最为流行，MySQL是一个开源的关系型数据库，其优势在于源代码开放，任何企业和个人都可以根据自身的需求对mysql的源码做修改。

NoSQL 数据库，全称为 Not Only SQL，意思就是适用关系型数据库的时候就使用关系型数据库，不适用的时候也没有必要非使用关系型数据库不可，可以考虑使用更加合适的数据存储。

Oracle，mysql 等传统的关系数据库非常成熟并且已大规模商用，为什么还要用 NoSQL 数据库呢？主要是由于随着互联网发展，数据量越来越大，对性能要求越来越高，传统数据库存在着先天性的缺陷，即单机（单库）性能瓶颈，并且扩展困难。这样既有单机单库瓶颈，却又扩展困难，自然无法满足日益增长的海量数据存储及其性能要求，所以才出现了各种不同的NoSQL产品。

虽然在云计算时代，传统数据库存在着先天性的弊端，但是 NoSQL 数据库又无法将其替代，NoSQL 只能作为传统数据的补充而不能将其替代，所以规避传统数据库的缺点是目前大数据时代必须要解决的问题。

为了解决现在MySQL等关系型数据库的这些问题，本文描述了如何设计和实现一个兼容MySQL协议的分布式数据库。他能自动的发现分布式数据库集群中的节点，自动的分配数据，支持海量的数据存储。同时考虑到如今数据库的安全变得越来越重要，不时的发生数据库管理员或者其他攻击者恶意更改数据库中的数据的情况，所以本文开发的数据库从底层加入数据库的审计功能，监控所有数据库的更改情况，而且可以以图形化的方式显示监控审计结果。

本论文涉及的主要工作如下：

基于java语言实现的兼容mysql通信协议的数据库；

实现了数据库集群，实现了数据库的高可用性，可扩展性，负载均衡等特性；

从数据库底层考虑了数据库安全，加入了数据审计等特性。

**关键词：** 分布式数据库，mysql，安全审计，OLTP，NOSQL

## ABSTRACT

Database, operating system and compiler and called the three systems, can be said that the cornerstone of the entire computer software. Which is closer to the application layer database, is a lot of business support. This field after decades of development, There are new developments. From the beginning of the hierarchy database and relational database, to the recent hot Nosql database, and then to the recent Google Spanner and F1 as the representative of the NewSql database.

In the Internet age, the storage and access of massive data becomes the bottleneck of system design and use. For mass data processing, King, divided into two types: online transaction processing (OLTP) and online analytical processing (OLAP).

Relational database is based on the relational model of the database, which by means of aggregation algebra and other mathematical concepts and methods to deal with the database data. Which is the most popular mysql, mysql is an open source relational database, the advantage lies in the open source code, any business and individuals can according to their own needs to modify the source code mysql.

NoSQL database, called Not Only SQL, meaning that when the relational database is used when the relational database, not applicable There is no need to use relational database is not necessary, you can consider the use of more appropriate data storage.

Oracle, mysql and other traditional relational database is very mature and has been large-scale commercial, why use NoSQL database? mainly With the development of the Internet, the amount of data is growing, the performance requirements are getting higher and higher, the traditional database of congenital defects, namely stand-alone (single Library) performance bottlenecks, and difficult to expand. This is a stand-alone single library bottleneck, but difficult to expand, naturally unable to meet the growing mass of data storage And its performance requirements, so there will be a variety of different NoSQL products.

Although in the cloud computing era, the traditional database there are congenital defects, but NoSQL database can not be replaced, NoSQL can only For the traditional data supplement can not be replaced, so to avoid the shortcomings of traditional databases is the current era of large data must be resolved.

In order to solve these problems, such as mysql and other relational database, this article describes how to design and implement a compatible mysql protocol distributed database. He can automatically find the distributed database cluster nodes, Automatically allocate data to support massive data storage. Taking into account the increasingly important security of the database, from time to time the occurrence of database administrators or other attackers malicious changes in the database data, So the database developed by the database from the bottom of the database to join the audit function.

The main work of this paper is as follows:

Based on the realization of java language compatible mysql communication protocol database;

The realization of the database cluster, to achieve the database of high availability, scalability, load balancing and other characteristics;

From the bottom of the database to consider the database security, joined the data audit and other features.

**Keywords:** mysql, java, mysql, java, mysql

## 目 录

|                        |          |
|------------------------|----------|
| <b>第一章 绪论</b>          | <b>1</b> |
| 1.1 研究工作的背景与意义         | 1        |
| 1.2 国内外研究历史与现状         | 1        |
| 1.3 本文的主要贡献与创新         | 1        |
| 1.4 本论文的结构安排           | 2        |
| <b>第二章 数据库系统的分类和架构</b> | <b>3</b> |
| 2.1 非关系型数据库            | 3        |
| 2.1.1 键值存储数据库          | 3        |
| 2.1.2 列存储数据库           | 3        |
| 2.1.3 面向文档数据库          | 3        |
| 2.1.4 图形数据库            | 3        |
| 2.2 关系型数据库             | 3        |
| 2.2.1 关系数据库重要理论        | 3        |
| 2.2.2 关系数据库架构          | 4        |
| 2.3 Mysql数据库体系结构       | 4        |
| 2.4 本章小结               | 4        |
| <b>第三章 分布式数据库概念和技术</b> | <b>5</b> |
| 3.1 分布式数据库的介绍          | 5        |
| 3.2 负载均衡               | 5        |
| 3.2.1 负载均衡的概念          | 5        |
| 3.2.2 负载均衡算法           | 5        |
| 3.2.3 一致性hash算法        | 5        |
| 3.3 数据分片技术             | 5        |
| 3.3.1 数据分片概述           | 5        |
| 3.3.2 数据库分片策略          | 6        |
| 3.4 数据库高可用技术           | 6        |
| 3.4.1 集群可靠性技术          | 6        |
| 3.4.2 数据复制             | 6        |
| 3.4.3 数据一致性            | 6        |
| 3.5 Hazelcast技术简介      | 6        |

|                                      |           |
|--------------------------------------|-----------|
| 3.6 本章小结 .....                       | 6         |
| <b>第四章 JAVA网络和多线程技术 .....</b>        | <b>7</b>  |
| 4.1 TCP/IP相关协议 .....                 | 7         |
| 4.1.1 TCP/ip协议概述 .....               | 7         |
| 4.1.2 TCP通信的连接和终止 .....              | 7         |
| 4.2 java流IO技术 .....                  | 7         |
| 4.3 Socket编程 .....                   | 7         |
| 4.4 NIO技术 .....                      | 7         |
| 4.5 线程池技术研究 .....                    | 8         |
| 4.6 Reactor模式 .....                  | 8         |
| 4.7 Netty简介 .....                    | 8         |
| 4.8 本章小结 .....                       | 8         |
| <b>第五章 Mysql通信协议的研究 .....</b>        | <b>9</b>  |
| 5.1 交互过程 .....                       | 9         |
| 5.1.1 握手认证阶段 .....                   | 9         |
| 5.1.2 命令执行阶段 .....                   | 9         |
| 5.2 协议基本类型 .....                     | 9         |
| 5.2.1 整型值 .....                      | 9         |
| 5.2.2 null结尾的字符串 .....               | 9         |
| 5.2.3 长度编码的字符串 .....                 | 9         |
| 5.2.4 二进制数据(长度编码) .....              | 10        |
| 5.3 mysql报文结构 .....                  | 10        |
| 5.3.1 报文头 .....                      | 10        |
| 5.3.2 报文数据 .....                     | 10        |
| 5.4 报文类型 .....                       | 10        |
| 5.4.1 登陆认证交互报文 .....                 | 10        |
| 5.4.2 客服端命令请求报文 .....                | 10        |
| 5.4.3 服务器响应报文 .....                  | 11        |
| 5.5 本章小结 .....                       | 11        |
| <b>第六章 分布式数据库JSQL总体设计目标和架构 .....</b> | <b>12</b> |
| 6.1 JSQL总体目标 .....                   | 12        |
| 6.2 JSQL的总体架构 .....                  | 12        |
| 6.2.1 数据库架构 .....                    | 12        |

|                                 |           |
|---------------------------------|-----------|
| 6.2.2 集群架构 .....                | 12        |
| 6.2.3 数据审计架构 .....              | 14        |
| 6.2.3.1 elasticsearch简介 .....   | 14        |
| 6.2.3.2 grafana简介 .....         | 14        |
| 6.2.3.3 审计的架构 .....             | 14        |
| 6.3 JSQL特性 .....                | 14        |
| 6.4 本章小结 .....                  | 14        |
| <b>第七章 JSQL各模块详细设计和实现 .....</b> | <b>15</b> |
| 7.1 数据库设计和实现 .....              | 15        |
| 7.1.1 前端mysql协议的实现 .....        | 15        |
| 7.1.2 sql解析模块 .....             | 15        |
| 7.1.3 存储引擎模块 .....              | 15        |
| 7.2 集群架构设计和实现 .....             | 15        |
| 7.2.1 通信模块的实现 .....             | 15        |
| 7.2.2 数据复制的实现 .....             | 15        |
| 7.2.3 负载均衡的设计和实现 .....          | 16        |
| 7.3 数据审计架构的设计和实现 .....          | 16        |
| 7.3.1 审计数据库设计 .....             | 16        |
| 7.3.2 审计管理器实现 .....             | 16        |
| 7.3.3 审计可视化模块的实现 .....          | 16        |
| 7.4 本章小结 .....                  | 16        |
| <b>第八章 数据库JSQL测试 .....</b>      | <b>17</b> |
| 8.1 测试环境 .....                  | 17        |
| 8.2 功能和性能测试 .....               | 17        |
| 8.2.1 基本功能测试 .....              | 17        |
| 8.2.2 数据库集群测试 .....             | 17        |
| 8.3 数据库审计和可视化测试 .....           | 17        |
| 8.4 本章小结 .....                  | 17        |
| <b>第九章 全文总结与展望 .....</b>        | <b>18</b> |
| 9.1 全文总结 .....                  | 18        |
| 9.2 后续工作展望 .....                | 18        |
| <b>致 谢 .....</b>                | <b>19</b> |
| <b>参考文献 .....</b>               | <b>20</b> |





## 第一章 绪论

在互联网时代，海量数据的存储与访问成为系统设计与使用的瓶颈问题，对于海量数据处理，按照使用场景，主要分为两种类型：联机事务处理（OLTP）和联机分析处理（OLAP）。

联机事务处理（OLTP）也称为面向交易的处理系统，其基本特征是原始数据可以立即传送到计算中心进行处理，并在很短的时间内给出处理结果。

联机分析处理（OLAP）是指通过多维的方式对数据进行分析、查询和报表，可以同数据挖掘工具、统计分析工具配合使用，增强决策分析功能。

### 1.1 研究工作的背景与意义

web2.0的到来，存储海量数据的需求越来越大。

在互联网时代，海量数据的存储与访问成为系统设计与使用的瓶颈问题，对于海量数据处理，按照使用场景，主要分为两种类型：联机事务处理（OLTP）和联机分析处理（OLAP）。

联机事务处理（OLTP）也称为面向交易的处理系统，其基本特征是原始数据可以立即传送到计算中心进行处理，并在很短的时间内给出处理结果。

联机分析处理（OLAP）是指通过多维的方式对数据进行分析、查询和报表，可以同数据挖掘工具、统计分析工具配合使用，增强决策分析功能。

### 1.2 国内外研究历史与现状

google spanner和t1等系统。在互联网时代，海量数据的存储与访问成为系统设计与使用的瓶颈问题，对于海量数据处理，按照使用场景，主要分为两种类型：联机事务处理（OLTP）和联机分析处理（OLAP）。

联机事务处理（OLTP）也称为面向交易的处理系统，其基本特征是原始数据可以立即传送到计算中心进行处理，并在很短的时间内给出处理结果。

联机分析处理（OLAP）是指通过多维的方式对数据进行分析、查询和报表，可以同数据挖掘工具、统计分析工具配合使用，增强决策分析功能。

### 1.3 本文的主要贡献与创新

结合nosql的关系型数据库的特点，同时从数据库自身考虑数据库安全。 ...

## 1.4 本论文的结构安排

本论文后续各章结构如下：

第二章介绍了

第三章介绍了

第四章描述了审计子系统是如何实现，同时将涉及的技术方法作了介绍和分析。

第五章对审计管理器模块进行了分析和设计。首先提出该模块的概述，并结合这些目标对系统总体结构和重要的功能进行了详细的分析和设计，从而保证设计目标的实现。

第六章主要通过介绍审计子系统的具体应用，说明了审计子系统的实用价值和地位。

结论部分详细总结了本论文的研究工作，并针对存在的主要不足指出了改进的重点和今后的方向。

## 第二章 数据库系统的分类和架构

本章主要写了又那些数据库，每种数据库的特点。

...

### 2.1 非关系型数据库

非关系型数据库包括下面3个

...

#### 2.1.1 键值存储数据库

主要是内存数据库

...

#### 2.1.2 列存储数据库

比如hbase数据库

...

#### 2.1.3 面向文档数据库

主要的是mangdb数据库

...

#### 2.1.4 图形数据库

主要有orientdb

...

### 2.2 关系型数据库

关系数据库是主要的数据库

...

#### 2.2.1 关系数据库重要理论

主要有acid 事务是数据库的主要特点

...

### 2.2.2 关系数据库架构

sql解析

...

### 2.3 Mysql数据库体系结构

从上到下

...

### 2.4 本章小结

存储部分的数据库主要分为两大类：关系型数据库与 NoSQL 数据库。

...

## 第三章 分布式数据库概念和技术

分布式数据库技术的概述

...

### 3.1 分布式数据库的介绍

分布式数据库技术的概述

...

### 3.2 负载均衡

...

#### 3.2.1 负载均衡的概念

分布式数据库技术的概述

...

#### 3.2.2 负载均衡算法

论云

...

#### 3.2.3 一致性hash算法

分布式数据库技术的概述

...

### 3.3 数据分片技术

...

#### 3.3.1 数据分片概述

分布式数据库技术的概述

...

### **3.3.2 数据库分片策略**

分布式数据库技术的概述

...

## **3.4 数据库高可用技术**

分布式数据库技术的概述

...

### **3.4.1 集群可靠性技术**

分布式数据库技术的概述

...

### **3.4.2 数据复制**

分布式数据库技术的概述 ...

### **3.4.3 数据一致性**

分布式数据库技术的概述

...

## **3.5 Hazelcast技术简介**

分布式数据库技术的概述

## **3.6 本章小结**

分布式数据库技术的概述

## 第四章 JAVA网络和多线程技术

### 4.1 TCP/IP相关协议

...

#### 4.1.1 TCP/ip协议概述

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

#### 4.1.2 TCP通信的连接和终止

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

### 4.2 java流IO技术

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

### 4.3 Socket编程

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

### 4.4 NIO技术

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

## 4.5 线程池技术研究

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。 ...

## 4.6 Reactor模式

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

## 4.7 Netty简介

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

## 4.8 本章小结

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...



## 第五章 Mysql通信协议的研究

...

### 5.1 交互过程

...

#### 5.1.1 握手认证阶段

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

#### 5.1.2 命令执行阶段

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

### 5.2 协议基本类型

#### 5.2.1 整型值

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

#### 5.2.2 null结尾的字符串

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

#### 5.2.3 长度编码的字符串

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

#### **5.2.4 二进制数据(长度编码)**

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

### **5.3 mysql报文结构**

...

#### **5.3.1 报文头**

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

#### **5.3.2 报文数据**

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

### **5.4 报文类型**

...

#### **5.4.1 登陆认证交互报文**

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

#### **5.4.2 客服端命令请求报文**

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

### 5.4.3 服务器响应报文

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

### 5.5 本章小结

本章主要说了java的相关技术，因为jsql是基于java开发的数据库。java是一个跨平台的语言。

...

## 第六章 分布式数据库JSQL总体设计目标和架构

### 6.1 JSQL总体目标

jsql是一个可以替代mysql的分布式java数据库

...

### 6.2 JSQL的总体架构

下面是分布式环境下的部署图：

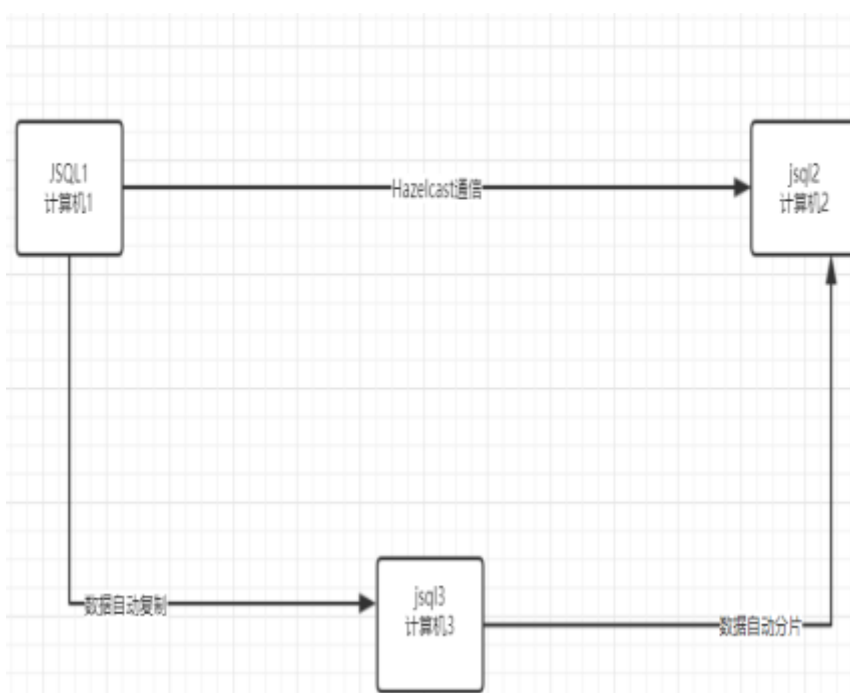


图 6-1 分布式环境下架构

图6-2是jsql的架构图

#### 6.2.1 数据库架构

数据库主要是sqk到粗存

...

#### 6.2.2 集群架构

主要用hazelcast

...

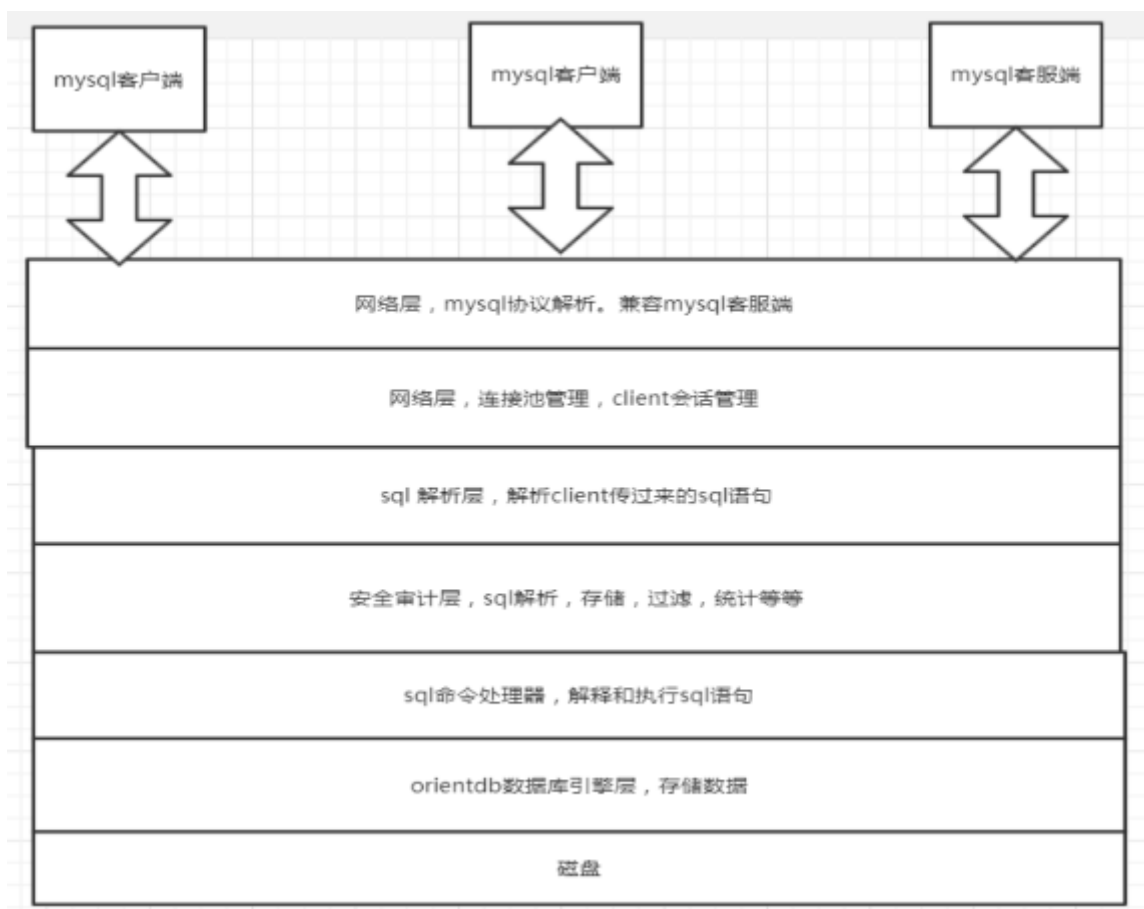


图 6-2 JSQL架构

### **6.2.3 数据审计架构**

数据库审计的

...

#### **6.2.3.1 elasticsearch简介**

...

#### **6.2.3.2 grafana简介**

...

#### **6.2.3.3 审计的架构**

...

### **6.3 JSQL特性**

分布式数据库

...

### **6.4 本章小结**

分布式数据库

...

## 第七章 JSQL各模块详细设计和实现

本文主要是

...

### 7.1 数据库设计和实现

主要分为3个部分

...

#### 7.1.1 前端mysql协议的实现

数据库主要是sqk到粗存

...

#### 7.1.2 sql解析模块

主要用hazelcast

...

#### 7.1.3 存储引擎模块

数据库审计的

...

### 7.2 集群架构设计和实现

主要用hazelcast

...

#### 7.2.1 通信模块的实现

数据库主要是sqk到粗存

...

#### 7.2.2 数据复制的实现

主要用hazelcast

...

### 7.2.3 负载均衡的设计和实现

数据库审计的

...

## 7.3 数据审计架构的设计和实现

...

### 7.3.1 审计数据库设计

数据库主要是sqk到粗存

...

### 7.3.2 审计管理器实现

主要用hazelcast

...

### 7.3.3 审计可视化模块的实现

数据库审计的

...

## 7.4 本章小结

分布式数据库

...



## 第八章 数据库JSQL测试

...

### 8.1 测试环境

...

### 8.2 功能和性能测试

...

#### 8.2.1 基本功能测试

.....

#### 8.2.2 数据库集群测试

.....

### 8.3 数据库审计和可视化测试

...

### 8.4 本章小结

...

## 第九章 全文总结与展望

...

### 9.1 全文总结

.....

### 9.2 后续工作展望

.....

## 致 谢

在攻读硕士学位期间，首先衷心感谢我的导师曹晟老师...

## 参考文献

- [1] 王浩刚, 聂在平. 基于mysql的分布式数据库的研究和实现[J]. 电子学报, 1999, 27(12):68–71
- [2] X. F. Liu, B. Z. Wang, W. Shao. A marching-on-in-order scheme for exact attenuation constant extraction of lossy transmission lines[C]. China-Japan Joint Microwave Conference Proceedings, Chengdu, 2006, 527–529
- [3] 竺可桢. 物理学[M]. 北京: 科学出版社, 1973, 56–60
- [4] 陈念永. 基于mysql的分布式数据库的研究和实现[D]. 成都: 电子科技大学, 2001, 50–60
- [5] 顾春. 基于mysql的分布式数据库的研究和实现[N]. 人民日报, 2012年3月31日
- [6] 冯西桥. 基于mysql的分布式数据库的研究和实现[R]. 北京: 清华大学核能技术设计研究院, 1997年6月25日
- [7] 肖珍新. 基于mysql的分布式数据库的研究和实现[P]. 中国, 实用新型专利, ZL201120085830.0, 2012年4月25日
- [8] 中华人民共和国国家技术监督局. GB3100-3102. 中华人民共和国国家标准—量与单位[S]. 北京: 中国标准出版社, 1994年11月1日
- [9] M. Clerc. Discrete particle swarm optimization: a fuzzy combinatorial box[EB/OL]. [http://clere.maurice.free.fr/pso/Fuzzy\\_Discrere\\_PSO/Fuzzy\\_DPSO.htm](http://clere.maurice.free.fr/pso/Fuzzy_Discrere_PSO/Fuzzy_DPSO.htm), July 16, 2010