

分布式 MiniSql 系统个人报告

课程名称 大规模信息系统构建技术导论

作业名称	个人模块报告
姓名	裴睿韬
学号	319016087
学院	计算机科学与技术学院
专业	软件工程
指导老师	鲁伟明
时间	2022-5-19

系统介绍

本次开发我们完成了一个基于底层数据库基于 MySQL，使用 zookeeper 集群。使用 java 进行开发，并用 maven 进行模块管理的分布式 MiniSql 项目。完成了对于简单的 SQL 语句执行，数据分布储存、副本管理、负载均衡、容错容灾等等功能。

项目简述

系统环境搭建

配置 maven (pom.xml)

挑选并测试如下 dependencies

1、Zookeeper 3.8.0

2、Curator 5.21

常用的 java zookeeper 框架，编写 ZookeeperTest.java 进行测试

3、log4j 1.2.17

apache 的 java log 输出库

4、junit 4.11

常用的 java test 组件

5、mysql-connector 8.0.23， 编写 MySQLTest.java 进行测试

java jdbc mysql 驱动

搭建 git 仓库

配置.gitignore 文件、Readme 文件、设置群成员共享等等

系统架构设计

```
├-.vscode
├-logs
├-sql
├-src
```

```

|   |main
|   |  |java
|   |   |com
|   |    |largescalesystem
|   |     |minisql
|   |      |client
|   |      |master
|   |      |regionserver
|   |      |socket
|   |      |zookeeper
|   |   |resources
|   |  |test
|   |   |java
|   |    |com
|   |     |largescalesystem
|   |      |minisql
|   |   |sql
|   |  |target
|   |   |classes
|   |    |com
|   |     |largescalesystem
|   |      |minisql
|   |       |client
|   |       |master
|   |       |regionserver
|   |       |socket
|   |       |zookeeper
|   |   |test-classes
|   |    |com
|   |     |largescalesystem
|   |      |minisql

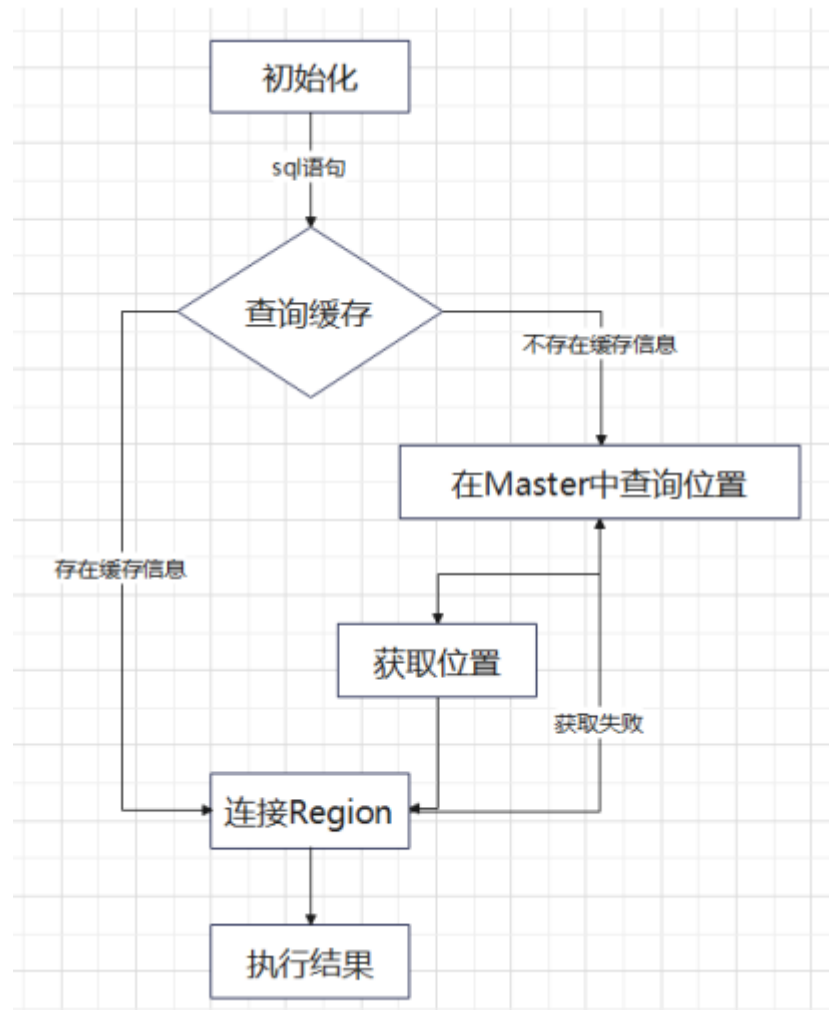
```

接口设计

详情请见 [github 文档](#)

[LargeScaleSystem/Document at feature/region-server · zwc233/LargeScaleSystem \(github.com\)](#)

Clint 流程图



个人工作

个人在本次项目中主要负责了 Client 端的编写与 Zookeeper 接口的封装，组织小组的成员进行会议与代码整合及测试，以及设计报告的总体设计部分与最终报告的撰写。整体工作量不大，在规定的时间内已完成个人负责的全部内容。

1. Zookeeper 接口封装

与赵文琛同学共同完善了 Zookeeper 相关功能调用接口的封装，在实现基本的增删改查功能的基础上增添了 listener，获取节点信息，获取子节点信息，创建客户端等功能。Listener 方法可以实现 ZNODE 状态发生改变时的实时广播，使得用户可以随时跟踪其状态。

2. Client 端编写

本次项目中我的主要工作为 Client 端的编写，Client 整体功能如下：与 Master 进行通信，将 sql 语句传输后从 Master 处获得反馈，依此定位到 region 坐标。然后与 region 进行通信，实现相关功能。整体开发思路如下所示：

- ① .由于与 Master 以及 Region 的通信整体模式一致，所以在 Client 类中实现了 getConnection 方法，利用 socket 实现通信，表明身份后发送 sql 语句，最后获取服务器端返回的字符串并将之 return。
- ② .由于副本管理的因素，服务器返回的字符串格式也有不同，对返回的字符串进行判断与处理，将返回的信息用 HashMap 储存并打印出来以便使用。
- ③ .采用多线程编写的方式，使得与 Master 保持通信的同时也可以与 Region 进行通信。

个人总结

本次大规模项目由我们小组的成员独立完成，看到成功的测试结果还是很有成就感，并对 zookeeper 以及分布式系统有了更进一步的了解。但是项目整体的开发进度略显落后，个人也因为其他课程大作业以及各类比赛的原因没能为小

组提供更多的帮助，容错容灾功能目前还不够完善。在答辩时会呈交一份更加完善的项目文件用于答辩。