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	赵文琛
学号	3190103179
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。

个人工作

个人在本次开发中承担了比较大的工作贡献（贡献详情也可查看git commit记录），主要开发以下模块

1、系统环境搭建、git环境搭建、基础工具类书写、系统架构、接口设计、并记录会议内容

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 8.023， 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)]

2、Region Server模块 设计开发

```
regionserver
    Client.java
    JdbcUtils.java
    RegionServer.java
    RegionServerToClientUtils.java
    ServerClient.java
    ServerMaster.java
    ServerThread.java
    SqlQuery.java
```

开发思路

备份管理

为每个表创建一个主表一个从表，主表和从表存在不同的RegionServer上，并且从表的名称以_slave结尾作为标识

代码文件详解

RegionServer入口，通过这个文件初始化和启动RegionServer，先连接MySQL和Zookeeper Client，并在Zookeeper集群上注册RegionServer信息。然后创建一个java 线程池来进行多线程socket处理，等待Master、Client、另一个RegionServer的连接

ServerThread，通过这个文件来处理socket多线程的Thread

socket连接成功后判断连接者的身份（Master、Client还是另一个RegionServer的连接），然后分别跳转到ServerMaster、ServerClient中的函数进行处理

ServerClient

主要处理正常的 创建、增删改查 语句

创建表：

在本机的MySQL数据库中创建对应表、并修改Zookeeper节点

删除表：

在本机的MySQL数据库中删除对应表、并修改Zookeeper节点

其他操作：

直接调用MySQL execute

ServerMaster

主要处理负载均衡和容错容灾部分，主要使用mysqldump直接远程dump目标数据库

负载均衡

接收到Master迁移命令后，迁移的RegionServer调用java runtime msyqldump命令，远程dump出某个表的创建插入语句等等，并在本地保存为sql，紧接着使用mysql导入指令，执行sql文件，将数据库同步到本地MySQL，然后调整zookeeper节点

然后被迁移的RegionServer删除表，并调整Zookeeper节点

容错容灾

当某个RegionServer宕机后，Master找到表的副本储存的RegionServer，并通知该RegionServer，将副本升级为主本。并通知另一个RegionServer，让之用mysqldump备份新的表。并修改Zookeeper。完成备份

测试模块

```
test
    AppTest.java
    LogTest.java
    MysqlTest.java
    RegionServerTest.java
    ServerMasterTest.java
    ZookeeperTest.java
```

3、工具类开发

1、JDBCUtils 在jdbc上做一层封装，方便使用

2、ZookeeperUtils 在Curator上做一层封装，方便使用

个人总结

本次大规模开发并没有完美完成要求的所有功能，究其原因为我们整体开发进度比较靠后。兼之我个人这个学期在网易实习，所以时间上非常紧迫。但截止2022/5/19。个人所负责的模块的完成度基本在90%以上了。并且剩下未完成部分，比较依靠于多人测试。在2022/5/27答辩之前我们将会将该项目的更加完善版本用于答辩。