浙江大学

大规模信息系统构建技术导论

分布式 MiniSQL 系统个人报告

2022 学年 春 学期

学号	3190105590
学生姓名	王晨璐
所在专业	软件工程
所在班级	软工 1902

2021 年 5 月 23 日

目　　录

1 引言

1.1 系统目标

本项目是《大规模信息系统构建技术导论》的课程项目，在大二春夏学期学习的《数据库系统》课程的基础上结合《大规模信息系统构建技术导论》所学知识实现的一个分布式关系型简易数据库系统。

该系统包含 ETCD 集群、客户端、主从节点等多个模块，可以实现对简单 SQL 语句的处理解析和分布式数据库的功能，并具有数据分区、负载均衡、副本管理、容错容灾等功能。

本系统使用 Go 语言开发，并使用 Github 进行版本管理和协作开发，由小组内的五名成员共同完成，每个人都有自己的突出贡献。

1.2 我的贡献

本项目的开发我主要负责其中主节点和客户端以及从节点的通讯架构，实现主节点对客户端请求的接收和处理、向从服务器发送请求、接收来自从服务器的请求结果、最后将结果返还给客户端的通讯流程。

2 系统设计与实现

2.1 系统总体架构

本系统的总体架构设计如下图所示：

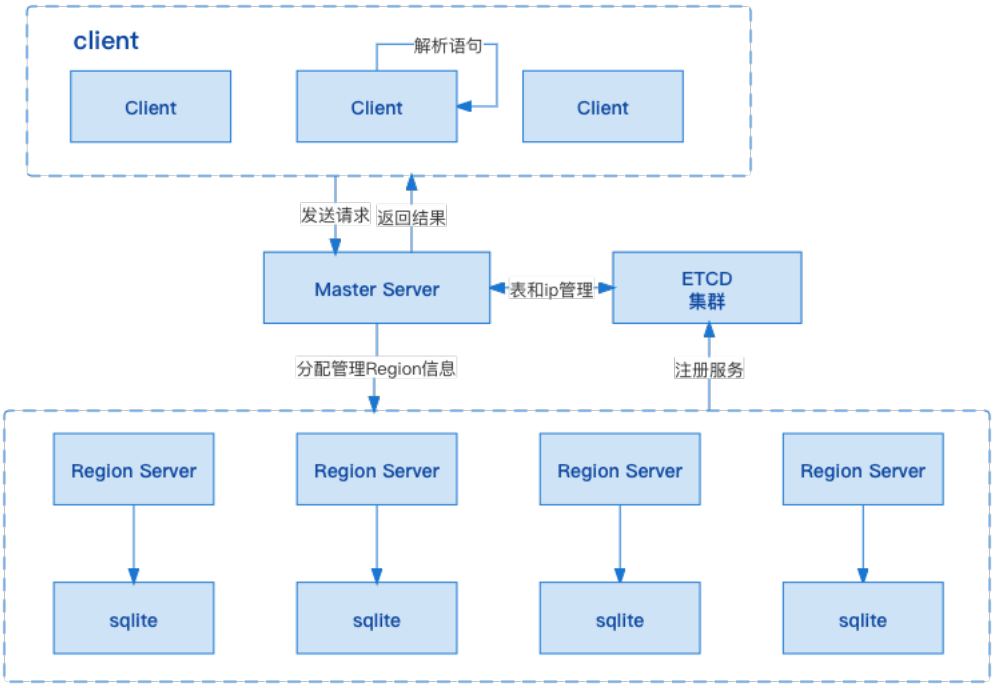


图 1 总体架构图

本系统将整个项目划分为三个模块，分别是 Client，Master Server 和 Region Server，分别对应分布式数据库系统的客户端、主节点和从节点，其中主节点作为通讯中心负责收发消息和任务分配，同时主节点和从节点通过 ETCD 集群，对数据表的信息进行统一的管理。

我主要负责的部分是 Master 中的通讯架构，接下来将对我负责的部分进行具体阐述。

3 我参与的核心功能模块

3.1 Socket 通信架构与通信协议

3.1.1 系统通信架构

本系统将 Master 作为通讯管理中心，通过 Socket 连接建立稳定可靠的通信与异步的 IO 流。Client 向 Master 请求服务，Master 接收消息后将任务分配给合适的 Region，并将从 Region 获取的数据信息返还给 Client。系统的总体通信架构如下图所示：

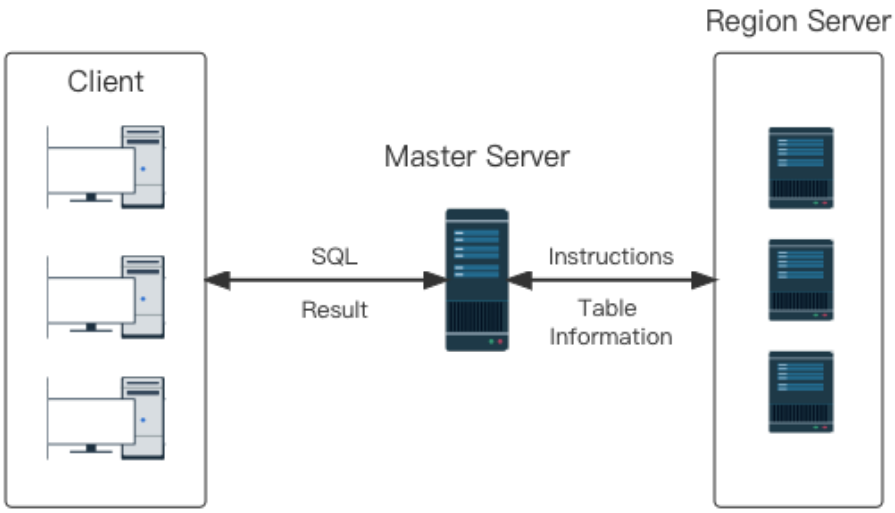


图 8 系统通信架构

其中 Master 和 Region、Client 的连接是一直存在的，直到客户端主动断开连接或从服务器宕机，连接断开。不同模块之间的通信基于 tcp 协议，都是全双工的。

3.1.2 通信协议设计

其中 Master 和 Region、Client 的通讯消息都是以 json 格式定义的，传输前将结构转为字节流，以字节流方式传输，在抵达终点后对其复原。

```
type regionRequest struct {
    TableName string
    IpAddress string
    Kind string
    Sql string
    File []string
}

type clientResult struct {
    Error string                    `json:"error"`
    Data  []map[string]interface{} `json:"data"`
}

type regionResult struct {
    Error string                    `json:"error"`
    Data  []map[string]interface{} `json:"data"`
    TableList []string              `json:"tableList"`
    Message string                  `json:"message"`
    ClientIP string                 `json:"clientIp"`
    File []string                   `json:"file"`
    TableName string                `json:"tableName`
}
```

图 9 通信协议

主节点作为通讯中心，和 Client 与 Region 共同维护特定的通讯协议。

Master 向 Region 发送的请求包含以下部分：待操作表名、发起请求的客户端 ip 地址、操作类型（删除、查询等）、完整的 sql 语句，还有 File 字段，该字段在备份策略时生效，即当某从节点宕机后，主节点找到与宕机从节点储存了同样表的其他从节点，获取遗失表的操作记录文件，并发送给重新选定的从节点。

Region 向 Master 返回的结果包含以下部分：是否出错、查询数据结果、该从节点包含的所有表名、发起请求的客户端 ip、当前请求的操作涉及的表名、还有用于执行备份策略的 File 字段。

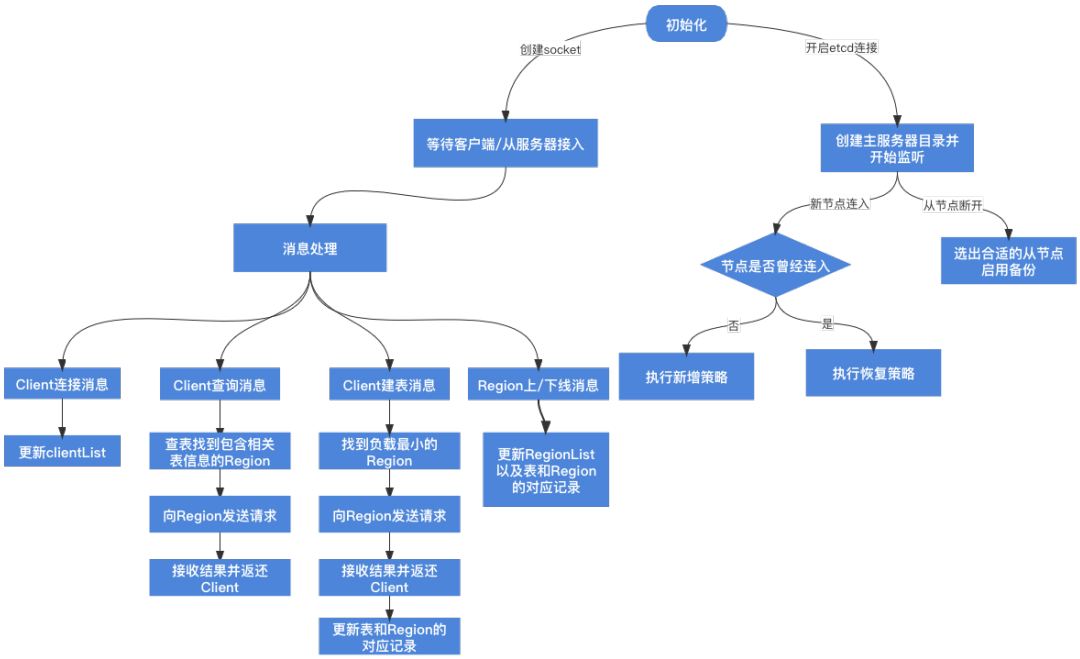
Master 向 Client 返回的结果则包含通讯是否错误以及请求返回的具体数据，供给 Client 输出操作结果。

3.1.3 通信中的可能情况及处理

在初始化时，Master 会监听客户端、服务端的端口，当监听到来自客户端或服务端的连接请求后与之建立稳定的连接，对于服务端，还需要与 etcd 模块沟通，完成服务注册。

来自 Client 的请求可以分为几种类型，当请求为建表操作时，Master 调用负载均衡相关函数，选择表记录最少的两个从服务器执行建表任务，向选择的服务器发送建表请求，并把接收到的值返回给客户端；当请求为查询操作时，Master 查表找到包含该表的从服务器，发送查询请求，并把返回结果转发给客户端；当请求为增删改时，Master 则需要找到包含该表的所有从服务器，向他们同时发送操作请求，当操作全部成功时再向客户端发送成功消息。

在运行过程中，可能出现从服务器宕机的情况，此时 Master 会查表找到与宕机从节点储存了同样表的其他从节点，获取遗失表的操作记录文件，并发送给重新选定的从节点。



4. 总结

本次课程项目由我们小组5人合作,完成了分布式关系型数据库系统的搭建，项目由Go语言完成，引入ETCD集群管理，实现了对分布式sqlite数据库系统的搭建，完成了分布式存储、负载均衡、副本管理和容错容灾等功能。通过共同设计和搭建我们自己的分布式数据库,我们在实践中加深了对于分布式系统的理解，学习并实践了编程语言Go，掌握了ETCD集群管理的方法，受益匪浅。