

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	3190103296
学生姓名	严昕辰
所在专业	软件工程
所在班级	软工 1902

2021 年 5 月 26 日

目 录

- 1 引言..... 1
 - 1.1 系统目标..... 1
 - 1.2 我的贡献..... 1
- 2 系统设计与实现..... 2
 - 2.1 系统总体架构..... 2
- 3 我参与的核心功能模块..... 3
 - 3.1 Client 模块..... 3
 - 3.1.1 流程设计..... 3
 - 3.1.2 通信协议与数据结构设计 4
- 4. 总结..... 5

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、主从架构数据库服务端等多个模块，可以实现对简单及部分复杂 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

本次项目开发中，我主要负责客户端（Client）模块的设计与实现。其功能包括与指定主节点服务器建立连接、输入与简单解析处理 SQL 语句、向服务端发送与接收消息及打印语句执行结果。此外，我也参与了 Client 与 Master 间通讯协议的设计。

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：

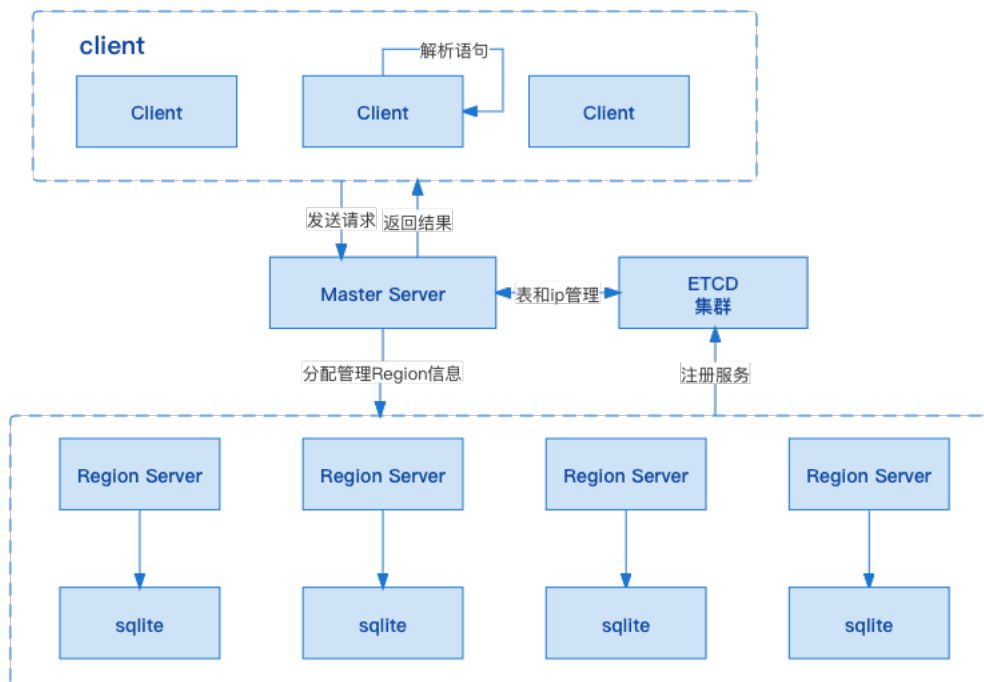


图 1 总体架构图

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客户端（Client）模块，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 Client 模块

3.1.1 流程设计

考虑到控制消息处理负载与维持系统实时性，客户端被设计为运行在单一线程上。打开客户端后，用户输入需要连接到的服务端（Master）IP 地址和端口号。建立连接后，用户即可输入 SQL 语句进行增、删、改、查等操作。系统读取 SQL 语句后，首先交由 SimpleInterpreter 进行简易解析处理。由于 Master 节点需要根据语句的操作类型和是否包含 join 决定处理方式，并根据表名决定处理消息的 Region，因此上述内容通过 SimpleInterpreter 进行预先解析处理，并将处理结果存放在键值对表中返回。该结果经过 marshal 处理后转化为字节流，通过 socket 连接发送到服务端。此后客户端等待直到服务端返回回复消息，并根据消息内容输出相应的错误信息或执行结果。Client 的流程图如下：

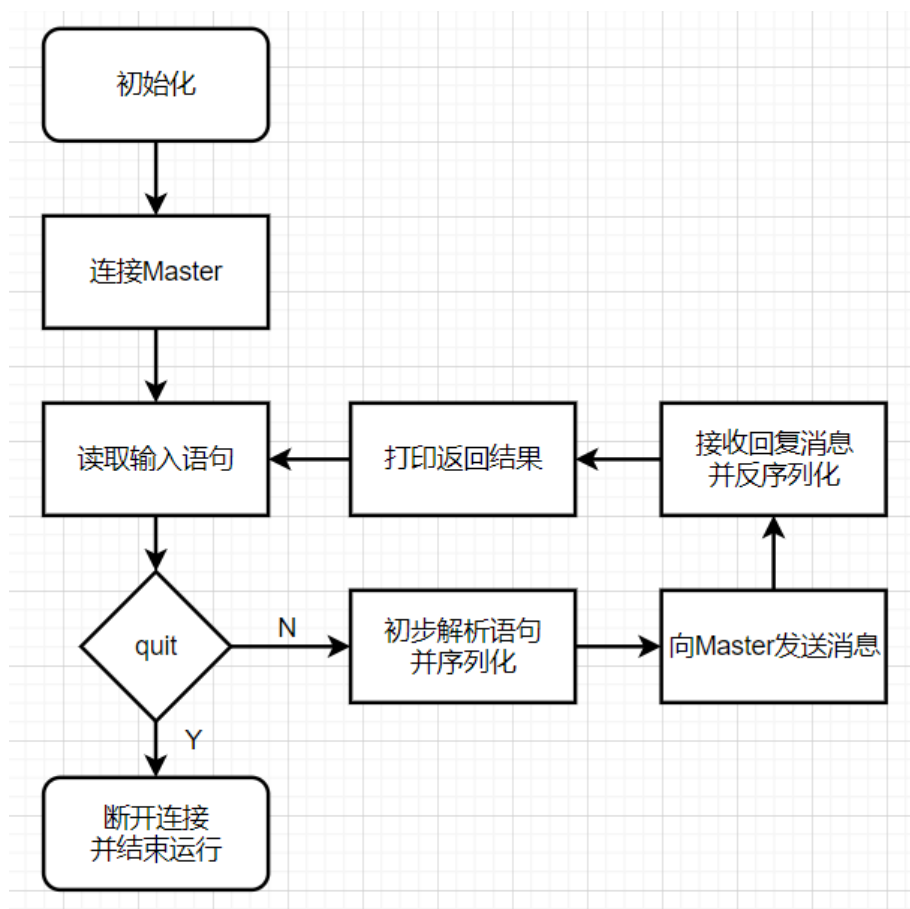


图 2 Client 流程图

3.1.2 通信协议与数据结构设计

Client 仅与 Master 建立 socket 连接，并通过字节流发送消息。通过 SimpleInterpreter 处理用户输入的 SQL 语句后，客户端对结果（键值对）进行序列化处理得到字节流。其中字段及含义设计如下：

字段名	含义
kind	操作类型
name	处理表名
join	是否包含 join 操作
sql	原始 SQL 语句

表 1 Client 消息格式

在接收 Master 返回的结果时，客户端对其进行反序列化，解析其中的错误信息以及请求返回的结果，并进行打印输出。返回结果格式如下：

```
type Result struct {  
    Error string          `json:"error"` //错误信息  
    Data  []map[string]interface{} `json:"data"`  //查询结果  
}
```

图 3 Master 返回结果结构

其中查询结果以键（字段名）值对形式存放在 Data 中，样例如下：

```
Data []map[string]interface{  
    {"field1": "a1", "field2": 2},  
    {"field1": "a2", "field2": 4}  
}
```

图 4 Data 查询结果结构

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编写。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。在我的个人模块中,我编写了简易的 SQL 语句解析器,学习了使用 Go 语言建立 socket 连接并传输消息的方法,同时进行了通信协议的设计,对相关知识有了更深理解和更进一步的实践。

