**基于Raft共识协议的KV数据库**

课程名称：中间件实现技术

学生：黄文俊

学号：2019104236

**一 项目介绍**

分布式一致性是构建容错系统的基础，它使得一些机器可以构成集群工作，并容许其中一些节点失效。Raft 是一个比较常见的分布式共识协议，Raft首先选举出一个server作为Leader，然后赋予它管理日志的全部责任。Leader从客户端接收日志条目，复制给其他server，并告诉他们什么时候可以安全的将日志条目应用到自己的状态机上。拥有一个Leader可以简化replicated log的管理。例如，leader可以决定将新的日志条目放在什么位置，而无需询问其他节点，数据总是简单的从leader流向其他节点。Leader可能失败或者断开连接，这种情况下会选出一个新的leader。具体的算法细节可以参考论文《In search of an Understandable Consensus Algorithm》。

本项目基于 Raft 分布式共识协议，实现一个简单 KV 数据库。该数据库支持3个操作，即 GET、PUT、APPEND。GET 操作能够获取一个 Key 对应的 Value。PUT 操作能够将一个 Key 对应的 Value 写入。APPEND 操作能够将参数中的 Value 追加写入的一个 Key 对应的 Value 上。

**二 系统架构**

图示

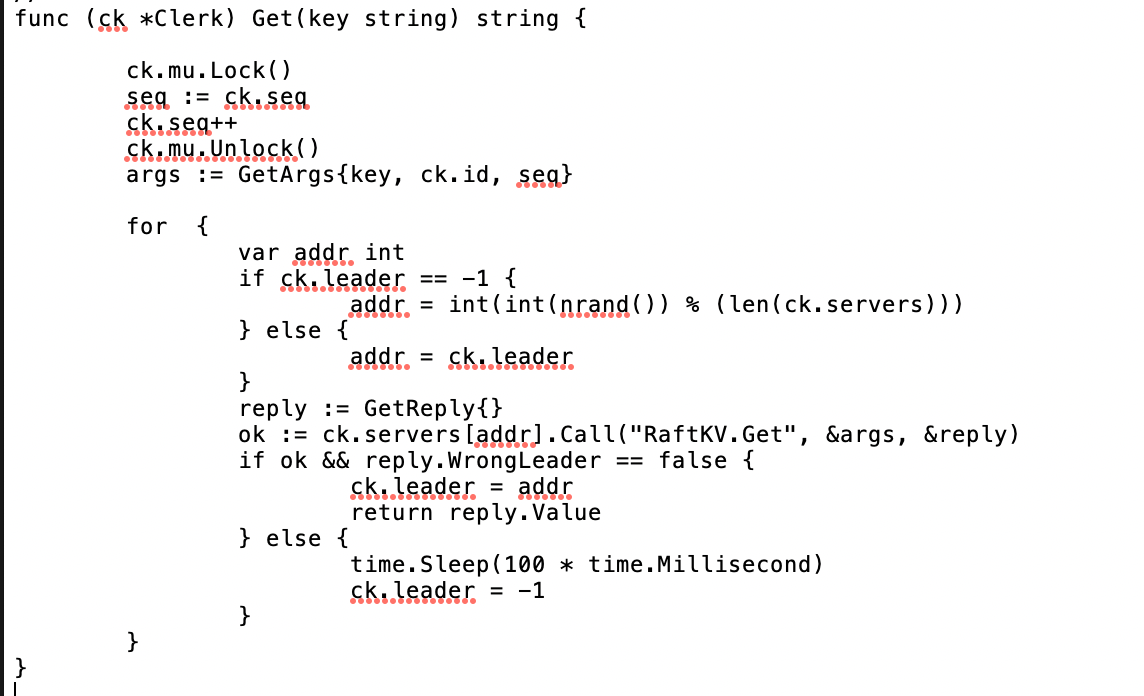
描述已自动生成

每一个server都有一个日志保存了一系列的指令，state machine会顺序执行这些指令。每一个日志都以相同顺序保存着相同的指令，因此每一个state machine处理相同的指令，state machine是一样的，所以最终会达到相同的状态及输出。

保证replicated log的一致是一致性算法的任务。server中的一致性模块接收客户端传来的指令并添加到自己的日志中，它也可以和其他server中的一致性模块沟通来确保每一条log都能有相同的内容和顺序，即使其中一些server宕机。 一旦指令被正确复制，就可以称作committed。每一个server中的状态机按日志顺序处理committed指令，并将输出返回客户端。

**三 源码分析**

**客户端代码：**



以 GET 操作为例。因为一个集群中，只有成为 Leader 的服务器能够处理用户的请求，所以客户端在发送请求给服务器时会缓存 Leader 的信息。如果返回的错误是 NotLeader，就会进行重试，将请求发给其他的服务器。

**服务器代码：**

文本

描述已自动生成

表格

描述已自动生成

同样以 GET 操作为例，服务器在接受到 Get 请求时，会检查自己是否是 Leader，如果是的话就注册一个回调函数。在另一个循环中，服务器会监听成功 Commit 的 Raft 日志，并返回GET的结果给客户端。

**Raft 模块代码：**

Raft 模块是最复杂的地方，其实现参考论文中的描述：

**图形用户界面, 文本, 应用程序

描述已自动生成**

在Raft层，一共有三种 RPC 调用，即sendRequestVote, sendAppendEntries, sendInstallSnapshot。其中 sendRequestVote 用来竞选 Leader，sendAppendEntries 用来同步 Raft 日志以及发送心跳，sendInstallSnapshot 用来进行快速恢复。

**四 测试部分**

为了验证 Raft 分布式共识算法实现的正确性已经服务器能够正确处理地处理来自客户端的请求，项目中有大量的测试用例。这些测试会模拟各种各样服务器宕机，网络隔离的情况。该KV数据库能够全部通过这些测试。