主要在于进行成员变更时,可能存在新旧配置的 2 个"大多数",导致集群中同时出现两个领导 者,破坏了 Raft 的领导者的唯一性原则,影响 了集群的稳定运行。

单节点变更是利用"一次变更一个节点,不会同时 存在旧配置和新配置 2 个'大多数'"的特性,实现 ———— 单节点变更方案

指令、

成员变更的问题

过程

保证日志的一致性

成员变更

日志复制

Raft

成员变更。

一条由客户端指定的状态机需要执行的指令

整数索引,用来标识日志项的,单调递增、连续 索引值 如如 组成,由日志项组成的整数

创建这条日志的领导者的任期编号 ——— 任期编号

优化后的二阶段提交协议

第一阶段:通过日志复制RPC,将日志项复制到集群中其他节点,接着如果领导者接收到大多数的"复制成功"响应后,它将日志项应用到它的状态机,并返回成功给客户端。如果领导者没有接收到大多数的"复制成功"响应,那么就返回错误给客户端。

无第二阶段,因为领导者的日志复制 RPC 消息或心跳消息,包含了当前最大的,将会被提交(Commit)的日志项索引值。所以通过日志复制RPC 消息或心跳消息,跟随者就可以知道领导者的日志提交位置信息。因此,当其他节点接受领导者的心跳消息,或者新的日志复制 RPC 消息后,就会将这条日志项应用到它的状态机。而这个优化,降低了处理客户端请求的延迟,将二阶段提交优化为了一段提交,降低了一半的消息延迟。

首先,领导者通过日志复制 RPC 的一致性检查,找到跟随者节点上,与自己相同日志项的最大索引值。也就是说,这个索引值之前的日志,领导者和跟随者是一致的,之后的日志是不一致的了。

然后,领导者强制跟随者更新覆盖的不一致日志 项,实现日志的一致

一切以领导者的日志为主,不会删除领导者的日 志项



领导者 三 节点状态 跟随者 三 候选者 三 仟期编号 Raft算法中的任期,不只是时间段,而且,任期 编号大小会影响领导者选举和请求处理 在 Raft 算法中约定,如果一个候选人或者领导 者,发现自己的任期编号比其他节点小,那么它 会立即恢复成跟随者状态。比如分区错误恢复 后、任期编号为 3 的领导者节点 B、收到来自新 领导者的,包含任期编号为4的心跳消息,那么 节点 B 将立即恢复成跟随者状态。 ·通讯 -RPC 领导者的心跳信息,能阻止跟随者发起新的选举 指定时间内, 没收到领导者的信息, 跟随者推举 自己为候选人,发起选举 赢得大多数选票的候选人,将晋升为领导者 在一次选举中,每一个服务节点,最多会对一个 任期编号,投出一张选票

> 当任期编号相同时,日志完整性高的跟随者,将 2拒绝投票给日志完整性低的获选人

> > 第一种含义,跟随者,等待领导者心跳信息超 时,然后发起选举的随机时间间隔

随机超时时间

第二种含义,当没有候选人赢得过半票数,选举 无效时,等待选举超时,然后再发起新一轮选举 的时间间隔

日志,一切以领导者为准