# 一致性协议Raft

## 引言

Raft是一种用来管理日志复制的协议，它和Paxos协议的功能是一样的，并且在性能上和Paxos相差无几。相信了解过Paxos协议的人都会觉得其晦涩难懂，工程实现和协议相差甚远。而Raft协议的核心设计理念就是简单，并且更易于建立实际的系统。Raft协议为了更便于理解，将问题拆分为Leader选举、日志复制、安全性这几个部分。还提供了动态改变集群成员的机制。

## 复制状态机

一致性协议都是在复制状态机的背景下提出的。基于这样一个理论：一组初始状态相同的状态机，如果按照相同顺序执行一批命令，那么这一组状态机的最终结果一定相同。通俗来讲，可以认为分布式系统中的每个节点内部有一个独立的状态机，每个节点通过执行自己日志中的命令，来达到系统中各个节点的一致性(如图1)。使用复制状态机的例子有Chubby和ZooKeeper等。Raft协议就是解决分布式系统中节点日志一致性问题的。

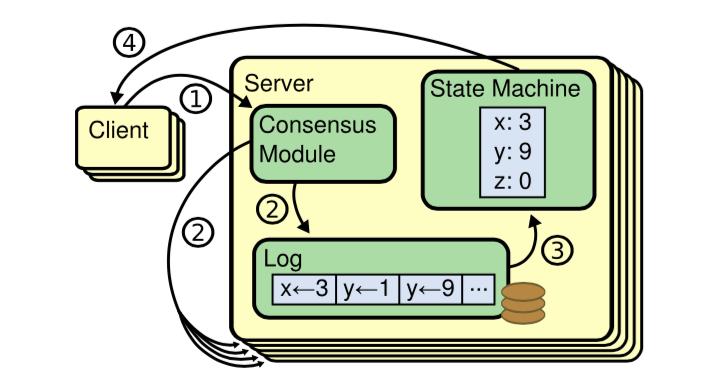


图1

## Paxos算法的不足

Paxos算法一直被认为分布式一致性算法中的鼻祖，无论是Raft协议还是其他的分布式协议，或多或少都借鉴了Paxos算法。

但是Paxos有两个缺点：

1. Paxos算法的晦涩难懂是一直以来被诟病的问题。
2. 难以在实际环境中实现，其中一个原因是对于多决策，大家还没有一个一致同意的算法。Lamport的描述大部分都是有关于单决策；他仅仅描述了实现多决策的可能的方法，缺少许多细节。有许多实现 Paxos 和优化 Paxos 的尝试，但是他们都和 Lamport 的描述有些出入。例如，Chubby 实现的是一个类似 Paxos 的算法，但是在许多情况下的细节没有公开。

还有一点是Paxos使用点对点的实现作为它的核心(尽管提出了一种弱Leader的形式来优化性能)，这种方法通常比较适用于那些只需要一个决策被制定的情况下。如果分布式系统需要做很多决策，那么通过选择一个Leader，由Leader来协调是更简单高效的方法。