# 分布式一致性算法应用场景

一致性问题，简单说就是一个或者多个进程提议了一个值后（这里一个值并不理解为一个数值，可以有很多的含义），使系统中所有进程对这个值达成一致意见。

如果在一个不出现故障的系统中，很容易解决分布式一致性问题。但是实际分布式系统一般是基于消息传递的异步分布式系统，进程可能会慢、被杀死或者重启，消息可能会延迟、丢失、重复、乱序等。

在一个可能发生上述异常的分布式系统中如何就某个值达成一致，形成一致的决议，保证不论发生以上任何异常，都不会破坏决议的一致性，这些正是一致性算法要解决的问题。

在分布式存储系统中经常使用多副本的方式容错，对于传统的主从同步方式，强一致性同步（主备副本都写成功返回）和异步同步（主副本写成功就返回再异步写备副本）。主从同步无法同时保证数据的一致性和可用性（CAP），而Paxos、Raft等分布式一致性算法则可在一致性和可用性之间取得很好的平衡，在保证一定的可用性的同时，能够对外提供强一致性，因此Paxos、Raft等分布式一致性算法被广泛的用于管理副本的一致性，提供高可用性。

# Paxos

Paxos算法是基于**消息传递**且具有**高度容错**特性的一致性算法，是目前公认的解决**分布式一致性**问题最有效的算法之一。

在常见的分布式系统中，总会发生诸如机器宕机或网络异常（包括消息的延迟、丢失、重复、乱序，还有网络分区）等情况。Paxos算法需要解决的问题就是如何在一个可能发生上述异常的分布式系统中，**快速且正确地在集群内部对某个数据的值达成一致**，并且保证不论发生以上任何异常，都不会破坏整个系统的一致性。这里**某个数据的值**并不只是狭义上的某个数，它可以是一条日志，也可以是一条命令（command）。。。根据应用场景不同，某个数据的值有不同的含义。

# raft

# 参考

Paxos算法原理与推导

<https://www.cnblogs.com/linbingdong/p/6253479.html>

分布式一致性最佳实践分享

<https://zhuanlan.zhihu.com/paxos>

理解这两点，也就理解了paxos协议的精髓

<https://zhuanlan.zhihu.com/p/29706905>

使用Paxos前的八大问题

<https://zhuanlan.zhihu.com/p/23811020>

Raft Consensus Algorithm

<https://raft.github.io/>

raft动画展示

<http://thesecretlivesofdata.com/raft/>