【我对paxos小岛的理解】

通过老师的介绍和网上看的科普，我首先认识到Paxos是位于希腊的一个小岛 。

1.

Paxos算法则是由分布式系统大神Leslie Lamport提出的一种基于消息传递并且高度容错的一致性算法。Paxos算法的命名由来，是因为Lamport最初是用一个发生在Paxos岛上的城邦议会故事阐述的，可惜并没有人能听懂他的故事。Lamport很可惜没人理解他的幽默 ，于是重新写了一篇关于Paxos算法的文章Paxos Made Simple。

2.

而之所以有paxos，在我看来首先是因为分布式系统要使用多台机器各司其职地为用户提供服务，而于用户而言，就像是使用单机系统一样方便，不需要也感受不到后端多台机器的运作 。

因此分布式系统中需要解决的问题也就是：如何使多台提供同一服务的机器协调一致？

我们知道，在一个分布式系统中，多台机器的地理位置可能不同（可能有的紧挨在一个机房，也可能有的横跨半个地球），每台机器的硬件性能不同，处理速度不同，实时网速不同，也能有的机器会宕机。

这个问题是使分布式系统正确运行的根本问题，也是分布式系统设计中的一大难题。

Paxos 算法则可以解决这个问题，毕竟它被誉为解决一致性问题的圣经 。

3.

在我的理解中可以将 Paxos 分为两类：

Basic Paxos：一个或多个机器提出指令，仅有一个指令会被选中

Multi-Paxos：由多个 Basic Paxos 共同组成，用于确定一系列指令

理解单个 value 一致性的 Basic Paxos 过程后，其他的 Paxos 形式就很好理解了。

而Basic Paxos 包含两个角色：

Proposers：处理 client 请求，提出一个proposal

Acceptors：处理 proposers 请求，回复当前proposal 的投票情况，保存已经选定的 proposal 和当前的投票状态

在原文中还有第三个角色Learner，它的作用仅仅是获取已被选中的proposal，在此为了方便表述将它与Acceptor合为一谈。

4.

至于它到底是如何工作的：

Paxos 是一个民主的算法，遵循少数服从多数的原则。当超过半数的 Acceptor 都 accept 了某个 proposal 时，这个 proposal 中的 value 就会被唯一选定，且不会再改变

除了基础的工作，paxos还有很多种分类面临不同的工作，我理解到：

Paxos 算法默认集群中没有坏消息或恶意消息，即应用在non- Byzantine fault情况下Proposer 的最终目的并不是全力是自己的初始 value 被选定，而是全力是集群尽快达成一致。

Accepter不会知道最终选定的 value 是什么，只有 Proposer 才知道，所以如果一个 Server 想知道最终被选定的 value ，它必须以 Proposer 的身份发起一次 proposal（value被选定后就唯一确定，不再更改，所以以获得选定 value 为目的的 proposal 可以携带任意 value）

总体看这个算法还是非常晦涩难懂的，我能理解到的以上。