第 4 章 区块链灵魂——共识算法

1.1 分布式系统的一致性

多路处理器和分布式系统中的一致性问题是一个非常难以解决的问题。难点在于：

- 分布式系统本身可能出故障。
- 分布式系统之间的通信可能有故障，或者有巨大的延迟。
- 分布式系统运行的速度可能大不相同，有的系统运行很快，而有些则很慢。

但是，分布式系统的一致性问题是设计分布式系统时必须考虑的问题。一致性问题历史悠久，而且臭名昭著。传统的处理一致性问题的方法有两段式提交、令牌环、时间戳等等，计算机专业的读者应该有所耳闻。本章将集中讨论与区块链相关的一致性问题和算法。

1.1.1 一致性的问题

下面我们用状态机来解释一致性问题。假设我们有 n 台机器，位于不同的位置的机器之间通过网络协同工作。所有机器的初始状态是一模一样的。给他们一组相同的指令，我们希望看到相同的输出结果，而且希望看到状态的变化也是一样的。比如机器甲的状态是用状态 A 到 B 再到 C，而如果机器乙的状态是由 A 直接到 C，这种情况就是不一致的。

总而言之，一致性要求分布式系统产生同一样的结果，看起来好像就是一台机器一样。同时还要具备以下特性：

- 分布式系统不应该返回错误的结果。
- 系统里的大部分机器正常，整个分布式系统就能有效运行。而不是一台机器出问题，整个系统就不工作了。
- 木桶原理不适用于分布式系统。木桶原理是指系统的最终性能由最短的那块木板决定。
- 分布式系统必须是异步的。即没有全序的时间顺序，只有相对的偏序。

1.1.2 两个原理——FLP&CAP

FLP 定理

Fischer, Lynch and Paterson 在论文 "Impossibility of distributed consensus with one faulty process" (1985)证明，在一个异步系统里面，只要有一个错误的进程，一致性就不可能达成. 这个也就是著名的"FLP"结论。在这里"一致性" 代表的是一堆进程同意同一个值的问题。这个问题的难点

在于你不能判断一个进程是停止了还是跑的非常的慢。处理一个在异步系统里面的错误几乎是不可能的。

CAP 定理

分布式计算系统不可能同时确保一致性、可用性和分区容错性。，这三者不可兼得。

- 一致性（Consistency）：所有节点在同一时刻能够看到同样的数据，即“强一致性”。
- 可用性（Availability）：确保每个请求都可以收到确定其是否成功的响应。
- 分区容错性（Partition Tolerance）：因为网络故障导致的系统分区不影响系统正常运行。

直觉上的论证很简单：如果网络分成了两半，并且我在一半的网络中“给 A 发送了 10 个币”，在另外一半的网络中“给 B 发送了 10 个币”，那么要么系统不可用，因为其中一笔交易或者全部两笔都不会被处理，要么系统会变得没有一致性，因为一半的网络会完成第一笔交易，而另外一半网络会完成第二笔交易。

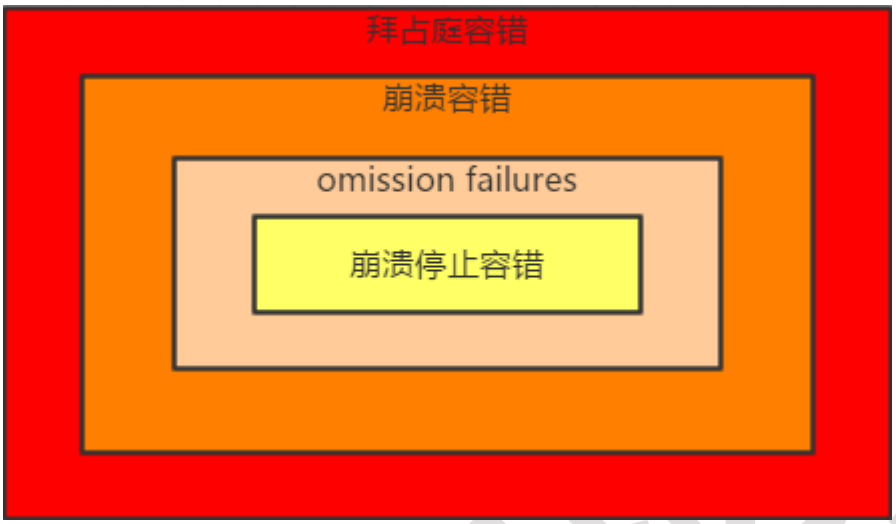
1.1.3 拜占庭将军问题

拜占庭将军的经典描述是：拜占庭的军队是由小分队组成的。每个小分队由一个将军指挥。将军们通过传令兵来策划一系列的行动。有些将军是叛徒，他们会有意的妨碍忠诚的将军们达成一个一致的计划划。这个问题的目标是使忠诚的将军达成一致的计划，即使背叛的将军们一直在诱使他们采用一个差的计划。已经证明，如果背叛的将军超过了将军总数的 1/3，达成上述目标是不可能的。特别要注意的是要把拜占庭问题和两军问题区分开。

拜占庭将军的问题的复杂性，可以用计算机容错学里的概念来表述：

1) 拜占庭容错：这是最难处理的情况，一个节点压根就不按照程序逻辑执行，对它的调用会返回给你随意或者混乱的结果。要解决拜占庭式故障需要有同步网络，并且故障节点必须小于 1/3，通常只有某些特定领域才会考虑这种情况，通过高冗余来消除故障.
2) 崩溃容错：它比拜占庭类故障加了一个限制，那就是节点总是按照程序逻辑执行，结果是正确的。但是不保证消息返回的时间。不能及时返回消息的原因可能是节点崩溃后重启了、网络中断了、异步网络中的高延迟等。
3) 遗漏容错（Omission Failure）：比崩溃容错多了一个限制，就是一定要非健忘。非健忘是指这个节点崩溃之前能把状态完整的保存在持久存储上，启动之后可以再次按照以前的状态继续执行和通信。比如最基本版本的 Paxos 要求节点必须把投票的号码记录到持久存储中，一旦崩溃，修复之后必须继续记住之前的投票号码。
4) 崩溃停止容错：它比遗漏容错多了一个故障发生后要停止响应的要求。简单讲，一旦发生故障，这个节点就不会再和其它节点有任何交互. 就像他的名字描述的那样, 崩溃并且停止。

由下图看出，拜占庭将军问题的复杂度。



1.1.4 共识算法的目的

在有错误的进程存在并且有可能出现网络分区的情况下，FLP 定理堵死了我们在传统计算机算法体系下提出解决方案的可能性。计算机科学家就想，如果我们把 FLP 定理的设定（Assumption）放松一点，问题是否有解呢？由社会学和博弈论中得到启发，科学家尝试引入了：

- **激励机制（Incentive）**

  比如，在拜占庭将军问题中给忠诚的将军以奖励。当背叛的将军发现背叛行为没有任何收益的时候，他们还有背叛的动机吗？这里我们引进了博弈论的概念：我们不再把节点或者说将军分成公正/恶意（忠诚/背叛）两方，我们认为每一个节点的行为是由激励机制决定了。就如 2 千年之前中国诸子百家热烈争论的话题：人之初，性本善焉，性本恶焉？我们认为人之初，性无善无恶。性的善恶由后天的激励机制决定。如果激励机制设置得当，考虑到每个节点都有最大化自己利益的倾向，大部分的节点都会遵守规则，成为公正的节点。

- **随机性（Randomness）**

  在拜占庭将军问题中，决定下一个行动需要将军们协调一致，确定统一的下一步计划。在存在背叛将军的条件下，忠诚的将军的判断可能被误导。在传统的中心化的系统中，由权威性大的将军作决定。比如现实世界里的政府，银行。在去中心化的系统中，研究者提出一种设想：是否可能在所有的将军中，随机的指定一名将军作决定呢？这个有点异想天开的设想为解决拜占庭将军问题打开了一扇门。根据什么规则制定作决定的将军呢？对应到金融系统里，就是如何决定谁有记账权。

  1) **根据每个节点（将军）的计算力（Computing Power）来决定。**

     谁的计算力强，解开某个谜题，就可以获得记账权（在拜占庭将军问题里是指挥权）。

     这是比特币里用的 PoW 共识协议。

2） **根据每个节点（将军）具有的资源（Stake）来决定。**

所用到的资源不能被垄断。谁投入的资源多，谁就可以获得记账权。这是 PoS 共识协议。

出于上面的考虑，科学家引入共识算法，试图解决拜占庭将军问题。分布式共识协议具有以下两点属性：

1) 如果所有公正节点达成共识，共识过程终止

2) 最后达成的共识必须是公正的

下面我们来谈谈共识算法的适用范围。区块链的组织方式一般有以下 3 种：

- 私有链：封闭生态的存储网络，所有节点都是可信任的，如某大型集团内部多数公司。
- 联盟链：半封闭生态的交易网络，存在对等的不信任节点，如行业内部的公司 A、B、C 等。
- 公有链：开放生态的交易网络，即所有人都可以参与交易，没有任何的限制和资格审核

由于私有链是封闭生态的存储网络，因此使用传统分布式一致性模型应该是最优的；由于联盟行业链其半封闭半开放特性，使用 Delegated Proof of XXX 是最优的。对于公有链，PoW 应该仍然是最优的选择

常见共识算法一览表：

共识算法	**应用**
PoW	比特币，莱特币，以太坊前三个阶段：即 Frontier（前沿）、Homestead（家园）、Metropolis（大都会）
PoS	PeerCoin， NXT， 以太坊的第四个阶段，即 Serenity（宁静）
DPos	BitShare
Paxos	Google Chubby， ZooKeeper

共识算法	应用
PBFT	Hyperledger Fabric
Raft	etcd

1.2 Paxos 算法

1998 年 Lamport 提出 Paxos 算法，后续又增添多个改进版本的 Paxos 形成 Paxos 协议家族。Paxos 协议家族有一个共同的特点就是不易于工程实现，Google 的分布式锁系统 Chubby 作为 Paxos 实现曾经遭遇到很多坑。

- Classic Paxos ：LeaderLess，又名 Basic Paxos，以下均为 Paxos 的变种，基于 CAP 定律，侧重了不同方向。
- Cheap Paxos
- Egalitarian Paxos
- Fast Paxos
- Multi-Paxos
- Byzanetine Paxos

"Byzantine Paxos adds an extra message (Verify) which acts to distribute knowledge and verify the actions of the other processors".Lamport 在 2011 年的论文《Leaderless Byzanetine Paxos》中表示不清楚实践中是否有效

1.3 Raft 算法

由于 Paxos 太难懂太难以实现，Raft 算法应运而生。其目的是在可靠性不输于 Paxos 的情况下，尽可能的简单易懂。斯坦福大学的 Diego Ongaro 和 John Ousterhout 以易理解为目标，重新设计了一个分布式一致性算法 Raft，并于 2013 年底公开发布。Raft 既明确定义了算法中每个环节的细节，也考虑到了整个算法的简单性与完整性。与 Paxos 相比，Raft 更适合用来学习以及做工程实现。下面，笔者将以通俗易懂的方式来描述这个过程。

```
百花村村长一人负责对外事务。比如县，乡两级的公文来往，公粮征收，工务摊派，
税收等等。
```

Raft 是一个强 Leader 的共识协议。我们想象百花村是一个服务器集群，而这个集群的 Leader 就是村长，村里的每户人家（Follower）对应一个服务器，每户人家都保存了一个数据副本。所有的数据副本都必须保证一致性。即上级官员下到村里视察时，从每户人家获得的信息应该是一样的。

百花村村长通过村户选举产生。谁得的票数多（简单多数）谁就当选村长。村长有任期概念（Term）。任期是一直向上增长的： 1，2，3... n，n+1，...

这里要处理的是平票（split vote）的情况。在平票的情况下，该次村长选举失败，每户人家被分配不同的睡眠值。在睡眠期间的村户不能发起选举，但是可以投票。而且只有选举权，但是没有被选举权。第一个走出睡眠期的村户发起新任期的选举。由于每户人家有不同长度的睡眠期，这保证了选举一定会选出一个村长，而不会僵持不下，不会出现每次选举都平票的情况。一旦村长产生，任何对百花村的"写"（比如政府政策宣示，普法教育）必须经过村长。

村长每天都要在村里转一圈，让所有人都看见。表明村长身体健康，足以处理公务。

村长选举出来后，要防止村长发生故障，必须定期检测村长是否失效。一旦发现村长发生故障，就要重新选举。

村长接收到上级命令，该命令数据处于未提交状态（Uncommitted），接着村长会并发向所有村户发送命令，复制数据并等待接收响应，确保至少所有村户中超过半数村户已接收到数据后再向上级确认数据已接收（命令已执行）。一旦向上级发出数据接收 Ack 响应后，表明此时数据状态进入已提交（Committed），村长再向村户发通知告知该数据状态已提交（即命令已执行）。

下面我们来测试各种异常情况：

1) 异常情况 1

上级命令到达前，村长挂了

这个很简单，重新选举村长。上级命令以及来自外面的请求会自动过时失效，他们会重发命令和请求的。

2) 异常情况 2

村长接到上级命令，还没有来得及传达到各村户就挂了

这个和异常情况 1 类似，重新选举村长。上级命令以及来自外面的请求会自动过时失效。他们会重发命令和请求的

3) 异常情况 3

村长接到上级命令，已传达到各村户，但是各村户尚未执行命令， 村长就挂了

这个异常情况下，重新选举村长。新村长选出后，由于已收到命令，就可以等待各村户执行命令（也就是 Commit 数据）。上级命令以及来自外面的请求会自动过时失效。有可能，他们会重发命令和请求的。 Raft 要求外部的请求可以自动去除重复。

4) 异常情况 4

村长接到上级命令，已传达到各村户，各村户执行了命令，但是村长并没有收到通知。就在这时候村长挂了

这种情况下，同上一种情况，新村长选出后，即可以等待通知，完成剩下的任务。外部也会接到通知命令已完成。

5) 异常情况 5

在命令执行过程中，村长身体不适，不能处理公务

因为百花村没有收到村长的心跳，百花村的村户就会自动选举（当前任期+1）任村长。这个时候就出现 2 个村长。这个时候新村长就会接过老村长角色，继续执行命令。即使原村长身体康复，也将沦为普通村户。

1.4 PBFT 算法（Practical Byzantine Fault Tolerance）

1999 年 Castro 和 Liskov 提出的 PBFT 是第一个得到广泛应用的 BFT 算法。在 PBFT 算法中，至多可以容忍不超过系统全部节点数量的 1/3 的拜占庭节点，即如果有超过 2/3 的正常节点，整个系统就可以正常工作。早期的拜占庭容错算法或者基于同步系统的假设，或者由于性能太低而不能在实际系统中运作。PBFT 算法解决了原始拜占庭容错算法效率不高的问题，将算法复杂度由指数级降低到多项式级，使得拜占庭容错算法在实际系统应用中变得可行。也许就是出于效率的考虑，央行推出的区块链数字票据交易平台用的就是优化后的 PBFT 算法。腾讯的区块链用的也是 PBFT。

在 PBFT 算法中，每个副本有 3 个状态： pre-prepare, prepared 和 commited。消息也有 3 种：pre-prepare, prepare 和 committed.收到 pre-prepare 消息并且接受就进入 prepared 状态。收到 commit 消息并且接受就进入 Committed 状态。下面以一个有 4 个节点/拷贝的例子说明，这个网络内，仅允许 1 个拜占庭节点（此处设 f=1）。

百花村小学举行百米赛跑比赛，3 年级第一组的选手只有 4 个人：Alice,Bob,Cathy 和 David（简称 A,B,C,D)。为了节省钱，比赛并没有请裁判，而是由 4 个选手中随机挑出一个做裁判。假设是 AliceA,B,C,D)．众所周知，百米跑的口令是 各就各位，预备，跑

这里各就各位就是 pre-prepare 消息，选手接受了就是脚踩进了助跑器，而这一动作被其他选手看到，就会认为该选手进入了 prepared 状态。相当于发了一个 prepare 消息给其他选手。同理，预备就是 prepare 消息，选手接受了就是双手撑起，身子呈弓形，而这一动作被其他选手看到，就会认为该选手进入了 committed 状态。

假设 A 是公正的。Alice 得到老师示意，3 年级第一组准备比赛。Alice 就喊 各就各位

老师的示意相当于一个外部消息请求。Alice 收到这个消息，给消息编一个号，比如编为 030101 号。必须编号，因为比赛有一个规则（假想），连续 4 次起跑失败，整个组都被淘汰。B,C,D 同学收到口令后,如果认为命令无误，便都把脚踩进助跑器（拜占庭的那个人例外）。而这一个动作又相当与互相广播了一个 prepare 消息。A,B,C,D 选手互相看到对方的动作，如果确认多于 f 个人（由于此处 f=1，所以是至少 2 个人）的状态和自己应有的状态相同，则认为大家进入 prepared 状态。选手会将自己收到的 PRE-PREPARE 和发送的 PREPARE 信息记录下来

假设 A 是公正的。Alice 看到至少 2 个人进入 prepare 状态，Alice 就接着喊“预备 跑”

发生的事类似上一步：B,C,D 同学收到口令后（相当于收到 commit 消息）,如果认为命令无误，便都双手撑起，身子呈弓形（拜占庭的那个人例外）。而这一个动作又相当与互相广播了一个 commit 消息。A,B,C,D 选手互相看到对方的动作，如果确认多于 f 个人（由于此处 f=1，所以是至少 2 个人）的状态和自己应有的状态相同，则认为大家进入 committed 状态。当大家都确认进入 Committed 状态后，就可以起跑了

假设 A 是不公正的。A 就会被换掉，重新选一个选手 B 发令

这时候，由于所有选手都记录了自己的状态和接受/发送的信息。对于换掉前已经是 Committed 状态的选手而言，开始广播 commit 消息。如果确认多于 f 个人（由于此处 f=1，所以是至少 2 个人）的状态和自己应有的状态相同，则认为大家进入 committed 状态。而对于换掉前是 Prepared 和 pre-prepare 状态的选手，则完全作废以前的命令和状态，重新开始。

因为非常适合联盟链的应用场景，PBFT 及其改进算法因此成为目前被使用最多的联盟链共识算法。改进主要集中在，修改底层网络拓扑的要求，使用 p2p 网络；可以动态的调整节点数量；减少协议使用的消息数量等。

不过 PBFT 仍然是依靠法定多数（quorum），一个节点一票，少数服从多数的方式，实现了拜占庭容错。对于联盟链而言，这个前提没问题，甚至是优点所在。但是在公有链中，就有很大的问题。

- PBFT 算法共识各节点由业务的参与方或者监管方组成，安全性与稳定性由业务相关方保证。
- 共识的时延大约在 2~5 秒钟，基本达到商用实时处理的要求。
- 共识效率高，可满足高频交易量的需求。

1.5 工作量证明——PoW

工作量证明机制（Proof of Work 简称 PoW）随着比特币的流行而广为人知。PoW 协议简述如下：

1) 向所有的节点广播新的交易
2) 每个节点把收到的交易放进块中
3) 在每一轮中，一个被随机选中的节点广播它所保有的块

4) 其他节点在验证块中的所有的交易正确无误后接受该区块

5) 其他节点将该区块的哈希值放入下一个他们创建的区块中，表示它们承认这个区块的正确性

节点们总是认为最长的链为合法的链，并努力去扩大这条链。如果两个节点同时广播出各自挖出的区块，其他节点以自己最先收到的区块为准开始自己的挖矿，但同时会保留另一个区块。所以就会出现一些节点先收到 A 的区块并在其上开始挖矿，同时保留着 B 的区块以防止 B 的区块所在的分支日后成为较长的分支。直到其中某个分支在下一个工作量证明中变得更长，之前那些在另一条分支上工作的节点就会转向这条更长的。

平均每 10 分钟有一个节点找到一个区块。如果两个节点在同一个时间找到区块，那么网络将根据后续节点的决定来确定以哪个区块构建总账。从统计学角度讲，一笔交易在 6 个区块(约 1 个小时)后被认为是明确确认且不可逆的。然而，核心开发者认为，需要 120 个区块(约一天)，才能充分保护网络不受来自潜在更长的已将新产生的币花掉的攻击区块链的威胁。

生物学上有一个原理叫做不利原理（the Handicap Principle），该原理可以帮助我们解释工作量证明的过程。这个原理说，当两只动物有合作的动机时，它们必须很有说服力地向对方表达善意。为了打消对方的疑虑，它们向对方表达友好时必须附上自己的代价，使得自己背叛对方时不得不付出昂贵的代价。换句话说，表达方式本身必须是对自己不利的。

定义可能很拗口，但是这是在历史上经常发生的事：在中国历史上，国家和国家之间签订盟约，为了表示自己对盟约的诚意，经常会互质。即互相送一个儿子（有些时候甚至会送太子，即皇位继承人）去对方国家做人质。在这种情况下，为取得信任而付出的代价就是君主和儿子的亲情和十几年的养育。

比特币的工作量证明很好的利用了不利原理解决了一个自己网络里的社会问题：产生一个新区块是建立在耗时耗力的巨大代价上的，所以当新区块诞生后，某个旷工要么忽视它，继续自己的新区块寻找，要么接受它，然后接着它再继续自己的更新区块的挖掘。显然前者是不明智的，因为在比特币网络里，以最长链为合法链，这个矿工选择忽视而另起炉灶，就不得不说服足够多的矿工沿着他的路线走，相反要是他选择接受，不仅不会付出额外的辛苦，而且照样可以继续自己的更新区块的挖矿，只利不害，而且这属于全网成员在遵守一个不成文的规定，对之后该矿工自己发现新区块更有利，不会再出现你走你的我走我的，是一个全网良性建设。比特币通过不利原理约束了节点行为，十分伟大，因为这种哲学可以用到如今互联网建设的好多方面，比如防垃圾邮件、防 DDoS 攻击。

PoW 共识协议的优点是完全去中心化，节点自由进出。但是依赖机器进行数学运算来获取记账权，资源消耗相比其他共识机制高、可监管性弱，同时每次达成共识需要全网共同参与运算，性能效率比较低，容错性方面允许全网 50%节点出错。

目前比特币已经吸引全球大部分的算力，其它再用 PoW 共识机制的区块链应用很难获得相同的算力来保障自身的安全

- ❑ 挖矿造成大量的资源浪费
- ❑ 共识达成的周期较长

1.6 股权权益证明——PoS

股权权益证明（Proof of Stack 简称 PoS）现在已经有了很多变种。最基本的概念就是选择生成新的区块的机会应和股权的大小成比例。股权可以是投入的资金，也可以是预先投入的其他资源。

PoS 算法是针对 PoW 算法的缺点的改进。Proof of Stake 由 Quantum Mechanic 2011 年在 bitcointalk 首先提出，后经 Peercoin 和 NXT 以不同思路实现。PoS 不像 PoW 那样无论什么人，买了矿机，下载了软件，就可以参与。POS 要求参与者预先放一些代币（利益）在区块链上，类似于财产储存在银行，这种模式会根据你持有数字货币的量和时间，分配给你相应的利息。用户只有将一些利益放进链里，相当于押金，用户才会更关注，做出的决定才会更理性。同时也可以引入奖惩机制，使节点的运行更可控，同时更好的防止攻击。

PoS 运作的机制大致如下：

1) 加入 PoS 机制的都是持币人，成为验证者（Validator）

2) PoS 算法在这些验证者里挑一个给予权利生成新的区块。挑选的顺序依据于持币的多少

3) 如果在一定时间内，没有生成区块，POS 则挑选下一个验证者，给予生成新区块的权利

4) 以此类推以区块链中最长的链为准

PoS 和 PoW 有一个很大的区别：在 PoS 机制下，持币是有利息的。众所周知，比特币是有数量限定的。由于有比特币丢失问题，整体上来说，比特币是减少的，也就是说比特币是一个通缩的系统。在股权证明 PoS 模式下，引入了币龄的概念，每个币每天产生 1 币龄。比如你持有 100 个币，总共持有了 10 天，那么，此时你的币龄就为 1000，这个时候，如果你发现了一个 PoS 区块，你的币龄就会被清空为 0。你每被清空 365 币龄，你将会从区块中获得一定的的利息。因此，PoS 机制下不会产生通缩的情况。

和 PoW 相比，PoS 不需要为了生成新区块而大量的消耗电力。也一定程度的缩短了共识达成的时间。但是，缺点是：PoS 还是需要挖矿。

1.7 委托权益人证明机制——DPoS

DPos 是委托权益人证明机制（Delegated Proof of Stake）的简写。它是 PoS 算法的改进。笔者试着以通俗易懂的方式来说明这个算法。

假设以下的场景：百花村旁有一座山叫区块链山，属村民集体所有。村外的 A 公司准备来开发区块链山的旅游资源。A 公司和村民委员会联合成立了百花旅游开发有限公司，签了股份制合作协议。以下是春节假期期间发生在村民李大和柳五之间的对话：

```
李大： 关于旅游开发区块链山，村民委员会和 A 公司签约了。
柳五： 那我们有什么好处？
```

李大： 我们都是区块链旅游有限公司的股东了。

由于村民都是股东，所有村民就是区块链山的权益所有人。

柳五：股东要干什么工作呢？

李大：关于区块链的开发的重要决定，股东都要投票的。

柳五：那可不成。春节后我要出去打工，在哪儿还不一定呢。哪有时间回来投票。

李大：不要紧，我们可以推选几个代表，比如王老师，他会一直留在村办小学教书，不会走的，而且人又可靠，讲信用。

柳五：我也推选王老师，代表我们在重大决议上投票。

王老师在这里就是委托权益人（也叫见证人）。DPoS 算法中使用见证人机制（witness）解决中心化问题。总共有 N 个见证人对区块进行签名。DPoS 消除了交易需要等待一定数量区块被非信任节点验证的时间消耗。通过减少确认的要求，DPoS 算法大大提高了交易的速度。通过信任少量的诚信节点，可以去除区块签名过程中不必要的步骤。DPoS 的区块可以比 PoW 或者 PoW 容纳更多的交易数量，从而使加密数字货币的交易速度接近像 Visa 和 Mastercard 这样的中心化清算系统。

李大：我们集体推举王老师的，每年给王老师一点补偿，因为代表我们参加 A 公司的董事会也很花时间，挺累人的。

柳五：成啊！

权益所有人为了见证人尽量长时间的在线，要付给见证人一定的报酬。

柳五：我还准备推荐陶大妈。文化高，人也好，也会一直留在村里。

李大：陶大妈身体不好，还是不要干这个差事了。

见证人必须保证尽量在线。如果见证人错过了签署区块链，就要被踢出董事会。不能担任见证人的工作。

村民选举出几个见证人后...

柳五：这次怎么选出了赖大这家伙。这家伙一贯不干好事。我退出！

如果权益所有人不喜欢选出来的见证人，可以选择卖出权益退场。

DPoS 使得区块链网络保留了一些中心化系统的关键优势，同时又能保证一定的去中心化。见证人机制使得交易只用等待少量诚信节点（见证人）的响应，而不必等待其他非信任节点的响应。见证人机制有以下特点

- 见证人的数量由权益所有者确定，至少需要确保 11 个见证人。
- 见证人必须尽量长时间的在线，以便做出响应。
- 见证人代表权益所有人签署和广播新的区块链。
- 见证人如果无法签署区块链，就将失去资格，也将失去这一部分的收入。
- 见证人无法签署无效的交易，因为交易需要所有见证人都确认。

1.8 共识算法的社会学探讨

对于分布式系统的拜占庭问题，从计算机科学的角度，FLP&CAP 定理已经告诉我们无解。研究人员及科学家只有从其他地方寻找灵感。其实并不用花太多时间，他们就会发现，真实的人类世界就是一个分布式系统。如果太阳系和三体人所在半人马座的星球同时发生了爆炸，对于我们地球人而言，肯定是太阳系的爆炸先发生，因为光肯定是先到达地球。而在三体人看来，他们会首先观测到半人马座的爆炸。对于同样的事件，不同的系统接收到事件的顺序是不一样的。不同的系统运行速度也是不一样的。再加上通信的信道是有问题的。在上面三体人的例子里，我们假设光线的传递是毫无障碍的。但是如果光线被传播途中的黑洞给吞噬了，消息永远接收不到怎么办？

比特币的天才之处在于参照人类社会的组织方式和运作方式，引入了共识机制。一个交易的成立与否，也就是分布式账本的记账权，经由特定共识机制达成的共识来决定。共识，是一个典型的社会学概念。本章中描述的各种共识算法，读者应该都有似曾相识的感觉：

传统的中心化系统要求我们信任中心服务器。就如生活中大家要信任政府、银行，或者其他庞然大物般的机构，因为他们是国家信用背书，是大到不能倒的。但是现实生活中，雷曼银行就是倒了，两房（两房即房利美与房地美，是带有政府性质的、两个美联邦住房贷款抵押融资公司）就是快垮了。如果没有政府救助的话，投资者损失惨重。比特币诞生于 2009 年，恰恰在 2008 金融危机之后，不能不说发轫于对中心化的系统，机构的不信任。

工作量证明机制（PoW），我们可以叫它做范进中举。范进用了大半辈子学习一种无用的八股文写作，如同比特币矿工用算力来算题，关键是算的题是毫无意义的。有朝一日，运气好，就飞黄腾达了。一朝权在手，便把令来行：算出题的矿工有权打包所有他认可的交易。

PoS 是用户要预先放入一些利益，这是不是很像我们现实世界中的股份制。人们把真金白银兑换成股份，开始创业。谁的股份多，谁的话语权就大。

DPoS 机制，特别像我们的董事会。选举出代表，代表股东的利益。被选出的代表，一般来说，成熟老练，阅历丰富。不但能快速的处理日常事务，同时也能很好的保护股东的利益

Paxos, Raft, PBFT 则很像我们生活中的操练队列，通过互相间的消息，口令来达成一致。每排的排头作为 Leader，而每排的其余人都以排头为目标，调整自己的行动。 瑞波共识算法，初始状态中有一个特殊节点列表。就像一个俱乐部，要接纳一个新成员，必须由 51%的该俱乐部会员投票通过。共识由核心成员的 51%权力投票决定，外部人员则没有影响力。由于该俱乐部由“中心化”开始，它将一直是“中心化的”，而如果它开始腐化，股东们什么也做不了。与比特币及点点币一样，瑞波系统将股东们与其投票权隔开，并因此比其他系统更中心化。

如果我们去看兰博特（Lamport）的关于分布式系统共识的论文，就会发现论文是以议员，法案和信使作为阐述理论的样例，读起来不太像一篇计算机论文。

在此可以做一个总结了。传统的纯正的计算机算法对分布式系统的拜占庭问题已经无处着力了（参考 FLP&CAP 定理）。所以在分布式系统的研究中引入了一些社会学的理论和概念，包括上述的博弈论，生物学原理等等。我们可以把每一个计算机节点想象成一个单元。而计算机网络就是一个一个单元组成的社会，我们该如何给这个计算机节点组成的社会设计规则呢，以保证：

- 少量节点太慢，或者故障崩溃的情况下，整个网络还能输出正确的结果
- 整个网络的响应不能太慢。买一杯咖啡要等一小时是不可接受的
- 计算机网络出现分区（网络上的某些节点和其余节点完全断开）的时候，仍然能够稳定输出正确的结果
- 整个系统能够稳定的运行，输出稳定的结果

我们可以借鉴人类历史上的社会机制，激励机制，达成上述的功能。我们有理由相信，互联网或者分布式网络系统与现实的社会运作有着千丝万缕的联系，正因为如此，区块链的发展并不是冥冥之中的产物。

1.9 有趣的争论（关于 POS 和 DPOS）

2017 年 7 月 29 日 Vitalik Buterin 现身深圳，第一次作为以太坊的代言人来到中国做相关宣传。期间，V 神回答了大家感兴趣的对以太坊交易量拥堵问题和对 EOS 的看法。引起了一场论战。下面收录这段有趣的争论，作者不持立场，读者自己分辨与思考。注释一下，EOS 采用的 DPOS 共识，而 V 神是为以太坊的 POS 算法 Casper 站台的。

提问 ：
如何理解您之前提到的"EOS 是中心化的"这一说法？

V 神回答：
1、ETH 拥有很多 EOS 没有的协议特点。其中之一就是默克尔树（merkle tree），默克尔树的一个好处是可以单独拿出一个分支来对部分数据进行校验，如果你要确认某个交易，不需要很多的电脑来进行验证。EOS 却不可以，这就意味着你在 EOS 的网络中，你没有全部的节点信息，而必须去相信整个节点。

2.、EOS 的整个节点数很少，运用的共识机制是 Dpos（股份授权证明），在 Dpos 共识机制下，整个网络中大概只需要 100 个节点，这些节点是所有节点投票产生的，所有的节点都必须相信这 100 个节点，这就是为什么 EOS 能够处理更多交易的原因，因为整个网络只有 100 个节点，每个节点的运算储存能力都很强，这是实现可扩展性的途径之一。正是由于整个网络中只有 100 个节点，这个系统就显得更中心化，这些节点很容易受到攻击，例如公司或者互联网供应商（ISP），政府可以以各种理由关闭它们。所以如果你想通过使得网络中的节点变大实现可扩展性，这样整个网络的节点数量越少，这个网络就越中心化。

隔日，EOS 做出回应，原文参见
https://steemit.com/eos/@dan/response-to-vitalik-buterin-on-eos

接着，Vitalik 开始攻击 DPOS 机制：

EOS 的完全节点的数量将会更小。所以 Dan Larimer 用了 dpos 的概念，他说在 dpos 中，网络只需要 100 个节点就够了，只需要这些完全节点能达成共识就行了，其他人就只是一个轻客户端。它(EOS)说它能处理更多交易的另一个原因，是这些完全节点需要的条件(带宽，算力)都更高。这是实现可拓展性的一个方法。

但问题是，如果你只有 100 个节点，那这个系统就更中心化。你可以对它们实施 dos 攻击。由于完全节点是选举产生的，被选出的节点，大家都知道它们是谁。这样一来，要对它们进行攻击就容易很多。isp 可以轻而易举地关掉它们，公司可以轻而易举地关掉它们，政府可以轻而易举地关掉它们。这种实现可拓展性的特殊方法，成本也会非常的高，这个成本就是，如果你通过缩小节点的数量，提升节点的能力来实现可拓展性，这会让系统变得更中心化。

他的观点基本就是：EOS 的完全节点数更少，他们很容易被辨识出来，也就很容易被政府关停。他还说，通过使用性能更强的节点来实现可拓展性，会让系统更中心化。这就涉及到一个比较的问题了，”比谁更中心化“。

这是以太坊的区块生产节点分布图。你可以看到，两个矿池控制了 51%的哈希算力，它们可以任意忽视其它所有矿池生产的区块。

而且，以太坊的完全节点也都是经过加强的，普通大众根本承受不起。所以，几乎所有的轻客户端根本不需要操心默克尔证明的问题，虽然 Vitalik 说默克尔证明多么的有价值。

对于区块生产者来说，事实是，以太坊和其它协议，都比 dpos 区块链更中心化。

黑市

Vitalik 最后认为 dpos 会被政府，isp，和企业轻而易举地关掉。这种观点是建立在对去中心的错误假设之上的，我在上面已经证明了。事实是以太坊和比特币都遭受过 dos 攻击，而 steem 和 bitshares 则运行良好。正如上图显示的那样，以太坊中 7 个节点的哈希算力就达到了整个网络的 90%，把这 7 个节点拿掉，就能轻松摧毁以太坊。

真正的问题是，所有这些公共区块链都依赖于点对点的发现过程。世界上的政府和 isp 清楚地知道每个以太坊节点在哪，也就可以轻松地把它们关掉。

我们已经说过，以太坊的完全节点在今天的情况下已不切实际。这意味着那些真正的应用必须依赖于公共的 api 终端。最近的 EOS 代币分发应用，就足以停掉所有的公共 api 终端(endpoint)。

政府关停的威胁，主要是基于非法活动的假设。我们认为，对于合法的区块链应用来说，不需要担心被关停的风险。区块生产者和 api 终端可以自由的设置。

1.10 知识点导图