区块链的共识机制

冯云飞

金融实验班18

2021.10.16

区块链：去中心化账本

- 去中心化也许是区块链技术**最为核心、前卫**的理念
- 在这台巨大的“超级计算机网络”中，每个人都是“超级计算机”的组成部分
- 典型应用：中心化金融VS去中心化金融
  - 如：去中心化换汇交易所Uniswap

怎样建立一个去中心化系统？

- 去中心化系统需要大家每个人的**共识**
- **共识算法（Consensus Algorithm）**——将“**大家都认可做同一件事**”抽象成计算机的语言
- 简单的“投票”可靠吗？

经典问题：拜占庭问题

- 拜占庭问题
  - Leslie Lamport在1982年提出的虚拟模型，用来解释**一致性**问题。拜占庭作为东罗马帝国的首都，地域辽阔，在首都周边有众多将军负责城防，将军之间通过信使来传递消息，达成某些一致的决定。但由于将军中**存在叛徒**，叛徒会想尽一切办法干扰**一致性**的达成，甚至是达成叛徒想要的共识从而实现攻击。拜占庭问题，假设节点总数是N，叛徒将军数为F，则当 N ≥ 3F+1 时，问题才有解，共识才能达成
- 三位将军的例子
- 四位将军的例子
- 九位将军的例子

一种算法——“国王算法”

算法 4.14 国王算法 (King Algorithm) ($f < n/3$)

```
1: x = 本节点的输入值
2: for 从第 1 到第 f+1 个阶段 do
     第 1 轮
3:   广播 value(x)
     第 2 轮
4:   if 接收到 value(y) 至少 n − f 次 then
5:     广播 propose(y)
6:   end if
7:   if 接收到 propose(z) 至少 f 次 then
8:     x = z
9:   end if
     第 3 轮
10:  设节点 v_i 是预先确定好的第 i 阶段的国王
11:  国王 v_i 广播它当前的值 w
12:  if 接收到 propose(x) 的次数严格少于 n − f then
13:    x = w
14:  end if
15: end for
```

引理 4.15. 算法 4.14 实现了全部相同有效性。

证明. 如果所有好节点初始时拥有相同的输入值，则好节点们在第 2 轮都会提议 (Propose) 这个值。所有好节点将会接收到至少 $n - f$ 个提案 (Proposal)，因此所有好的节点将保持这个值，并且不会切换到国王的值。这个结论对所有阶段都适用。

引理 4.16. 在 $n > 3f$ 的情况下，如果一个好节点提议 x，不会有其他好节点提议另一个值 y ($y \neq x$)。

引理 4.17. 至少存在一个阶段，该阶段的国王是好节点。

引理 4.18. 当 $n > 3f$, 如果某一轮的国王是好节点，所有的好节点们在这轮之后都不会改变它们的值 v。

截自《区块链核心算法解析》（Roger Wattenhofer， 2017）

为什么区块链需要共识？

- 想象在区块链系统中，如果有叛徒……
- 区块链中的经典问题：“分叉问题”

比特币区块链对拜占庭问题的回答

- 每个挖矿节点收到定量的交易后，将这些交易打包（新区块），并整合其他某些必要信息，得到区块头c，包含version、perhash、merkleroot、ntimenbits等等……
  - 通常认为每个正常的矿工的c是一样的
- 大家开动显卡（或ASIC、CPU等硬件），穷举探索这样一个数学难题的答案：

$$SHA-256\big(SHA-256(c,x)\big) < 2^{224}/d$$

- 其中，d为难度系数(思考：d越大，此题越难还是越简单？)
- x为难题的答案
- 谁先解出x,此节点将会把这一消息告诉全世界：“我是矿工XX号，我的答案是YY”
- 大家将答案x代入原式，若成立，大家就达成了共识，大家将新区块链接在自己的账本末尾。区块奖励和矿工费全部交给矿工XX号
- 所有人将**互联网中最长的比特币区块链**当作正确的区块链
  - 为什么我是正直的矿工，却计算得到不满足该式？
    - 在收集交易信息的时候出了些差错，导致c和大多数人不一致

散列函数

- 给定x，计算是$f(x)$很容易的
- 给定$f(x)$，计算x是极其困难
  - 在实践上甚至不可能
- 对于$x1 \neq x2$，$f(x1) \neq f(x2)$
  - 或概率极低
- 对于$f(x1) \neq f(x2), x1 \neq x2$
- 演示

天地玄黄，宇宙洪荒	fdae5d93b5da6802e71c8420055455312 9e85e4119f61a19e82cbdd08bef1f98
天地玄黄，宇宙洪荒.	353ec487a6ad1cf7bf4fac2cdd0f4e9d979224e0f88e149f05b33c5b8b1e94463

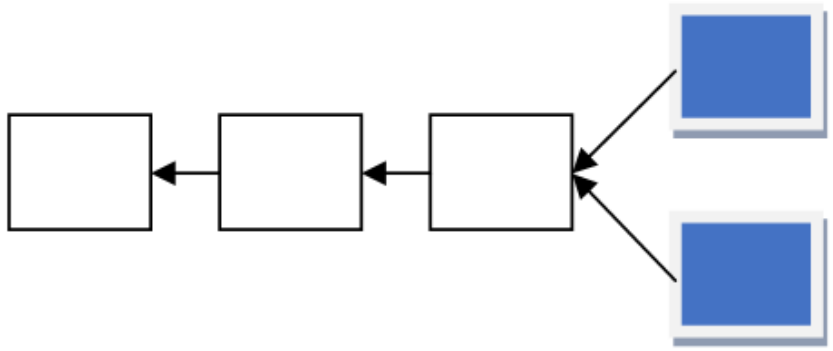
比特币区块链对拜占庭问题的回答

- 每个挖矿节点收到定量的交易后，将这些交易打包（新区块），并整合其他某些必要信息，得到区块头c，包含version、perhash、merkleroot、ntimenbits等等……
  - 通常认为每个正常的矿工的c是一样的
- 大家开动显卡（或ASIC、CPU等硬件），穷举探索这样一个数学难题的答案：

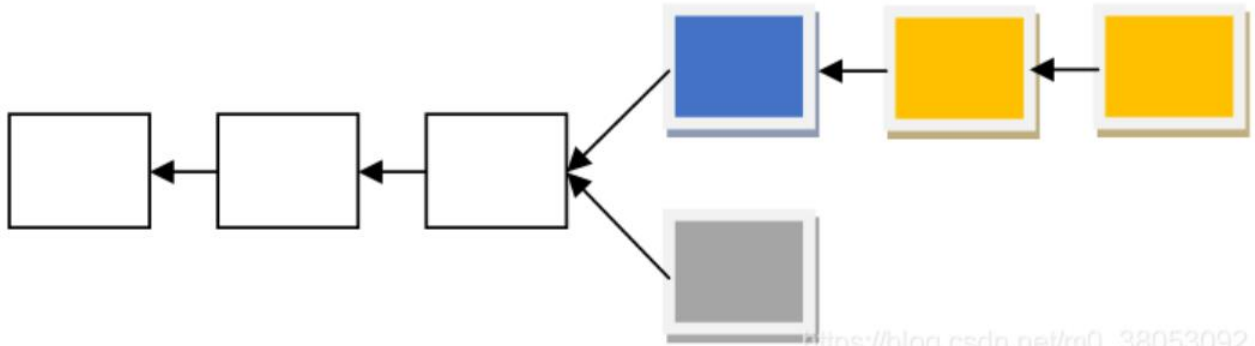
$$SHA-256\big(SHA-256(c,x)\big) < 2^{224}/d$$

- 其中，d为难度系数(思考：d越大，此题越难还是越简单？)
- x为难题的答案
- 谁先解出x,此节点将会把这一消息告诉全世界：“我是矿工XX号，我的答案是YY”
- 大家将答案x代入原式，若成立，大家就达成了共识，大家将新区块链接在自己的账本末尾。区块奖励和矿工费全部交给矿工XX号
- 所有人将**互联网中最长的比特币区块链**当作正确的区块链
  - 为什么我是正直的矿工，却计算得到不满足该式？
    - 在收集交易信息的时候出了些差错，导致c和大多数人不一致

在比特币区块链中，如果有“坏人”……



- 坏人声称：“我的账本才是正确的！”
- 坏人给不出他的x，就不会有人相信他。在比特币区块链中，大家只认同那个数学难题的答案。坏人必须付出50%以上的算力追赶正确的区块链



比特币区块链共识机制的问题

- 比特币区块链的问题
  - 为了达成共识，牺牲了太多效率
  - 电力消费太大

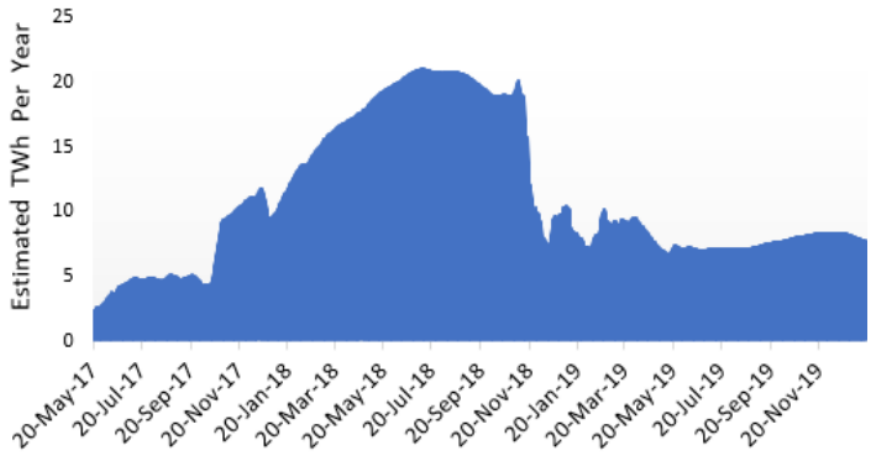


Figure 9: Ethereum energy consumption over the last year.

现有的一些共识机制

- Proof of Work
  - 起源：Dwork and Naor(1993)
    - 用PoW解决垃圾邮件问题：发件人需要在发信时解答数学问题
  - Compute-bond PoW
    - 类型：CPU、GPU、ASIC
    - 实例：Bitcoin
  - Memory-bond PoW
    - 类型：主存（内存）
    - 实例：Bytecoin、Monero……
  - Chained PoW
    - 实例：Ethereum
  - FileCoin

现有的一些共识机制

- Proof of Stake
  - 起源：某论坛(2011)
  - “Proof of Stake 的世界里没有矿机，不再是消耗大量的电力创造工作量证明进行挖矿，取而代之 PoS 里是用币来挖出更多的币，币可以类比为 PoW 里的矿机，谁拥有的币越多谁能挖出区块获得奖励的概率就越高”
  - 在 POS 权益证明共识机制裡有个专有名次叫做币龄。在 POS 权益证明共识系统中的每个货币每天都会产生 1 币龄，若你在权益证明机制中拥有 100 枚货币并存放了 10 天，你的币龄就为 1,000。若你成功被系统挑选出挖掘新区块，你的币龄会归 0 并重新开始累积计算，你会获得的奖励公式如下：
    - 奖励 = 币龄 * 年利率 / 365
    - 意味你每被清空 365 币龄即会从区块中会得 N% 年利率的货币奖励。假使在一个当前年利率为 5% 的系统中，你每成功帮忙打包一个新区块会获得的奖励为 1,000 * 5% / 365 = 0.137 个系统货币
- DPoS：大家用代币投票选出记账人
  - 实例：EOS

现有的一些共识机制

- 公有链与联盟链
- 联盟链中的共识机制
  - Hyperledger Fabric
  - FISCO BCOS