2.6 共识算法和工作量证明机制

01 共识机制

共识，从语文的角度进行理解，即许多不同的人对同一件事情达成一样的或者至少说方向一致的看法。这个解释同样适用于比特币网络。

解决两个问题：

- 不同的人在这个网络里对应的是什么？
- 他们需要对什么东西达成一致的看法？

02 共识机制

- 达成共识的主体

当前这个区块链中的一些节点，到底哪些节点需要达成一致，这是一个需要考虑的问题。

- 对什么达成一致

共识机制涉及了区块该如何生成以及生成之后如何选择的问题。

——区块和交易

在区块链当中，每个节点都是平等的，没有一个中心机构的存在，因此这时候就需要通过共识机制来达成节点间的一致。

03 共识算法

是为了达成共识所依据的一种规则，是筛选出具有代表性的节点的方法。

为此，区块链设计了一定的底层算法，通过这个特定的算法来选出那个可以生成新区块的节点，同时对于每一笔在这条区块链上进行的交易是否准许完成进行了约束和规定，也就是共识算法。

共识算法规定了，下一个新区块由哪个矿工生成，同时，在这条区块链上一笔交易要达成，需要被共识算法选出的部分节点达成一致的观点。

04 共识算法

- PoW（Proof of Work，工作量证明）
- PoS（Proof of Stake，权益证明）
- DPoW（Delegate Proof of Work，委托工作量证明）
- DPoS（Delegate Proof of Stake，委托权益证明）
- PBFT（Practical Byzantine Fault Tolerance，实用拜占庭容错算法）

05 工作量证明机制

工作量证明（PoW， Proof of Work），用来确认你做过一定量的工作。

如何证明呢?

通过工作结果——当用户做一定难度的工作得出一个结果时，通过展现出这个结果证明你完成了一定的工作量——也就是说PoW作为一种共识机制，它是**结果导向**而非过程导向的。

监测工作的整个过程来证明工作量，这是极为低效的，为了保证一定的效率，就设计了通过验证结果的方式证明工作量的方法。

06 工作量证明机制

特征：不对称性

进行实际工作的用户需要付出很多的工作量才能得到一个符合已定条件的结果，但是作为验证方却可以根据用户提供的材料很容易重算这个结果，同时验证这个结果是否满足提前规定好的条件。

07 工作量证明机制

举例：给定一个基本的字符串"Hello, world!"，我们给出的工作量要求是，可以在这个字符串后面添加一个叫做nonce的整数值，对变更后的字符串进行哈希运算，如果得到的哈希结果是以"0000"开头的，则验证通过。为了达到这个工作量证明的目标，我们需要不停的递增nonce值，对得到的新字符串进行哈希运算。按照这个规则，我们需要经过4251次计算才能找到恰好前4位为0的哈希散列。

```
"Hello, world!0" => 1312af178c253f84028d480a6adc1e25e81caa44c749ec81976192e2ec934c64

"Hello, world!1" => e9afc424b79e4f6ab42d99c81156d3a17228d6e1eef4139be78e948a9332a7d8

"Hello, world!2" => ae37343a357a8297591625e7134cbea22f5928be8ca2a32aa475cf05fd4266b7

...

"Hello, world!4248" => 6e110d98b388e77e9c6f042ac6b497cec46660deef75a55ebc7cfdf65cc0b965

"Hello, world!4249" => c004190b822f1669cac8dc37e761cb73652e7832fb814565702245cf26ebb9e6

"Hello, world!4250" => 0000c3af42fc31103f1fdc0151fa747ff87349a4714df7cc52ea464e12dcd4e9
```

08 工作量证明机制

所以，在工作量证明中，你工作的时间越长，工作时采用的设备越先进，你的工作量就越高，你收获的也会越多。虽然短期看可能有运气因素，但是宏观长期来看是公平的，谁工作付出的多，谁得到的就多。

09 工作量证明机制

在比特币网络中，首先，生成要加入到区块链中的一笔新的交易信息，也就是新区块时必须满足的要求。然后在基于工作量证明机制构建的区块链网络中，所有节点开始不断尝试和计算，直到第一个节点找到了那个随机哈希散列的数值解，于是它就得到了生成新区块的权利。

优点：完全去中心化

缺点：1. 挖矿行为造成了大量的资源浪费

2. 达成共识所需要的周期较长