以太坊的未来

前言

> 凯文·凯利在《失控》一书中像我们描绘了蜜蜂是如何自下而上的构建出复杂智能的蜂群系统，他对于分布式系统有着深刻而富有远见的理解，区块链技术正是借助于分布式赖以存在，而要理解以太坊的前提是理解区块链。

让我们先闭上眼睛回到人类文明的早期阶段，想象一个猎人狩猎归来，他收获颇丰，在饱餐一顿后，他手里的鹿肉还有一大半。如果置之不理，一晚上就会腐烂。于是他想出一个好注意，用鹿肉来跟同伴交换兔肉。

第一次交换进行的很顺利，他吃到了从未吃过的兔肉。并立即喜欢上了兔子的味道，于是开始频繁的交换。久而久之，他想要交换更多的东西，但他也发现了一个问题，不是所有的人都愿意用鹿肉来跟他交换，有人喜欢兔肉，于是他用鹿肉换来兔肉，再用兔肉来交换他喜爱的铃铛。

但这样的交换成本太高，而且并不总是幸运的交换到自己喜欢的东西，是不是存在一种东西，大家都愿意用自己手上的东西跟它交换呢？没错，**它就是货币**。

货币的一个重要的特征就是流通。它建立在所有人都**信任**的基础上，并且货币应该是极难伪造的。 当货币在流通的时候，遇到的一个挑战是，如何防止货币被伪造。因为总有人想要不劳而获，通过伪造货币来交换想要的东西，一旦货币的制造成本很低，那么，通货膨胀将慢慢的蚕食掉货币的价值，最后变成一文不值的东西。同样没有完善的验证机制，每个人都可以私自制造货币，最终货币泛滥，再也没有人愿意用自己手里真真实实的大米来跟它交换了。所以，货币的另一个重要的特征就是，可以分辨真伪。

> 假设有一个账本，记录了 [张三持有1000W BCT] 这样的信息。大多数的参与者都认可这条信息，证明张三确实持有了1000W的BTC(可以把BTC理解为一种数字货币，其实就是比特币)。

现在，我们的目的就是让绝大多数人都认可这个账本。我们可以使用投票机制，假设有100台计算机都各自持有一个账本，其中51台计算机记录了[张三持有1000W BCT] 这样的信息，49台持有[李四 持有1000W BCT]这样的信息。那么，我们认为，张三拥有1000W BCT这条信息是合法的。

这里有一个明显的漏洞，当有不怀好意的人控制了51台计算机，那么，他就可以篡改信息。（著名的**拜占庭将军问题**）。

现在，我们把所有的交易都想象成一个巨大的账本，这个账本记录了所有的历史交易。在最开始的时候，在这个账本上记录 [张三得到1000W BTC] 这条的信息。 现在大家都认同了这个账本。

接下来，交易开始，张三持有了1000W。

这个账本将自己的信息通过广播的形式发送到互联网，所有的其他客户端都接收到这个信息，并进行验证。验证成功后，将自己的账本更新为最新的。

每个客户端都想写入一笔交易到账本上，并且他要获得绝大多数客户端的认可。几乎每个客户端（当然会有不怀好意的客户端私自修改了算法）都运行着一样的算法检测。当客户端获取到最近的交易信息的时候，他会试图解决一个数学问题，来证明自己的运算能力更快。

以比特币为例

证明方法是：

Hash(历史的交易信息+当前的最新需要打包交易+随机数）生成的Hash满足某个特定的条件，这个条件决定了获取这个Hash的难易程度。很明显，在这个Hash算法中，历史的交易信息和当前的最新交易，对于客户端来说都是共享的，他们都持有这样的信息，唯一不同的是，他们要在其他客户端算出满足条件的Hash之前算出它，并打上时间戳，广播出去。

这样一组包含了hash和交易的数据结构就是一个区块。作为奖励，你可以在里面增加一条你拥有25BTC的记录。其他客户端在接收到你的广播后，会验证是否有效。而这样你就拥有了25个BTC（如果你私自改动这个值，其他客户端将会拒绝这个区块，并且这个值在每四年会减少一半）。

但2140年之后，打包区块将不会获得BTC。因为总量被设置了2100万个被分配完毕，收益只是交易创建者支付的手续费。这里每个CPU都真实的付出了劳动（proof-of-work,POW），它们消耗了电力和资源来产生一个满足要求的随机数。并没有任何投机取巧的办法能获得这个随机数，唯一的方案就是暴力遍历匹配。

POW的灵感来自于Hashcash,它很难被找到但很容易被验证。

所以，最快算出结果的那个客户端，就创建了一个区块。这个区块里包含了若干条交易记录。 比特币的区块大约每十分钟产生一个。一旦A接收到来自B的广播，声称自己发现了新的满足要求的随机数，A就会进行验证，通过后（指得到了一定数量节点的确认后），将新的区块连接到旧的区块上，并将这个消息广播出去。而连接若干个的区块形成一个链条。

当诚实的节点站绝大多数的时候，这个由于无数个计算机节点构成的P2P网络，是可以信任的。而每一台客户端都拥有一个完整的账本。比特币本身代表了互联网架构下无数人构建起的、没有中央集权的信任网，这个信任不被任何政府和机构控制。所有的节点

都扮演了监督的决策，每一笔交易都有迹可循。就像是对黄金的信任是因为黄金极难获得且难以伪造，比特币需要消耗大量的计算才能获得。

当一个比特币被重复支付的时候，究竟哪一条交易被接受取决于被最先加入到链上的区块包含的交易信息，剩下的交易信息将会丢弃。

接下来，让我们比较正式的了解一下下面的概念。

什么是区块链

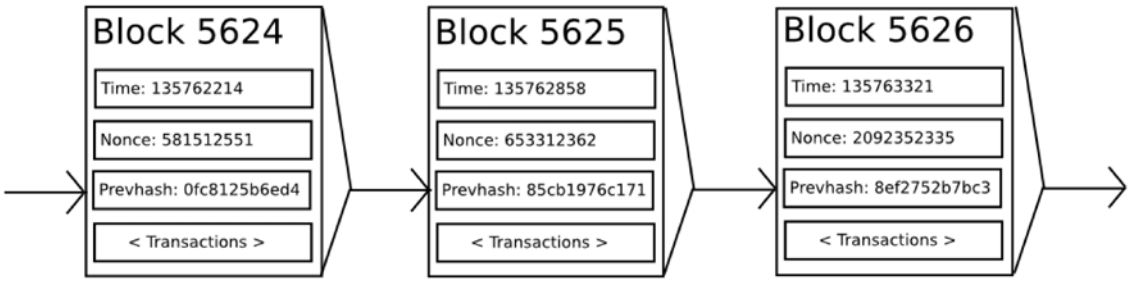
区块链是一个分布式的”账本”，每一个区块都是一些交易的集合，所有的区块按照时间顺序（timestamp）被某个Hash值串连在一起，形成一个不可篡改的交易历史。

Hash的计算是这样的：

> 获取需要打包的历史交易，之前区块的Hash,并进行Hash函数计算，求出一个新的Hash值，是的这个值满足特定前缀（如必须以0000开始）。

意思是，求出的这个Hash值满足某个特定的条件，要满足这个条件的唯一办法就是不断的尝试使用随机数来撞击。最先找到这个Hash的节点将获取打包区块的权利。它有权利凭空捏造一笔交易发送给自己（这笔交易将被其他节点承认，但数量是受到限制的），而其他矿机都将验证这笔交易。如果验证通过，则将区块包含追加到历史区块上，进行下一次数学计算的赛跑。这些区块，通过Hash首尾相连，就是区块链。

Hash的计算需要消耗大量的算力，并且真实的消耗了电力，所以被称做挖矿。而这个过程是Proof of Work(PoW)共识机制。



什么是数字货币

在知道了什么是区块链之后，现在需要了解一下货币或者说数字货币究竟是什么东西。跟你猜测的一样，数字货币本身并不存在，简单点，它只是打包区块的一个数字奖励而

已（比特币的货币总量被设置为2100万个），但注意，它本身包含了自身存在的证据，即存在于区块链的共识历史证明了它的存在。

货币的存在是基于共识的，它并没有真实的物理实体做支撑，比如黄金或是白银。我们信任人民币或是美元，同样也是对贵重金属的信任。

在很长一段时间，达成一个共识是需要中央权力的介入，比如央行。所有的人都信任它，就可以产生交易。但是带来的问题是，任何一笔交易都需要经过一个中央机构确认。

一笔交易由A->B的过程其实至少经历了**A->C->B**，可能中间的确认环节会更多。

数字货币的诞生并不是简单的臆造一个数字，如果一个节点宣称自己拥有了1000BTC（比特币），它将这个消息向网络广播出去，其他节点在接收到这个信息后并不会承认它。因为缺乏足够的证据，此时网络没有对此交易达成共识，验证失败，信息被丢弃。因为基于PoW的共识机制，产生的Hash很难被破解，但很容易被验证。

什么是以太坊

> 在确保你理解了区块链和区块链是如何产生货币的之后，我们现在来讨论以太坊（ETH）。

如果是比特币是一个巨大的账本，那以太坊则是一个应用平台。人们可以在以太坊上构建应用分布式应用，以太坊是可编程的区块链。

以太坊的基本单元是账号而非交易，任何的资产的转移都是通过账号来激活的，

账号分为:

- 合约账号
- 外部账号

> Tips: 在比特币的区块链中，维护这一个简单的交易列表，这一点在以太坊中变得复杂了。以太坊提供了一个外部账号，这个账号被私人的密码控制（这段密码用来加密你的私钥，而你的私钥是唯一可以确认你对这个地址的所有权）。 一些代码会被存储在合约账号里面，这些代码被称做智能合约。

智能合约

> 智能合约即在一个图灵完备的编程环境下（EVM，以太坊虚拟机）执行的一些代码。这些代码像是现实生活中的合约一样，但不需要在物理世界中的那样，人工

> 的干预或是中央机构的介入。只需要在关键的时刻由外部账号或是一个合约来激活该合约即可。

智能合约的概念在1995年就已经被密码学家提出了，但直到区块链的出现，才使得这种思想得到了应用。这是因为，智能合约依赖的共识只有在区块链出现后才得到真正的解决。在这之前，一个分布式的网络环境中，各个节点无法真正的在不信任彼此的情况下达到共识。

智能合约的本质就是一些可以访问区块链数据的代码，这些代码被部署在区块链上。可以在一个分布式环境下正确的执行。

智能合约一般采用一种叫Solidity的编程语言编写。Solidity的详细介绍在这里，你可以很快的浏览一下其中的语法。

一个非常简单且没有意义的智能合约如下：

```
prpragma solidity ^0.4.0;

contract Hellosmartcontract{

    address owner;
    function whoami () returns(address){
        owner=msg.sender;
        return owner;
    }
}
```


在这个简单的例子中，采用了solidity来编写一个智能合约，通过一个whoami接口，返回一个账号地址。虽然这没有任何意义，但足以说明智能合约是什么东西，它将被编译成字节码最终在EVM上运行。你可以把这个智能合约部署在区块链上，当然，你得负担一部分的Gas。

通过Web3.js 的API 你可以访问合约的接口，这里是whoami。

什么是Gas

Gas是一个非常形象的比喻，在以太坊中，部署的应用运行在区块链的共识引擎是需要消耗”汽油“的，就像是轿车发动那样。它就是Gas。Gas是以太坊生态中提出的一个概念。智能合约在执行的过程中，总是在消耗着算力或是内存。比如要进行一个Sha3操作，比如要存储等。 你必须的负担这部分的费用。这部分奖励将会被打包这个区块的矿机获取，作为它的奖励。Gas需要用Ether购买。

Ether是以太坊内部发行的一种货币（ETH，有各种单位，最小的单位是Wei，可以跟比特币类比），你可以用来支付一次合约所必须的gas。每一个合约能消耗的Gas是有上限

的，并且全网动态调整。如何在Gas消耗完毕后合约还未执行完毕，则将此次合约进行事务回滚，但并不会退还你已经消耗的Gas。

智能合约提供了一种能力可以改变区块链的中某个账号的状态，它可以指示状态是如何在区块链上透明的转移的。

DApp

基于智能合约的应用叫DApp(Decentralized App),DApp通过一组轻量级的API访问区块链。DApp是可视化的“智能合约”。比如，你可以简单的用它来发行自己命名的代币（ Token system ）并提供友好的客户端。

将应用部署在区块链上，借助智能合约可以做一些非常振奋人心的事，比如去中心化组织（ DAO ），它是一个令人惊叹的尝试。虽然在2016年遭遇了可怕的灾难：智能合约的漏洞被利用，数以亿计的资金被洗劫一空。

尽管如此，智能合约仍旧带来了不可思议的可能性，它通过共识引擎了来瓦解权利方程。金融交易的纠纷和确认不在依赖于某个机构，所有的信息都透明并且被大多数人承认。合约开源，任何人都可以审查它，提出改进的建议，等待大多数人的接受。它可以构建一个完全自给自足的市场，例如募集资金进行项目的投资，所有的交易将被平等的对待。

使用智能合约可以做包含但不仅限于以下的事件：

- 储蓄钱包
- 代币系统
- 对赌合约
- 对冲基金
- 风险基金管理
- 慈善
- 选举
- 婚姻合约
- ….

以太坊上的应用具有非常前瞻性的思路，它接受各种创新，并且依赖自身的机制完成闭环，比如在一个以太坊应用上产生的一些付费需求可以用以太币完成支付。显然，在金融方面，它带来的改变是非常明显和强大的。

区块链的未来

区块链对金融和经济的影响是变革性的，所有依赖共识或是中央信任的事物都能在区块链上找到一种新的可能性：即利用分布式可以分散权利，并且在一个系统中达到完全共识。

> 跟比特币一样，以太坊是区块链上另一个璀璨的实践。于此同时，基于区块链的新事物正在不断的发生。
>
> 没有人知道以太坊的未来在哪里，就像是没有人知道区块链上会诞生以太坊那样。
>
> 正如以太坊创始人杰弗里维尔克说的那样，”也许我们需要预言机”。

最后：

我希望你通过阅读本文对区块链和以太坊产生强烈的兴趣，就像是对所有的新事物那样抱有热忱。

常见的资料整理

- 官方blog，最新的消息都在这里：https://blog.ethereum.org/
- Etherenm:https://ethereum.org/

编程技术相关：

- web3.js Javascript API:

  http://forum.ethereum.org/categories/ethereum-js

- remix浏览器IDE 在线编译智能合约:

  https://remix.ethereum.org

- solidity github :

  https://github.com/ethereum/solidity

- ethereum github :

  https://github.com/ethereum

- solidity文档：

https://solidity.readthedocs.io/en/develop/

- solidity中文文档 ：

  http://wiki.jikexueyuan.com/project/solidity-zh/

- Gas小知识：

  https://github.com/ethereum/wiki/wiki/Design-Rationale#gas-and-fees