### DDA（Decentralized Data Assets）概述

DDA是基于 staking Proof Of Capacity的新型加密货币。其主要的特点是使用硬盘存储空间作为共识的参与者，采用 MHF（Memory Hard Function）哈希函数，抵制了POW的算力竞争，大幅降低加密货币对能耗依赖，绿色环保，低能耗，低噪音。同时采用硬盘挖矿模式，抗ASIC，无需专业设备即可挖矿，降低矿工参与门槛，让其生产方式更趋向去中心化方式，并更加安全可信，让人人都能参与到加密货币的开采。

* **低能耗**

与使用ASIC来挖矿的PoW矿工相比，PoC非常高效。据测试，PoC共识所耗费的能量平均是比特币交易的1/500。

* **低门槛**

PoW挖矿需要昂贵、专用的ASIC矿机或GPU，与此相比，PoC只需通过额外的笔记本电脑和外置HDD便能进行挖矿。

* **真正去中心化**

由于多余的储存空间很常见，硬件便宜，竞争也不那么激烈，更多的人可以参与到PoC挖矿中，这意味着网络是更加分散的。

## DDA链技术方案

容量证明（Proof of Capacity），也被称作空间证明，于2013年在空间证明（Proof of Space）白皮书中首次提出。PoC与PoW差别不大，事实上，它可以看作一种采用 MHF（Memory Hard Function，一种计算代价取决内存的哈希算法）的 PoW 协议。而在PoC中，这MHF的计算工作是在一个叫作“测绘”的过程中提前完成，然后再用此过程的结果验证每个区块，而不是花费巨大精力去验证每个区块。其基本思想是：在PoC中，繁重的计算工作只在测绘期间完成一次，而不是每一个新区块都要进行，使得PoC比PoW系统大大地节约了能源。矿工最终完成的“工作”量取决于他们能够投入测绘过程的可用磁盘空间。

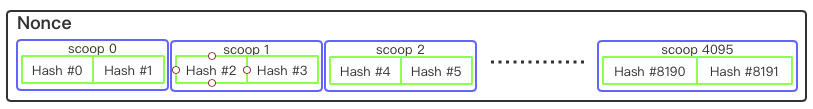
测绘（Plotting）使用的是一种很慢的哈希函数，名叫Shabal ，它和比特币使用的 SHA-256 哈希函数不同。因为 Shabal 哈希非常难计算，需要对它进行“预计算”（precompute），所以Shabal哈希很难做成ASIC矿机。这里我们采用256 bit的Shabal哈希函数Shabal256（）。

DDA公链采用SPOC（staking Proof of Capacity）共识机制。项目挖矿采用抵押Token挖矿机制。抵押标准暂定1TB容量空间需要抵押1000项目Token。

矿工的存储空间达到100%抵押标准，将会获得全部铸块收益，不然，铸块收益将会相应地打折。

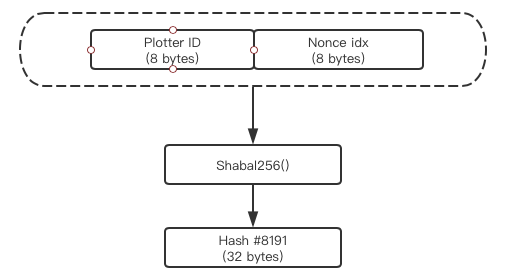
#### 创建Plot文件

生成Plot文档的过程称为测绘（Plotting），这些Plot文档储存了大量预先计算过的哈希。每个plot文档包含多个8192哈希组，这些组被称为随机数（Nonce）。单个Nonce的大小正好是256 kb（哈希为32字节，8192哈希组大小为256 kb）。另外，一个nonce又被分为4096对哈希，每对称之为一个scoop。具体可参见下图Nonce的结构示意图

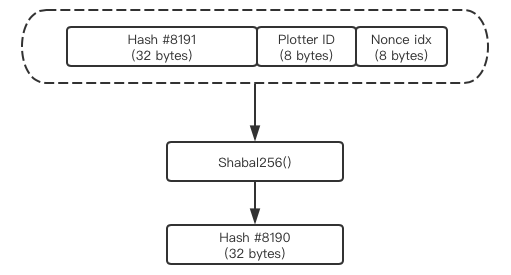


Nonce也可通过其索引号（index）来识别，这里采用64bit整型（8 bytes）来记录，范围是从0到264。

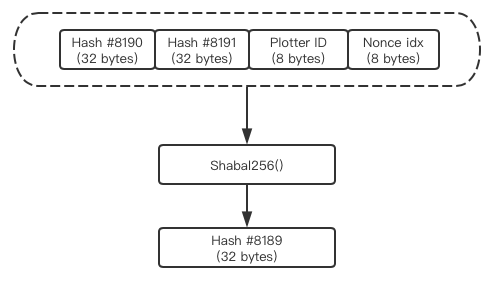
测绘过程需要输入矿工的测绘编号（Plotter ID），确保plot文档数据仅能由该矿工使用。



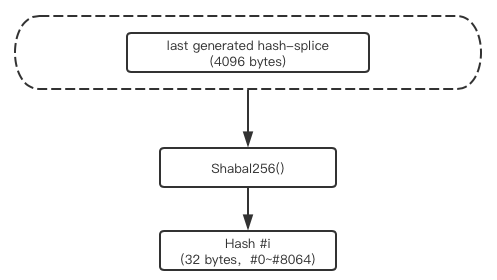
根据Plotter ID和Nonce 索引号，计算Shabal256哈希，得到Nonce中的最后一个哈希#8191。 将哈希#8191，Plotter ID， Nonce索引号作为下一轮shabal256哈希计算的输入，得到哈希#8190（如下图所示）。



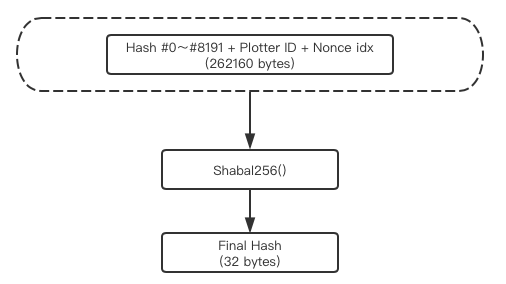
将最新所得的哈希，前置附加到上一轮shabal256的输入，作为新一轮shabal256函数的输入（输入字符串长度小于4096bytes时），依次可以计算得到哈希#8189～#8065。



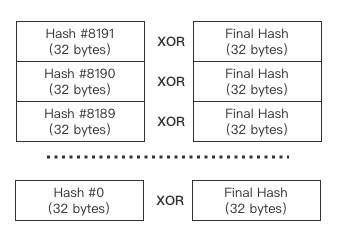
从计算哈希#8064开始，不断前置附加的输入字符串长度超过了4096 bytes，则保留最新产生的4096 bytes作为shabal256哈希计算的输入，依次迭代计算，得到哈希#8064～#0。如下图所示。



当得到Nonce所有8192个哈希，可以通过以下来计算最终hash



Nonce 中最终保存的 Hash 值是Hash #0～#8191与最终Hash 的 XOR 操作结果。



这样，Nonce的所有最终数据获得。

#### 挖矿过程

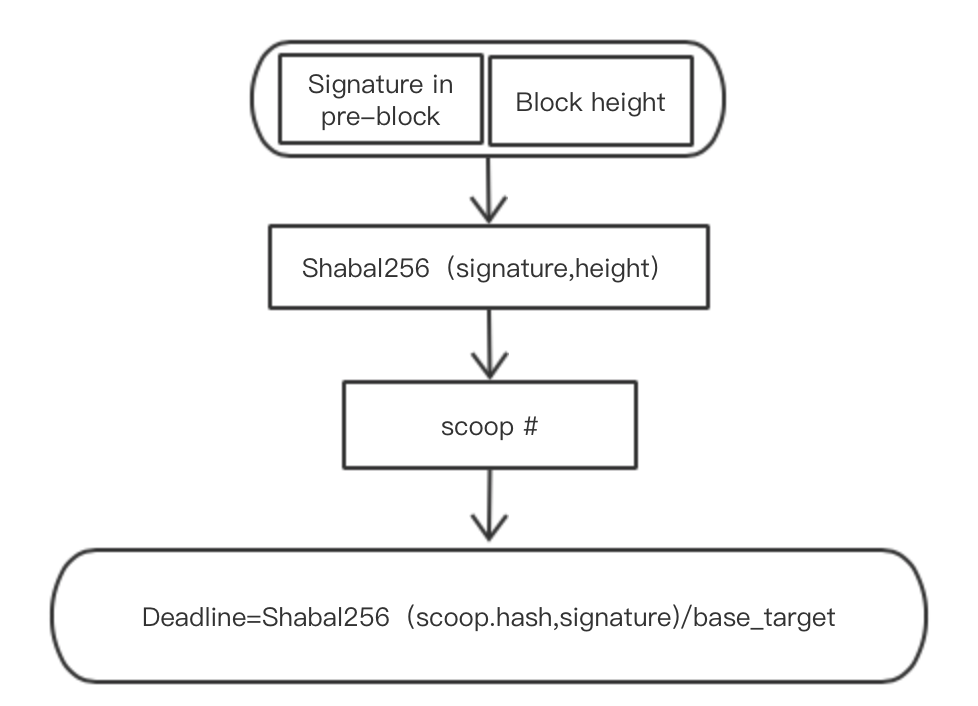
一旦生成了plot文件，就可以进入实际的挖矿过程。矿工首先获取当前最高区块的相关信息，包括生成签名（32 bytes），待铸区块高度，以及基于最后288个区块计算出的base\_target，被认为是区块的“难度等级”。

Base target

Base Target 是根据最后24小时出块情况（288个块）计算得出的铸块难度。 该值调整了矿工的难度。 基准目标越低，矿工越难找到数值小的deadline。 它的调整方式是尽量让DDA每个区块平均间隔时间为5分钟。

Deadline

当矿工挖掘并处理Plot文件时，最终会产生称为deadline的数值。 这些值表示在允许铸造区块之前，自上一个块被铸造以来必须经过的秒数。 如果没有其他矿工在这段时间内铸造一个区块，则该矿工可以铸造一个区块并获得区块奖励。

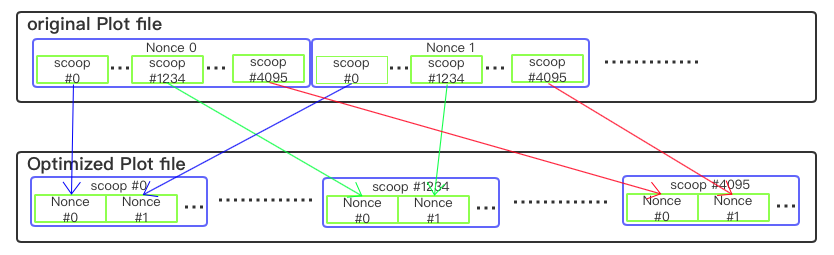


挖矿流程大致如上图所示：

1. 获取当前最高区块的生成签名以及待铸块高
2. 以生成签名和块高作为shabal256输入，计算得到的哈希作为随机数种子，产生[0,4095]的随机数，确定scoop的索引号。
3. 遍历该索引号的scoop中的所有hash，依次计算Deadline=Shabal256（scoop.hash,signature)/base\_target，选取最小的Deadline作为本轮的结果。
4. 若本轮的deadline幸运地成为全网最小，则负责打包交易签名，产生候选区块，向全网广播，并获得铸块奖励。
5. 不然，等待别人的新块，进行验证，确保接受最小的deadline作为新块。

#### 优化Plot文件结构

由上述挖矿流程容易知道：当确定本轮铸块所用到的scoop索引号后，需要遍历所有该索引号的scoop内的hash值。如下图所示：原始plot文件格式没有优化，为了读取在不同 nonce 中的同一编号的 scoop，需要依次访问不同的 nonce。很显然，这样的操作是耗时以及效率低的。将所有nonce按scoop索引来存储，当确定本轮铸块所用到的scoop索引号后，它只寻求一次并按顺序读取所有数据，因此比较高效，符合机械硬盘慢寻址特性。



|  |  |
| --- | --- |
| 代币总量 | 80亿枚DDA |
| 出块时间 | 5～6分钟 |
| 初始铸块奖励 | 4400枚DDA/区块 |
| 铸块奖励衰减周期 | 每80万个区块，铸块奖励减半。 |

## DDA代币分配方案

| 公链 映射规则 | |
| --- | --- |
| 总量 | 8,000,000,000 |
| 社区 | 10% |
| 团队 | 2% |
| 基金会 | 4% |
| 生态 | 4% |
| 矿工 | 80% |