POCETHEREUM系统说明

POWERED BY **POC LAB**, @2019

- 向poc共识算法的先驱burstcoin致敬
- 向智能合约的完美实现ethereum致敬

是什么：

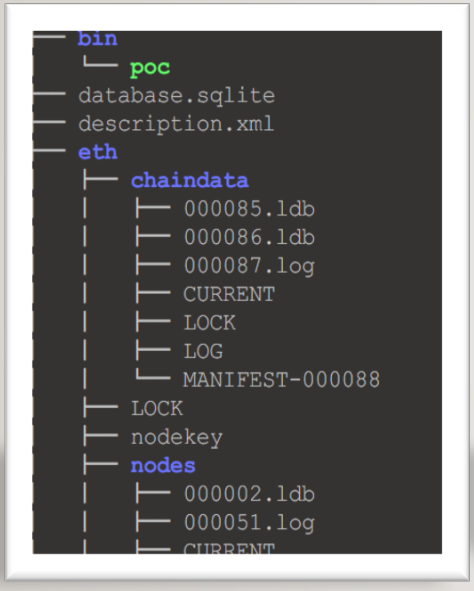
- 去中心化技术系统

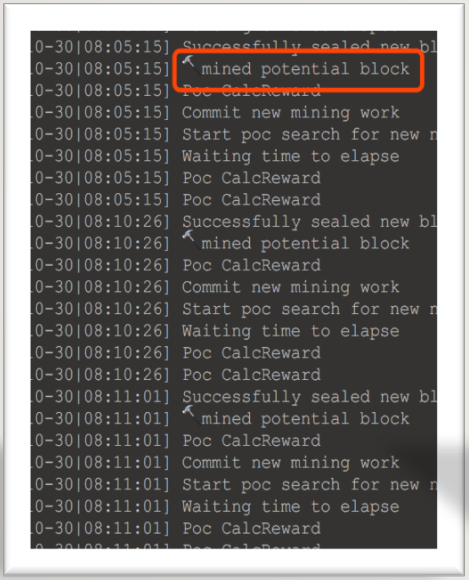
包含什么：

- 核心共识挖矿系统
- 外部生态辅助系统

• 系统组成

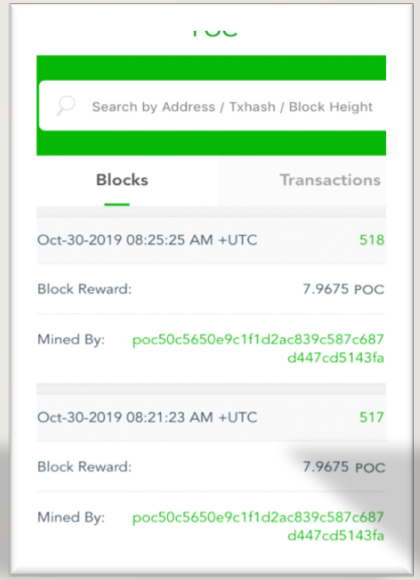
• POC共识挖矿系统【去中心化】





- ✓ POC共识挖矿主程
- ✓ 挖矿设备配置程序（搜索、P盘等）
- ✓ 控制台操作子模块

• 外部生态辅助系统



- ✓ APP基础钱包应用程序【转账、抵押】
- ✓ 区块链浏览器
- ✓ 桌面端管理程序【矿机管理】

1.	**POC共识为什么有效**
2.	挖矿奖励参数设置
3.	抵押系数如何起作用
4.	如何控制出块时间的均值为3分钟
5.	如何评估全网算力T数

- 核心原理讲解--1.POC共识为什么有效

POC挖矿过程：

过程一 P盘：利用用户的账户信息事先生成大量挖矿中间数据，然后存盘

过程二 挖矿：从磁盘读取挖矿中间数据，用来解算一定难度的数学难题

POC所设计的算法保证了：

那怕从最慢的磁盘读取挖矿中间数据也远比临时用CPU生成所需的时间少

- 读磁盘：3分钟时间可以读取20G以上的有效数据
- 临时算：3分钟时间只能计算出1M不到的有效数据

• 核心原理讲解--2. 挖矿奖励参数设置

- ✓ 3分钟出一个块
- ✓ 每4年减半
- ✓ 总量2.1亿

根据几何级数求和的特点

- ➢ 每天出块480，
- ➢ 每四年出块数：4*365*480
- ➢ 头四年出1.05亿

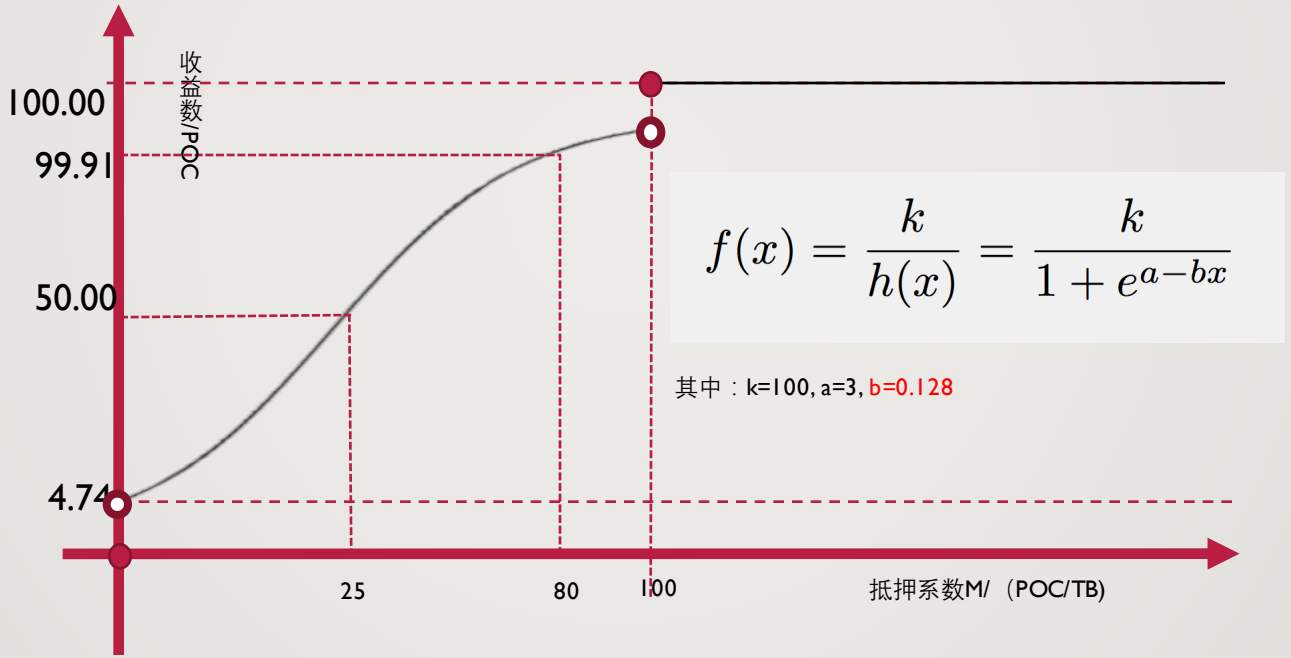
所有人抵押满后理论出块奖励数=$\frac{1.05亿}{4*365*480}=\frac{105000000}{700800}=149.8287\ldots\approx150$

假设抵押满率为：90%

理论出块奖励数 = $\frac{150}{90\%}=166.666\approx168$ (POC)

- 核心原理讲解--2. 挖矿奖励参数设置

抵押系数越高收益率越高



$$f(x) = \frac{k}{h(x)} = \frac{k}{1 + e^{a-bx}}$$

其中：k=100, a=3, b=0.128

注：以上计算按照k=100进行，最终k系数根据需求模型进行调整

• 核心原理讲解--2. 挖矿奖励参数设置

抵押系数越高收益率越高

观测抵押系数(POC/T)	128T抵押数(POC)	期望收益（%）	期望出块收益数（POC）	倍率
0	0	~~4.7426~~	~~7.967546693831220495~~	~~1.00~~
10	1280	15.1871	25.514355494318760265	3.20
20	2560	39.1741	65.812482834585580349	8.26
30	3840	69.8465	117.342156288426494370	14.73
40	5120	89.2832	149.995764158263284571	18.83
50	6400	96.7705	162.574361930660330700	20.40
60	7680	99.0806	166.455457498295629648	20.89
70	8960	99.7427	167.5676901101598730 29	21.03
80	10240	99.9283	167.879578004415236592	21.07
90	11520	99.9801	167.966500857588528106	21.08
~~**100**~~	~~12800~~	~~99.9945~~	~~167.9906846478465638 48~~	~~21.08~~
100	12800	100.0000	168.000000000000000000	21.09
	……			
~~**310**~~	~~39680~~	~~100.0000~~	~~167.999999999999971578~~	~~21.09~~
~~**312**~~	~~40000~~	~~100.0000~~	~~168.000000000000000000~~	~~21.09~~

注：以上计算按照k=100进行，最终k系数根据需求模型进行调整

- 核心原理讲解--3.抵押系数如何起作用

$$\text{抵押系数} M = \frac{\text{抵押数}}{\text{挖矿容量}} \cong \frac{\text{抵押数}}{\text{观测挖矿容量}}$$

- 用户提交区块的nonce值的期望正比于用户磁盘大小；
- 用nonce的观察值替代nonce的期望值是可行的；

共识抵押系数

$$M = \lambda \frac{Mof(addr)}{Storage|_{real}} = \lambda \frac{Mof(addr)}{Storage|_{estimate}} \quad (\text{EHD/TB})$$

$$= \lambda \frac{Mof(addr)}{E(Nonce) * 2^{18} B} = \lambda \frac{Mof(addr)}{nonce * 2 * 2^{18} B} \quad (\text{EHD/B})$$

$$= 2.0 * \frac{Mof(addr)}{nonce * 2 * 2^{18} * 2^{-40} TB} = \frac{2^{22} * Mof(addr)}{nonce} \quad (\text{EHD/TB})$$

注：λ用来调节系数大小，当前取λ=2.0，用以规避用户对抵押系数的困惑

- 核心原理讲解--4.如何控制出块时间的均值为3分钟

$$CalcDeadline = \frac{Target(nonce)}{Basetarget} = \frac{Target(nonce)}{C_{max64}/Difficulty} = \frac{Target(nonce) * Difficulty}{C_{max64}}$$

要求: $Target(nonce) = \frac{CalcDeadLine * C_{max64}}{Difficulty} <= \frac{MaxDeadLine * C_{max64}}{Difficulty} = \frac{180 * C_{max64}}{Difficulty}$

通过负反馈调节系统的难度值,
从而控制需要搜索的nonce的期望次数,
难度调整公式参考如下：

$$D_{i+1} = \begin{cases} GenesisDifficulty & \text{if } i < 5, \\ \frac{5*4*Deadline_{max}}{(\sum_{j=i-4}^{i} \frac{1}{D_j})*(T_i - T_{i-4})}\Big|_{max\ 10\ \%\ change} & \text{if } i < 25, \\ ..... & \text{if } i >= 25, \\ \text{REF: consensus/poc/poc.go(func CalcDifficulty)} & \end{cases}$$

• 核心原理讲解--5.算力估计问题

假设: 均匀分布 $N_{ns} \in (0, C_{max64})$

要求: $N_{ns} <= L = \frac{C_{max64} * Deadline_{max}}{B_{difficulty}}$ 等价于: $N_{ns} \in (0, L)$

一次尝试需要的存储是256KB，故求算力容量问题可转化为：

平均需要尝试 多少次(X次)，才能使上述要求成立。

$$X = \frac{C_{mac64}}{L}$$
$$= \frac{C_{mac64}}{\frac{C_{max64} * Deadline_{max}}{B_{difficulty}}}$$
$$= \frac{B_{difficulty}}{Deadline_{max}}$$

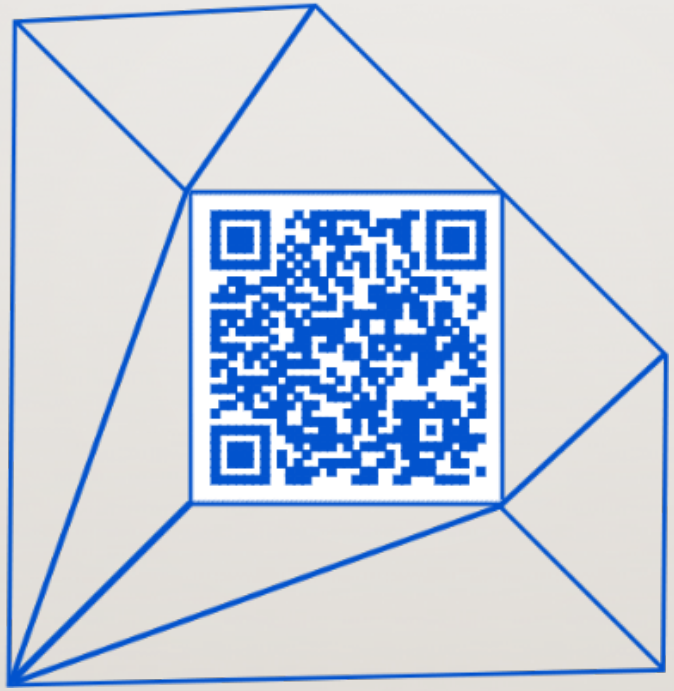
每Nonce大小为256KB

总算力：T = X * 256KB

算力估计

$$T = \frac{Difficulty * 256KB}{180}$$
$$= 1456 * Difficulty(B)$$

结束



https://github.com/pocethereum