

奖励算法模型

matrix工作

1.矿工奖励

- 输入：
 - a.分配金额M0，BigInt存储，单位wei；
 - b.比例，uint64存储，如1代表1/10000包括：出块矿工分配比例R0，参与矿工分配比例R1，基金会分配比例R2
- 输出：
 - a.出块矿工奖励M1
 - b.参与矿工分配金额M2
 - c.基金会节点奖励F0
- 算法：

```
M1 = floor(M0*R0/10000)
M2 = floor(M0*R1/10000)
F0 = floor(M0*R2/10000)
```

- 备注：整形计算，后续计算比例都是先乘分子后除10000;floor为向下取整

2.验证者奖励

- 输入：
 - a.分配金额V0，BigInt存储，单位wei；
 - b.比例，uint64存储，如1代表1/10000包括：leader分配比例R0，参与验证者分配比例R1，基金会验证分配比例R2
- 输出：
 - a.leader分配工奖励V1

b.参与验证者分配金额V2

c.基金会节点奖励F1

• 算法：

```
V1 = floor(V0*R0/10000)
V2 = floor(V0*R1/10000)
F1 = floor(V0*R2/10000)
```

3.参与奖励

• 输入：

a.分配金额 M0, BigInt存储，单位wei

b. 初选且是当前顶点抵押列表D0、初选不在顶点抵押列表D1，不在初选是当前顶点抵押列表D2，切片，map[common.Address]BigInt存储，单位wei

c. 替补节点成为顶点后分配比例R， uint64存储，如1 代表 1/10000

• 输出：(D0、D1、D2) 账户对应奖励，精度为100wei

• 算法：

```
DepositAcc = sum(D0)+ sum(D1* (10000-R) /10000)+ sum(D2*R/10000)
reward0 = floor(M0*D1/100*DepositAcc)*100
reward1 = floor( M0*D1* (10000-R) /10000/100*DepositAcc)*100
reward2 = (M0*D2*R/10000/100*DepositAcc)*100
```

4.彩票奖励

###4.1选取交易

- 输入： a.交易列表T0，交易类型

- b.选取个数 n

- c 随机种子 seed

- 输出： 交易列表

• 算法：

```

T1= sort.Sort(T0)//按照交易hash排序
for i< MIN(len(T1),n) ; i++
    rand = rand.Uint64(seed)
    j = rand% (len(T1)-1)
    T1 = append ( T1,T1[j])

```

4.2 选取奖项

- 输入：a 奖项配置，结构体切片，结构体包含人数和奖励金额，下标对应奖项 b 交易列表 T
- 输出：map[common.Address]BigInt
- 算法：按照顺序选取 对应奖项的名单

5.固定奖励折半

- 输入：a.发放账户的初始金额B0 b.发放账户的当前金额 B1 c.固定奖励金额M0 d.减半金额 S
- 输出：当前发放固定区块奖励金额M1
- 算法：

```

B2 = B1-B0
if B2<=0
M1=B0
else
n = B2 /S
M1 = B0/2的n次方

```

6.惩罚

- 输入：currentuptimeList 当前的uptime账户列表
preuptimeList 上一个广播区块uptime账户列表
interest 当前区块产生的利息
- 输出：惩罚金额
- 算法：

```
rate = 1 - float64(currentuptimeList-preuptimeList)/97  
slash = interest*rate
```