fileCoin mint coin mechanism

baseline 和 simple supply 总供应量

$$
\begin{aligned}
&DefaultSimpleTotal = 3.3 * 10^8\ FIL \\
&DefaultBaselineTotal = 7.7 * 10^8\ FIL \\
&FilecoinPrecision = 10^{18} \\
&EpochDurationSeconds = 30 \\
&SecondsInHour = 60 * 60 \\
&SecondsInDay = 24 * SecondsInHour \\
&EpochsInHour = SecondsInHour / EpochDurationSeconds \\
&EpochsInDay = SecondsInDay / EpochDurationSeconds \\
&daysInYear = 365 \\
&epochsInYear = daysInYear * EpochsInDay
\end{aligned}
$$

simple supply 6年减半

$$
\begin{aligned}
&\lambda = \ln(2) / (6 * epochsInYear) \\
&expLamSubOne = e^{\lambda} - 1 \\
&epochSimpleReward(epoch) = DefaultSimpleTotal * expLamSubOne * e^{-epoch * \lambda}
\end{aligned}
$$

每个epoch的simple reward也可以通过下面公式计算. 上面公式为具体实现所用

$$
\begin{aligned}
&SimpleSupplyShouldMint(epoch) = (1 - \tfrac{1}{2}^{epoch/6 * epochsInYear}) * DefaultSimpleTotal \\
&epochSimpleReward(epoch) = SimpleSupplyShouldMint(epoch) - \\
&SimpleSupplyShouldMint(epoch - 1)
\end{aligned}
$$

baseLine 全网基线power初始值为2.5 EB，年增长率为200%

$$
\begin{aligned}
&BaselineInitialValue = 2.5057116798121726EiB \\
&BaselineExponent = IntegerPart(e^{\ln(1+200\%)/epochsInYear}) \\
&CurBaseLinePower = prevEpochBaselinePower * BaselineExponent
\end{aligned}
$$

baseline的供应逻辑如下:

https://ipfser.org/2020/10/13/filecoinjixian/

基本原理为，根据累计的实际算力和累计的baseLine算力，计算高效的网络时间，根据高效的网络时间计算当前epoch的baseline区块奖励。

令全网power变化曲线为 f1(epoch). 由于系统reward actor保存了到当前区块为止，系统的CumsumBaseline和CumsumRealized. 因此可以根据当前epoch的起始状态，计算当全网power变化曲线为f1时，baseline区块奖励的变化情况。

baseline power的变换曲线 f2(epoch)为

$$
\begin{aligned}
&f2(epoch) = \\
&ThisEpochBaselinePower * BaselineExponent^{epoch - currentEpoch}, \quad epoch >= currentEpoch
\end{aligned}
$$

当前epoch相关状态信息可以从链上获取，使用下面给定的相关初始信息进行计算

$$
\begin{aligned}
&currentEpoch = 184775 \\
&CumsumBaseline = 69641262587082009056424\ bytes \\
&CumsumRealized = 69639970155545799688192\ bytes \\
&ThisEpochBaselinePower = 3263203739195877712\ bytes \\
&EffectiveNetworkTime = 23917
\end{aligned}
$$

$$
\begin{aligned}
&EffectiveBaselinePower = 2934809387732535699\ bytes \\
&startEpoch = currentEpoch \\
&f3(epoch) = \\
&EffectiveBaselinePower * BaselineExponent^{epoch-EffectiveNetworkTime}, \quad epoch >= EffectiveNetwork \\
&CumsumRealized + \int_{startEpoch}^{x} f1(epoch) = CumsumBaseline + \int_{EffectiveNetworkTime}^{y} f3(epoch)
\end{aligned}
$$

根据上述方程，求出 y,x变换函数: y=f(x). 即为指定epoch根据全网算力变换估算的高效网络时间。在其具体实现中，y用Theta表示，即:

$$
\Theta = f(epoch)
$$

将 y=f(x)代入下速6年减半的铸币逻辑，即可得到每个epoch的baseline区块奖励。其逻辑和simple supply的铸币逻辑相同

$$
\begin{aligned}
&BaselineSupplyShouldMint(epoch) = (1 - e^{-\lambda * \Theta}) * DefaultBaselineTotal \\
&epochBaselineReward(epoch) = BaselineSupplyShouldMint(epoch) - \\
&BaselineSupplyShouldMint(epoch - 1)
\end{aligned}
$$

总出块奖励变化及IPFSMain块奖励计算

那么每个epoch系统的总铸币奖励为simple和baseline之和，即:

$$
epochReward(epoch) = epochSimpleReward(epoch) + epochBaselineReward(epoch)
$$

假设IPFSMAIN算力占全网总算力的比例为:

$$
ipfsMainPowerProportation(epoch) = p(epoch)
$$

那么在不考虑手续费奖励的情况下，ipfsmain从startEpoch到endEpoch的预期收益如下:

$$
\begin{aligned}
&ipfsMainExpectedReward(epoch) = \\
&\int_{startEpoch}^{endEpoch} ipfsMainPowerProportation(epoch) * epochReward(epoch)
\end{aligned}
$$

circulated supply及扇区抵押估算

circulated_supply = filVested+filMined+filReserveDisbursed-fileBurnt-filLocked

- filVested ico当前释放量.
- fileMined 矿工挖矿总奖励
- filReserveDisbursed 预留百分之15资金，总额为3亿，从转账信息看，spaceRace期间水龙头及后续补贴的钱均从此账户发出。目前剩余282,933,381，即释放量为17066619
- fileBurnt 为系统燃烧的币
- filLocked为系统总锁定的钱，包含市场和矿工抵押及奖励锁定

下述为epoch 185276时全网流通量状态

```
2020-10-28
epoch:185276
Circulating supply: 18078188.299786322196736941 FIL
Mined: 8560267.726552724031282506 FIL
Vested: 9931056.5980739160513691 FIL
Burnt: 3729707.888220421132127314 FIL
Locked: 13750047.098393308643850397 FIL
```

扇区的抵押和全网流通量相关，而全网流通量又和扇区的总抵押相关。
令指定epoch扇区大小为SectorSize的抵押和全网流通量分别为InitialPledge(epoch)和circulatedSupply(epoch). 具体公式表示如下:

$$
SectorSize = 32G = 32 * 1024 * 1024\ bytes
$$

$$
\begin{aligned}
&InitialPledge(epoch) = \int_{epoch}^{epoch+20} (\frac{SectorSize}{f1(epoch)} * epochReward(epoch)) + \\
&30\% * circulatedSupply(epoch) * \frac{SectorSize}{f1(epoch)}
\end{aligned}
$$

流通量计算逻辑为起始流通量加上从当前epoch起ICO释放加出块铸币，减去燃烧和锁仓。
燃烧量为外部输入，目前暂定为当前值不变，实际值应该是每天燃烧交易费加系统多种惩罚燃烧费用。
锁仓为，扇区抵押，奖励锁定和market锁定之和，目前假设后续market锁定全为0
ICO相关，释放按线性释放处理，每天释放量，通过lotus 命令获取后，除以10月15号开始的天数。

> $icoEveryEpochRelease = 9931056.5980739160513691 * 10^{18}/13/2880 = 2.652525801 \times 10^{20} = 265.2525801\ FIL$
> $EveryEpochBurn = 0\ FIL$
> $startCirculatedSupply = 18078188.299786322196736941\ FIL$
> $startEpoch = 185276$

出块铸币释放分为历史区块锁定释放和当前出块奖励释放。所有miner的历史奖励锁定，通过链上遍历所有miner, 将其LockedFunds相加获取。

ILRealse为从startEpoch开始，每个epoch已被锁定的历史奖励释放的币， R(epoch)为每个epoch的奖励，为了计算前180天在当前epoch的释放量，当epoch < startEpoch时，将其设置为 0 ,因为当epoch < startEpoch时，此部分被包含在ILRealse中计算.

> $InitTotalMinerLocked = TBD$
> $$R(epoch) = \begin{cases} 0, & epoch < startEpoch \\ \\ epochReward(epoch), & epoch \geq startEpoch \end{cases}$$

> $$ILRealse(epoch) = \begin{cases} 0, & epoch < startEpoch \parallel epoch > startEpoch + 180 * EpochsInDay \\ \\ InitTotalMinerLocked * \frac{180*EpochsInDay-(epoch-startEpoch)}{180*EpochsInDay} * \frac{1}{180*EpochsInDay}, & startEpoch + 180 * Epoch \end{cases}$$

> $blockMinedRelease(epoch) = \int_{startEpoch}^{epoch} ILRealse(epoch) +$
> $25\% * R(epoch) + \int_{epoch-180*EpochsInDay}^{epoch}(\frac{1}{180*EpochsInDay} * R(epoch))$

> $circulatedSupply(epoch) = startCirculatedSupply + blockMinedRelease(epoch) +$
> $(icoEveryEpochRelease - EveryEpochBurn) * (epoch - startEpoch) -$
> $\int_{startEpoch}^{epoch} InitialPledge(epoch) * \frac{f1(epoch)-f1(epoch-1)}{SectorSize}$

计算初值设置

若要估算系统流通量，抵押和区块奖励，需要设置以下参数：

- 系统power变换f1(epoch)
- 计算的起始高度startEpoch
- startEpoch时的全网miner总锁定InitTotalMinerLocked
- startEpoch时全网流通量startCirculatedSupply
- 每个epoch ico 释放量　icoEveryEpochRelease
- 每个epoch 燃烧的币　EveryEpochBurn
- IPFSMAIN算力占全网总算力的比例 ipfsMainPowerProportation(epoch)