

計算 用紙

まい月

はじめに

本書は著者が試してみたいと思うことをリストするという体でLaTeXとTikZを練習する文書である。

目次

はじめに	ii
目次	iii
1 ステ이블コインのようなもの	1
1.1 状況	2
1.2 ためしてみたい解決策	2
2 インポートランスの計算の効率化	4
2.1 目標と方針	5
2.2 準備	5
2.3 計算	6
2.4 導出過程	6
3 難易度とヴァリデーターの抽選	8
3.1 目標	9
3.2 具体的な計算	9

第1章

ステーブルコインのようなもの

1.1 状況

ブロックチェーンで流通するトークン🍎を裏付けとして、ある価値🍎と連動したトークン🍌を発行したい。

1.2 ためしてみたい解決策

参加者の財産	DAOの財産	レバレッジ	🍌/🍎
• 現物1000🍎	0	0.5倍	1🍌
参加者は1000🍎を担保に10🍌铸造する			
• 1000🍎を担保に10🍌铸造したポジション • 現物10🍌	• 現物1000🍎	0.5倍	1🍌

Table 1.1: 铸造する手続き

参加者の財産	DAOの財産	レバレッジ	🍌/🍎
• 1000🍎を担保に10🍌铸造したポジション • 現物10🍌	• 現物1000🍎	0.5倍	1🍌
参加者は10🍌焚いて1000🍎を返してもらう			
• 現物1000🍎	0	0.5倍	1🍌

Table 1.2: 焚く手続き

参加者の財産	DAOの財産	レバレッジ	🍋/🍊
<ul style="list-style-type: none"> • 1000🍊を担保に10🍋鑄造したポジション • 現物10🍋 	<ul style="list-style-type: none"> • 現物1000🍊 	0.5倍	1🍊
<p>1🍋の価格が100🍊になり、1000🍊にレバレッジの0.5倍を乗じた額を超えて🍋を鑄造している状態になり強制ロスカする</p>			
<ul style="list-style-type: none"> • 現物10🍋 	<ul style="list-style-type: none"> • 現物1000🍊 	0.5倍	100🍊

Table 1.3: 強制決済

第2章

インポートタンスの計算の効率化

2.1 目標と方針

この節は理想についての演説にほかならず中身がない。

まずSymbolやNEMではブロックを検証するのに所定の条件を満たした適格なアカウントであることを要求していた。これは検証するための抽選とは別に、その抽選に参加するために必要なものであった。

そこでSymbolのPoS+を参考にしつつSymbolのPoSより計算の効率と開かれ具合をすっごく上げたい。特にSymbolで採用されているPoS+と今回やりたいことの違いは：

- 所持金の下限を撤廃
 - ブロックチェーンに載っていれば適格
- ブロック毎に計算できる効率
- 毎回すべての適格なアカウントのインポートانسを計算しなくていい

であり、そのための原材料は：

- 残高
- 検証したブロックの数
- 支払った手数料

であり、ここまではSymbolと同じで、ここからが肝心なところで、これらすべての適格なアカウントについての総和がブロックヘッダに残ること、そしてこれがいたって自然な活動であれば尚良い。

2.2 準備

ブロック i ごと	アカウント $j \in \mathcal{A}$ ごと	パラメータ
<ul style="list-style-type: none">• アカウトの数$\#\mathcal{A}$• 総供給量$S_1(i)$• ブロックの高さ$S_2(i)$• 手数料の合計$S_3(i)$	<ul style="list-style-type: none">• 残高$s_1(i, j)$• 検証した回数$s_2(i, j)$• 手数料の合計$s_3(i, j)$	<ul style="list-style-type: none">• 遡るブロックH• ハッシュの大きさN• 理想の間隔T• 重み$\alpha_1, \alpha_2, \alpha_3$

Table 2.1: オンチェーンの情報

Table2.1のとおりブロックチェーンに書き込むものとして、色々計算する。

2.3 計算

そしてみんなここにしか興味なしであろう計算の部分です。

$$\begin{aligned}\tilde{I}(i, j) &:= \frac{\sum_{h=0}^{\min\{H, i\}-1} \frac{\sum_{k=1}^3 \alpha_k \frac{s_k(h, j)}{S_k(h)}}{\sum_{k=1}^3 \alpha_k}}{\min\{H, i\}} \\ &= \frac{1}{\min\{H, i\}} \sum_{h=0}^{\min\{H, i\}-1} \frac{1}{\sum_{k=1}^3 \alpha_k} \sum_{k=1}^3 \alpha_k \frac{s_k(h, j)}{S_k(h)}\end{aligned}$$

$$M := 3NH\#\mathcal{A} + 1$$

$$I(i, j) := \frac{1}{M} \lfloor M\tilde{I}(i, j) \rfloor$$

この $I(i, j)$ 、あるいはお好みの方法で減衰させたもの（e.g., $\tilde{I}(i, j)$ の代わり

に $\frac{\sum_{h=0}^{\min\{H, i\}-1} \gamma^h \frac{\sum_{k=1}^3 \alpha_k \frac{s_k(h, j)}{S_k(h)}}{\sum_{k=1}^3 \alpha_k}}{\sum_{h=0}^{\min\{H, i\}-1} \gamma^h}$ ($\gamma \in (0, 1)$) を使う、すると適当な M も変わりうる) をインポートランス (i.e., 議決権) として使おうと企てている。

2.4 導出過程

この節では、みんな興味ないかもしれないけど、 $M := 3NH\#\mathcal{A} + 1$ とする理由を説明する。 $\forall i = 1, 2, \dots; \left| 1 - \sum_{j \in \mathcal{A}} I(i, j) \right| < \frac{1}{N}$ を満たせばよいから、

$$\begin{aligned}\forall i = 1, 2, \dots; \\ \left| 1 - \sum_{j \in \mathcal{A}} I(i, j) \right| &< \frac{1}{N} \\ \Leftrightarrow \sum_{j \in \mathcal{A}} \left| \frac{\sum_{h=0}^{\min\{H, i\}-1} \frac{\sum_{k=1}^3 \alpha_k \frac{s_k(h, j)}{S_k(h)}}{\sum_{k=1}^3 \alpha_k}}{\min\{H, i\}} - \frac{1}{M} M \frac{\sum_{h=0}^{\min\{H, i\}-1} \frac{\sum_{k=1}^3 \alpha_k \frac{s_k(h, j)}{S_k(h)}}{\sum_{k=1}^3 \alpha_k}}{\min\{H, i\}} \right| &< \frac{1}{N}\end{aligned}$$

$$\begin{aligned}
& \Leftrightarrow \sum_{j \in \mathcal{A}} \left(\frac{\sum_{h=0}^{\min\{H,i\}-1} \frac{\sum_{k=1}^3 \alpha_k \frac{s_k(h,j)}{S_k(h)}}{\sum_{k=1}^3 \alpha_k} - \frac{1}{M} \left[M \frac{\sum_{h=0}^{\min\{H,i\}-1} \frac{\sum_{k=1}^3 \alpha_k \frac{s_k(h,j)}{S_k(h)}}{\sum_{k=1}^3 \alpha_k} \right] \right) < \frac{1}{N} \\
& \Leftrightarrow \sum_{j \in \mathcal{A}} \sum_{h=0}^{\min\{H,i\}-1} \left(\frac{\sum_{k=1}^3 \alpha_k \frac{s_k(h,j)}{S_k(h)}}{\sum_{k=1}^3 \alpha_k} - \frac{1}{M} \left[M \frac{\sum_{k=1}^3 \alpha_k \frac{s_k(h,j)}{S_k(h)}}{\sum_{k=1}^3 \alpha_k} \right] \right) < \frac{1}{N} \\
& \Leftrightarrow \sum_{j \in \mathcal{A}} \sum_{h=0}^{\min\{H,i\}-1} \sum_{k=1}^3 \left(\frac{\alpha_k \frac{s_k(h,j)}{S_k(h)}}{\sum_{k=1}^3 \alpha_k} - \frac{1}{M} \left[M \frac{\alpha_k \frac{s_k(h,j)}{S_k(h)}}{\sum_{k=1}^3 \alpha_k} \right] \right) < \frac{1}{N} \\
& \Leftrightarrow \sum_{j \in \mathcal{A}} \sum_{h=0}^{\min\{H,i\}-1} \sum_{k=1}^3 \left(\frac{M \alpha_k \frac{s_k(h,j)}{S_k(h)}}{\sum_{k=1}^3 \alpha_k} - \left[M \frac{\alpha_k \frac{s_k(h,j)}{S_k(h)}}{\sum_{k=1}^3 \alpha_k} \right] \right) < \frac{M}{N} \\
& \Leftrightarrow \sum_{j \in \mathcal{A}} \sum_{h=0}^{\min\{H,i\}-1} \sum_{k=1}^3 1 < \frac{M}{N} \\
& \Leftrightarrow 3N \# \mathcal{A} \min\{H, i\} < M \\
& \Leftrightarrow 3NH \# \mathcal{A} < M
\end{aligned}$$

となる。例えば、ハッシュには Keccak512 を使い、アカウントは1000こあり、ブロックは100000ブロック前まで遡って良いものとして、 $M = 3 \cdot 2^{512} \cdot 100000 \cdot 1000 + 1$ となる。

第3章

難易度とヴァリデーターの抽選

3.1 目標

みんなにブロックチェーンの維持をお願いするために、やらなきゃいけないこと、それが「だれにしようかな」である。誰たちのなかからどうやってどんなひとを選ぶか、どのくらいのペースで条件がゆるくなるか、それを決めなければならない。

3.2 具体的な計算

現在の時刻を t 、出力の大きさ N のハッシュを $h: \{0, 1\}^* \rightarrow \{1, 2, \dots, N\}$ 、 i 番目のブロック b_i の難易度 $d(i)$ を、ブロック生成の時刻を $t(i)$ 、難易度を導出するために遡るブロックの高さを D 、目標のブロック生成の間隔を T 、アカウント $j \in \mathcal{A}$ の議決権を全体の $\tilde{I}(i, j)$ 倍とすると、

$$d(i+1) := \frac{T \sum_{g=1}^{\min\{i, D\}} d(g)}{t(i) - t(i-D)}$$

と更新すると合理的な感じがして、 $p(s) := \frac{h(s)}{N}$ とすると、この p とある定数 $C > 1$ を用いて、それっぽい式 $p(b_{i-1} \parallel j) < C^{\frac{t-t(i-1)-T}{d(i)}}$ を変形して、

$$\begin{aligned} p(b_{i-1} \parallel j) &< C^{\frac{t-t(i-1)-T}{d(i)}} \\ \Leftrightarrow \log_C p(b_{i-1} \parallel j) &< \frac{t-t(i-1)-T}{d(i)} \\ \Leftrightarrow \log_C \frac{h(b_{i-1} \parallel j)}{N} &< \frac{t-t(i-1)-T}{d(i)} \\ \Leftrightarrow d(i) \log_C \frac{h(b_{i-1} \parallel j)}{N} &< t-t(i-1)-T \\ \Leftrightarrow d(i) (\log_C h(b_{i-1} \parallel j) - \log_C N) + T &< t-t(i-1) \end{aligned}$$

を得る。ここで $C = 2$ と h としてKeccak512の値を整数と思って1を足したものを選ぶと $N = 2^{512}$ となり、きりのいいところで端数処理を行うと、

$$\begin{aligned} p(b_{i-1} \parallel j) &< C^{\frac{t-t(i-1)-T}{d(i)}} \\ \Leftrightarrow d(i) (\lceil \log_2 h(b_{i-1} \parallel j) \rceil - 512) + T &< t-t(i-1) \end{aligned}$$

を得る。すなわち、各アカウントでハッシュの先頭の0の個数と概ね連動する。やったね✨