Bitcoin Cash Upgrade 2024-05-15

自己紹介

- haryu703
- コインチェック株式会社 (2020〜)

今回のアップグレードの概要

今回は下記の 1 件のプロトコルアップデートが行われた

- CHIP-2023-04 Adaptive Blocksize Limit Algorithm for Bitcoin Cash
  - ブロックサイズの上限を需要に応じて自動で増減させる

ブロックサイズ上限の歴史

- 2010 年-2017 年: 1MB
- 2017 年-2018 年: 8MB
- 2018 年-2024 年: 32MB
- 2024 年-: 32MB -

背景

- ブロックサイズの上限はネットワークを健全に維持するために必要
- 需要や技術の進歩に合わせて上限を変更する必要はある
- ブロックサイズを上げるには関係者の合意を取るなどのコストがかかる

目的

- 今後のブロックサイズに関する合意コストをなくす
- 需要に応じたブロックサイズの上限を自動で設定する

方法

「control function」と「elastic buffer function」という 2 つの要素で制御される

$$y_n = \varepsilon_n + \beta_n$$

$$\varepsilon_n = \begin{cases} \varepsilon_0 & \text{if } n \leq n_0 \\ \varepsilon_{n-1} + \gamma \cdot \left(\zeta \cdot x_{n-1} - \varepsilon_{n-1} - \zeta \cdot \beta_{n-1} \cdot \frac{\zeta \cdot x_{n-1} - \varepsilon_{n-1}}{\zeta \cdot y_{n-1} - \varepsilon_{n-1}} \right) & \text{if } n > n_0 \text{ and } \zeta \cdot x_{n-1} > \varepsilon_{n-1} \\ \max(\varepsilon_{n-1} + \gamma \cdot (\zeta \cdot x_{n-1} - \varepsilon_{n-1}), \varepsilon_0) & \text{if } n > n_0 \text{ and } \zeta \cdot x_{n-1} \leq \varepsilon_{n-1} \end{cases}$$

$$\beta_n = \begin{cases} \beta_0 & \text{if } n \leq n_0 \\ \max(\beta_{n-1} - \theta \cdot \beta_{n-1} + \delta \cdot (\varepsilon_n - \varepsilon_{n-1}), \beta_0) & \text{if } n > n_0 \text{ and } \zeta \cdot x_{n-1} > \varepsilon_{n-1} \\ \max(\beta_{n-1} - \theta \cdot \beta_{n-1}, \beta_0) & \text{if } n > n_0 \text{ and } \zeta \cdot x_{n-1} \leq \varepsilon_{n-1} \end{cases}$$

- y: ブロックサイズの上限
- n: ブロック高
- ε: control function
- β: elastic buffer function
- $n_0, \varepsilon_0, \beta_0$: 初期値
- x: 実際に作られたブロックのサイズ
- γ: control function の「forget factor」
  - ブロックサイズの増減率を調整する
- ζ: control function の「asymmetry factor」
  - ブロックサイズの増加率と減少率の差を調整する
- θ: elastic buffer の減少率
  - control function 側の増加率がθより低いと elastic buffer は減少する
- δ: elastic buffer の「gearing ratio」
  - elastic buffer と control function をどの程度連動させるかを調整する

mainnet のパラメータ

- $\varepsilon_0 = 16000000$
- $\beta_0 = 16000000$
- $n_0 =$ ハードフォーク時点のブロック高
- $\zeta = 1.5$
- $\gamma = \frac{1}{37938}$
- $\delta = 10$
- $\theta = \frac{1}{37938}$

- ブロックサイズ上限の初期値は $y_0 = \varepsilon_0 + \beta_0 = 32000000$ でハードフォーク前と同じ
- このアルゴリズムが有効化されている testnet はε_0、β_0およびn_0が異なる
- その他、一時的に$y_{\text{temporary_max}} = 2000000000 (2\ \text{GB})$が設定されている
  - 32-bit アーキテクチャや p2p プロトコルの制約
  - 2028 年 5 月までに取り除かれるらしい

パラメータの特徴

control function

- ブロックサイズ増加率の上限: $\left(\frac{\varepsilon_n - \varepsilon_{n-1}}{\varepsilon_{n-1}}\right)_{\max} = \gamma \cdot (\zeta - 1) = \frac{1}{75876}$
  - 年間で最大 +200 %
- ブロックサイズ減少率の上限: $\left(\frac{\varepsilon_n - \varepsilon_{n-1}}{\varepsilon_{n-1}}\right)_{\min} = -\gamma = -\frac{1}{37938}$
  - 年間で最大 −75 %

elastic buffer function

- ε に対する β の上限: $\left(\frac{\beta_n}{\varepsilon_n}\right)_{\max} = \delta \cdot \frac{\gamma}{\theta} \cdot \frac{\zeta - 1}{\frac{\gamma}{\theta} \cdot (\zeta - 1) + 1} = 3.33$
- βの減少率はθで、年間で最大 −75 %
  - 半減するのは$\frac{\log(0.5)}{\log(1-\theta)} = 26296$ブロックで 10 分間に 1 ブロックなら約 6 ヶ月
- 急激なブロックサイズの上昇に対応できるような増加をするらしい

パラメータの選び方

Asymmetry factor (ζ)

「Asymmetry factor」は control function の増減率の関係を決める

$\zeta = 2$ だと増加率と減少率は同じになる

- ハッシュレートを 50 % 持った攻撃者によるスパム TX でブロックサイズを増やす攻撃に対し、防御側が空ブロックを作ることが強制される
  - 防御側は fee を得られないため攻撃側に対して不利

$\zeta = 1.5$だと上記の攻撃シナリオに耐性がつく

- ハッシュレートが 50:50 の場合、防御側は上限の 33% のブロックサイズまで作れる
- 空ブロックでブロックサイズを小さくする攻撃はしやすくなる
  - こちらは防御側が fee を受け取れるため有利

Forget factor (γ)

「Forget factor」は control function の増減率の関係を決める

- 1 年間(52959 ブロック)の最大増加率が +100 % になるように設定された
- BIP-101 で提案されていた増加率よりは高いが、最大増加率を維持するのは現実的ではないため実際に BIP-101 のブロックサイズを超えることは考えにくい

Gearing ratio (δ) と Decay rate (θ)

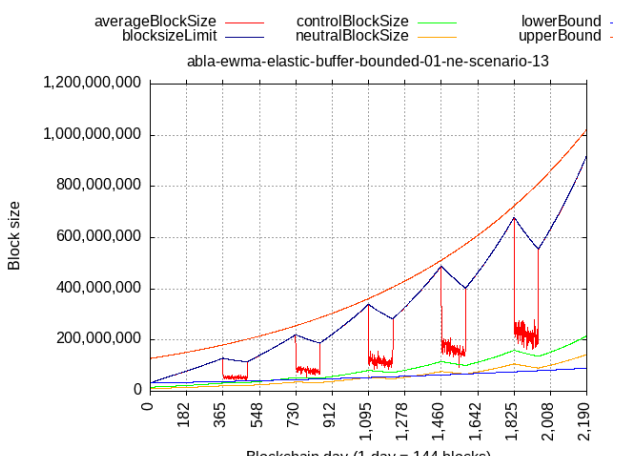
「Gearing ratio」と「Decay rate」は elastic buffer の大きさと増減速度を決める

- 大きさは最大でβがεの 3.33 倍になるように設定されている
  - 数ヶ月でブロックサイズが倍になっても大丈夫らしい
- 減少速度は半年で半分になるように設定されている

シナリオ

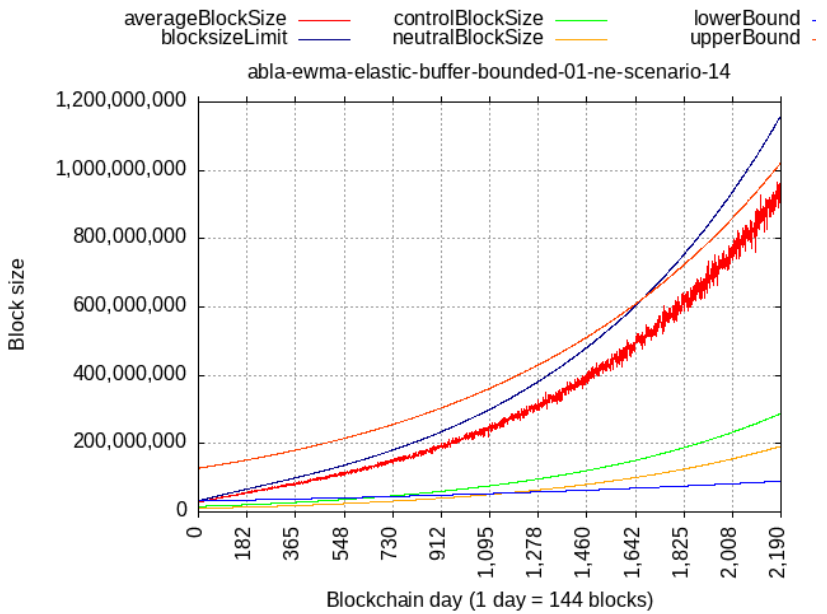
下記を繰り返す場合[1]

- すべてのブロックがブロックサイズ上限で 8 ヶ月
- 33% のブロックがブロックサイズ上限、他のブロックが 21MB で 4 ヶ月



[1] https://gitlab.com/0353F40E/ebaa/-/raw/9606b73b10551e4ef56e238c7a7bedc4f95236dd/simulations/results/abla-ewma-elastic-buffer-bounded-01-ne-scenario-13.png

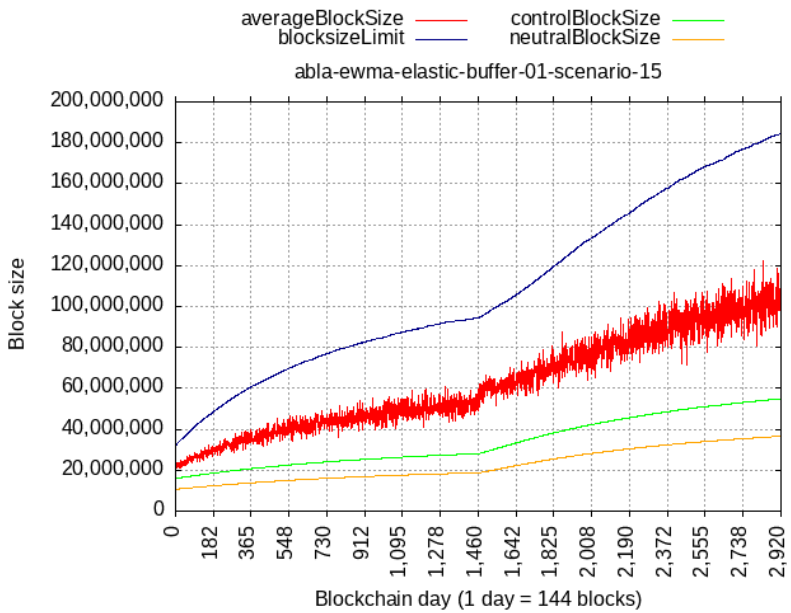
90% のブロックがブロックサイズ上限の 90%、残りが 21MB が続く場合[2]



[2] https://gitlab.com/0353F40E/ebaa/-/raw/9606b73b10551e4ef56e238c7a7bedc4f95236dd/simulations/results/abla-ewma-elastic-buffer-bounded-01-ne-scenario-14.png

ハッシュレートの 50% がスパム攻撃をする場合[3]

- 1-4 年目: 50% のブロックがブロックサイズ上限、残りが 10.67MB
- 5-8 年目: 50% のブロックがブロックサイズ上限、残りが 21.33MB



[3] https://gitlab.com/0353F40E/ebaa/-/raw/9606b73b10551e4ef56e238c7a7bedc4f95236dd/simulations/results/abla-ewma-elastic-buffer-01-scenario-15.png

まとめ

- ブロックサイズの上限を自動で変更するアルゴリズムが導入された
- 今後ブロックサイズの上限は需要に応じてハードフォーク無しに変更される