Vol.2017-CSEC-79 No.6
2017/12/4

計算資源の有効利用を目的とした ブロックチェーンシステムの提案と設計

中山和也[†1]　金子晃介[†2]　城島翔太[†3]
西田裕輝[†3]　堤優亮[†4]　櫻井幸一[†5]

概要：この報告書の趣旨はビットコインの根幹を成しているコンセンサスアルゴリズムであるプルーフオブワーク(以下 PoW)のハッシュ計算を遺伝的アルゴリズムの計算に置き換え、ビットコインのシステムの維持に利用されている膨大な計算資源をより有効活用するシステムの提案と設計である。

キーワード：ブロックチェーン、合意形成アルゴリズム、遺伝的アルゴリズム

Blockchain System Aiming for Effective Use of Computing Resources

KAZUYA NAKAYAMA[†1]　KOSUKE KANEKO[†2]　SYOTA JOJIMA[†3]
YUKI NISHIDA[†3]　YUSUKE TSUTSUMI[†4]　KOUICHI SAKURAI[†5]

Abstract: This report proposes a Blockchain system aiming for effective use of computing resources by applying Genetic Algorithm into the consensus algorithm to generate a new block of it.

1. はじめに

1.1 背景と目的

近年、ビットコインを中心にブロックチェーンを活用した様々なタイプの非中央集権型仮想通貨が流通し注目を集めている。2017 年 10 月 26 日現在、仮想通貨の時価総額は、１位がビットコイン、２位がイーサリウムという状況になっている[1]。この二つの仮想通貨は、非中央集権型のネットワーク上で、仮想通貨のやり取りの内容（以下、トランザクション）を記録するために Proof-of-work（以下、PoW）と呼ばれるコンセンサスアルゴリズムを採用している[2]。

ブロックチェーンを構成するブロックには、一定の期間内で発生したトランザクションの内容が記されており、それらのブロックがチェーンのようにつながってブロックチェーンというデータベースを構成している。つまり、ブロックチェーンは過去から現在までのトランザクションの履歴になっている。新しいトランザクションの内容を含むブロックを追加する際には、非中央集約型のネットワークを構成するノード同士による合意形成アルゴリズム（以下、コンセンサスアルゴリズム）が適用される。コンセンサスアルゴリズムの一つである PoW では、ハッシュ計算を行うことで分散ノード間での合意形成を実現している。PoW におけるブロックチェーンの各ブロックには、直前に生成されたブロックのハッシュ値、選択したトランザクション、ナンスと呼ばれる値を含んでいる（図 1）。 PoW では、マイナーと呼ばれるノードがこのブロックをハッシュ化し、ハッシュ値が設定された値以下であるようなナンスを探す作業を行う。条件を満たすナンスを見つけたノードは、新たなブロックを配信し、全てのノードのブロックチェーンが更新される。新たなブロックを配信したノードはブロック生成の報酬を受け取る。このハッシュ計算をして条件を満たすナンスを探す作業はマイニングと呼ばれている。この報酬獲得競争を勝ち取るためには、より早くハッシュ計算を行える計算資源が必要である。ビットコインの場合、10 分ごとにマイニングが発生し、世界中のマイナーが参加する報酬獲得競争によって非常に膨大な計算資源が浪費されている。

そこで本研究では、このハッシュ計算をより有意義な計算に置き換えることで計算資源を有効活用し、実社会の問題の解決に役立てることが可能なブロックチェーンシステムを提案する。

†1 九州大学 工学部
Faculty of Information Science and Electrical Engineering, Kyushu University
†2 九州大学　サイバーセキュリティセンター
Cybersecurity Center, Kyushu University
†3 九州大学　大学院システム情報科学府
Graduate School of Information Science and Electrical Engineering, Kyushu University
†4 九州大学 理学部
Faculty of Science, Kyushu University
†5 九州大学　大学院システム情報科学研究院
Graduate School of Information Science and Electrical Engineering, Kyushu University

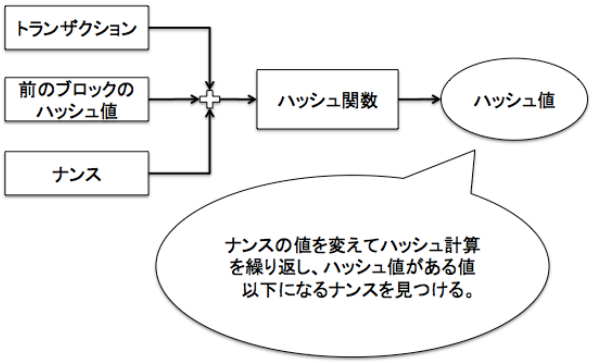

図 1　PoW のハッシュ計算

2. 関連研究・関連技術

2.1 コンセンサスアルゴリズムの先行研究

ブロックチェーンは、Peer-to-Peer（以下、P2P）をベースとした非中央集権型のネットワーク上での運用が想定されている。このような非中央集権型のネットワークでは、悪意のある情報を流すノードやノードの故障等によって情報伝達の不具合が生じ合意形成が困難な問題が起きる可能性がある。この問題は、ビザンチン将軍問題[3]と呼ばれている。ブロックチェーンは、トランザクションの内容を含んだブロックをつないだ分散台帳と新しいブロックを生成するためのコンセンサスアルゴリズムによって、非中央集権型のネットワークを運用する上で課題となるビザンチン将軍問題を解決する手段を提供している。このコンセンサスアルゴリズムは、ビットコインで利用されている PoW をはじめとして、ブロックチェーンを利用する仮想通貨で、様々なアルゴリズムが考案されている[4][5]。例えば、Proof-of-Stake(以下、PoS)は、PoW のハッシュ計算の難易度を通貨の所有量によって変化させ、通貨所有量の多いものがブロックを生成できる確率が高くなり報酬を受け取りやすくなるコンセンサスアルゴリズムである[6]。また、近年は、Hashgraph のようなマイニングを行わない合意形成アルゴリズムも考案されており、合意形成のための計算資源を抑えることできるため、IoT (Internet of Things)のような計算能力の低いデバイスのネットワークで構成される分野にも適用できる期待が高まっている[7]。

2.2 ブロックチェーンを利用したアプリケーション

(1) ビットコイン

ビットコインは、Satoshi Nakamoto によって考案されたブロックチェーンを利用した仮想通貨である[2]。ビットコインは、2017 年 10 月現在、時価総額 1 位の仮想通貨システムで、コンセンサスアルゴリズムに PoW を採用している。ハッシュ計算の難易度は直近の一週間の 1 ブロックあたりの平均ブロック生成時間を基に 1 ブロックの生成時間が 10 分程度になるように設定される。

(2) イーサリウム

イーサリウムは、Vitalik Buterin によって考案されたブロックチェーンプラットフォームである[8]。イーサリウムは、2017 年 10 月現在、時価総額 2 位の仮想通貨システムで、コンセンサスアルゴリズムに PoW を採用している。このコンセンサスのアルゴリズムは、将来的には PoS に移行される予定である。 また、スマートコントラクトと呼ばれる用意されたプログラミング言語で記述されたプログラムをブロックチェーンに載せることができる機能がある。このスマートコントラクトを利用することで、通貨のやりとりだけでなく実在品物や著作権などのやりとりも可能となっており、分散コンピューティングのシステムを構築することも可能である。

(3) Hyperledger Fabric

Hyperledger Fabric は、ブロックチェーンの技術を仮想通貨以外の分野にも利用することを目的として生まれたブロックチェーン技術の推進コミュニティーの Hyperledger プロジェクトのブロックチェーンシステムの一つである[9]。Hyperledger Fabric には、PBFT(Practical Byzantine Fault Tolerance)と呼ばれる特定の管理者を介してコンセンサス形成を行い、特定のタイミングでブロックが生成され、またハッシュ計算がなくスループットが高いコンセンサスアルゴリズムを採用している。このため、高速で安全なデータ通信が要求される IoT への活用が期待できる。

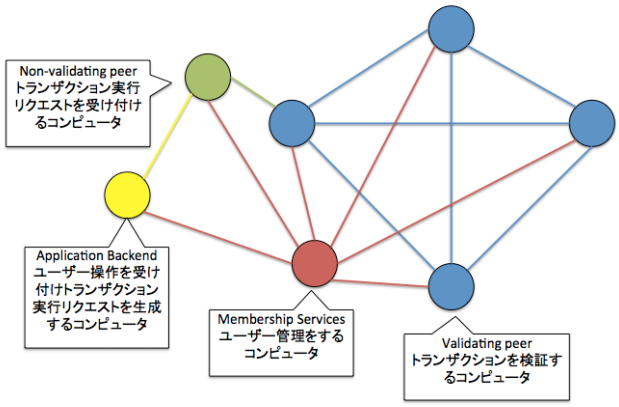

図 2　Hyperledger Fabric のネットワーク構成図

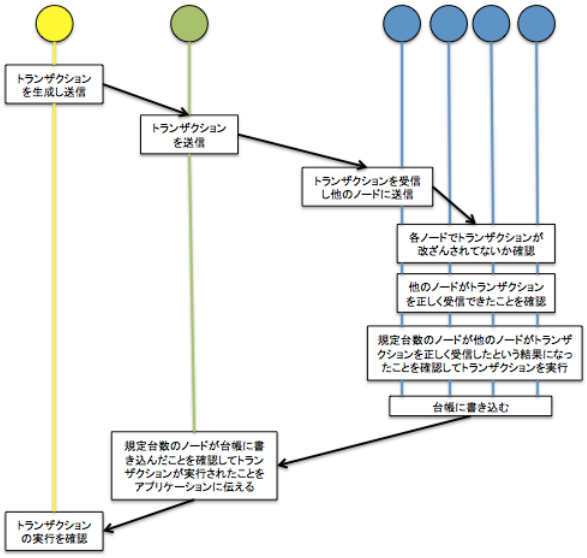

図３　PBFT の流れ

(4)　Golem

Golem は、P2P ネットワーク状でノードとなるコンピューターの計算資源の貸し借りを可能にすることを目標にしたプロジェクトである[10]。GNT と呼ばれる通貨で計算資源の売買を行い、通貨の売買はイーサリウムのスマートコントラクトを利用し、計算する問題の配布や計算への参加等は別の専用に構築された独自のネットワークを利用する。現在もプロジェクトが進められており、2016 年のプロジェクト開始から 4 年のロードマップが設計されている。

3.　本システムの構成

本章では、計算資源の有効活用を目的としたブロックチェーンシステム（以下、本システム）を構成する要素について説明する。

3.1　遺伝的アルゴリズム

本システムでは、PoW において新しいブロックを生成する際に必要となるハッシュ計算を遺伝的アルゴリズムの計算に置き換え、ブロック生成のアルゴリズムを実行する。この遺伝的アルゴリズムで解く最適解問題は、各ノードから提供される(以下、問題の投下)。投下された問題に対し、各ノードが遺伝的アルゴリズムによる最適解計算を行う。その結果、より効果的な遺伝子を生成したノードに新しいブロック生成の報酬を与える。遺伝的アルゴリズムの計算は、ランダムな操作の試行錯誤で最適解を導くのでハッシュ計算と同様に確率的に正解者(このシステムでは最も結果の良かった者)になることができ、操作から結果を導き出し、不正がないことを確かめるのが容易で時間もかからない。計算する問題は組み合わせ最適化問題で、3D モデルの動作の最適化などが考えられる。

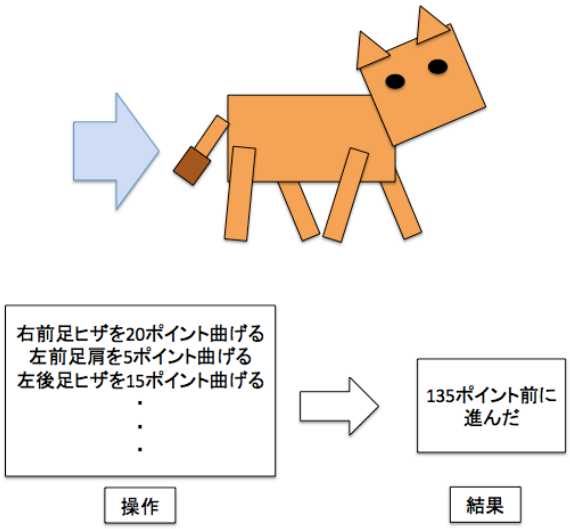

図４　遺伝的アルゴリズムの計算

3.2　ネットワーク構成

本システムのネットワークの構成図を図 5 に示す。本システムのネットワークは、コアノードとリーフノードで構成される。コアノードは、問題とトランザクションの承認と決定､報酬を受け取るノードの決定を行うノードである。リーフノードは、コンセンサスアルゴリズムのための遺伝的アルゴリズムを計算するノードである。図 5 中の青のノードはコアノードを表し、緑のノードはリーフノードを表している。コアノード同士で構成されるネットワークは、コアネットと呼ぶ。

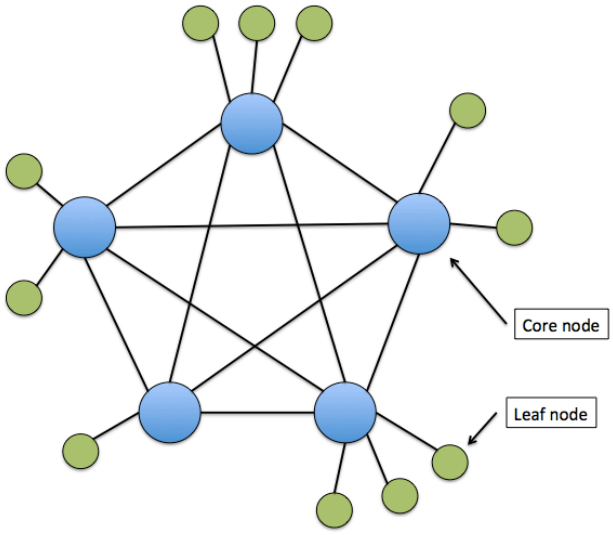

図５　ネットワーク構成

コアノードは、複数のリーフノードとつながっている。コアノード、リーフノードは共に秘密鍵を持ち、秘密鍵から生成された公開鍵とそれに紐付いたアドレスを持っている。

3.3 トランザクションと問題の構成

(1)トランザクション

本システムのトランザクションの内容は、以前のトランザクションのハッシュ値及び送金元アドレス送金先アドレス、送金量、タイムスタンプ、電子署名から構成されている（図 6)。トランザクションの際に、本システムで利用される仮想通貨を便宜上ジーンと呼ぶことにする。送金元のアドレスから送金先のアドレスに送金量分のジーンが送られる。タイムスタンプの時刻から一定時間経過したトランザクションはプール（以下、トランザクションプール）から破棄され、改ざんを防止するために電子署名をつける。

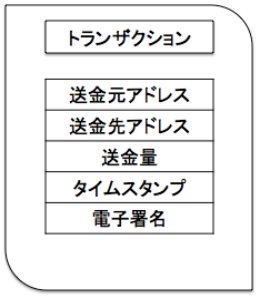

図 6　トランザクションの構成

(2)投下問題

本システムでは、各ノードが遺伝的アルゴリズムで解答可能な最適解問題を投下する。この各ノードが遺伝的アルゴリズムで投下問題を解く作業をトライングと呼ぶことにする。これは PoW におけるマイニングに当たる作業になる。問題がトライングされると投下元のアドレスから消費量分のジーンが消費され、このジーンは消滅するが、計算時間に対して一定のジーンが報酬として生成され送られる。トランザクションと同様にタイムスタンプの時刻から一定時間経過した投下問題はプールから破棄され、改ざんを防止するために電子署名をつける。投下問題は時間あたりの消費量が大きいものが優先して選ばれる。

図 7　投下問題の構成

(3)ランダム問題

ランダム問題は、ランダムに生成された問題で目標流通量よりも実際のジーンの流通量が少ない時、問題プールに問題がない時にトライングされ、計算時間に対して一定のジーン(目安は 1 分あたり 100 ジーン)が報酬として生成される。計算時間に対して一定のジーンが報酬として生成され送られる。ジーンの流通量を安定させるため、目標流通量が設定され(図 9)、これを目安にランダム問題をトライングするかどうか選択される。

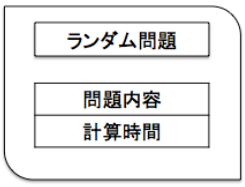

図 8　ランダム問題の構成

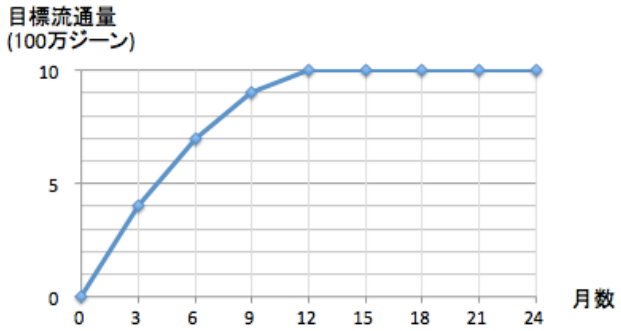

図 9　時間経過と目標の通貨量

3.4 ブロックの構成

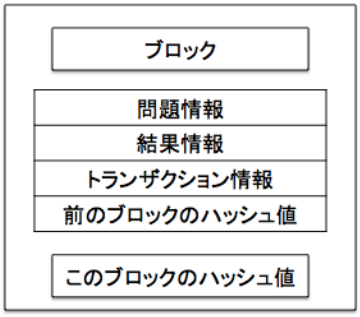

図 10　ブロックの構成

問題情報にはそのブロックでトライングした問題の内容が入っており、結果情報にはその問題の結果と報酬を受け取ったリーフノードのアドレスが入っている。トランザクション情報にはいくつかのトランザクションが入っている。また、前のブロックのハッシュ値も載せることでブロック同士がつながっている。

4. コンセンサスアルゴリズムの手順

4.1 システムのコンセンサスの流れ

システム全体の流れは以下のステップに分けられる。

(1)トランザクションと問題の決定

ブロックに載せるトランザクションとトライングさせる問題をコアネット上で決める。

(2)問題の配信とトライング

決定した問題をコアノードが自身につながっているリーフノードに配信し、受け取ったリーフノードがトライングする。

(3)回答の集計とチェック

リーフノードの回答をコアノードが集計しチェックする。

(4)報酬を受け取るリーフノードとブロックの決定

コアネット上で報酬を受け取るリーフノードを決定し、ブロックを生成する。

また、システムはこの(1)~(4)を繰り返すが、リーフノードから問題とトランザクションの投下があるたびにこれらのステップと同時に(5)トランザクションと問題の承認を行う。

4.2 各手順の詳細

(1)トランザクションと問題の決定

ランダムに決められたコアノードがトランザクションプールからブロックに載せるトランザクションを任意に選び、現在の流通量と問題プールの状況を基に問題を選ぶ、そしてそれらをまとめてコアネットに配信する。他のコアノードはそれを受け取った後、内容を確認し異常がなければ仮承認したことをコアネットに配信する。コアノードは一定の時間内にコアネット内の３分の２より多くのコアノードから承認を受け取ると本承認したとみなし次のステップに移行する(図 11)、一定の時間内に仮承認の数がコアネット内の３分の２を越えなければトランザクションと問題を決めるコアノードを別のコアノードに変更して再度行う。

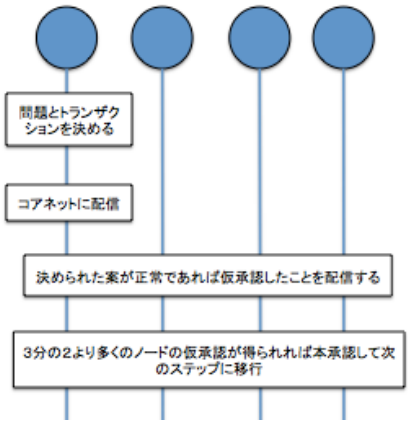

図１１　トランザクションと問題の決定の手順

(2)問題の配信とトライング

コアノードは自分につながっているリーフノードに問題を配信し制限時間が経過するまで待つ。問題を受け取ったリーフノードはトライングをして制限時間内に回答をコアノードに送る。この際に、リーフノードからコアノードに送られるトライング結果のデータの構成を図 12 に記す。

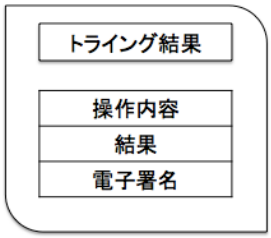

図１２　リーフノードのトライング結果の構成

(3)回答の集計とチェック

コアノードは制限時間が経過するとリーフノードの回答をチェックし一番結果の良かったものとその回答者のアドレス(図 13)をコアネットに配信し一定時間が経過するまで待ち、他のコアノードから結果を受け取るとその結果をチェックし異常がなければ仮承認したことをコアネットに配信する。３分の２より多くのノードの仮承認が得られた結果は本承認する（図 14)。

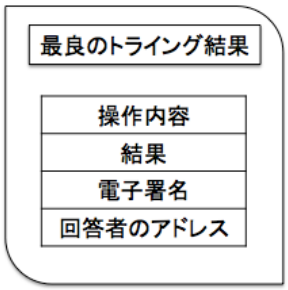

図１３　最良のトライング結果の構成

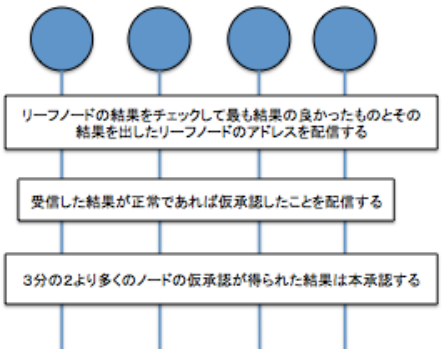

図１４　解答の集計とチェックの手順

(4) 報酬を受け取るリーフノードとブロックの決定

一定時間経過後にランダムに決められたコアノードが本承認を得られた結果の中から最も良かったものを回答したリーフノードを、報酬を受け取るリーフノードとした新たなブロックの候補としてコアネットに配信する。これを受け取ったコアノードは内容を確認し異常がなければ仮承認

したことをコアネットに配信する。一定時間内にコアネット内の３分の２より多くの承認を受け取ったコアノードはブロック候補を正式なブロックであるとし自身のブロックチェーンに追加する。一定の時間内に仮承認の数がコアネット内の３分の２を越えなければブロック候補を決めるコアノードを変更し再度行う（図 15）。

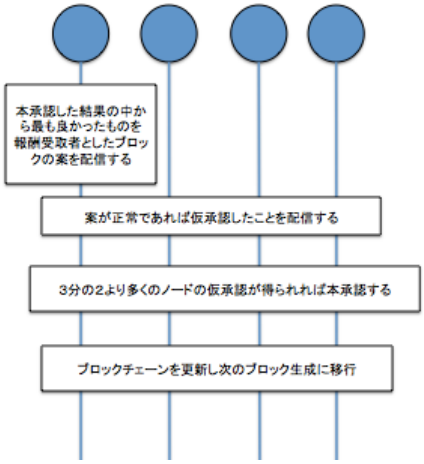

図１５ ブロックの決定の手順

(5) トランザクションと問題の承認

リーフノードから問題、トランザクションを受け取ったコアノードはそれをコアネットに配信する。これを受け取ったコアノードは内容をチェックし異常がなければ仮承認したことをコアネットに配信する。コアノードは一定時間内にコアネット内の３分の２より多くの仮承認を得られた問題、トランザクションを随時各プールに追加する（図 16）。

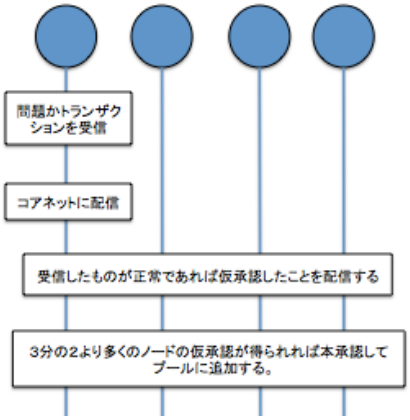

図１６ プール追加の流れ

5. 想定される問題と対策

本章では、本システムの設計を実装するにあたって想定されうる問題点とその解決方法について説明する。

5.1 コアノードの障害発生

コアノードが攻撃される、故障するなどにより障害が発生した場合でも、コアネットの３分の２より多くのコアノードが正常であればシステムを維持できる。

5.2 不正なトランザクションの投下

トランザクションの改ざん、二重送金、残高不足の送金はコアネットによるチェックで防止できる。

5.3 有利な問題の投下によるジーンの独占

自分に有利な問題を投下し、その報酬を受け取り続け、ジーンを独占できないように、投下問題のジーン消費量に下限を設定して、ジーンの流通量が減少するようにする。

5.4 ジーン流通量の過減少

投下問題によってジーンの流通量が減少しすぎることを防止するためにその時点の目標流通量に達していない場合は優先してランダム問題をトライングさせる。

5.5 ジーン流通量の過増加

問題とトランザクションの投下がない状態が続くとジーン流通量が増えるのみとなる、そのため流通量が一定量を超えた場合に報酬量を減らす、所有されているジーンを少量の割合減らす等の対策が必要になる。

5.6 不正なトライング結果の回答

不正なトライング結果を回答してジーンを取得できないようにリーフノードにつながっているコアノードが結果の不正がないかチェックする。そのコアノードが正常に働いていない時のためにコアネットでもチェックする。

5.7 大量の問題とトランザクションの投下

大量の問題とトランザクションの投下によってネットワーク、システムに負荷をかけることを避けるために、投下の度に少量の手数料が支払われるようにし、一定時間の投下量が閾値を越えた時に手数料を増加させる。

6. 終わりに

本報告書では、計算資源の有効活用を目的としたブロックチェーンシステムを提案した。今後、本システムで提案した設計やアルゴリズムを元に実装を行い、今の時点で想定していない障害や攻撃が起こる可能性がないか検討する。本システムの実装を通じて、システムの設計を改善し、本研究の目的である仮想通貨ネットワーク内の計算資源の有効活用と安全な分散コンピューティングの実現ができると期待している。

謝辞

This research was supported by Strategic International

Research Cooperative Program, Japan Science and Technology Agency (JST). 本研究は JSPS 科研費 JP15H02711 の助成を受けたものです。

参考文献

[1] “Cryptocurrency Market Capitalizations”. https://coinmarketcap.com
[2] Nakamoto, S., (2008), “Bitcoin: A Peer-to-Peer Electronic Cash System”, white paper, https://bitcoin.org/bitcoin.pdf.
[3] Lamport, L., Shostak, R., Pease, M., 1982, “The Byzantine Generals Problem”, ACM Transactions on Programming Languages and Systems, Vol. (4/3), pp. 382-401.
[4] Zheng, Z., Xie, S., Dai, H., Chen, X. and Wang, H., (2016), “Blockchain Challenges and Opportunities: A Survey”, International Journal of Web and Grid Services.
[5] 赤羽喜治, 愛敬真生, 2016, “ブロックチェーン　仕組みと理論 サンプルで学ぶ Fintech のコア技術”, リックテレコム, ISBN-13: 978-4865940404.
[6] Pavel, V., (2015). "BlackCoin’s Proof-of-Stake Protocol v2", white paper, http://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf.
[7] Baird, L., (2016), “THE SWIRLDS HASHGRAPH CONSENSUS ALGORITHM: FAIR, FAST, BYZANTINE FAULT TOLERANCE”, white paper, http://www.swirlds.com/downloads/SWIRLDS-TR-2016-01.pdf.
[8] Wood, D., (2014), “ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER”, white paper, https://bravenewcoin.com/assets/Whitepapers/Ethereum-A-Secure-Decentralised-Generalised-Transaction-Ledger-Yellow-Paper.pdf.
[9] Cachin, C., (2016), “Architecture of the Hyperledger Blockchain Fabric*”, white paper, https://www.zurich.ibm.com/dccl/papers/cachin_dccl.pdf.
[10] Zawistowski, J., et al., (2016), “The Golem Project”, crowdfunding whitepaper, http://golemproject.net/doc/DraftGolemProjectWhitepaper.pdf,.