Constellation（コンステレーション：星座）

ブロックチェーン･マイクロサービス･オペレーティングシテム

2017年11月25日

要点

Bitcoin、Ethereumなどの現在のブロックチェーンテクノロジーには、システム的な同期制限があるため、現在の分散型消費者用アプリケーションの技術構造に向けた大量採用には適しておりません。また、コンセンサス･メカニズムとスマートコントラクト構造は、消費者用グレードアプリケーションを機能させるために必要なレベルまで拡張する事が、できないのが実情です。私たちは、耐障害性があり、水平方向にスケーラブルな分散オペレーティング・システムを提案することで、モバイル・クライアントに向けてフル・ノードの実装されたサービスを提供いたします。したがって、私たちは現代のサーバーレス構造を暗号化し、安全なコンセンサスを再構成します。その際、非同期のExtendedTrustChain（拡張済みトラストチェーン）やProof-of-Memeによるコンセンサスモデルを使用し、また、アクターベースの有限状態マシンとして最初に実装されたJVMによる構成可能なマイクロサービスとしてのスマートコントラクトを取り入れいています。この構造により、高いトランザクション･スループットが保証されると共にConstellation（コンステレーション）を通しコンシューマーグレード（消費者向け）の分散アプリケーションを構築することが可能となります。

序説

現在の暗号化通信は、地方分権化を目指しているにも関わらず、ネットワークのセキュリティを制御することで、さらに集中化されたネットワークと鉱業組織により運用されるようになり、保護されているのが実情です[1]。また、Ethereumのようなポスト Bitcoinネットワークの登場に伴い、金融取引処理の初期目標が拡大され、EVMや、スマートコントラクト･ロジックシステムによる信頼できない分散型コンピューティングシステムが提供されるようになっています。これらの分散型プロトコルは多様で創造的ではありますが、エントリーの際に大きな障壁が存在するため既存の主流のソフトウェアからは除外されています。[2]。企業が慣れ親しんでいないコーディング言語やデザインパターンを使用し、堅牢な分散型の環境を開発、展開、および保守するためには、多くの費用と時間がかかります。

[1]デジタル通貨グループの‘コンセンサス2017でのBitcoin スケーリング･アグリーメント medium.com/@DCGco

[2]J.エバンスのブロックチェーンは新しいインターネットではなく、新しいLinuxです。techcrunch.com

さらに、このようなアプリケーションを、現在の最先端の公共ブロックチェーンに加えて、適度なエンタープライズユースケースのパフォーマンス要求に合わせることは事実上不可能です。これらの技術的課題を克服するために、ブロックチェーンテクノロジーは、水平にスケーラブルな上に、コスト効率が高く、効率的な分散オペレーティングシステムとして機能する自立プロトコルを必要としています。

Bitcoinは、財務取引に関する分散したコンセンサスの問題を解決するために作成されましたが、プルーフ・オブ・ワーク（pow）として知られるエネルギー集約的なコンセンサスプロセスに依存していました。これにより、選択された少数の個人から垂直方向の計算能力がプールされる事になります。[3]従って、ネットワークのセキュリティおよび報酬は、同様に、選択された数少ないユーザーの手に渡されます。スケーリングの議論が拡大するにつれて、Bitcoin キャッシュやBitcoin ゴールドなどのBitcoinフォークがいくつか作成されるようになりました。これらの別々の勢力は酷く分裂しており、エコシステムに害を及ぼします。また、時間のかかるアルゴリズムと組み合わされた旧式のスループット要件（言い換えれば、キャップされた1MBブロックサイズ）は、取引時間および取引費用を急増させることとなりました。Ethereumは、ASICに似たASICに耐性のある作業証明コンセンサスアルゴリズムを使用しており、確証証明のコンセンサスメカニズムであるCasperに移行しています。プルーフ・オブ・ステークには、民主的な不均衡が生じ、最も資金のある人が、コンセンサスを通じてネットワークの状態を選ぶことができるようになっています。Ethereumはスマートコントラクトの使用においてパイオニアとなりましたが、同期実行システムであるために、消費者向けの分散アプリケーションが深刻なボトルネックとして取り上げられています。同時実行を実施することが、消費者向けの製品を提供するための鍵となっており、Constellation（コンステレーション）はマイクロサービス構造としてのスマートコントラクトでこれを実現しています。

Constellation（コンステレーション）は何が違うのですか？

ブロックチェーン開発コミュニティは、スケーラビリティの問題を解決するための分散型元帳テクノロジーの新しい実施方法を探しておりましたが、いままで、解決できていませんでした。この問題は依然として残っています：より多くの取引をより少ないコストおよび短時間で処理し、多くのサービスを実行するにはどうすればよいのでしょうか？私たちは、MapReduceに似たテクニックをを使用する水平スケーリング手法を提案します。

水平スケーラビリティとは、並行プログラミングのアプリケーションの事です。これにより、ユーザーがネットワークに参加するにつれて、取引のスループットを向上させる事ができます。MapReduceとは、コンピューテーションを単純な操作に分割するプロセスであり、それらによる計算を非同期DAG（有向非巡回グラフ）に供給できるため、すでに並行しているプログラムの効率を向上させることが可能です。

Constellation（コンステレーション）のプロトコルは、モバイルデバイス上に展開できるゴシッププロトコルとして知られているピアツーピア層を備えた拡張トラストチェーンと呼ばれる、水平方向にスケーラブルなブロックチェーン構造を実施しています。

[3] Bitcoinは集中化したのでしょうか？' www.trustnodes.com

Constellation（コンステレーション）は、マイクロサービス構造を採用したスマートコントラクトにアプローチすることで、各マイクロサービスのSLA（サービスレベルアグリーメント）および/またはタイプシグネチャを認識するだけで、連鎖的な可用性の高いサービスを分散アプリケーションに組み込むことができるのです。

ゴシッププロトコルにより、大規模なネットワークは、既存のブロックチェーンテクノロジーよりも桁違いに高いスケールでネットワークの全状態を通信できます[4]. このタイプのネットワークでは、ネットワークの各メンバーが隣接する人物を追跡し、新しいメッセージを受信すると、そのメッセージを隣接するすべての人物に順番に伝達します。これには興味深い数学的特性がありますが、私たちのケースにおいては、接続されたデバイスの大規模なプールがネットワークの状態を共有しており、現在のブロックチェーンの実施では知られていないスケールでコンセンサスを達成できるのです。（図 1を参照).

Transaction Throughput (tps)			
Bitcoin	Ethereum	IOTA (Tangle)	Hylochain (1200 nodes - fixed neighbors)
3-4	15	500-800	4000-4800

図1：ブロックチェーンのスループットの比較

水平スケーラビリティを可能にするブロックチェーンコンセンサスの1つのアプローチにExtendedTrustChain（拡張済みトラストチェーン）が上げられます。
ExtendedTrustChain（拡張済みトラストチェーン）には、プロトコルのさまざまな側面を非同期に管理するために、それぞれ独自の責任と役割を持つ複数のノードタイプがあります。

[4]R. ヴァン・レネッセ「ブロックチェーンはゴシップに基づいているのか？」zurich.ibm.com

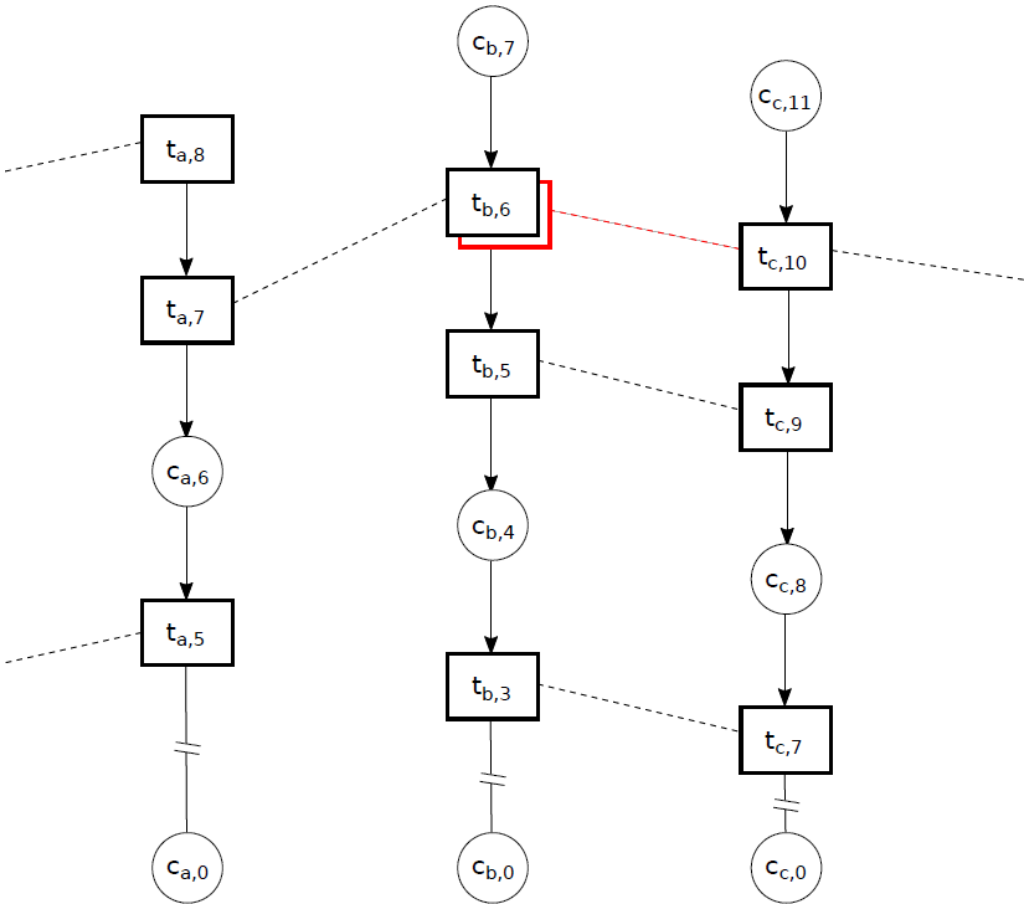

図2：Extended Trust Chain（拡張済みトラストチェーン）の図、K. Cong　及びその他の者

ノード

具体的には、取引を送信し、メンバーの個々のチェーンをホストする基本ノードがあります（各個人のチェーンは、取引が作成され、リニアブロックにハッシュされるときにネットワーク上の他のチェーンとDAGを形成します）。

チェックポイントプロセス

取引（バッファを満たし、チェックポイントブロックにハッシュされる）に基づきコンセンサスを実行するノードプロセスがあります。 これらの取引の集合はハッシュされ、メインのチェーンの次のブロックになります。

バリデーター（検証コントロール）プロセス

最後に、バリデーター（検証コントロール）プロセスを実行し、現在のチェーン状態をホストし、また、ネットワーク上のノードにチェーンの状態/履歴をシードする検証コントロールノードが存在する事となります。

EVM 対 JVM

スマートコントラクトのMapReduce操作としての分散アプリケーションの組織立ては、Ethereum[5]へのプラズマ・アップグレードによって提案されました。しかしながら、EVMへの応用では、費用効果が高くレイテンシーの低い方法におけるスケーリングの同期構造の落とし穴がまだ残っています。Solidityプログラミング言語は、スマートコントラクトの開発における非決定論的な指示におけるリスクに対応するために作成されました。決定論的な指示の必要性は、スマートコントラクトの実行コストを計算する事、そしてDDoS攻撃に対する予防措置として、そしてネットワークへのスパムを送信するために生じます。しかしながら、コンパイルされたバイトコードがチェーン上で公証され、JVMで検証されると、決定論的コンパイルの要件は緩くなります。[6] そこで、燃料消費の削減、燃料制限の回避、およびマイクロサービスの複雑さを伴うスマートコントラクトの可能性が生じます。

各アプリケーションが既存のマイクロサービスであるレゴを組み立てるなどして、共通のビルディングブロックの構成として分散アプリケーションを設計することは理にかなっています。私たちは、ここからさらなる展開を進めることで、チェーン上の相互運用性を実現化します。これにより、ブロックチェーン企業は、より複雑な分散アプリケーションのビルディングブロックとして、他者が再利用できる分散アプリケーションを提供できるようになるのです。 真に分散したエコシステムは画一主義的なスマートコントラクトでは実践できません。このため、この種のコードの再利用は不可能で非効率的だといえます。

分散アプリケーションのコンジットとしてのJVMの使用は、大容量データコミュニティにおける業界標準とされています。消費者用アプリケーションは、JVMが提供する簡単なプロビジョニングに依存する自動スケーリンググループ上のマイクロサービスの集合として配置されます。これらのシナリオでは、数千のノードクラスターをAWS仮想インスタンスにプロビジョニングして自動スケーリングすることができます。Constellation（コンステレーション）のマイクロサービス構造では、同様のスケーラビリティを実現することができ、必要に応じてリソースをプロビジョニングでき、また、動的にスケーラブルな分散アプリケーション構造を提供することが可能なのです。

このタイプの分散型構造は、本質的に、関数型プログラミングが提供するモジュール性を必要とします。したがって、私たちは、Scalaプログラミング言語を使用して、任意のJVM言語で実施できるインタフェースを構築しています。

---

[5] V. Buterin 及びその他の者. https://plasma.io/plasma.pdf

[6] M. Hearn 'Corda: A 分散元帳' docs.corda.net

マイクロサービスが具体的なタイプとして設計されている場合、分散アプリケーションをソースコードで直接構成することができ、コンパイル時に検証が可能です。開発者は、適切なタイプのスマートコントラクトを組み合わせて複合アプリケーションにすることで、新しい機能を作成することが可能です。 したがって、計算ロジックのConstellation（コンステレーション）は、1つ以上のマイクロサービスまたは分散アプリケーションとして構成されます。Constellation（コンステレーション）のスマートコントラクトのインターフェースは、共分散および反変的な型を念頭に置いて設計されます。つまり、互換性のないマイクロサービスまたは分散アプリケーションが混在させた場合、すぐに認識されます。このため私たちは、他のブロックチェーンからのスマートコントラクトを含むConstellation（コンステレーション）を作成することもでき、また、Polka− dot[7]のようなクロス・チェーンの流動性などの未解決問題の解決策を見出すことも出来るのです。

私たちの機能プログラミングの精神におけるコンセンサスプロトコルは、質料形相論（ハイロモフィズム）のカテゴリの理論的定義によって記述することができます。

HyloChain − コンセンサス構造

Constellation（コンステレーション）のコンセンサス構造がハイロモーフィックであるため、私たちはコンセンサス構造にHyloChainと名前を付けました。コンセンサスのラウンドは、前回のラウンドのハッシュブロックを取得し、それを通常の取引としてトランザクションプールに追加します。トランザクションプールの充填はアンフォールド（展開）操作と同形となります。このチェックポイントブロックが前回のラウンドと新しいトランザクション（取引）で満たされると、ハッシュされます。これはフォールド（折り畳み）操作と同形となります。

定義: HyloChain

hylomorphism（ハイロモフィズム） h : A →C は、その独立したアナモルフィックな部分とカタモルフィックな部分に関して定義することができます。Hylomorphism（ハイロモフィズム）の定義を参照してください。

$$h = [(c, \oplus), (g, p)] \qquad (1)$$

アナモルフィック部分は、単項関数gによって定義することができます。
g : A → BA: 反復適用または展開を介してB内の要素のリストを定義します
述部p: A→終了条件を提供するブール値

カタモルフィック部分は、初期値の組み合わせとして定義することができます。
収束（折り畳み）のための$\in C$ と2項演算子$\oplus : B \times C \to C$ は収束（折り畳み）ために使用されます

[7]G. Wood 及びその他の者 https://github.com/w3f/polkadot-white-paper/raw/master/PolkaDotPaper.pdf

私たちのケースでは、メッセージのゴシップは私たちのアナモルフィックとなり、それらのメッセージのハッシングと前のブロックの結果は私たちのカタモルフィズムとなります。

演算子が暗号化ハッシュ関数として定義されている場合、⊕ = CHash
n番目の反復でg n =(n、n－1)かつp n = False(誤り)(私たちのチェーンは、終了条件を持たないため) HyloChain は分類上は下記のように定義されます。

$$HyloChain = [(\text{GenesisBlock}, \oplus), (g, p)] \qquad (2)$$

GenesisBlock(ジェネシスブロック)は私たちの出発点であり、つまり、HyloChainを導入する際に使用される起源ブロックとなります。

ExtendedTrustChain(拡張済みトラストチェーン)との関係

当社のコンセンサス構造であるHyloChainは、ExtendedTrustChain(拡張済みトラストチェーン)構造をベースにしており、次のように拡張しています。

ExtendedTrustChain(拡張済みトラストチェーン)では、各ノードはアカウントとして動作し、独自のチェーンの履歴を保持し、取引の順序に基づいて妥当性を確認します(ハッシュグラフ[8]と同様)。取引は、(ゴシップを介して)ネットワークにブロードキャストされたイニシエーターおよびカウンターパーティによって署名されます。デリゲートは、チェックポイントブロック内のトランザクション(取引)についてコンセンサスを行い、評判に基づいて選択されます。コンセンサス結果は再びブロードキャストされ、通常のトランザクション(取引)として次のブロックに追加されます。[9]

---

[8] L BAIRD およびその他の者http://www.swirlds.com/downloads/SWIRLDS-TR-2016-01.pdf

[9] Kelong Cong, およびその他の者 https://repository.tudelft.nl/islandora/object/uuid:86b2d4d8-642e-4d0f-8fc7-d7a2e331e0e9?collection=education

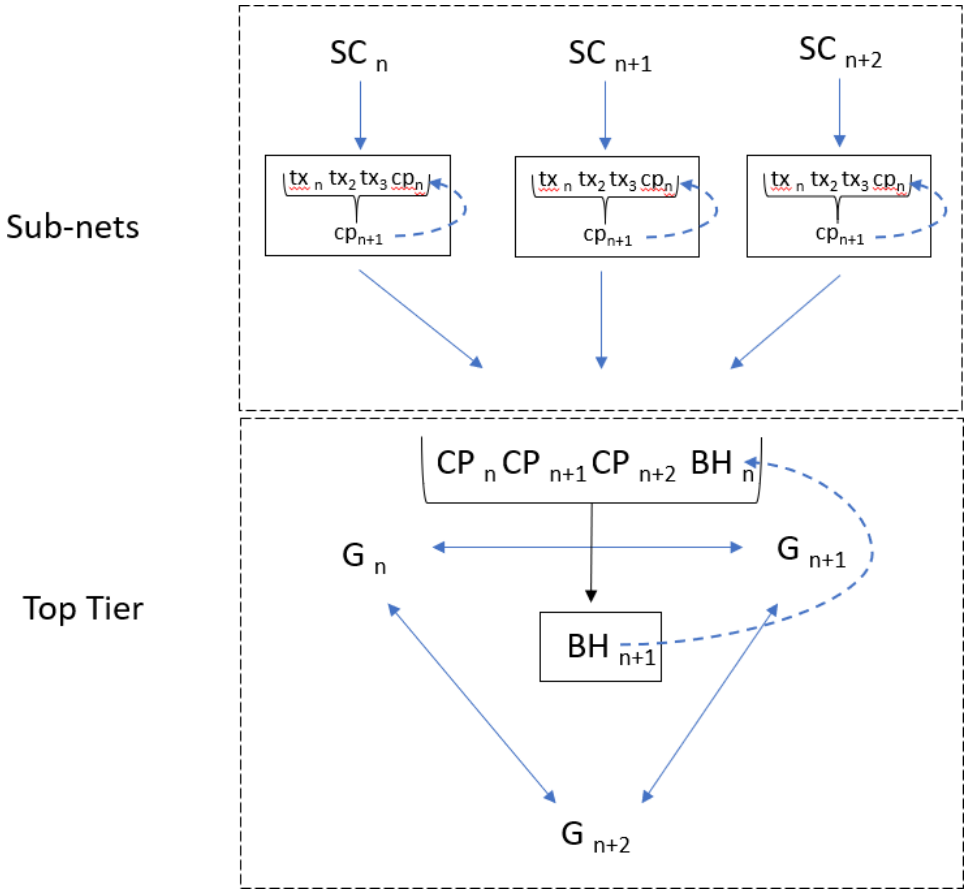

図3: HyloChain

フォールトトレランス

Constellation（コンステレーション）のHyloChainのフォールトトレランスは、典型的といえるビザンチン（Byzantine）コンセンサスモデルに従っています。コンセンサス参加者のコントロールが1/3に近づくにつれて敵対者がネットワークコンセンサスをコントロールする確率が1/2に近づきます。[10]. ネットワークのセキュリティを確保するために、私たちは、これらの境界を使用して、特定のパラメーターを選択することで、フォールトトレランスを調整し、トランザクションのスループットを最大化し、レイテンシーを削減することができます。

[10]M. Pease, R. Shostak and L. Lamport,「障害における合意形成」 ACM ジャーナル(JACM), vol. 27, no. 2

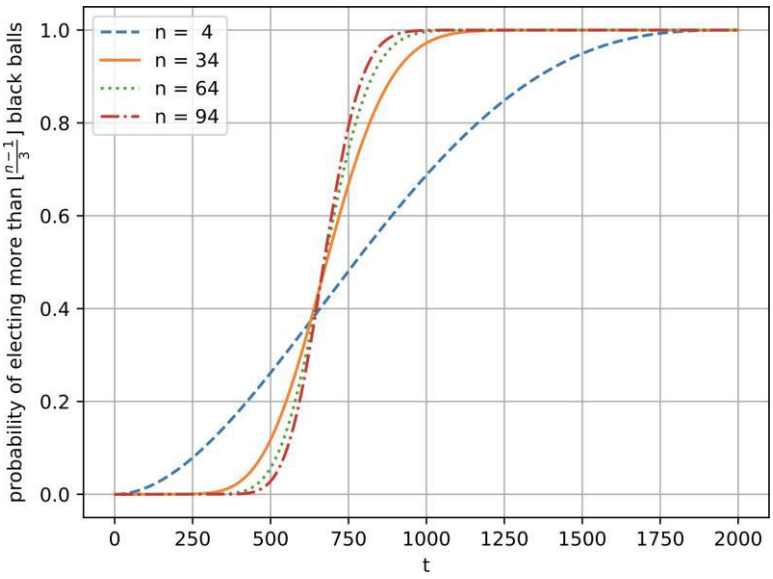

図4：母集団サイズが固定されたtの異なる値（ tはビザンチンアクターの数）に対して、（n－1）／ 3以上のものを選択する確率（超幾何分布の説明に由来する黒い球として示される） 2000年、K. Cong などその他の者

また、私たちのビザンチン（Byzantine）コンセンサスの手続き（Honeybadger BFTによって使用されているM. Ben-Orらが実施したACSを暫定的に使用しています）と取引の二重署名に基づき、ネットワークをフォークすることは不可能であるといえます。不正確なコンセンサスが起きる唯一の原因としてあげられるのは、チェックポイント･ブロックからの個々のトランザクション（取引）の検閲が行われた結果だといえます。すなわちコンセンサスの間に失われた取引がある事を意味します。これにより、資本の損失を引き起こす可能性はなく、この状態の緩和はネットワークに再送信するのと同じくらい簡単に行えます。

スループットの関係は、水平方向のスケーラビリティを考慮すると、参加者数に対して線形であるといえますが、フォールトトレランスの線形増加は通信コストの多項式増加につながります[11]。したがって、境界のあるフォールトトレランスにおいて、最大のスループットを保証するコンセンサスパラメーターおよび取引における確認時間を決定する必要があります。

そのような範囲において私たちは、ネットワークのスケール変更をどのように行うのでしょうか？私たちのアプローチでは、ネットワークのブロックチェーンをサブネットの階層（図1）として構成し、それぞれが独自のコンセンサスを実行し、その結果が通常の取引としてこれらのチェックポイント･ブロックを処理する上位層にバブリングされるのです。

[11] pp 50, K. Cong, およびその他の者

このアプローチは、Chain Fibers（検証コントロールを使用する場合も同様）と非常に似ていますが、ローカリティセンシティブハッシュの次元削減テクニックを適用してトランザクションプールのサイズを増やす点で異なります。この構造の効果は、Honeybadger BFTでコンセンサスアルゴリズムをデリゲート[12]ごとに$O(1)$に減らすことができるという事実を利用して、$O(n^3)$の平均的なケースまたはノードごとの二次的なものからコンセンサスの複雑さを減らすことであるため、下記の方程式が成り立ちます。

$$O(n) \rightarrow O(m) : m << n \tag{3}$$

したがって、これは、サブネット内のノードの数に関して劣線形であるといえます。

この場合、これらのチェックポイントブロックは、ローカリティセンシティブハッシュブロックと呼ばれます。各サブネットはローカリティセンシティブハッシュブロックを形成し、それを親ネットに送信します。（ギャラクシー、以下で説明します）。この親ネットは、次のラウンドのコンセンサスで通常のトランザクション（取引）としてローカリティセンシティブハッシュブロックを扱います。ネットワークの性能を向上させるために、サブネット内の隣接ノードの選択は、各ノードが自身の取引履歴を追跡するので、レイテンシーの最小化のみに依存する必要があります。

各レベルのスケールでは、ゴシップ・メッセージ・パッシングの同じ自己類似構造が採用されているため、ノードによって提供されるリソースの変更をするだけで、同じノードが各レベルに参加することができます。

Constellation（コンステレーション：星座）のすべてのノードは「スリープ」にすることができます。つまり、いつでもネットワークに参加したり離れることができるのです。新しいノードがネットワークに加わると、それらのリソースはサブネットに割り当てられます。上記で概説したように、サブネットがトランザクションスループットと暗号化セキュリティ（参加者数対進行役の数）の閾値に達すると、ノードを入力すると新しいサブネットを形成する必要があります。したがって、メンバーがネットワークを離れて参加するときに、新しいサブネットが動的に割り当てられます。この自己類似構造と以下に概説する私たちのデリゲート選択モデルの目標は、ネットワークリソースに基づいて動的自動スケーリングを可能にし、一貫したスループットを可能にすることです。
この構造が、分散コンピューティングなどの複雑なシステムでフォールトトレランスに適用されることで知られているスケールフリーのネットワークの本質的な指数法則を示すことは自明であるといえます。[13]

---

[12]M. Ben-Or and R. El-Yaniv. 一定の時間内における弾力的で最適なインタラクティブな一貫性。 分散コンピューティング16(4):249?262, 2003

[13]Ravasz, E.; Barabási (2003). ”複雑なネットワークにおける階層的組織”. Phys. Rev. E. 67: 026112.

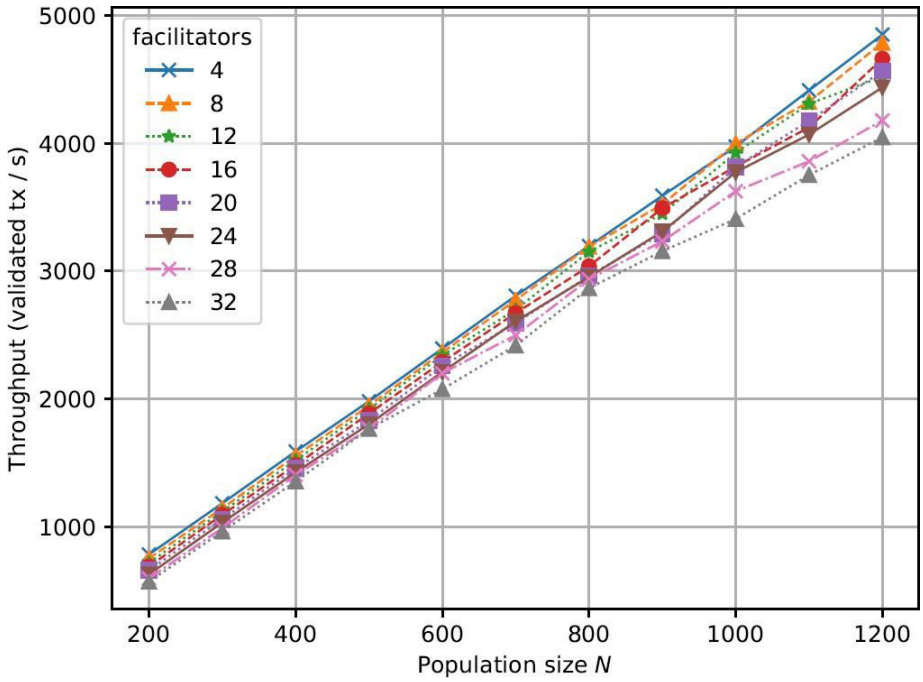

図5：HyloChainスループット対人口 K. Cong およびその他の者

デリゲート選択モデル：Proof-of-Meme（プルーフ･オブ･ミーム）

デリゲートを選択するための評判システムを使用することで、TrustChainsのフォールトトレランスを向上させることができます。[14] それゆえに、私たちはデリゲート選択に向けたノードの履歴に基づく参加形式を組み込んだ分散型コンセンサス方式であるProof-of-Meme（プルーフ･オブ･ミーム）を提示します。Meme（ミーム）は、普及可能な模範的なアイデアや行動を表す確かな単位だといえます。したがって、このConstellation（コンステレーション）全体の善意ある行為は報酬を与えられるに値するものであると共に、システム内のノード全体の評判を向上させるために模倣すべきであるといえるのです。Proof of Meme（プルーフ･オブ･ミーム） は、金権政治的（Plutocracy）であるProof of Stake （プルーフ・オブ・ステーク)に比べて、メリトクラシー（Meritocracy）的であるといえます。

私たちは、meme（ミーム）を各ノードのアカウントに対応する特徴ベクトルと見做しています。最も単純なケースでいうと、決定論的機械学習アルゴリズムへの入力として使用される浮動小数点値の行列であるといえます。私たちはネットワーク内のノードの評判を記述するために機能のオントロジーを使用しているREGRETの足跡をたどります。技術的には、これは特徴空間のテンソル積であるといえます。[15].

[14] pp 50, K. Cong, およびその他の者

[15] J. Sabater, REGRET: 大衆社会の評判モデル

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.8.7554rep=rep1type=pdf

このオントロジー（meme）を特徴空間として使用して対応する決定論的機械学習モデルを使用して、ノードの評判スコアを決定し、前記ノードがコンセンサスに参加する確率を推移的に決定します。

meme（ミーム）は評判スコアの基盤となるものであり、評判スコアはコンセンサスのために選択される確率を示唆します。

確率論的なデリゲートの選択は、非同期コンセンサスメカニズムにおけるBFTを改善するという文脈の中で広く研究されてきました。[16]. 具体的には、GURUに関する作業でA. Biryukovらは、ビザンチンコンセンサスにおける1/3の悪意のあるノードの典型的なフォールトトレランスを1/2まで（場合によってはそれ以上）改善できることを示しています。 私たちは、デリゲートの選択のための検証フレームワークをproof of meme（プルーフ･オブ･ミーム）に取り入れています。

また、Sybil 抵抗モデリングの問題を解決するExtended Trust Chain（拡張済みトラストチェーン）に関連する作業が存在します。具体的にいうと、P. Otte[17]は、ネットフローのアルゴリズムを使用して、シビル攻撃やページランクの改ざん、つまり一時的なページランクを防止し、評判状態を更新することが可能だと示唆しています。Constellation （コンステレーション）はまず一時的にページランクを複製し、パフォーマンス解析中に可能な限り改善を促します。

私たちは、各ノードの個別アカウントとしての役割とともに、Proof-of-Meme（プルーフ･オブ･ミーム）を使用したデリゲート選択の評判ベーススコアリングを使用することで、良好な行動を促す透明性を強制することができ、無許可ネットワークに許可されたシステムの利点がもたらされるように促しています。各ノード（および経緯的にマイクロサービス）に関するすべての取引およびコンセンサスの履歴は公に公証されます。 これにより、取引手数料や不平等なコンセンサス･メカニズムにおいて既存の技術では無視されている良好な行動を強制することができ、ある種の信頼を得ることができます。

Proof-Of-Meme （プルーフ･オブ･ミーム）の概要

私たちは、分散コンセンサスでにおけるなれあいの問題を解決する3つの作業体について説明してきました。 これらは、次に記載されているように、デリゲート選択のすべての要素となっています。私たちはREGRETで使用されているアプローチに従って、評判スコアのための特徴空間となるmeme（ミーム）のためのオントロジーを開発します。私たちは、評判状態を更新するために一時的なページランクに適応させています。また、私たちはGURUの検証フレームワークをデリゲート選択メソッドに組み込んでいます。詳細事項は、開発プロセス中に追加されます。評判オントロジーを構築し、評判スコアの決定論的モデルのトレーニングを実施し、デリゲート選択のためのサンプリングアルゴリズムを実施することが、私たちの開発の重要な要素なのです。

[16] Guru:分散コンセンサスプロトコルのためのユニバーサル評判モジュール、, A. Biryukov etなど他
https://eprint.iacr.org/2017/671.pdf

[17]P. Otte " 分散システムにおけるSybil-レジスタントトラストメカニズム"
https://repository.tudelft.nl/islandora/object/uuid:17adc7bd-5c82-4ad5-b1c8- a8b85b23db1f/datastream/OBJ/

ここでは、このスコアがデリゲート選択にどのように使用されるのかを詳しく説明します。各機能は、ネットワークへのノードの有用性を表しています（コンセンサスの際に重大な欠陥がある、コンセンサスにどれくらいのメモリを割り当てることができるかなど）meme（ミーム）は、上記のような関数、すなわちトレーニングを行ったモデルに適用することによって、数値に変換できます。これは、meme（ミーム）の評判をどのように数値化するかという事を意味します。新しいノードを促進してそのmeme（ミーム）を改善するメカニズムを提供するために、meme（ミーム）スコアの確率分布が一定の大きさのバケットに描かれます。

この分布は、特定のバケット内のスコアのmeme（ミーム）が選択される確率を示します。計算上の観点からは、これはおそらくノードをサブルーチンとして再クラスタリングするクラスタリングアルゴリズムとして実装されます。これは、同じレベルのフォールトトレランスおよび確認時間を維持しながら、ファシリテーター（進行役）の数を緩和してスループットを最大化することを意味します。評判の良いmeme（ミーム）には優先権が与えられますが、新しいmeme（ミーム）にも参加してネットワーク上で良い評判を築く機会が与えられます。パフォーマンスのログは連鎖的に公証されるため、欠陥のあるリソースを提供したり、敵対的な方法で行動しているmeme（ミーム）は、評判が低下し、高いパフォーマンスをしたり信頼性の高いmeme（ミーム）は増加します。帰納的ケースでのデリゲート選択における私たちの考え方は次のとおりです。:

1)コンセンサスが実施されます。

2)ハッシュブロックは、各デリゲートの更新されたmeme（ミーム）スコアを出力する決定論的アルゴリズムに供給されるます。これは、ファシリテーター（進行役）の選定を担当するコンセンサス層（ギャラクシー）で実施されます。

3)この定数はディストリビューション（分布）をシャッフルするために使用されます（パフォーマンスの新しいデータに基づいてバケット間でmemeを移動させます）。

4）前回のブロックハッシュ（最後のコンセンサスの結果）を使用して各バケットの内容をソートし、各バケットからの上位N（確率分布に従って）をこのラウンドで選択します。失敗した、または有害である可能性のある（ログ内で検証可能な）コンセンサスに参加したmeme（ミーム）は、次のコンセンサスへの参加に向けた選択でドッキングされます。

Meme（ミーム）の評判がより価値を持ったものになると、取引手数料の必要性が置き換えられることとなります。 マイクロサービスは、サービスホスティングを行うために評判の良いmemeを探すかもしれません。なぜなら、コンセンサスを実施するために取引手数料を稼ぐよりも利益を上げられる可能性があるからです。レイテンシーが高いときには価格を上げる代わりに、低い評判スコアを

持つmemeは抑制されます。 つまり、その取引はより低い優先度で処理されることとなります。この場合、memeはリソースを提供し（コンセンサスに参加する）、それによってネットワークのスループットが向上し、ネットワークの帯域幅の問題が解決されます。

あなた自身のmeme（ミーム） を所有する

Constellation（コンステレーション）にはネットワーク取引手数料がかからず、また、レイテンシーや敵対的攻撃を防ぐことができるため、私たちのメカニズムは良い評判を得ています。私たちのネットワークをシードするために、トークンの作成が終了する予定の日まで、リターンを減らしながら、最初のジェネシスブロック（ICO参加者のためのトークンを割り当てる）の後に、新しく作成されたトークンの形でネットワークインセンティブが提供されます。
新しく作成されたトークンは、それぞれのmeme（ミーム）の評判に基づいて割り当てられます。

販売管理システムのポイントについて、特に自動販売機のシナリオに焦点をあてて考えてみましょう。Constellation（コンステレーション）で動作する自動販売機は、コンセンサスに参加することを選択するマイクロサービスを実施するのに使用できます。meme（ミーム）の評判は時間とともに増大し、ハイ·トラフィックな状況下でも短い取引時間が保証されます。これは、実世界で競争力のあるアプリケーションのサービスを提供するために、高いスループットが不可欠な要素である事に起因しています。このように自動販売機が、高い評判を有しているという事は、たとえ自動販売機の顧客が低いmeme（ミーム）の評判を有していても、ハイ·トラフィック期間中にも、顧客が便宜的に商品を受け取れることを保証することを意味します。

ピアツーピア構造

前のセクションでは、ピアツーピア層の構造に触れましたが、ここでは、この構造を詳細に見ていきましょう。

Stars（スター）

Stars（スター）はConstellation（コンステレーション）の基本オブジェクトです。ネットワークで直接、情報を交換するためには、ユーザーが、デバイス上のスターノードをインスタンスを作

成し、このスターによるインスタンスを通じて取引が発行されます。各スターには、ネットワーク上の履歴から構成されるローカルチェーンが含まれています。このローカルチェーンは順序付けを強制するために使用され、またプライベートのカウンターパートが取引に署名するために使用されるパブリックキーによって識別されます。スター自体は、ユーザーがネットワークとやりとりするための軽量なクライアントであり、モバイルデバイスと元々互換性があります。しかしながら、スターの希望により、コンセンサスに参加することを選ぶことができ、それによりmeme（ミーム）の評判を理解することが可能となります。.スターがコンセンサスに参加することを選択した場合、スタークラスターと呼ばれるスターの集合体に参加することとなります。

Star Clusters（スタークラスター）

スタークラスターは、コンセンサスに参加することを決めたスターの集まりのことです。スターの総数は、上記で概説した十分な実験的および統計的分析の後に決定される上限によって制限されます。この閾値に達すると、新しいスタークラスターが作成され、それぞれのスタークラスターがローカリティセンシティブハッシュブロックを形成します。これらのローカリティセンシティブハッシュブロックは、通常の取引のように扱われ、ギャラクシー、すなわちブラックホールによってハッシュされます。

Galaxies（ギャラクシー）

ギャラクシーは、ExtendedTrustChain（拡張済みトラストチェーン）でバリデーターと同型の役割をしますが、オートスケーリンググループとしても機能し、新しいスタークラスターのリソースをプロビジョニングし、meme（ミーム）の評判を維持します。Meme（ミーム）の評判は、コンセンサスのパフォーマンスのログを調べることによって計算されます。提案されたブロックの内容に関するメタデータ、レイテンシー、および不履行は、ローカリティセンシティブハッシュブロックとともにギャラクシーに送信されます。ギャラクシーがそのデータを受け取ると、次のコンセンサスのために新しいサンプルが選ばれる前に、meme（ミーム）の評判が更新されます。ギャラクシーはバリデーターであるため、無効な取引を取り除き、スタークラスターのデリゲートを決定する真のソースとなります。ギャラクシーのパフォーマンスに関するメタデータはブラックホールに保存されています。.スターが蓄積され、評判の閾値に達すると、ギャラクシーノードとして機能する権利を得ることができます。

Black Holes（ブラックホール）

ブラックホールは、ハッシュされたローカリティセンシティブハッシュブロックのブロックです。これらは、ブロックチェーンのブロックを呼ぶのと同じ意味で使用されます。ギャラクシーはブロックチェーンの履歴を完全に保存しています。

マイクロサービスとしてのスマートコントラクト

高可用性および弾性のある分散システムは、サーバーレスの構造で成長をみせます。分散オペレーティングシステムの場合、これは分散型マイクロサービスのネットワークを使用することで、実現させることができます。したがって、Constellation（コンステレーション）では、スマートコントラクト自体がJVM上で動作するマイクロサービスとなっております。それらは取引を送信し、また、カウンターパーティーとして署名し、コンセンサスを行うことができます。

目標は、マイクロサービス自体が、合意された金額のサービスを提供すると共に、対応するmeme（ミーム）を持つスターとして機能することです。それらは、Ethereumまたはカウンターパーティーのスマートコントラクトと同じ役割を果たすことができますが、JVMエコシステムの既存のコードベースを利用することにより、さらに複雑なロジックを提供できます。さらに、RPCインターフェイスを使用して外部プログラムと通信することもできます。これらのマイクロサービスが具体的なサービスレベルでのアグリーメントまたはそれより優れた署名により構築されている場合、それらのコンポジットロジックをチェーン化して分散アプリケーションに直感的に組み立てることができます。これは、MapReduceオペレーターが参加し操作する場所です。.スマートコントラクトマイクロサービスは、計算上の複雑さを改善できるデータモデル（非同期構造を想定）を送受信するように設計でき、新しいアプリケーションに再利用することができます。

上記の天体によるメタファーを考慮すると、Constellation（コンステレーション）上の分散アプリケーションは星座そのものであることが分かります。各マイクロサービスはスターであるため、連鎖および/または合成されたマイクロサービスのコレクションは、アプリケーションを表す線で接続することが可能です。これらが描かれるとき、星座が生成されるのが明白に分かります。

ブロックチェーンオペレーティングシステムとしてのConstellation（コンステレーション）

上記のサーバーレス構造は、分散オペレーティング・システムの実例です。オペレーティングシステムの目標は、基礎となるハードウェアのリソースを利用するためのインターフェースを提供することです。これは、高度なプログラミング言語と直感的なユーザーインターフェイスを可能にするアプリケーション開発する際、不可欠な事だと言えます。分散オペレーティングシステムはこれとは異なり、分散コンピューティングクラスターの基礎となるリソースのインターフェイスを提供することを目的としています。ブロックチェーンは本質的に分散システムであるため、ブロックチェーン互換アプリケーションを作成するには、基盤となるクラスターをフルパワーで利用するための分散オペレーティングシステムが必要となります。通常、オペレーティングシステムには、基本となるハードウェアの状態を維持するプログラムが必要です。これはカーネルとして知られています。サーバーレス構造で分散オペレーティングシステムを構築することは可能です。MicroOsのようなシナリオでは、オペレーティングシステムはマイクロサービスの集合体となっています。これをアプリケーション開発者の観点から推定して見てみると、マイクロサービスはConstellation（コンステレーション）で構築された包括的なアプリケーションのためのレゴのようなビルディングブロックであるといえます。これの目標は、技術者でなくても、既存のマイクロサービスを利用してアプリケーションを構築できるようにすることです。Constellation（コンステレーション）では、技術者以外のユーザーでも、直感的なユーザーインターフェイスと組み合わせ、スムーズに分散アプリケーションを開発することができます。

上記で説明したように、アイデアをアプリケーションにシームレスに（非技術的に）実現する能力は正に、ノウアスフィアから物質世界への潜在的エネルギーの変容であると言えます。このため、私たちは、私たちの目標とアイデアを構成する非球面の潜在的なエネルギーに照らして、私たちの通貨の名前をnoOs,と決定しました。

結論

私たちは、現代のサーバーレス構造を暗号化し、安全性を提供できるコンセンサスを再構成することを提案しております。それにより、メインストリームアプリケーションでブロックチェーンテクノロジーを使用することが可能となります。私たちは、インターネットスケールにおけるトラストチェーンを可能にする分裂と征服のメカニズムとして、ローカリティセンシティブハッシングの次元削減テクノロジーを適用させています。 私たちのmeme （ミーム）経済は、暗号セキュリティにおける深いレイヤーおよび取引手数料を廃止する可能性を提供します。さらに、私たちは、分散アプリケーションと自然界における自己相似スケーリングパター

ンとの間の直感的な相関関係を描く、分散構造の天体によるメタファーを提供しています。