(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7361103号
(P7361103)

(45)発行日　令和5年10月13日(2023.10.13)

(24)登録日　令和5年10月4日(2023.10.4)

(51)国際特許分類
Ｇ０６Ｆ　21/64　(2013.01)

ＦＩ
Ｇ０６Ｆ　21/64

請求項の数　23　（全40頁）

(21)出願番号　特願2021-512757(P2021-512757)
(86)(22)出願日　令和1年8月29日(2019.8.29)
(65)公表番号　特表2022-500738(P2022-500738A)
(43)公表日　令和4年1月4日(2022.1.4)
(86)国際出願番号　PCT/EP2019/073066
(87)国際公開番号　WO2020/057930
(87)国際公開日　令和2年3月26日(2020.3.26)
審査請求日　令和4年1月25日(2022.1.25)
(31)優先権主張番号　16/135,260
(32)優先日　平成30年9月19日(2018.9.19)
(33)優先権主張国・地域又は機関
米国(US)

(73)特許権者　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国１０５０４　ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
Ｎｅｗ　Ｏｒｃｈａｒｄ　Ｒｏａｄ，　Ａｒｍｏｎｋ，　Ｎｅｗ　Ｙｏｒｋ　１０５０４，　Ｕｎｉｔｅｄ　Ｓｔａｔｅｓ　ｏｆ　Ａｍｅｒｉｃａ
(74)代理人　100112690
弁理士　太佐 種一

最終頁に続く

(54)【発明の名称】　演算および信用できる確認のための分散型プラットフォーム

(57)【特許請求の範囲】

【請求項１】

演算システムであって、

シミュレーションのデータを受信し、前記シミュレーション・データ内のチェックポイントを識別し、前記識別されたチェックポイントに基づいて複数の順次データ構造を生成するように構成されたプロセッサであって、各データ構造が、前記順次データ構造のうちの前のシミュレーションの状態のデータ構造に対して前記シミュレーションの状態の進展を識別する、前記プロセッサと、

データ・ブロックのハッシュリンク・チェーン内の１つまたは複数のデータ・ブロック内に含めるために、前記順次データ構造をブロックチェーン・ネットワークのノードに送信するように構成されたネットワーク・インターフェースと

を備える演算システムであって、

前記ネットワーク・インターフェースが、前記生成された順次データ構造のうちのデータ構造に関連付けられたシミュレーション・データの状態が無効になったことを示すメッセージを、ブロックチェーン・ピア・ノードから受信するようにさらに構成されている、演算システム。

【請求項２】

各データ構造の幅が、前記プロセッサによって、前記受信した前記シミュレーションのデータ内の前記それぞれのデータ構造に対応するチェックポイントの頻度に基づいて、適応的に決定される、請求項１に記載の演算システム。

【請求項３】
前記シミュレーションが、モデルの反復シミュレーションと、入力と出力のペアのセットを含む非反復シミュレーションとのうちの１つを含む、請求項１に記載の演算システム。
【請求項４】
前記プロセッサが、前記シミュレーションの演算された連続状態のうちの各データ構造内に反復ＩＤを格納するようにさらに構成され、前記反復ＩＤが、前記それぞれのデータ構造に関連付けられた反復シミュレーションのそれぞれの反復を識別する、請求項１に記載の演算システム。
【請求項５】
前記プロセッサが、前記無効のデータ構造に含まれる前記シミュレーション・データの前記状態を精製して、更新済みデータ構造を生成し、かつ前記ネットワーク・インターフェースを制御して、前記更新済みデータ構造を確認のために前記ブロックチェーン・ピア・ノードに送信するようにさらに構成されている、請求項４に記載の演算システム。
【請求項６】
前記プロセッサが、１つまたは複数のモデルをシミュレートして前記シミュレーション・データを生成するようにさらに構成されている、請求項１に記載の演算システム。
【請求項７】
方法であって、
シミュレーションのデータを取得することと、
前記シミュレーション・データ内のチェックポイントを識別することと、
前記識別されたチェックポイントに基づいて複数の順次データ構造を生成することであって、各データ構造が、前記順次データ構造のうちの前のデータ構造に対して前記シミュレーションの進展状態を識別する、前記生成することと、
データ・ブロックのハッシュリンク・チェーン内の１つまたは複数のデータ・ブロック内に含めるために、前記生成された順次データ構造をブロックチェーン・ネットワークのノードに送信することと、
前記生成された順次データ構造のうちのデータ構造に関連付けられたシミュレーション・データの状態が無効になったことを示すメッセージを、ブロックチェーン・ピア・ノードから受信することと
を含む方法。
【請求項８】
各データ構造の幅が、前記取得した前記シミュレーションのデータ内の前記それぞれのデータ構造に対応するチェックポイントの頻度に基づいて、適応的に決定される、請求項７に記載の方法。
【請求項９】
前記シミュレーションが、モデルがトレーニングされる反復シミュレーションと、入力と出力のペアのセットが処理される非反復シミュレーションとのうちの１つを含む、請求項７に記載の方法。
【請求項１０】
前記シミュレーションの演算された連続状態のうちの各データ構造内に反復ＩＤを格納することをさらに含み、前記反復ＩＤが、前記それぞれのデータ構造に関連付けられた反復シミュレーションのそれぞれの反復を識別する、請求項７に記載の方法。
【請求項１１】
前記無効のデータ構造に含まれる前記シミュレーション・データの前記状態を精製して、更新済みデータ構造を生成することと、前記更新済みデータ構造を確認のために前記ブロックチェーン・ピア・ノードに送信することとをさらに含む、請求項１０に記載の方法。
【請求項１２】
前記シミュレーションを実行して、前記シミュレーションの前記データを生成することをさらに含む、請求項７に記載の方法。
【請求項１３】

コンピュータ・プログラムを記録した非一過性コンピュータ可読媒体であって、
請求項１ないし１２のいずれか１項に記載の方法の各ステップをコンピュータに実行させるための、前記コンピュータ・プログラムを記録した非一過性コンピュータ可読媒体。
【請求項１４】
演算システムであって、
反復シミュレーションの順次データ構造の中から、前記反復シミュレーションの状態データを格納するデータ構造を受信するように構成されたネットワーク・インターフェースと、
前の反復シミュレーションのデータ構造から前に確認された反復シミュレーションの状態データに基づいて、前記データ構造についての状態データをローカルに演算で生成し、かつ状態データの前記ローカルに演算した状態データと前記受信済みデータ構造内の前記状態データとの類似性を判定するように構成されたプロセッサと
を備え、
前記プロセッサが、前記類似性が所定の閾値内にあるという判定に応答して、前記ネットワーク・インターフェースを制御して、データ・ブロックのハッシュリンク・チェーンのうちのデータ・ブロック内に含めるために、前記データ構造のエンドースメントをブロックチェーン・ネットワークに送信するようにさらに構成されている、
演算システム。
【請求項１５】
前記反復シミュレーションがモデルの深層学習シミュレーションを含み、前記順次データ構造が、反復による前記モデルの状態の変化を格納する、請求項１４に記載の演算システム。
【請求項１６】
前記所定の閾値が、前記ローカルに演算した状態データと前記受信済みデータ構造内の前記状態データとの偏差の許容レベルを識別する、請求項１４に記載の演算システム。
【請求項１７】
前記プロセッサが、前記類似性が前記所定の閾値を超えるという判定に応答して、前記ネットワーク・インターフェースを制御して、前記状態データが無効であることを示すメッセージを、前記データ構造をサブミットしたクライアント・ノードに送信するようにさらに構成されている、請求項１４に記載の演算システム。
【請求項１８】
前記ネットワーク・インターフェースが、精製された状態データを含む更新済みデータ構造を受信するようにさらに構成され、前記プロセッサが、前記状態データの演算した位置と前記更新済みデータ構造内の前記精製された状態データとの類似性を判定するように構成されている、請求項１７に記載の演算システム。
【請求項１９】
方法であって、
反復シミュレーションの順次データ構造の中から、前記反復シミュレーションの状態データを格納するデータ構造を受信することと、
前の反復シミュレーションのデータ構造から前に確認された反復シミュレーションの状態データに基づいて、前記データ構造についての状態データをローカルに演算で生成することと、
状態データの前記ローカルに演算した状態データと前記受信済みデータ構造内の前記状態データとの類似性を判定することと、
前記類似性が所定の閾値内にあるという判定に応答して、データ・ブロックのハッシュリンク・チェーンのうちのデータ・ブロック内に含めるために、前記データ構造のエンドースメントをブロックチェーン・ネットワークに送信することと
を含む方法。
【請求項２０】
前記反復シミュレーションが、モデルがトレーニングされる深層学習シミュレーション

を含み、前記順次データ構造が、反復による前記モデルの状態の変化を格納する、請求項１９に記載の方法。
【請求項２１】
前記所定の閾値が、前記ローカルに演算した状態データと前記受信済みデータ構造内の前記状態データとの偏差の許容レベルを識別する、請求項１９に記載の方法。
【請求項２２】
前記類似性が前記所定の閾値を超えるという判定に応答して、前記状態データが無効であることを示すメッセージを、前記データ構造をサブミットしたクライアント・ノードに送信することをさらに含む、請求項１９に記載の方法。
【請求項２３】
精製された状態データを含む更新済みデータ構造を受信することと、前記状態データの演算した位置と前記更新済みデータ構造内の前記精製された状態データとの類似性を判定することとをさらに含む、請求項２２に記載の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本出願は、一般に、シミュレーションおよび学習システムに関し、より詳細には、ブロックチェーンなどの非集中型データベースであって、演算の信用できる確認および検証において複数のノードが共有するブロックチェーンに、進展中の（ｅｖｏｌｖｉｎｇ）演算状態が格納される、非集中型データベースに関する。
【背景技術】
【０００２】
集中型データベースは、１つの位置において単一のデータベース（例えば、データベース・サーバ）にデータを格納して維持する。この位置は、多くの場合、中央コンピュータ、例えば、デスクトップ中央処理ユニット（ＣＰＵ）、サーバＣＰＵ、またはメイン・フレーム・コンピュータである。集中型データベースに格納されている情報には、通常、複数の異なる点からアクセス可能である。複数のユーザまたはクライアント・ワークステーションが、例えば、クライアント／サーバ構成に基づいて、集中型データベース上で同時に作業することができる。集中型データベースは、その単一の位置により、特にセキュリティの目的で管理、維持、および制御が容易である。集中型データベース内では、すべてのデータの単一の格納場所が、データの所与のセットが１つのプライマリ・レコードのみを有することをも意味するため、データの完全性が最大化され、データの冗長性が最小化される。これは、データをできるだけ正確に、かつ一貫性を有して維持することを助け、データの信頼性を高める。
【０００３】
しかしながら、集中型データベースは大きな欠点を有する。例えば、集中型データベースは単一障害点を有する。特に、フォールトトレランスが設定されておらず、ハードウェアの故障が発生した場合、データベース内のすべてのデータが失われ、すべてのユーザの作業が中断される。加えて、集中型データベースは、ネットワーク接続性に大きく依存する。そのため、インターネット接続が遅いほど、データベース・アクセスごとに必要な時間が長くなる。別の欠点は、単一の位置に起因して、集中型データベースの通信量が多いときに障害が生じることである。さらに、集中型データベースは、データの１つのコピーのみがデータベースによって維持されるため、データへのアクセスが限られる。そのため、複数のユーザが、格納されたデータに上書きするなどの問題を生じさせることなく、同じデータに同時にアクセスすることができない場合がある。さらに、中央データベース・システムにはデータ冗長性が最小限しかない、またはまったくないため、データのセットが予期せず失われた場合、バックアップ・ディスク・ストレージからの手動による動作以外の方法でそれを取り出すことは困難である。
【０００４】
ブロックチェーン・システムなどの非集中型データベースは、集中型データベースの欠

点に対処可能なストレージ・システムを提供する。ブロックチェーン・システムでは、複数のピア・ノードが分散型台帳を格納する。クライアントは、ピア・ノードと対話してブロックチェーンにアクセスすることができる。ピア・ノードは、異なる関心を有する異なるエンティティによって制御され得るため、互いに信用できるエンティティではない。さらに、ブロックチェーンには中央集権がない。したがって、信用できる方法で分散型台帳にデータを追加する、または分散型台帳においてデータを変更するために、ピア・ノードのコンセンサスを行わなければならない。コンセンサスは、信用できないピア・ノードのブロックチェーン・システムにおいて信用を実現するための方法を提供する。

【０００５】

一方、演算および学習システムが、大規模な含意（ｉｍｐｌｉｃａｔｉｏｎ）を有するシナリオについて、政策規模で理解し、推論し、かつ最も重要なことには、決定を行うために、様々な環境においてますます導入されている。例えば、シミュレーション環境を使用して、病気の伝播、ならびに、異なる地理的および人口統計学的状況における様々な制御機構の影響を理解することができる。同様に、深層学習は、システムの基本的な状態を学習するために使用することができ、自動運転車、分析論、サプライ・チェーンなどの日常的な適用、および多くの他の適用においてますます導入されている。

【０００６】

別の例として、疫学および気象学などの分野に関連する演算には、数週間に及ぶ実行時間がかかる場合がある。例えば、マラリア・データ科学者（ＭＤＳ）は、２つ以上の演算モデル・エージェント（例えば、ＯｐｅｎＭａｌａｒｉａおよび疫学モデリング・ソフトウェア（ＥＭＯＤ））を使用して、病気の拡大をシミュレートするための実験を行うことがある。モデルは、多くの別個のシミュレーションを実行して、モデル構造またはパラメータの変化の影響を比較する。実験の目的は、病気を根絶するための有効な政策および介入戦略を考え出すことである。

【０００７】

これらの大規模シミュレーション・ベースのシステムでは、低リソースのソーシャルグッドな状況で適時に決定を行うために、複数の異質な当事者間で入力および出力を共有する必要がしばしばある。しかしながら、これらの演算および推論（例えば、深層ニューラル・ネットワークのトレーニングなど）の多くは、個人によって、およびしばしば独立したエージェントによって、ローカルに行われるため、信用できない対話システムの体系となる。そのようなシステムが有する影響力の規模を考慮すると、個々のエージェントにより推論される結果が信用でき、引き継ぐことができる（ｔｒａｎｓｆｅｒａｂｌｅ）ことが重要である。さらに、エンドーサ（ｅｎｄｏｒｓｅｒ）が、誤っている、対抗している（ａｄｖｅｒｓａｒｉａｌ）、または演算が遅いことがある。したがって、確認は、各フレームをエンドースするための時間に関して大きな犠牲を払うことがある。例えば、エンドーサがストラグラ（ｓｔｒａｇｇｌｅｒ）である場合には、エンドースメント取得が過度に遅れることがある。したがって、システム内のエンティティが、個々のエージェントによって得られた結論の有効性を信用することができ、そのような有効性が効率的に実行されることが、その結論をさらなる研究および開発のためにまとめて使用するために必須である。透明性もそのようなシステムにおける出所（ｐｒｏｖｅｎａｎｃｅ）を保証し、誤りの発生源を識別することができることが重要である。

【発明の概要】

【０００８】

１つの例示的な実施形態は、シミュレーションのデータを受信すること、シミュレーション・データ内のチェックポイントを識別すること、および、識別されたチェックポイントに基づいて複数の順次データ構造を生成することであって、各データ構造が、順次データ構造のうちの前のデータ構造に対してシミュレーションの状態の進展を識別する、生成すること、のうちの１つまたは複数を行うように構成されたプロセッサと、データ・ブロックのハッシュリンク・チェーン内の１つまたは複数のデータ・ブロック内に含めるために、順次データ構造をブロックチェーン・ネットワークのノードに送信するように構成さ

れたネットワーク・インターフェースと、のうちの１つまたは複数を備える演算システムを提供することができる。

【０００９】

別の例示的な実施形態は、シミュレーションのデータを取得することと、シミュレーション・データ内のチェックポイントを識別することと、識別されたチェックポイントに基づいて複数の順次データ構造を生成することであって、各データ構造が、順次データ構造のうちの前のデータ構造に対してシミュレーションの進展状態を識別する、生成することと、データ・ブロックのハッシュリンク・チェーン内の１つまたは複数のデータ・ブロック内に含めるために、生成された順次データ構造をブロックチェーン・ネットワークのノードに送信することと、のうちの１つまたは複数を含む方法を提供することができる。

【００１０】

さらなる例示的な実施形態は、プロセッサによって読み取られたとき、シミュレーションのデータを取得することと、シミュレーション・データ内のチェックポイントを識別することと、識別されたチェックポイントに基づいて複数の順次データ構造を生成することであって、各データ構造が順次データ構造のうちの前のデータ構造に対してシミュレーションの進展状態を識別する、生成することと、データ・ブロックのハッシュリンク・チェーン内の１つまたは複数のデータ・ブロック内に含めるために、生成された順次データ構造をブロックチェーン・ネットワークのノードに送信することと、のうちの１つまたは複数をプロセッサに実行させる命令を含む、非一過性コンピュータ可読媒体を提供することができる。

【００１１】

別の例示的な実施形態は、反復シミュレーションの順次データ構造の中から、反復シミュレーションの状態データを格納するデータ構造を受信するように構成されたネットワーク・インターフェースと、前のデータ構造から前に確認された状態データに基づいて、データ構造についての状態データのローカル演算を生成すること、および状態データのローカル演算と受信済みデータ構造内の状態データとの類似性を判定すること、のうちの１つまたは複数を行うように構成されたプロセッサと、のうちの１つまたは複数を備え、プロセッサが、類似性が所定の閾値内にあるという判定に応答して、ネットワーク・インターフェースを制御して、データ・ブロックのハッシュリンク・チェーンのうちのデータ・ブロック内に含めるために、データ構造のエンドースメントをブロックチェーン・ネットワークに送信するようにさらに構成され得る、演算システムを提供することができる。

【００１２】

別の例示的な実施形態は、反復シミュレーションの順次データ構造の中から、反復シミュレーションの状態データを格納するデータ構造を受信することと、前のデータ構造から前に確認された状態データに基づいて、データ構造についての状態データのローカル演算を生成することと、状態データのローカル演算と受信済みデータ構造内の状態データとの類似性を判定することと、類似性が所定の閾値内にあるという判定に応答して、データ・ブロックのハッシュリンク・チェーンのうちのデータ・ブロック内に含めるために、データ構造のエンドースメントをブロックチェーン・ネットワークに送信することと、のうちの１つまたは複数を含む方法を提供することができる。

【図面の簡単な説明】

【００１３】

【図１】例示的な実施形態による、シミュレーション・データを確認し格納するためのブロックチェーン・ネットワークを示す図である。

【図２Ａ】例示的な実施形態による、アセット共有シナリオについてのピア・ノード・ブロックチェーン・アーキテクチャ構成を示す図である。

【図２Ｂ】例示的な実施形態による、ブロックチェーン・ネットワークのノード間の通信シーケンスを示す図である。

【図３】例示的な実施形態による、許可型ブロックチェーン・ネットワークを示す図である。

【図４Ａ】例示的な実施形態による、クライアント・ノードがシミュレーション・データのフレームを生成するプロセスを示す図である。
【図４Ｂ】例示的な実施形態による、エンドーシング・ノードがシミュレーション・データのフレームを確認するプロセスを示す図である。
【図４Ｃ】例示的な実施形態による、順序付けノードがシミュレーション・データのフレームをブロック内に配置するプロセスを示す図である。
【図５Ａ】例示的な実施形態による、ブロックチェーンに格納するためのシミュレーション・データのフレームを生成する方法を示す図である。
【図５Ｂ】例示的な実施形態による、シミュレーション・データのフレームをエンドースする方法を示す図である。
【図５Ｃ】例示的な実施形態による、並列エンドースメントのためにシミュレーション・データの連続フレームを割り当てる方法を示す図である。
【図５Ｄ】例示的な実施形態による、並列エンドースメント処理の結果としてのシミュレーション・データを順序付けする方法を示す図である。
【図５Ｅ】例示的な実施形態による、シミュレーション・コンテンツを圧縮する方法を示す図である。
【図５Ｆ】例示的な実施形態による、シミュレーション・コンテンツをエンドースする方法を示す図である。
【図６Ａ】例示的な実施形態による、本明細書に記載の１つまたは複数の動作に従ってブロックチェーン上で様々な動作を実行するように構成された物理的インフラストラクチャを示す図である。
【図６Ｂ】例示的な実施形態による、ブロックチェーン上でスマート・コントラクトの条項を執行するように構成された、契約当事者と仲介サーバとの間のスマート・コントラクト構成を示す図である。
【図６Ｃ】例示的な実施形態による、ブロックチェーン上でスマート・コントラクトの条項を執行するように構成された、契約当事者と仲介サーバとの間のスマート・コントラクト構成を示す図である。
【図６Ｄ】例示的な実施形態による、別の例示的なブロックチェーン・ベースのスマート・コントラクト・システムを示す図である。
【図７Ａ】例示的な実施形態による、新しいブロックをブロックチェーン台帳に追加するプロセスを示す図である。
【図７Ｂ】例示的な実施形態による、ブロックチェーンのためのデータ・ブロック構造のコンテンツを示す図である。
【図８】例示的な実施形態のうちの１つまたは複数をサポートするように構成された例示的なコンピュータ・システムを示す図である。

【発明を実施するための形態】

【００１４】

本明細書の図面に概略的に説明され示されているように、本明細書の構成要素を様々な異なる構成で配置および設計してもよいことが、容易に理解されるだろう。したがって、添付図面に表された方法、装置、非一過性コンピュータ可読媒体、およびシステムのうちの少なくとも１つの実施形態に関する以下の詳細な説明は、特許請求されている出願の範囲を限定することを意図したものではなく、選択された実施形態を代表するものに過ぎない。

【００１５】

本明細書全体を通して説明される特徴、構造、または特性は、１つまたは複数の実施形態において、任意の適切な方法で組み合わせることができる。例えば、「例示的な実施形態」、「一部の実施形態」という語句、またはその他の同様の言葉の使用は、本明細書全体を通じて、実施形態に関連して説明される特定の特徴、構造、または特性が少なくとも１つの実施形態に含まれ得ることを指す。したがって、「例示的な実施形態」、「一部の実施形態において」、「その他の実施形態において」という語句、またはその他の同様の

言葉の出現は、本明細書全体を通じて、必ずしもすべてが実施形態の同じグループを指しておらず、説明される特徴、構造、または特性は、１つまたは複数の実施形態において、任意の適切な方法で組み合わせることができる。

【００１６】

加えて、「メッセージ」という用語が実施形態の説明において使用されていることがあるが、本出願は、パケット、フレーム、データグラムなどの多くのタイプのネットワーク・データに適用することができる。「メッセージ」という用語は、パケット、フレーム、データグラム、および任意のこれらと同等のものも含む。さらに、特定のタイプのメッセージおよび信号伝達が例示的な実施形態において示されることがあるが、それらは特定のタイプのメッセージに限定されず、本出願は特定のタイプの信号伝達に限定されない。

【００１７】

大規模演算実験を使用して、機械学習モデル、深層学習システム、予測分析論などを確立することにより、政策レベルの決定を行うことができる。これらの演算は、多くの場合、複数の独立機関の個々の研究および開発の統合から導き出される。しかしながら、これらの機関は、必ずしも互いを信用していない。さらに、各機関は、通常、その機関独自のローカル・バージョンのモデルおよびデータを維持しているため、異なるエージェンシー間でモデルのずれが生じることがある。例示的な実施形態は、異なる機関およびエンティティに関連付けられ得るノード間に信用を確立する、分散型演算環境をサポートするブロックチェーン・ネットワークを提供する、方法、システム、非一過性コンピュータ可読媒体、デバイス、またはネットワーク、あるいはその組合せを提供する。本明細書に記載のシステムはまた、各エンティティが決定のソースを追跡することができるように出所を確立する。システムはまた、システム状況の分散型演算システムなどのマルチエージェント・システムにとって重要なアカウンタビリティおよび透明性を提供する。システムは、正しいものとして合意された一貫性のあるローカル演算を保証することによって、演算の確認を確実なものにし、かつ記録された監査を通じて演算の一貫性を追跡することによって、演算の検証を確実なものにする。

【００１８】

非集中型データベースは、互いに通信する複数のノードを含む分散型ストレージ・システムである。ブロックチェーンは、相互に信用できない当事者間においてレコードを維持することのできる分散型台帳に類似した、追加専用の不変データ構造を含む非集中型データベースの一例である。信用できない当事者は、本明細書において、ピアまたはピア・ノードと呼ばれる。各ピアはデータベース・レコードのコピーを維持しており、分散したピア間でコンセンサスに達しなければ、ピアのうちの誰一人として、データベース・レコードを変更することができない。例えば、ピアは、ブロックチェーン・ストレージ・トランザクションを確認し、ストレージ・トランザクションをブロックにグループ化し、ブロックを介してハッシュ・チェーンを構築するために、コンセンサス・プロトコルを実行することができる。このプロセスは、必要に応じて、一貫性のために、ストレージ・トランザクションを順序付けすることによって台帳を形成する。パブリックな、または許可のないブロックチェーンにおいては、誰でも特定のアイデンティティなしで参加することができる。パブリック・ブロックチェーンは、多くの場合、ネイティブな暗号通貨に関与し、プルーフ・オブ・ワーク（ＰｏＷ）に基づくコンセンサスを使用する。一方、許可型ブロックチェーン・データベースは、資金、物資、情報などを交換するビジネスのような、共通の目的を共有し得るが、互いに完全には信用することのできないエンティティのグループ間における対話を保護することのできるシステムを提供する。

【００１９】

ブロックチェーンは、非集中型ストレージ・スキームに合わせて調整され、「スマート・コントラクト」または「チェーンコード」と呼ばれる任意のプログラム可能なロジックを動作させる。場合により、システム・チェーンコードと呼ばれる専用チェーンコードが、管理機能およびパラメータのために存在し得る。スマート・コントラクトは、ブロックチェーン・データベースの改ざん防止特性と、エンドースメントまたはエンドースメント

・ポリシーと呼ばれるノード間の基礎をなす合意とを活用する、信用できる分散型アプリケーションである。一般に、ブロックチェーン・トランザクションは、通常、ブロックチェーンにコミットされる前に「エンドース」されなければならず、エンドースされないトランザクションは無視される。通常のエンドースメント・ポリシーは、チェーンコードが、エンドースメントに必要とされるピア・ノードのセットの形態のトランザクションについて、エンドーサを規定することを許容する。クライアントが、エンドースメント・ポリシーで規定されたピアにトランザクションを送信すると、トランザクションは、トランザクションを確認するように実行される。確認後、トランザクションは、順序付けフェーズに入り、このフェーズにおいて、ブロックにグループ化されたエンドース済みトランザクションの順序付けされたシーケンスを生成するために、コンセンサス・プロトコルが使用される。

【００２０】

ノードは、ブロックチェーン・システムの通信エンティティである。「ノード」は、異なるタイプの複数のノードが同一の物理的サーバ上で実行可能であるという意味で論理的な機能を実行することができる。ノードは、信用できるドメイン内でグループ化され、それらのノードを様々な方法で制御する論理的なエンティティと関連付けられる。ノードは、トランザクション呼出しをエンドーサ（例えば、ピア）にサブミットし、トランザクション提案を順序付けサービス（例えば、順序付けノード）にブロードキャストするクライアントまたはサブミット・クライアント・ノードなどの、異なるタイプを含むことができる。別のタイプのノードは、クライアントがサブミットしたトランザクションを受信し、トランザクションをコミットし、ブロックチェーン・トランザクションの台帳の状態およびコピーを維持することができるピア・ノードである。ピアは、エンドーサの役割を有することもできるが、これは、必須要件ではない。順序付けサービス・ノードまたはオーダラ（ｏｒｄｅｒｅｒ）は、すべてのノードのための通信サービスを実行するノードであり、トランザクションをコミットするとき、およびブロックチェーンの世界状態（通常は制御情報および設定情報を含む、初期ブロックチェーン・トランザクションの別の名称）を変更するときの、システム内のピア・ノードの各々に対するブロードキャストなどの配信保証を実装する。

【００２１】

台帳は、ブロックチェーンのすべての状態遷移の順序付けられた改ざん防止レコードである。状態遷移は、参加している当事者（例えば、クライアント・ノード、順序付けノード、エンドーサ・ノード、ピア・ノードなど）によってサブミットされたチェーンコード呼出し（すなわち、トランザクション）から生じ得る。トランザクションは、１つまたは複数のオペランド（作成、更新、削除など）として台帳にコミットされているアセットのキーと値（バリュー）のペアのセットをもたらすことができる。台帳は、不変の順序付けされたレコードをブロックに格納するために使用されるブロックチェーン（チェーンとも呼ばれる）を含む。台帳は、ブロックチェーンの現在の状態を維持する状態データベースも含む。通常、１つのチャネルごとに１つの台帳が存在する。各ピア・ノードは、そのピア・ノードがメンバになっているチャネルごとに、台帳のコピーを維持する。

【００２２】

チェーンは、ハッシュリンク・ブロックとして構造化されたトランザクション・ログであり、各ブロックはＮ個のトランザクションのシーケンスを含み、Ｎは１以上である。ブロック・ヘッダは、ブロックのトランザクションのハッシュ、および前のブロックのヘッダのハッシュを含む。このようにして、台帳のすべてのトランザクションが順序付けられ、暗号によって互いにリンクされ得る。したがって、ハッシュリンクを壊さずに台帳データを改ざんすることはできない。直近で追加されたブロックチェーン・ブロックのハッシュは、それ以前に発生したチェーン上のすべてのトランザクションを表し、すべてのピア・ノードが一貫性のある信用できる状態にあることを確実にすることができる。チェーンは、ブロックチェーンの作業負荷の追加専用という性質を効率的にサポートする、ピア・ノード・ファイル・システム（すなわち、ローカル、取り付けられたストレージ、クラウ

ドなど）に格納されてよい。

【００２３】

不変台帳の現在の状態は、チェーンのトランザクション・ログに含まれるすべてのキーの最新の値を表す。現在の状態は、チャネルに知られている最新のキーの値を表すため、世界状態（ｗｏｒｌｄ　ｓｔａｔｅ）と呼ばれることもある。チェーンコード呼出しは、台帳の現在の状態データに対してトランザクションを実行する。これらのチェーンコードの対話を効率的にするために、最新のキーの値が状態データベースに格納されてよい。状態データベースは、単にチェーンのトランザクション・ログへのインデックス付きビューであってよく、したがって、いつでもチェーンから再生成することができる。状態データベースは、ピア・ノードの起動時に、トランザクションが受け取られる前に、自動的に回復され（必要な場合は、生成され）てよい。

【００２４】

本明細書で説明し図示する解決策のいくつかの利点は、反復シミュレーションの確認済み状態を反復のフレームに格納することを含む。すなわち、状態が、クライアントによって反復により演算され、フレームに圧縮され、エンドーサによって確認される。確認時に、これらのフレームはブロックチェーンに順次追加される。計数（ｅｎｕｍｅｒａｔｉｖｅ）実験（所与の入力についての出力の非反復評価）の場合、ブロックチェーンは、入力と出力のペアの確認済みフレームを格納することができる。システムの状態は、反復により進展するとき、反復のフレーム内でブロックチェーンに格納され得る。フレームは、開始時にチェックポイントにより特徴付けられ、更新（差）が定義された許容閾値内にある限り、状態を含むように適応的に定義される。フレームの最大サイズを適切に選択することができ、各フレームは、サイズにより許容されるできるだけ多くの状態を含むことができる。さらに、エンドーサによって実行される確認は、フレームを並列して確認することができるように並列化される。非反復実験も並列化することができるが、反復シミュレーションはクライアントによって順次実行される。

【００２５】

加えて、例示的な実施形態はまた、オーバヘッド削減のためのフレーミングおよび圧縮フレームワークを提供する。各フレームは、シミュレーションが反復であるか非反復に基づいて異なって圧縮されてよい。システムはまた、パラメータにとらわれない（ｐａｒａｍｅｔｅｒ　ａｇｎｏｓｔｉｃ）設計の近似誤差を訂正するように、無効状態（エンドースされていない）の連続的な精製（ｒｅｆｉｎｅｍｅｎｔ）を提供する。非反復シミュレーションの場合、計数（非反復）シミュレーションのための最小全域木（ＭＳＴ）ベースのフレーム構成を、圧縮のために実行してもよい。システムはまた、確認時間短縮のためのコード化された演算ベースのタスク割当て、および確率的確認保証を用いた外的ランダム性によるシミュレーションのための確認を提供する。

【００２６】

ブロックチェーンは、中央ストレージではなく、むしろ、非集中型の不変で安全なストレージであるという点において、従来のデータベースとは異なっており、この場合、ノードは、ストレージ内のレコードに対する変更を共有しなければならない。ブロックチェーンにおいて固有であり、ブロックチェーンの実装を助けるいくつかの特性は、本明細書でさらに説明する、不変台帳、スマート・コントラクト、セキュリティ、プライバシ、非集中型、コンセンサス、エンドースメント、アクセス可能性などを含むが、これらに限定されない。様々な態様によれば、分散型プラットフォームを介した演算プロセスの確認済み状態は、ブロックチェーンに固有かつ特有の不変性、セキュリティ、非集中型、コンセンサス、エンドースメント、アクセス可能性に起因して実装される。

【００２７】

特に、確認済み状態のレコードが不変であるため、演算の一貫性のためのシミュレーションの検証は、ブロックチェーン上のハッシュ・チェーンの一貫性のチェックに相当する。これは、シミュレーション全体を実行するよりもはるかに容易であるため、検証を保証することが簡単になる。演算の発生源を追跡するだけでなく、演算およびその原点（ｏｒ

ｉｇｉｎ）における誤りを識別することが実現可能であるため、アカウンタビリティをブロックチェーンによって保証することができる。これにより、演算システムの出所を確立する。

【００２８】

システムは、演算、エンドースメント、およびストレージにおける対抗者からのセキュリティをさらに提供する。これは、状態が確認されると、ブロックチェーンを参照することによって、その後の任意の時点においてこの状態を信用することができるからである。これにより、シミュレーション結果の使用インスタンスにまたがって信用を引き継ぐことができる。さらに、確認、演算、およびストレージのプロセス全体が非集中型であるため、シミュレーション結果を損う中央攻撃点が存在しない。また、分散型の性質により、プロセスの並列可能性による演算コストおよび時間の削減が可能になる。

【００２９】

少なくとも十分な数のピアが演算を正しいものとして受け入れること、かつ台帳のコピーに格納されたデータがピアにまたがって最新であることが必須であるので、ピア間における確認のコンセンサスが、信用を可能にするために重要である。さらに、ネットワーク内のエンドーサのランダムな割当てが、対抗するピアの影響を冗長性によって打ち消すことを示すので、ネットワーク内の独立したエンドーサによる演算のステップのエンドースメントによって、確認プロセスにおける信用が可能になり、ネットワークの全体的な性質を誠実に機能させることが可能になる。ブロックチェーン台帳は分散され、すべてのピア間で共有されるため、シミュレーションの結果へのアクセス、または反復からの推論の引出しを望むピアは誰でも、台帳のローカル・コピーを参照することによって容易にこれを行うことができる。

【００３０】

例示的な実施形態は、従来のデータベースを超える多くの利点を提供する。例えば、システムの設計は、ネットワーク内のピア間において一貫性のある方法で共有可能な、不変の、かつ分散して更新可能なデータ構造を必要とする。これらはすべて、まさにブロックチェーンに相当する。加えて、ブロックチェーンにより、シミュレーションの有効性の検証に相当する簡単な一貫性チェックも可能になる。別のデータ構造がこれらの特徴を提供することは技術的に実現可能であり得るが、上記の要件すべてを提供する簡単な統合機構は存在しない。この意味で、ブロックチェーンは最適な選択肢であることがわかる。

【００３１】

確認済み状態の新規の圧縮を最初に実行する圧縮スキーマと組み合わせてブロックチェーンを使用して、実装の拡張性を可能にしてもよい。圧縮スキーマが存在しない場合、ストレージおよび通信オーバヘッドが極めて大きくなり、大規模シミュレーションおよび大型ネットワークに合わせた設計のスケーリングを妨げる。さらに、ここで定義された確認プロセスは、暗号通貨およびスマート・コントラクトにおいて適用可能なエンドースメントの概念と比較すると、レポート偏差を検査する機能の再演算に基づくという点で、リソースの利用可能性のチェックのようなことに対して新しい。

【００３２】

また、例示的な実施形態は、データをブロックチェーンのブロック構造内に格納することのできる方法を変化させる。例えば、関連するブロックチェーンは、トランザクション（資産譲渡）、およびスマート・コントラクト（実行すべき命令のセット）などの情報を格納することができる。一方、例示的な実施形態において、格納されているデータは、演算プロセスの確認済み状態である。すなわち、各ブロックは、演算の状態の圧縮されたチェックポイント、およびフレームにおける連続反復のデルタ符号化から得られた、量子化された状態更新を含む、１つまたは複数のフレームから構成される。格納され得る追加のメタデータは、シミュレーションを実行しているクライアントのＩＤ、状態を確認したエンドーサのＩＤ、反復インデックス、およびシミュレーションを実行するために使用される他の外部情報を含むことができる。ブロックチェーンのデータ・ブロック内に演算の確認済み状態を格納することによって、演算の確認済み状態が、ブロックのハッシュリンク

・チェーンを通じて不変台帳に追加され得、その後の演算の反復を確認し、圧縮を実行するために使用され得る。

【００３３】

本明細書に記載の例示的な実施形態によって、様々な技術的利点が提供される。例えば、実施形態は、シミュレーションの状態を、状態を並列に確認するエンドーサのサブセットにレポートすることによって、大規模反復シミュレーションのための分散型確認フレームワークを提供する。加えて、チェックポイントを識別し、シミュレーションの状態のフレームを構成し、確認を並列にディスパッチする前にデルタ符号化および格子ベクトル量子化を実行してフレームを圧縮する、圧縮スキーマが提供される。加えて、オーダラを使用してエンドースメントの一貫性をチェックするシミュレーションによる状態の確認および圧縮済みフレームを格納するために、ブロックチェーン台帳が設けられる。さらに、状態の木を作成して計数実験において圧縮を実行するために、ＭＳＴベースのフレーム構成方法が提供される。さらに、コード化された演算ベースのエンドースメント割当てが、状態確認のための演算時間を短縮するために提供される。

【００３４】

図１は、例示的な実施形態による、シミュレーション・データを確認し格納するためのブロックチェーン・ネットワーク１００を示す。図１を参照すると、ブロックチェーン・ネットワーク１００はクライアント・ノード１１０を含み、クライアント・ノード１１０は、モデルをトレーニングするための大規模演算を実行することができ、または大規模演算データを別の１つまたは複数のシステムから受信することができる。大規模演算は、モデル／データをトレーニングするため、または非反復プロセスを演算するためのシミュレーション・データを生成することができる。シミュレーション・データ内のチェックポイントに基づいて、シミュレーション・データをフレーム、データ・ポイントなどのデータ構造に変換することができ、データ構造を圧縮してオーバヘッドを削減することができる。

【００３５】

チェックポイントを使用して、次のフレームなどを識別することができる。チェックポイントは、反復データ、非反復入力／出力ペアなどに基づいて、クライアント・ノードによって決定されてよい。フレームは、モデルの状態が進展するときに、データの連続フレームを含むことができる。ブロックチェーン・ネットワーク１００はまた、エンドーシング・ノード１２０、１３０、１４０の複数のサブセットを含み、これらのサブセットは相互に排他的であり得、これらのサブセットを使用して、シミュレーション状態データのフレームまたは他のデータ構造を並列に確認することができる。データのフレームが、エンドーシング・ノード・サブセット１２０、１３０または１４０あるいはその組合せによって確認されると、本明細書でブロックのハッシュリンク・チェーンとも呼ばれるブロックチェーン１６０のうちのデータ・ブロック内に含めるために、フレームをオーダラ１５０に送信することができる。

【００３６】

シミュレーションは、複数の演算ノードから利益を得る実験であり得る。非限定的な例として、マラリア・データ科学者（ＭＤＳ）は、１つまたは複数の演算モデル・エージェント（例えば、ＯｐｅｎＭａｌａｒｉａおよび疫学モデリング・ソフトウェア（ＥＭＯＤ））を使用して、病気の拡大をシミュレートするための実験を行うことがある。この例において、モデルは、多くの別個のシミュレーションを実行して、モデル構造またはパラメータの変化の影響を比較する。実験の目的は、病気を根絶するための有効な政策および介入戦略を考え出すことである。本明細書のシステムを使用して、シミュレーションの演算結果において、アカウンタビリティおよび透明性を確立することができる。

【００３７】

再び図１を参照すると、エンドーシング・ノード１２０、１３０、１４０のサブセットによって分散型確認が実行される。エンドースするローカル動作の再演算のコンセンサスは、サブセットおよびレポートされた演算によって生成されてよい。サブセットによって確認されると、データの確認済みフレームは、ブロックチェーン１６０に格納され、クラ

イアント・ノード１１０、エンドーシング・ノード（サブセット１２０～１４０）、他のピア・ノードなどを含み得る、ブロックチェーン・ネットワーク上のすべてのノード間で共有することができる。ブロックチェーン１６０は、連続するチェックポイント間に、確認済みフレームの、共有された不変の追加専用のレコードを作成する。さらに、不可逆圧縮を使用して、各フレーム内の状態データを圧縮することにより、ストレージのコストおよび演算オーバヘッドを削減する。

【００３８】

クライアント・ノード１１０を、演算を実行するエージェントと呼ぶこともできる。クライアント・ノード１１０は、フレームの形態のチェックポイントに、モデル、データなどの状態を格納することができる。フレームは、フレーム内の圧縮済み状態更新（差）と、前のフレーム内の状態データとを含むことができる。フレームは、連続フレームとして生成されたより大きいフレームのグループの一部であってよい。フレームは、再演算により、エンドーシング・ノード・サブセット１２０、１３０、１４０内のエンドーサ（他のエージェント）によって検証することができる。確認済みフレームは、ブロックチェーン・ネットワークのエンドーサ、ピア、クライアントなどの間で共有されるブロックチェーン１６０に追加することができる。

【００３９】

システムの状態は、

【数１】

$$X_{t+1} = f(X_t, \theta_t)\ X_t \in \mathbb{R}^d$$

としてのシミュレーションの反復によって進展し、システム状態ベクトル

【数２】

$$\theta_t \in \mathbb{R}^{d'}$$

は、（場合によりランダムな）情報の共有ソースであり、

【数３】

$$f\colon \mathbb{R}^d \times \mathbb{R}^{d'} \to \mathbb{R}^d$$

すべてのピアによって共有されるアトミック操作

シミュレーションの有効性　⇔　中間状態｛X_1，X_2…｝の有効性

ｆ（・）はＬ、リプシッツ連続であり、

すなわち、

$\| F(x_1) - f(x_2) \| \leqq L \| x_1 - x_2 \|$

であると仮定する。

【００４０】

シミュレーションは、状態$x_{t+1} = f(x_t, \theta_t)$の反復進展であり得る。クライアント・ノード１１０は、チェックポイントｔ∈｛T_1，T_2，…｝における状態X_tをレポートすることができる。チェックポイントｔ∈（T_1，T_2）間の反復について、圧縮された更新

【数４】

$$\widetilde{\Delta X_t}$$

をレポートする。これに応答して、ノードのエンドースメント・サブセットは、クライアント・ノード１１０によってレポートされた、前に確認済みの状態に基づいて、データのフレームを再演算し、それを受信済みフレームと比較することができる。さらに、オーダラ１５０は、確認済みフレームをブロックチェーン１６０に順次追加することができる。

【００４１】

さらに、クライアント・ノード１１０は、不可逆圧縮（映像符号化プロセスに類似）を使用して、小さい∈についての状態更新（差）、

【数５】

$$\left\| \widetilde{\Delta X}_t - \Delta X_t \right\| \leq \epsilon$$

をレポートすることができる。ここで、フレーム内のすべてのスロットについての$\| \Delta X_t \| \leqq \Delta_{quant}$およびフレーム内のいくつかの反復が最大数によって限定され得るように、フレームの幅（チェックポインティング頻度）が適応的に決定されてよい。現在の状態は、Ｌｅｍｐｅｌ－Ｚｉｖのような不可逆圧縮を使用して、フレーム（チェックポイント）の開始時に格納されてよい。圧縮率に基づいて、通信およびストレージのコストが変化する。シミュレーションがカオス・シミュレーションである場合、システムは、より小さいフレームおよび大きい圧縮率を必要とし得るが、実施形態はこれに限定されない。

【００４２】

図１のネットワーク１００において、個々のフレームを、ばらばらのエンドーサ・サブセットによって並列に確認することができるため、エンドースメント並列化が実施される。ここで、すべてのエンドーサが、シミュレーションによって生成された全フレームのうちのごく一部のみを確認することにより、演算オーバヘッドを大幅に削減する。さらに、エンドーサは、大きな通信遅延または不十分な演算能力を有するピアを待つ必要がなく、エンドースによりタスクしたフレームに集中することができる。エンドーシング・サブセットがデータのフレームを確認することができないとき、エンドーシング・サブセットはクライアント・ノード１１０に通知することができる。確信があれば、クライアント・ノード１１０は、再演算なしで連続精製を使用してレポート更新を通信することができる。そうでない場合、クライアント・ノード１１０は、フレームを再演算し、エンドースのためにそのフレームを再びサブセットに送信する。

【００４３】

フレーム・エンドースメントを決定するために、エンドーシング・ピア・ノードが、シミュレーションについての許容可能な偏差マージンΔ_{marg}を決定することができる。したがって、クライアント・ノード１１０によってレポートされたモデル／シミュレーションの状態が、再演算された状態についての許容可能な偏差マージン内にある場合のみ、演算が有効であると宣言することができる。ピアのサブセットは、フレーム内のアトミックな演算をエンドースし、コンセンサスに従ってブロックチェーンに追加する。さらに、マージンを適応的に削減することができ、シミュレーションの終了（または状態の収束）に近い、より厳しい要件を得るために、圧縮スキームを適切に更新することができる。

【００４４】

オーダラ１５０は、エンドーシング・ノード１２０、１３０、１４０の異なるサブセットから異なるフレームについてのエンドースメントを受信することができる。エンドーサ１５０は、ブロックチェーン１６０上に格納するためのデータ・ブロックを生成する、または他の方法で初期化することができる。ここで、エンドーサは、コンセンサスに従って、ブロックチェーン１６０に追加すべきフレームを順序付けすることができる。格納されるフレームは、不変であり、エンドーシング・ノードによってチェックポイントおよびフレームを検証するために使用され得る。また、フレーム内の状態更新は、オーダラ１５０によってサブサンプリングされ、より少ないストレージ・コストのために格納され得る。

【００４５】

一部の実施形態において、演算は、深層学習モデル、機械学習アルゴリズムなどの精製のような反復演算を伴う。別の例として、実験は、アトミックな演算ブロックを使用し、入力ごとに出力を生成する、入力パラメータの大きなセットを含む非反復シミュレーションを伴うことができる。この例において、シミュレーションの状態は、入力と出力のペアであってよく、これは、圧縮を困難にする、フレームへの自然なグループ化をもたらさない。

【００４６】

しかしながら、フレームを、最小全域木（ＭＳＴ）を介して構成し、その後に圧縮してもよい。ここで、効率的なデルタ符号化に必要な状態更新のフレームの構成、および状態更新の圧縮レポートを実行してもよい。この例において、ペアの距離マトリックス

【数６】

$$W = \left[\left\| Z_i - Z_j \right\|\right]_{i,j}$$

ここで、$Z_i = (X_i, Y_i)$、を利用してもよい。クライアント・ノード１１０は、重み付きグラフＧ（［ｎ］，Ｗ）のＭＳＴを生成することができる。閾値エッジはΔ_{quant}により重み付けし、せいぜいＭ個のノードを各々含むように木を刈り込む（ｐｒｕｎｅ）。さらに、各木は、フレームに対応し得、木の根元を用いてチェックポイントを作成すること、および木のエッジに沿って量子化された状態更新を通信することを含み得る。さらに、反復データについて説明したものと同じ圧縮スキームを使用してもよい。

【００４７】

非反復シミュレーション・データの例において、クライアント・ノード１１０は、圧縮済みフレームおよびＭＳＴ構造に関連する状態をエンドーシング・サブセットにレポートすることができる。これに応答して、エンドーシング・ノードは、状態

【数７】

$$\left(\tilde{X}_i, \tilde{Y}_i\right)$$

を解凍することができる。エンドーサは、レポートされた入力

【数８】

$$\hat{Y}_i = f\left(\tilde{X}_i\right)$$

からの出力を再演算し、

【数９】

$$\left\| \hat{Y}_i - \tilde{Y}_i \right\| \leq \Delta_{\mathrm{marg}}$$

である場合に、フレーム内の状態レポートを確認することができる。さらに、エンドーサは、すべての状態が有効である場合にフレームを確認することができる。演算（入力）の数が大きいとき、フレーム構成、ローカル確認、およびエンドースメント並列化は、コストを削減し、演算の有効性を確実なものにする。

【００４８】

シミュレーションは、通常、ランダム性（θ_t）の外部ソースに影響されやすく、すなわち、$X_{t+1}=f(X_t,\theta_t)$である。ランダム性へのアクセスがないと、個々のエンドーサは、レポートされた状態を再演算することができない。可能な解決策は、各反復においてθ_tを格納することであるため、非常に高いストレージ・コストを生じる。別の例として、独立したエンドーサにより再演算された状態の平均からの、レポートされた状態の偏差を検査することによって確認する。ここで、ランダム性のソースが同じである場合、エンドーサは、正常に動作する機能について、予想されるパスに沿って集合的に動作する。

【００４９】

本明細書の説明は、エンドーサが誠実であること、および適時の確認のための演算の均一性を仮定する。しかしながら、エンドーサが誤っている、対抗している、または演算が遅い場合、確認は、各フレームをエンドースするための時間に関して大きな犠牲を払うことがある。例えば、エンドーサがストラグラである場合には、エンドースメント取得が過度に遅れることがある。信用できないピアを使用する分散型信用の場合、状態確認ごとに複数のエンドーサを割り当てることによる、確認の冗長性が必要である。したがって、エンドースメント・ポリシー（少なくともＴ個の確認が必要など）を、システムにおける対抗するピアの数に応じて定義することができる。

【００５０】

一部の実施形態において、フレームにおいてエンドーサにより確認すべき状態の割当てのためにコード化演算を使用することにより、各エンドーサにより実行される演算の数を減少させることができる。そうすることで、システムは、閾値エンドースメント・ポリシーの下で確認のための時間を短縮させるが、より多くのエンドーサを使用する。

【００５１】

本明細書に記載のシステムは、従来のシミュレーション・ベースのシステムについて、いくつかの利点をもたらす。例えば、設計における出所、およびマルチエージェント・システムにおいてローカル演算を追跡する方法を環境が保証するため、アカウンタビリティが確実になる。ブロックチェーンを介した定期監査およびローカル演算確認の形態の透明性により、ノード間に確立される信用によって透明性が形成される。フレーム設計、エンドースメント、および確認は、状態の進展に従って適応されてよい。ここで、確認マージンを変化させることにより、信用の要件を適切に変更することができる。プラットフォームは、かなり一般的な要素を使用し、様々な設計パラメータおよびアルゴリズムのうちの任意の１つを使用して実装されてよい。さらに、設計はシミュレーションの詳細にとらわれず、シミュレーションが再現可能なアトミックな演算に分解可能である限り、設計を実装することができる。システムは、既存の圧縮およびブロックチェーン技術から適切に発展し得るプラットフォームを作成するために、かなり簡単な構築ブロックを必要とする。さらに、システム状態の中間評価を格納することにより、プラットフォームは、確実なデータおよびモデル共有、ならびに共同研究を保証する。

【００５２】

図２Ａは、例示的な実施形態による、ブロックチェーン・アーキテクチャ構成２００を示す。図２Ａを参照すると、ブロックチェーン・アーキテクチャ２００は、特定のブロックチェーン要素、例えば、ブロックチェーン・ノード２０２のグループを含むことができる。ブロックチェーン・ノード２０２は、１つまたは複数のノード２０４～２１０を含むことができる（単に例として、４つのノードが示されている）。これらのノードは、ブロックチェーン・トランザクションの追加（例えば、シミュレーション・データ・フレームなど）および確認プロセス（コンセンサス）などのいくつかの活動に参加する。ブロックチェーン・ノード２０４～２１０のうちの１つまたは複数は、エンドースメント・ポリシーに基づいてトランザクションをエンドースすることができ、アーキテクチャ２００内のすべてのブロックチェーン・ノードに順序付けサービスを提供することができる。ブロックチェーン・ノードは、ブロックチェーン認証を開始し、ブロックチェーン層２１６に格納されたブロックチェーンの不変台帳に書き込もうとすることができ、この書込みのコピ

ーが、基盤になる物理的インフラストラクチャ２１４にも格納されてよい。ブロックチェーン構成は、格納されたプログラム／アプリケーション・コード２２０（例えば、チェーンコード、スマート・コントラクトなど）にアクセスして実行するためにアプリケーション・プログラミング・インターフェース（ＡＰＩ）２２２にリンクされた１つまたは複数のアプリケーション２２４を含むことができる。プログラム／アプリケーション・コード２２０は、参加者によって要求されてカスタマイズされた構成に従って作成することができ、それら自身の状態を維持し、それら自身のアセットを制御し、外部情報を受信することができる。このブロックチェーン構成は、トランザクションとしてデプロイし、分散型台帳に追加することによって、すべてのブロックチェーン・ノード２０４～２１０にインストールすることができる。

【００５３】

ブロックチェーン・ベースまたはプラットフォーム２１２は、ブロックチェーン・データの様々な層と、サービス（例えば、暗号信用サービス、仮想実行環境など）と、新しいトランザクションを受信して格納し、データ・エントリにアクセスしようとしている監査人にアクセスを提供するために使用され得る、基盤になる物理的コンピュータ・インフラストラクチャとを含むことができる。ブロックチェーン層２１６は、プログラム・コードを処理し、物理的インフラストラクチャ２１４に関与するために必要な仮想実行環境へのアクセスを提供するインターフェースを公開することができる。暗号信用サービス２１８は、アセット交換トランザクションなどのトランザクションを検証し、情報をプライベートに保つために使用することができる。

【００５４】

図２Ａのブロックチェーン・アーキテクチャ構成は、ブロックチェーン・プラットフォーム２１２によって公開された１つまたは複数のインターフェースおよび提供されたサービスを介して、プログラム／アプリケーション・コード２２０を処理および実行することができる。コード２２０は、ブロックチェーンのアセットを制御することができる。例えば、コード２２０は、データを格納および転送することができ、スマート・コントラクトおよび条件を含む関連するチェーンコードまたは実行の対象になるその他のコード要素の形態で、ノード２０４～２１０によって実行することができる。非限定的な例として、リマインダ、更新、または、変更、更新などの対象になるその他の通知、あるいはその組合せを実行するために、スマート・コントラクトを作成してもよい。スマート・コントラクト自体は、権限付与およびアクセスの要件ならびに台帳の使用に関連付けられたルールを識別するために使用することができる。例えば、読取りセット２６６は、ブロックチェーン層２１６に含まれる１つまたは複数の処理エンティティ（例えば、仮想マシン）によって処理されてよい。書込みセット２２８は、キー・バリューに対する変更を含み得る。物理的インフラストラクチャ２１４は、本明細書に記載のデータまたは情報のいずれかを取り出すために利用されてよい。

【００５５】

チェーンコード内で、高水準のアプリケーションおよびプログラミング言語を使用して、スマート・コントラクトが作成され、その後、ブロックチェーン内のブロックに書き込まれ得る。スマート・コントラクトは、ブロックチェーン（例えば、ブロックチェーン・ピアの分散型ネットワーク）への登録、格納、または複製、あるいはその組合せが実行される実行可能コードを含むことができる。トランザクションは、スマート・コントラクトが満たされていることに関連付けられた条件に応答して実行され得る、スマート・コントラクト・コードの実行である。スマート・コントラクトの実行は、デジタル・ブロックチェーン台帳の状態に対する信用できる変更をトリガすることができる。スマート・コントラクトの実行によって引き起こされるブロックチェーン台帳に対する変更は、１つまたは複数のコンセンサス・プロトコルを通じて、ブロックチェーン・ピアの分散型ネットワーク全体に自動的に複製されてよい。

【００５６】

スマート・コントラクトは、データをキーと値のペアの形式でブロックチェーンに書き

込むことができる。さらに、スマート・コントラクト・コードは、ブロックチェーンに格納された値を読み取り、それらをアプリケーションの動作において使用することができる。スマート・コントラクト・コードは、様々な論理演算の出力をブロックチェーンに書き込むことができる。このコードを使用して、仮想マシンまたはその他の演算プラットフォーム内の一時データ構造を作成することができる。ブロックチェーンに書き込まれたデータは、パブリックになること、またはプライベートとして暗号化されて維持されること、あるいはその両方が行われ得る。スマート・コントラクトによって使用／生成される一時データは、提供された実行環境によってメモリ内に保持され、ブロックチェーンに必要なデータが識別された後に削除される。

【００５７】

チェーンコードは、追加の特徴と共に、スマート・コントラクトのコード解釈を含むことができる。本明細書に記載されているように、チェーンコードは、演算ネットワーク上にデプロイされるプログラム・コードであってよく、コンセンサス・プロセス中に、チェーン・バリデータによって一緒に実行され確認される。チェーンコードは、ハッシュを受信し、予め格納された特徴エクストラクタを使用して作成されたデータ・テンプレートに関連付けられたハッシュをブロックチェーンから取り出す。ハッシュ識別子のハッシュと、格納された識別子テンプレート・データから作成されたハッシュとが一致する場合には、チェーンコードは、要求されたサービスに権限付与キーを送信する。チェーンコードは、暗号の詳細に関連付けられたデータをブロックチェーンに書き込むことができる。

【００５８】

図２Ｂは、例示的な実施形態による、ブロックチェーン・ネットワークのノード間の通信シーケンス２５０の例を示す。図２Ｂの例において、クライアント・ノード２６０は演算を実行してシミュレーション・データを生成し、これをフレームに格納する。さらに、シミュレーション・テータのフレームは、複数のエンドーシング・ピア・ノード２７１、２７２、２７３間で分散される。フレームがエンドーシング・ピア・ノード２７１、２７２、２７３によって確認されると、フレームは、順序付けノード２７４に転送され、順序付けノード２７４は、エンドース済みフレームを１つまたは複数のブロックに順序付けし、ノードにまたがって複製された分散型台帳に格納するために、順序付けされたブロックをブロックチェーン・ネットワークのノードに送信する。

【００５９】

図２Ｂを参照すると、２８１において、クライアント・ノード２６０は、第１のフレームＮを、エンドーシング・ピア・ノードの第１のサブセットに含まれるエンドーシング・ピア・ノード２７１に送信する。同様に、２８２において、クライアント・ノード２６０は、第２のフレームＮ＋１を、エンドーシング・ピア・ノードの第２のサブセットに含まれるエンドーシング・ピア・ノード２７２に送信する。さらに、２８３において、クライアント・ノード２６０は、第３のフレームＮ＋２をエンドーシング・ピア・ノード２７３に送信する。フレームＮ、Ｎ＋１、Ｎ＋２などは、シミュレーション・データをクライアント・ノード２６０に提供するクライアント・ノード２６０または別のシステムによって生成されてよい。各フレームは、フレームＮ、Ｎ＋１、Ｎ＋２のうちの各フレームによって表される各ステップまたはステージにおいてシミュレートされているモデル／データの状態を含むことができる。これに応答して、クライアント・ノード２６０は、シミュレーション・データ内のチェックポイントを識別し、データを圧縮し、識別されたチェックポイントに基づいてデータ・フレームＮ、Ｎ＋１、Ｎ＋２などを生成することができる。

【００６０】

２８４において、エンドーシング・ピア・ノード２７１は、フレームＮを確認しようとする。同様に、２８５において、エンドーシング・ピア・ノード２７２は、フレームＮ＋１を確認しようとし、２８６において、エンドーシング・ピア・ノード２７３は、フレームＮ＋２を確認しようとする。ここで、２８４、２８５、および２８６における確認は並列に実行されてよい。エンドースメント・プロセスを実行するために、各エンドーシング・ピア・ノード２７１～２７３は、前に確認されたシミュレーションの状態を識別し、シ

ミュレーションを再演算してモデルのローカル状態を生成し、ローカルで演算された状態を、クライアント・ノード２６０から受信したそれぞれのフレームに含まれる状態と比較することができる。演算された状態が受信済み状態との偏差の許容範囲内にあることに応答して、エンドーシング・ピア・ノードは、フレームが有効であると判定し、フレームをエンドースする。

【００６１】

この例において、エンドーシング・ピア・ノード２７２およびエンドーシング・ピア・ノード２７３は、フレームＮ＋１およびＮ＋２がそれぞれ有効であると判定する。したがって、２８９および２９０において、エンドーシング・ピア・ノード２７２、２７３は、エンドースメントを順序付けノード２７４に送信する。一方、エンドーシング・ピア・ノード２７１は、フレームＮに対応するチェックポイントにおけるモデルの状態のローカル演算に基づいて、フレームＮが有効でないと判定する。ここで、エンドーシング・ピア・ノード２７１は、ローカルで演算された状態と、フレームＮに含まれる受信済み状態との差が、偏差の許容範囲を超えて異なっていることを判定する。したがって、２８８において、エンドーシング・ピア・ノード２７１は状態レポートを却下し、クライアント・ノード２６０に通知を送信する。これに応答して、クライアント・ノード２６０は、場合により、再演算なしでシミュレーションを精製することができ、または、シミュレーションの状態を再演算することができる。２８９において、クライアント・ノード２６０は、更新／精製された状態に基づいて、フレームＮをエンドーシング・ピア・ノード２７１に再サブミットする。これに応答して、２９１において、エンドーシング・ピア・ノード２７１は、２８９において受信された、更新済み状態を確認しようとし、フレームＮを確認することを決定する。したがって、２９２において、エンドーシング・ピア・ノード２７１は、フレームＮのエンドースメントを順序付けノード２７４に送信する。

【００６２】

すべての連続フレームＮ、Ｎ＋１、Ｎ＋２などのエンドースメントを受信したことに応答して、順序付けノード２７４は、オリジナル・フレームのタイムスタンプに基づいてフレームを順序付けする。例えば、順序付けノード２７４は、フレームをキューなどに格納することができる。さらに、フレームＮ、Ｎ＋１、Ｎ＋２を含む順序付けされたフレーム・データは、２９３においてデータ・ブロックに格納され、２９４においてエンドーシング・ノード２７１、２７２、２７３に送信され、２９５において、順序付けされたフレーム・データを含むデータ・ブロックが分散型台帳にコミットされ得るようにする。ここで、データ・ブロックは、エンドーシング・ピア・ノード２７１～２７３（ならびに、場合により、クライアント・ノード２６０、順序付けノード２７４、および他の図示しないノードなどのその他のノード）間で分散され共有される分散型台帳（例えば、ブロックチェーン、世界状態ＤＢなど）の複製コピーにコミットされてよい。

【００６３】

図３は、分散型の非集中型ピアツーピア・アーキテクチャ、ならびにユーザの役割および許可を管理する認証局３１８を特徴とする、許可型ブロックチェーン・ネットワーク３００の例を示す。この例において、ブロックチェーン・ユーザ３０２は、トランザクションを許可型ブロックチェーン・ネットワーク３１０にサブミットすることができる。この例において、トランザクションは、デプロイ、呼出し、または問合せであってよく、ＳＤＫを利用するクライアント側のアプリケーションを介して、ＲＥＳＴ　ＡＰＩなどを介して直接的に発行されてよい。信用できるビジネス・ネットワークは、監査人（例えば、米国株式市場における証券取引委員会）などの規制者システム３１４にアクセスを提供することができる。一方、ノード３０８のブロックチェーン・ネットワーク運用者システムは、規制者システム３１４を「監査人」として登録し、ブロックチェーン・ユーザ３０２を「クライアント」として登録するなど、メンバの許可を管理する。監査人を、台帳への問合せのみに制限することができ、一方、特定のタイプのチェーンコードのデプロイ、呼出し、および問合せを行うための権限をクライアントに与えることができる。

【００６４】

ブロックチェーン開発者システム３１６は、チェーンコードおよびクライアント側のアプリケーションを書き込む。ブロックチェーン開発者システム３１６は、ＲＥＳＴインターフェースを介して、チェーンコードをネットワークに直接的にデプロイすることができる。従来のデータ・ソース３３０からの認証情報をチェーンコードに含めるために、開発者システム３１６は、帯域外接続を使用してデータにアクセスすることができる。この例において、ブロックチェーン・ユーザ３０２は、ピア・ノード３１２を介してネットワークに接続する。ピア・ノード３１２は、いずれかのトランザクションを開始する前に、ユーザの登録およびトランザクション証明書を認証局３１８から取り出す。場合により、ブロックチェーン・ユーザは、許可型ブロックチェーン・ネットワーク３１０上でトランザクションを実行するために、それらのデジタル証明書を有していなければならない。一方、チェーンコードを動作させようとしているユーザは、従来のデータ・ソース３３０上のそれらの認証情報を検証することが必要になることがある。ユーザの権限付与を確定するために、チェーンコードは、従来の処理プラットフォーム３２０を介して、このデータへの帯域外接続を使用することができる。

【００６５】

図４Ａは、例示的な実施形態による、クライアント・ノード４１０がシミュレーションの状態のフレーム４１２を生成するプロセス４００Ａを示す。図４Ａを参照すると、クライアント・ノード４１０は、複数の連続フレーム４１２を生成する、深層学習モデル、大規模非反復実験などを実行することができる。各フレーム４１２は、シミュレーション・データ内でクライアント・ノード４１０によって識別されたチェックポイント４１３に基づいて生成され得る。さらに、フレーム４１２は、圧縮されて、圧縮済みデータ・フレーム４１４を生成することができる。

【００６６】

様々な実施形態によれば、シミュレーションは、より簡単で確認が容易なステップ／反復に分解することができ、これは、ネットワーク内のピアにまたがる分散型確認の実行に不可欠であり得る。特に、分解によって、複数のフレームを独立したエンドーサにまたがって並列に確認することが可能になり、確認時間要件の短縮が可能になる。分解によって、確認をシミュレーション・プロセスと共に実行することが可能になるため、演算の確認が同時に実行されるだけでなく、信用保証または信用点が作成され、これらの信用保証または信用点は、後に続くフレームのエンドースメントのためにシミュレーションが参照され得る、後の時点に引き継ぐことができる。

【００６７】

モデルの状態データのステップのシーケンスは、一緒にフレームにグループ化され、ブロックチェーンに追加されてよい。しかしながら、これは必ずしも、フレームごとに別個のブロックを追加するように実装を限定するものではない。フレームが順次追加される（反復である）限り、マークル木構造などの考え方を使用して、一般性を失うことなく、単一ブロック内の複数のフレームをグループすることができる。映像の圧縮と同様に、シミュレーション・データのフレームは、デルタ符号化、連続精製、またはベクトル量子化、あるいはその組合せを使用して圧縮されて、通信を実行するためのストレージおよびオーバヘッドに必要なデータ量を減少させることができる。シミュレーションまたは演算は、反復であっても非反復であってもよい。例えば、反復の場合は、フレームを構成し、デルタ符号化を実行する簡単な方法に相当する。一方、非反復シミュレーションの場合には、最小全域木ベースのポリシーを使用して、近接性（ｃｌｏｓｅｎｅｓｓ）に従って状態をグループすることができるため、効率的な圧縮が可能になる。

【００６８】

様々な実施形態によれば、チェックポイント４１３は、複数の異なるシナリオの下で生成または決定されてよい。例えば、フレームが予め設定された最大数の状態を含む場合には、各エンドーサが、各フレーム内の、せいぜい一定数の状態を確認することを確実にするように、新しいフレームが作成されてよい（最初の状態がチェックポイントとなる）。別の例として、選択された最大値を超える大きさだけ連続状態が逸脱しているときに、チ

ェックポイントが作成されてよい。すなわち、状態がかなりの程度異なっているときに、チェックポイントが作成される。

【００６９】

圧縮済みフレーム４１４の生成後、個々の圧縮済みフレーム４１４は、確認のためにエンドーシング・ピア・ノードの複数のサブセット間で分散されてよい。ここで、エンドーシング・ピア・ノードのサブセットは、少なくとも１つのエンドーシング・ピア・ノードを含むことができる。エンドースメントを複数のサブセットにまたがって分散させることによって、エンドースメント・プロセス（解凍、再演算、偏差の決定などを含む）を並列に実行することができるため、処理時間が大幅に短縮される。さらに、各エンドーシング・ノードは、すべてのフレームを演算／確認する必要がない。むしろ、エンドーシング・ノードは、ブロックチェーンから信用できる確認データを取り出しながら、フレームの一部のみをエンドースすることができる。

【００７０】

図４Ｂは、例示的な実施形態による、エンドーシング・ノード４２０がシミュレーション・データの圧縮済みフレーム４２２を確認するプロセス４００Ｂを示す。この例において、圧縮済みデータのフレームは、シミュレーションの状態情報を含むことができる。状態は、演算の反復を継続する必要がある中間値である。疫学的シミュレーションの場合、例えば、状態は、病気の拡大を説明する値のセットを含んでもよい。深層ニューラル・ネットワークのトレーニングの場合、例えば、状態は、ニューラル・ネットワークのノードに適用されるすべての重みのセットなどであってもよい。一部の実施形態において、トレーニング・データが、すべてのピアに予め送信されていてもよい。

【００７１】

エンドースメント・プロセス４２４中、各エンドーシング・ピア・ノードは、フレーム４２２を解凍して、クライアントから提供されたシミュレーションの状態データを明らかにすることができる。エンドーシング・ピア・ノード４２０は、前のシミュレーションの状態をブロックチェーンから取り出すことができる。ここで、前の状態とは、順調に確認され、かつエンドーシング・ピア・ノード４２０にローカルで格納され得るブロックチェーンにコミットされた、最新の状態であってよい。エンドーシング・ピア・ノード４２０は、圧縮済みフレーム４２２内の受信済み状態に対応するシミュレーションの現在の状態を再演算し、ローカルで生成された状態が圧縮済みフレーム４２２内の状態に十分に類似しているかどうかを判定して、エンドースメントのためにフレーム４２２を確認することができる。状態の値が所定の閾値を超えて異なっている場合、エンドーシング・ピア・ノード４２０は、エンドースメントが拒否されていることを示す通知をクライアント・ノードに送信することができる。これに応答して、クライアント・ノードは、フレーム４２２の状態を精製し、精製に基づいて状態を更新した後にフレーム４２２を再サブミットすることができる。このプロセスは、フレーム４２２が確認されるまで繰り返されてよい。一方、確認されると、確認済みフレーム４２６は、ブロックチェーンに含めるために、順序付けノードにサブミットされてよい。

【００７２】

図４Ｃは、例示的な実施形態による、順序付けノード４３０がシミュレーション・データのフレームをブロック内に配置するプロセス４００Ｃを示す。図４Ｃを参照すると、複数のフレームが、エンドーシング・ピアの複数のサブセットによって確認され、順序付けノード４３０に提供される。これに応答して、順序付けノード４３０は、フレームがクライアントによって生成されたときに追加されたフレーム内のタイムスタンプまたは他の情報に基づいて、キュー４３２内にフレームを順次（または、１、２、３などの連続する順番で）配置することができる。さらに、配置されたフレームは、１つまたは複数のデータ・ブロック４３４内で順序付けされてよく、データ・ブロック４３４は、ブロックチェーン・ネットワークのノードに送信され、かつノード間で共有される分散型台帳にコミットされてよい。

【００７３】

図５Ａは、例示的な実施形態による、ブロックチェーンに格納するためのシミュレーション・データのフレームを生成する方法５１０を示す図である。例えば、方法５１０は、クライアント・ノードなどのブロックチェーンのピア・ノードによって実行することができる。別の例として、方法は、プロセッサおよびストレージを有する、サーバ、データベース、クラウド・プラットフォーム、複数のデバイスなどの演算デバイスによって実行することができる。図５Ａを参照すると、５１１において、方法は、シミュレーションのデータを取得することを含むことができる。例えば、反復演算または非反復演算などのシミュレーションが実行され得、シミュレーション・データを生成することができる。一部の実施形態において、シミュレーションは、シミュレーション・データをピア・ノードなどに送信する別のデバイスによって実行されてよい。

【００７４】

方法は、５１２において、シミュレーション・データ内のチェックポイントを識別することを含むことができ、５１３において、方法は、識別されたチェックポイントに基づいて、シミュレーション・データの複数の連続フレームを生成することを含むことができ、各フレームは、連続フレームのうちの前のフレームに対してシミュレーションの進展状態を識別する。例えば、チェックポイントは、シミュレーションの反復、非反復シミュレーションの入力／出力ペアなどに基づいて識別されてよい。例えば、シミュレーション・データのフレームが予め設定された最大数の状態を含む場合には、各エンドーサが、各フレーム内の、せいぜい一定数の状態を確認することを確実にするように、新しいフレームが作成される（最初の状態がチェックポイントとなる）。別の例として、選択された最大値を超える大きさだけ連続状態が逸脱しているときに、チェックポイントが作成されてよい。すなわち、状態が、かなりの程度異なっているときに、チェックポイントが作成される。チェックポインティングは、演算データ内で前のフレームの終了および次の連続フレームの開始を識別するために使用されてよい。一部の実施形態において、フレームは、演算のタイプ（例えば、反復、非反復など）に基づいてさらに圧縮されてよい。

【００７５】

一部の実施形態において、各フレームの幅が、受信したシミュレーションのデータ内のそれぞれのフレームに対応するチェックポイントの頻度に基づいて、適応的に決定されてよい。一部の実施形態において、シミュレーションは、モデルがトレーニングされる反復シミュレーションの１つを含むことができ、各反復は、１つまたは複数のアルゴリズム、ニューラル・ネットワークなどに基づいて、モデルの状態をさらに精製する。別の例として、シミュレーションは、入力と出力のペアのセットが処理されることを含む、非反復シミュレーションを含むことができる。

【００７６】

方法は、５１４において、データ・ブロックのハッシュリンク・チェーン内の１つまたは複数のデータ・ブロック内に含めるために、生成された連続フレームをブロックチェーン・ネットワークのノードに送信することを含むことができる。例えば、各フレームは、ブロックチェーン・ネットワーク内の異なるエンドーシング・ピア・グループまたはノードのサブセットに送信されてよい。一部の実施形態において、方法は、生成された連続フレームのうちの各フレーム内に反復ＩＤを格納することをさらに含むことができ、反復ＩＤは、それぞれのフレームに関連付けられた反復シミュレーションのそれぞれの反復を識別する。一部の実施形態において、方法は、生成された連続フレームのうちのフレームに関連付けられた状態が無効になったことを示すメッセージを、ブロックチェーン・ピア・ノードから受信することをさらに含むことができる。方法は、これに応答して、状態を再演算すること、無効のフレームの状態を精製して、更新済みフレームを生成すること、および更新済みフレームを確認のためにブロックチェーン・ピア・ノードに送信することのうちの１つまたは複数をさらに含むことができる。

【００７７】

図５Ｂは、例示的な実施形態による、シミュレーション・データのフレームをエンドースする方法５２０を示す図である。例えば、方法５２０は、ブロックチェーン・ネットワ

ークの１つまたは複数のエンドーシング・ノードによって実行することができる。ノードは、サーバ、データベース、クラウド・プラットフォームなどの演算システムを含むことができる。図５Ｂを参照すると、５２１において、方法は、反復シミュレーションの連続データ・フレームの中からデータ・フレームを受信することを含むことができる。ここで、データ・フレームは、モデル（アルゴリズムなど）、トレーニング・データなどの現在の状態の識別を含む、反復シミュレーションの状態データを含むことができる。一部の実施形態において、データ・フレームは、圧縮スキームを使用して圧縮されてよい。一部の実施形態において、データ・フレームを解凍して、圧縮前のシミュレーションの状態を明らかにしてもよい。

【００７８】

方法は、５２２において、前のフレームから前に確認された状態データに基づいて、データ・フレームの状態データのローカル演算を生成することを含むことができる。ここで、前に確認された状態データは、ブロックチェーンのブロック内の前のフレームに格納されていてよい。方法は、５２３において、状態データのローカル演算と受信済みデータ・フレーム内の状態データとの類似性を判定することを含むことができる。方法は、５２４において、類似性が所定の閾値内にあるという判定に応答して、データ・ブロックのハッシュリンク・チェーンのうちのデータ・ブロック内に含めるために、データ・フレームのエンドースメントをブロックチェーン・ネットワークに送信することを含むことができる。

【００７９】

一部の実施形態において、反復シミュレーションは、ニューラル・ネットワークなどを介してモデルがトレーニングされる深層学習シミュレーションを含むことができる。さらに、連続フレームは、反復によるモデルの状態の変化を格納する。一部の実施形態において、所定の閾値は、状態データのローカル演算と受信済みデータ・フレーム内の状態データとの偏差の許容レベルを識別することができる。一部の実施形態において、方法は、類似性が所定の閾値を超えるという判定に応答して、状態データが無効であることを示すメッセージを、データ・フレームをサブミットしたクライアント・ノードに送信することをさらに含むことができる。一部の実施形態において、方法は、精製された状態データを含む更新済みデータ・フレームを受信することと、状態データの位置演算と更新済みデータ・フレーム内の精製された状態データとの類似性を判定することとをさらに含むことができる。

【００８０】

図５Ｃは、例示的な実施形態による、並列エンドースメントのためにシミュレーション・データの連続フレームを割り当てる方法５３０を示す。例えば、方法５３０は、ブロックチェーン・ネットワークのクライアント・ノードによって実行することができる。ノードは、サーバ、データベース、クラウド・プラットフォームなどの演算システムを含むことができる。図５Ｃを参照すると、５３１において、方法は、既定のチェックポイントに基づいて反復シミュレーションの複数の連続データ点を生成することを含むことができる。例えば、各データ点は、連続データ点のうちの前のデータ点に対する反復シミュレーションの進展状態の識別を格納する、データのフレームまたはウィンドウを含むことができる。ここで、シミュレーションは、ニューラル・ネットワークなどを介して機械学習アルゴリズムがトレーニングされるような深層学習シミュレーションであってよい。データ点は、圧縮されて演算オーバヘッドを削減することができる。

【００８１】

方法は、５３２において、複数の連続データ点のうちの第１のデータ点内の状態データを確認するためのブロックチェーン要求を、ブロックチェーン・ネットワークのエンドーシング・ノードの第１のサブセットに送信することを含むことができ、方法は、５３２において、複数の連続データ点のうちの第２のデータ点内の状態データを確認するためのブロックチェーン要求を、エンドーシング・ノードの第２のサブセットに送信することをさらに含むことができ、第２のサブセットは、第１のデータ点および第２のデータ点の並列エンドースメントのために、ブロックチェーン・ネットワークのエンドーシング・ノード

の第１のサブセットとは相互に排他的である。クライアント・ノードは、フレームをエンドースするためのエンドーシング・ピア・ノードのサブセットのみを使用し、かつ異なるフレームをエンドースするためのエンドーシング・ノードの異なるサブセットを使用することによって、エンドーシング・ピア・ノードの演算を減少させることができる。

【００８２】

一部の実施形態において、第１のデータ点の状態データは、反復シミュレーションの第１の反復に基づいて生成され、第２のデータ点の状態データは、反復シミュレーションの次の反復に基づいて生成される。一部の実施形態において、第２のデータ点は、第１のデータ点の反復シミュレーションの状態と第２のデータ点における反復シミュレーションの状態との差を格納する。一部の実施形態において、方法は、第２のデータ点の状態データが無効であることを示すメッセージを、エンドーシング・ノードの第２のサブセットから受信することをさらに含む。

【００８３】

一部の実施形態において、方法は、無効である第２のデータ点の状態データを精製して、第２のデータ点の更新済み状態データを生成することと、更新済み状態データを確認のためにエンドーシング・ノードの第２のサブセットに送信することとをさらに含むことができる。一部の実施形態において、方法は、連続データ点のうちの各データ点内に反復ＩＤを格納することをさらに含むことができ、反復ＩＤは、それぞれのデータ点に関連付けられた反復シミュレーションのそれぞれの反復を識別する。一部の実施形態において、方法は、反復シミュレーションを実行して複数の連続データ点を生成することをさらに含むことができる。

【００８４】

図５Ｄは、例示的な実施形態による、並列エンドースメント処理の結果としてのシミュレーション・データを順序付けする方法５４０を示す。例えば、方法５４０は、ブロックチェーン・ネットワークの順序付けノードによって実行することができる。ノードは、サーバ、データベース、クラウド・プラットフォームなどの演算システムを含むことができる。図５Ｄを参照すると、方法は、５４１において、ピア・ノードの第１のサブセットを介して、反復シミュレーションによって生成された複数の連続データ点のうちの第１のデータ点の状態データの検証を受信することを含むことができ、５４２において、ピア・ノードの第１のサブセットと相互に排他的である、エンドーシング・ピア・ノードの第２のサブセットを介して、反復シミュレーションの複数の連続データ点のうちの第２のデータ点の状態データの検証を受信することを含むことができる。ここで、第１のデータ点および第２のデータ点は、順序付けノードを含む同じブロックチェーン・ネットワークのエンドーシング・ピア・ノードの異なるサブセットによって、並列に検証され、または他の方法でエンドースされてよい。

【００８５】

方法は、５４３において、確認済み状態データを含む第１のデータ点および第２のデータ点を含む１つまたは複数のデータ・ブロックを生成することを含むことができ、方法は、５４４において、データ・ブロックのハッシュリンク・チェーンに格納するために、１つまたは複数のデータ・ブロックをブロックチェーン・ネットワーク内のピア・ノードに送信することを含むことができる。例えば、順序付けノードは、各ピア・ノードによって保持されたブロックチェーンのローカル・レプリカに格納するために、データ・ブロックをブロックチェーン・ネットワーク内のピア・ノードに送信することができる。

【００８６】

一部の実施形態において、方法は、第１のデータ点および第２のデータ点に含まれるタイムスタンプに基づいて、第１のデータ点および第２のデータ点をキュー内に配置することをさらに含むことができる。この例において、１つまたは複数のデータ・ブロックを生成することが、キュー内における第１のデータ点および第２のデータ点の位置に基づいて、１つまたは複数のデータ・ブロック内の第１のデータ点および第２のデータ点を順序付けすることを含むことができる。一部の実施形態において、各検証は、それぞれのデータ

点の状態データが、既定の閾値からの偏差の許容範囲内にあることを示すことができる。一部の実施形態において、第１のデータ点および第２のデータ点は各々、それぞれのデータ点に関連付けられた反復シミュレーションのそれぞれの反復を識別する反復ＩＤを含むことができる。

【００８７】

図５Ｅは、例示的な実施形態による、シミュレーション・コンテンツを圧縮する方法５５０を示す図である。例えば、方法５５０は、演算データを生成または受信あるいはその両方を行うブロックチェーン・ネットワークのクライアント・ノードによって実行することができる。ノードは、サーバ、データベース、クラウド・プラットフォーム、複数のシステムなどの演算システムを含むことができる。図５Ｅを参照すると、方法は、５５１において、シミュレーション・コンテンツを格納するデータ・フレームを生成することを含むことができる。ここで、データ・フレームは、演算モデルの連続して生成された状態データ、および深層学習モデル、大規模シミュレーションなどのデータを含む複数のデータ・フレームのうちの１つであってよい。

【００８８】

方法は、５５２において、別のデータ・フレームに格納された前のシミュレーション・コンテンツに基づいて、データ・フレーム内のシミュレーション・コンテンツを圧縮して、圧縮済みデータ・フレームを生成することを含むことができる。例えば、シミュレーション・コンテンツは、反復シミュレーションによる状態データであっても、非反復シミュレーションによる状態データであってもよい。シミュレーションが、複数の入力／出力ペアを有する非反復シミュレーションである場合、圧縮することは、入力／出力ペアの近接性に基づく最小全域木（ＭＳＴ）プロセスに基づいて、非反復シミュレーション・コンテンツを圧縮することを含むことができる。別の例として、シミュレーションが、反復して進展する状態を有する反復シミュレーションである場合、圧縮することは、反復シミュレーションの前の反復との状態の差に基づいて、反復シミュレーション・コンテンツを圧縮することを含むことができる。前の状態は、ノードによって識別され、状態データ全体を含むのではなく、状態データのデルタを２つのフレーム間に挿入するために使用されてよい。

【００８９】

方法は、５５３において、圧縮済みデータ・フレームをブロックチェーン・ネットワークのブロックのハッシュリンク・チェーン内に含めるために、ブロックチェーン要求を介して、圧縮済みデータ・フレームをブロックチェーン・ネットワークの１つまたは複数のエンドーシング・ピア・ノードに送信することを含むことができる。例えば、ブロックチェーン要求は、ブロックチェーンに格納するためのトランザクションをエンドースする要求を含むことができる。一部の実施形態において、データ・フレームは、シミュレーション・コンテンツ内の１つまたは複数の既定のチェックポイントに基づく適応サイズを含むことができる。一部の実施形態において、圧縮済みデータ・フレームは、シミュレーション・コンテンツの前の圧縮済みデータ・フレームに対する状態更新を含むことができる。

【００９０】

図５Ｆは、例示的な実施形態による、圧縮済みシミュレーション・コンテンツをエンドースする方法５６０を示す図である。例えば、方法５６０は、ブロックチェーン・ネットワーク内のエンドーシング・ピア・ノードまたはエンドーシング・ピア・ノードのサブセットによって実行することができる。図５Ｆを参照すると、方法は、５６１において、前のデータ・フレームに格納された前のシミュレーション・コンテンツに基づいて圧縮されたシミュレーション・コンテンツを含むデータ・フレームを受信することを含むことができる。一部の実施形態において、圧縮済みコンテンツは解凍されてよい。方法は、５６２において、データ・ブロックのハッシュリンク・チェーンから、前のデータ・フレームに格納された前のシミュレーション・コンテンツを抽出することを含むことができる。前のシミュレーション・コンテンツは、前に確認されたシミュレーションの状態データをブロックに格納するブロックチェーンから抽出されてよい。

【００９１】
方法は、５６３において、抽出された前のシミュレーション・コンテンツに基づいて、ローカル・シミュレーション・コンテンツを生成することを含むことができる。ここで、ローカル・シミュレーション・コンテンツは、ノードにより格納された、前に確認されたシミュレーションの状態を用いてシミュレーションの状態の値を再演算して、状態の推定値を決定することによって、生成されてよい。方法は、５６４において、受信した圧縮済みシミュレーション・コンテンツとローカル・シミュレーション・コンテンツとの差に基づいて、ブロックのハッシュリンク・チェーンに含めるために受信済みデータ・フレームをエンドースするかどうかを決定することを含むことができる。決定することは、シミュレーション・コンテンツの受信済み状態データとローカルで生成されたシミュレーション・コンテンツの状態データとの類似性に基づいていてよい。状態データが既定の閾値偏差内にある場合、エンドースメントは権限付与されてよい。方法は、５６５において、受信済みデータ・フレームをエンドースすると決定したことに応答して、エンドースメントをブロックチェーン・ネットワーク内のピア・ノードに送信することを含むことができる。
【００９２】
一部の実施形態において、受信した圧縮済みシミュレーション・コンテンツは、データ・フレームに格納されてよく、複数の入力／出力ペアを有する非反復シミュレーションによるコンテンツを含むことができる。この例において、非反復シミュレーション・コンテンツは、入力／出力ペアの近接性に基づくＭＳＴプロセスに基づいて圧縮されてよい。一部の実施形態において、受信した圧縮済みシミュレーション・コンテンツは、データ・フレームに格納されてよく、反復して進展する状態を有する反復シミュレーションによるコンテンツを含むことができる。一部の実施形態において、反復シミュレーション・コンテンツは、反復シミュレーションの前の反復との状態の差に基づいて圧縮されてよい。
【００９３】
図６Ａは、例示的な実施形態による、例示的な動作方法のうちの１つまたは複数に従ってブロックチェーン上で様々な動作を実行するように構成された例示的な物理的インフラストラクチャを示す。図６Ａを参照すると、例示的な構成６００Ａは、ブロックチェーン６２０およびスマート・コントラクト６３０を有する物理的インフラストラクチャ６１０を含み、例示的な実施形態のいずれかに含まれる動作ステップ６１２のうちのいずれかを実行することができる。ステップ／動作６１２は、１つまたは複数のフロー図または論理図あるいはその両方に記載または図示されるステップのうちの１つまたは複数を含むことができる。ステップは、コンピュータ・システム構成の物理的インフラストラクチャ６１０上にある１つまたは複数のスマート・コントラクト６３０またはブロックチェーン６２０あるいはその両方に書き込まれた、またはこれらから読み出された、出力されたもしくは書き込まれた情報を表すことができる。データは、実行されたスマート・コントラクト６３０またはブロックチェーン６２０あるいはその両方から出力されてよい。物理的インフラストラクチャ６１０は、１つまたは複数のコンピュータ、サーバ、プロセッサ、メモリ、または無線通信デバイス、あるいはその組合せを含むことができる。
【００９４】
図６Ｂは、例示的な実施形態による、ブロックチェーン上でスマート・コントラクトの条項を執行するように構成された、契約当事者と仲介サーバとの間の例示的なスマート・コントラクト構成を示す。図６Ｂを参照すると、構成６５０Ｂは、通信セッション、資産譲渡セッション、あるいは１つまたは複数のユーザ・デバイス６５２もしくは６５６またはその両方を明確に識別するスマート・コントラクト６３０によって動かされるプロセスまたは手続きを表すことができる。スマート・コントラクト実行に関する実行、動作、および結果は、サーバ６５４によって管理されてよい。スマート・コントラクト６３０のコンテンツは、スマート・コントラクト・トランザクションの当事者であるエンティティ６５２、６５６の１つまたは複数によるデジタル署名を要求することができる。
【００９５】
図６Ｃは、例示的な実施形態による、ブロックチェーン上でスマート・コントラクトの

条項を執行するように構成された、契約当事者と仲介サーバとの間の例示的なスマート・コントラクト構成を示す。図６Ｃを参照すると、構成６５０は、通信セッション、資産譲渡セッション、あるいは１つまたは複数のユーザ・デバイス６５２もしくは６５６またはその両方を明確に識別するスマート・コントラクト６３０によって動かされるプロセスまたは手続きを表すことができる。スマート・コントラクト実行に関する実行、動作、および結果は、サーバ６５４によって管理されてよい。スマート・コントラクト６３０のコンテンツは、スマート・コントラクト・トランザクションの当事者であるエンティティ６５２、６５６の１つまたは複数によるデジタル署名を要求することができる。スマート・コントラクト実行の結果は、ブロックチェーン・トランザクションとしてブロックチェーン６２０に書き込まれてよい。この例において、スマート・コントラクト６３０はブロックチェーン６２０上にあり、ブロックチェーン６２０は１つまたは複数のコンピュータ、サーバ、プロセッサ、メモリ、または無線通信デバイス、あるいはその組合せ上にあってよい。

【００９６】

図６Ｄは、例示的な実施形態による、ブロックチェーンのロジックおよびデータにアクセスするための共通インターフェースを示す。図６Ｄの例を参照すると、アプリケーション・プログラミング・インターフェース（ＡＰＩ）ゲートウェイ６６２が、ブロックチェーン・ロジック（例えば、スマート・コントラクト６３０または他のチェーンコード）およびデータ（例えば、分散型台帳など）にアクセスするための共通インターフェースを提供する。この例において、ＡＰＩゲートウェイ６６２は、１つまたは複数のエンティティ６５２、６５６をブロックチェーン・ピア（すなわち、サーバ６５４）に接続することによって、ブロックチェーン上でトランザクション（呼出し、問合せなど）を実行するための共通インターフェースである。サーバ６５４は、ブロックチェーン・ネットワーク・ピア・コンポーネントであり、世界状態のコピーおよび分散型台帳を保持し、クライアント６５２、６５６が世界状態上のデータを問い合わせると共に、トランザクションをブロックチェーン・ネットワークにサブミットすることを可能にする。ブロックチェーン・ネットワークでは、スマート・コントラクト６３０およびエンドースメント・ポリシーに応じて、エンドーシング・ピアがスマート・コントラクト６３０を実行することになる。

【００９７】

図７Ａは、例示的な実施形態による、新しいブロック７３０を分散型台帳７２０に追加するプロセス７００を示す。図７Ｂは、例示的な実施形態による、ブロックチェーンのためのブロック構造７３０のコンテンツを示す。図７Ａを参照すると、クライアント（図示せず）は、トランザクションをブロックチェーン・ノード７１１、７１２または７１３あるいはその組合せにサブミットすることができる。クライアントは、いずれかのソースから、ブロックチェーンに対して活動するように命令されることがある。例として、クライアントは、ブロックチェーンに対するトランザクションを提案するデバイス、人、またはエンティティなどの要求者の代理として動作する、（ＳＤＫに基づく）アプリケーションであってよい。複数のブロックチェーン・ピア（例えば、ブロックチェーン・ノード７１１、７１２、７１３）は、ブロックチェーン・ネットワークの状態および分散型台帳７２０のコピーを維持することができる。

【００９８】

異なるタイプのブロックチェーン・ノード／ピアが、ブロックチェーン・ネットワークに存在してもよい。ブロックチェーン・ネットワークは、クライアントにより提案されたトランザクションをシミュレートおよびエンドースするエンドーシング・ピアと、エンドースメントの検証、トランザクションの確認、および分散型台帳７２０へのトランザクションのコミットを行うコミッティング・ピアとを含む。この例において、ブロックチェーン・ノード７１１、７１２、７１３は、エンドーサ・ノード、コミッタ・ノード、またはその両方の役割を果たすことができる。

【００９９】

分散型台帳７２０は、不変の順序付けされたレコードをブロックに格納するブロックチ

ェーン７２２と、ブロックチェーン７２２の現在の状態（キー値）を維持する状態データベース７２４（現在の世界状態）とを含む。１つのチャネルごとに１つの分散型台帳７２０が存在し得、各ピアは、そのピアがメンバになっているチャネルごとに、分散型台帳７２０のそれ自身のコピーを維持する。ブロックチェーン７２２は、ハッシュリンク・ブロックとして構造化されたトランザクション・ログであり、各ブロックはＮ個のトランザクションのシーケンスを含む。ブロック（例えば、ブロック７３０）は、図７Ｂに示すような様々なコンポーネントを含むことができる。ブロックのリンク（図７Ａに矢印で示す）は、現在のブロックのブロック・ヘッダ内に前のブロックのヘッダのハッシュを追加することによって生成されてよい。このようにして、ブロックチェーン７２２上のすべてのトランザクションが順序付けられ、暗号によって互いにリンクされるため、ハッシュリンクを壊さずにブロックチェーン・データを改ざんすることはできない。さらに、リンクにより、ブロックチェーン７２２内の最新のブロックは、それ以前に発生したすべてのトランザクションを表す。ブロックチェーン７２２は、追加専用のブロックチェーンの作業負荷をサポートするピア・ファイル・システム（ローカルまたは取り付けられたストレージ）に格納されてよい。

【０１００】

ブロックチェーン７２２および分散型台帳７２０の現在の状態は、状態データベース７２４に格納されてよい。ここで、現在の状態データは、ブロックチェーン７２２のチェーン・トランザクション・ログにおいて過去に含まれているすべてのキーの最新の値を表す。チェーンコードの呼出しは、状態データベース７２４内の現在の状態に対してトランザクションを実行する。これらのチェーンコード対話を効率的なものにするために、すべてのキーの最新の値が状態データベース７２４に格納されてよい。状態データベース７２４は、ブロックチェーン７２２のトランザクション・ログへのインデックス付けされたビューを含むことができ、したがって、任意の時点においてチェーンから再生成され得る。状態データベース７２４は、トランザクションが受け付けられる前に、ピアのスタートアップ時に自動的に回復されてよい（または必要に応じて生成されてよい）。

【０１０１】

エンドーシング・ノードは、クライアントからトランザクションを受信し、シミュレートされた結果に基づいてトランザクション（例えば、シミュレーション状態の変化など）をエンドースする。エンドーシング・ノードは、トランザクション提案をシミュレートするスマート・コントラクトを保持することができる。トランザクションをエンドースするために必要なノードは、チェーンコード内で規定され得るエンドースメント・ポリシーに依存する。エンドースメント・ポリシーの一例は、「エンドーシング・ピアの大多数がトランザクションをエンドースしなければならない」というものである。異なるチャネルは、異なるエンドースメント・ポリシーを有することができる。エンドース済みトランザクションは、クライアント・アプリケーションにより順序付けサービス７１０に転送される。

【０１０２】

順序付けサービス７１０は、エンドース済みトランザクションを受け付け、これらをブロックに順序付けし、ブロックをコミッティング・ピアに配信する。例えば、順序付けサービス７１０は、トランザクションの閾値に達したとき、タイマがタイムアウトしたとき、または別の状態のときに、新しいブロックを開始することができる。順序付けサービス７１０は、ブロックチェーン・ノード７１１～７１３などからのタイムスタンプの重み付け平均に基づいて、トランザクションごとに最終タイムスタンプを計算するなど、本明細書に記載のタイムスタンプ合意プロセスに基づいて動作することができる。図７Ａの例では、ブロックチェーン・ノード７１２は、ブロックチェーン７２２に格納するための新しいデータ・ブロック７３０を受信したコミッティング・ピアである。

【０１０３】

順序付けサービス７１０は、オーダラのクラスタから構成されてよい。順序付けサービス７１０は、トランザクション、スマート・コントラクトを処理せず、または共有型台帳を維持しない。むしろ、順序付けサービス７１０は、エンドース済みトランザクションを

受け付け、トランザクションの最終タイムスタンプを決定し、最終タイムスタンプに基づいて、それらのトランザクションが分散型台帳７２０にコミットされる順序を規定する。ブロックチェーン・ネットワークのアーキテクチャは、「順序付け」の特定の実装（例えば、Ｓｏｌｏ、Ｋａｆｋａ、ＢＦＴ、など）がプラグ接続可能なコンポーネントとなるように設計されてよい。

【０１０４】

トランザクションは、一貫性のある順序で分散型台帳７２０に書き込まれる。トランザクションがネットワークにコミットされたときに、状態データベース７２４に対する更新が有効であることを確実にするように、トランザクションの順序が確立される。順序付けが暗号パズルの解決、すなわちマイニングによって発生する暗号通貨ブロックチェーン・システム（例えば、ビットコインなど）とは異なり、この例においては、分散型台帳７２０の当事者は、時系列の順序付けなどのネットワークに最良に適した順序付け機構を選択することができる。

【０１０５】

順序付けサービス７１０が新しいブロック７３０を初期化すると、新しいブロック７３０は、コミッティング・ピア（例えば、ブロックチェーン・ノード７１１、７１２、７１３）にブロードキャストされてよい。これに応答して、各コミッティング・ピアは、読取りセットおよび書込みセットが状態データベース７２４内の現在の世界状態に依然として一致していることを確かめるようにチェックすることによって、新しいブロック７３０内のトランザクションを確認する。詳細には、コミッティング・ピアは、エンドーサがトランザクションをシミュレートしたときに存在した読取りデータが状態データベース７２４内の現在の世界状態と同一であるかどうかを判定することができる。コミッティング・ピアがトランザクションを確認すると、トランザクションは分散型台帳７２０上のブロックチェーン７２２に書き込まれ、状態データベース７２４は、読取り－書込みセットからの書込みデータにより更新される。トランザクションが失敗した場合、すなわち、コミッティング・ピアが、読取り－書込みセットが状態データベース７２４内の現在の世界状態に一致しないことを見出した場合、ブロックに順序付けされたトランザクションは、依然としてそのブロックに含まれることになるが、無効であるとしてマークされ、状態データベース７２４は更新されない。

【０１０６】

図７Ｂを参照すると、分散型台帳７２０のブロックチェーン７２２に格納されているブロック７３０（データ・ブロックとも呼ばれる）は、ブロック・ヘッダ７３２、ブロック・データ７３４、およびブロック・メタデータ７３６などの複数のデータ・セグメントを含むことができる。図７Ｂに示すブロック７３０およびそのコンテンツなどの、図示される様々なブロックおよびそのコンテンツは、例示を目的としたものに過ぎず、例示的な実施形態の範囲を限定することを意図したものではないことを理解されたい。場合により、ブロック・ヘッダ７３２およびブロック・メタデータ７３６の両方が、トランザクション・データを格納するブロック・データ７３４より小さくてもよいが、これは必須要件ではない。ブロック７３０は、ブロック・データ７３４内にＮ個のトランザクション（例えば、１００個、５００個、１０００個、２０００個、３０００個など）のトランザクション情報を格納することができる。様々な実施形態によれば、各トランザクションは、エンドーシング・ピア・ノードによって確認されたシミュレーション（例えば、反復、非反復など）の状態の変化を含む、フレーム・データ７３５を含むことができる。

【０１０７】

メタデータ７３６は、データ・フレームを生成したクライアント・ノード／ワーカの識別、フレームをエンドースするエンドーシング・ノード、状態データが引き出されるシミュレーションの１つまたは複数の反復（例えば、反復インデックスなど）、フレームを生成するために実行されたシミュレーションのインスタンスに対する環境ＩＤポインティングなどを含むことができる。メタデータは、すべてのトランザクションについての有効／無効インジケータに加えて、ブロック・メタデータ・セクション７３６に格納されてよい

。格納されたメタデータは、データ・ブロックに格納された情報のインジケータを再確認するのに有用である。

【０１０８】

従来のブロックチェーンは、トランザクション（資産譲渡）、およびスマート・コントラクト（実行される命令のセット）などの情報を格納する。本実施形態において、格納されているデータは、演算プロセスの確認済み状態を含むこともできる。すなわち、各ブロックは、圧縮済みチェックポイントと、フレームにおける連続反復のデルタ符号化から得られた、量子化された状態更新とを含むフレームを含むことができる。格納される追加のメタデータは、シミュレーションを実行するクライアントのＩＤ、状態を確認したエンドーサのＩＤ、反復インデックス、およびシミュレーションを実行するために使用される他の外部情報を含むことができる。

【０１０９】

ブロック７３０は、ブロック・ヘッダ７３２内の（例えば、図７Ａのブロックチェーン７２２上の）前のブロックへのリンクを含むこともできる。特に、ブロック・ヘッダ７３２は、前のブロックのヘッダのハッシュを含むことができる。ブロック・ヘッダ７３２は、固有のブロック番号、現在のブロック７３０のブロック・データ７３４のハッシュなどを含むこともできる。ブロック７３０のブロック番号は、固有であってよく、ゼロから始まる増分的／順次的な順序に割り当てることができる。ブロックチェーン内の最初のブロックは、ブロックチェーン、そのメンバ、その内部に格納されているデータなどについての情報を含む、ジェネシス・ブロックと呼ばれることがある。

【０１１０】

ブロック・データ７３４は、ブロック７３０内に記録される各トランザクションのトランザクション情報を格納することができる。例えば、ブロック・データ７３４内に格納されたトランザクション・データは、トランザクションのタイプ、バージョン、タイムスタンプ（例えば、最後に計算されたタイムスタンプなど）、分散型台帳７２０のチャネルＩＤ、トランザクションＩＤ、エポック、ペイロードの可視性、チェーンコード・パス（デプロイ・トランザクション）、チェーンコード名、チェーンコード・バージョン、入力（チェーンコードおよび機能）、パブリック・キーおよび証明書などのクライアント（クリエータ）・アイデンティティ、クライアントの署名、エンドーサのアイデンティティ、エンドーサの署名、提案ハッシュ、チェーンコード・イベント、応答ステータス、名前空間、読取りセット（トランザクションによって読み取られたキーおよびバージョンのリストなど）、書込みセット（キーおよび値のリストなど）、スタート・キー、エンド・キー、キーのリスト、マークル木クエリ・サマリなどのうちの１つまたは複数を含むことができる。トランザクション・データは、Ｎ個のトランザクションの各々に格納されてよい。

【０１１１】

ブロック・メタデータ７３６は、複数のメタデータのフィールドを（例えば、バイト・アレイなどとして）格納することができる。メタデータ・フィールドは、ブロック生成に対する署名、最後の構成ブロックに対する参照、ブロック内の有効および無効トランザクションを識別するトランザクション・フィルタ、ブロックを順序付けした順序付けサービスの存続する最後のオフセットなどを含むことができる。署名、最後の構成ブロック、およびオーダラ・メタデータは、順序付けサービス７１０によって追加されてよい。一方、（ブロックチェーン・ノード７１２などの）ブロックのコミッティング・ノードは、エンドースメント・ポリシー、読取り／書込みセットの検証などに基づいて、有効性／無効性情報を追加することができる。トランザクション・フィルタは、ブロック・データ７３４内のトランザクションの数に等しいサイズのバイト・アレイと、トランザクションが有効／無効であったかどうかを識別する確認コードとを含むことができる。

【０１１２】

上記の実施形態は、ハードウェアにおいて、プロセッサによって実行されるコンピュータ・プログラムにおいて、ファームウェアにおいて、または、上記のものの組合せにおいて実装することができる。コンピュータ・プログラムは、ストレージ媒体などの、コンピ

ュータ可読媒体上で具体化することができる。例えば、コンピュータ・プログラムは、ランダム・アクセス・メモリ（ＲＡＭ）、フラッシュメモリ、読出し専用メモリ（ＲＯＭ）、消去可能なプログラム可能な読出し専用メモリ（ＥＰＲＯＭ）、電気的に消去可能なプログラム可能な読出し専用メモリ（ＥＥＰＲＯＭ）、レジスタ、ハード・ディスク、リムーバブル・ディスク、コンパクト・ディスク読出し専用メモリ（ＣＤ－ＲＯＭ）、または当技術分野において既知のストレージ媒体の任意の他の形態内に存在することができる。

【０１１３】

例示的なストレージ媒体は、プロセッサがストレージ媒体との間において情報を読取りおよび書込みし得るように、プロセッサに結合することができる。代替案において、ストレージ媒体は、プロセッサと一体であってもよい。プロセッサおよびストレージ媒体は、特定用途向け集積回路（ＡＳＩＣ）内に存在することができる。代替案においては、プロセッサおよびストレージ媒体は、別個のコンポーネントとして存在することができる。例えば、図８は、例示的なコンピュータ・システム・アーキテクチャ８００を示し、これは、上記のコンポーネントなどのうちのいずれかを表していても、またはその内部に統合されていてもよい。

【０１１４】

図８は、本明細書に記載されているアプリケーションの実施形態の使用または機能性の範囲について、何らかの限定を示唆することを意図したものではない。そうではなく、演算ノード８００は、実装されること、または上記の機能性のうちのいずれかを実行すること、あるいはその両方が可能である。例えば、演算ノード８００は、図５Ａ～図５Ｆに関して図示し説明した方法５１０～５６０のいずれかを実行することができる。

【０１１５】

演算ノード８００内には、多数のその他の汎用目的または特殊目的演算システム環境または構成と共に動作可能なコンピュータ・システム／サーバ８０２が存在する。コンピュータ・システム／サーバ８０２と共に使用するのに適切であり得る周知の演算システム、環境、または構成、あるいはその組合せの例は、パーソナル・コンピュータ・システム、サーバ・コンピュータ・システム、シン・クライアント、シック・クライアント、ハンドヘルドまたはラップトップ・デバイス、マルチプロセッサ・システム、マイクロプロセッサ・ベースのシステム、セット・トップ・ボックス、プログラム可能な消費者電子装置、ネットワークＰＣ、ミニコンピュータ・システム、メイン・フレーム・コンピュータ・システム、および上記のシステムまたはデバイスのうちのいずれかを含む分散型クラウド演算環境などを含むが、これらに限定されない。

【０１１６】

コンピュータ・システム／サーバ８０２は、コンピュータ・システムによって実行されるプログラム・モジュールなどの、コンピュータ・システム実行可能命令の一般的な文脈において説明することができる。一般に、プログラム・モジュールは、特定のタスクを実行する、または特定の抽象データ型を実装する、ルーチン、プログラム、オブジェクト、コンポーネント、ロジック、データ構造などを含むことができる。コンピュータ・システム／サーバ８０２は、分散型クラウド演算環境内において実施されてもよく、この場合、タスクは、通信ネットワークを通じてリンクされたリモート処理デバイスによって実行される。分散型クラウド演算環境において、プログラム・モジュールは、メモリ・ストレージ・デバイスを含む、ローカルおよびリモート・コンピュータ・システム・ストレージ媒体の両方に配置されてよい。

【０１１７】

図８に示すように、クラウド演算ノード８００内のコンピュータ・システム／サーバ８０２は、汎用演算装置の形態で示されている。コンピュータ・システム／サーバ８０２のコンポーネントは、１つまたは複数のプロセッサまたは処理ユニット８０４、システム・メモリ８０６、およびシステム・メモリ８０６を含む様々なシステム・コンポーネントをプロセッサ８０４に結合するバスを含むことができるが、これらに限定されない。

【０１１８】

バスは、メモリ・バスまたはメモリ・コントローラ、周辺バス、アクセラレーテッド・グラフィクス・ポート、およびプロセッサ、あるいは様々なバス・アーキテクチャのうちのいずれかを使用するローカル・バスを含む、いくつかのタイプのバス構造のいずれかのうちの１つまたは複数を表す。例として、限定されることなく、このようなアーキテクチャは、ＩＳＡ（Ｉｎｄｕｓｔｒｙ Ｓｔａｎｄａｒｄ Ａｒｃｈｉｔｅｃｔｕｒｅ）バス、ＭＣＡ（Ｍｉｃｒｏ Ｃｈａｎｎｅｌ Ａｒｃｈｉｔｅｃｔｕｒｅ）バス、ＥＩＳＡ（Ｅｎｈａｎｃｅｄ ＩＳＡ）バス、ＶＥＳＡ（Ｖｉｄｅｏ Ｅｌｅｃｔｒｏｎｉｃｓ Ｓｔａｎｄａｒｄｓ Ａｓｓｏｃｉａｔｉｏｎ）ローカル・バス、およびＰＣＩ（Ｐｅｒｉｐｈｅｒａｌ Ｃｏｍｐｏｎｅｎｔ Ｉｎｔｅｒｃｏｎｎｅｃｔｓ）バスを含む。

【０１１９】

コンピュータ・システム／サーバ８０２は、通常、様々なコンピュータ・システム可読媒体を含む。このような媒体は、コンピュータ・システム／サーバ８０２によってアクセス可能な任意の利用可能な媒体であってよく、揮発性および不揮発性の媒体、リムーバブルおよび非リムーバブルの媒体の両方を含む。システム・メモリ８０６は、一実施形態において、その他の図のフロー図を実施する。システム・メモリ８０６は、ランダム・アクセス・メモリ（ＲＡＭ）８１０またはキャッシュ・メモリ８１２あるいはその両方などの、揮発性メモリの形態のコンピュータ・システム可読媒体を含むことができる。コンピュータ・システム／サーバ８０２は、その他のリムーバブルおよび非リムーバブルの、揮発性／不揮発性のコンピュータ・システム・ストレージ媒体をさらに含むことができる。例としてのみ、ストレージ・システム８１４は、非リムーバブルの不揮発性の磁気媒体（図示されておらず、通常「ハード・ドライブ」と呼ばれる）との間における読取りおよび書込みのために提供されてよい。図示されていないが、リムーバブルの不揮発性磁気ディスク（例えば、「フロッピー（Ｒ）・ディスク」）との間における読取りおよび書込みのための磁気ディスク・ドライブ、ならびにＣＤ－ＲＯＭ、ＤＶＤ－ＲＯＭ、またはその他の光媒体などの、リムーバブルの不揮発性光ディスクとの間における読取りまた書込み用の光ディスク・ドライブを提供することができる。このような例において、１つまたは複数のデータ媒体インターフェースにより、それぞれをバスに接続することができる。以下でさらに図示し説明するように、メモリ８０６は、本出願の様々な実施形態の機能を実行するように構成されたプログラム・モジュールのセット（例えば、少なくとも１つ）を有する少なくとも１つのプログラム製品を含むことができる。

【０１２０】

プログラム・モジュール８１８のセット（少なくとも１つ）を有する、プログラム／ユーティリティ８１６、ならびに、限定されないが、例として、オペレーティング・システム、１つまたは複数のアプリケーション・プログラム、その他のプログラム・モジュール、およびプログラム・データを、メモリ８０６に格納することができる。オペレーティング・システム、１つまたは複数のアプリケーション・プログラム、その他のプログラム・モジュール、およびプログラム・データ、またはそれらの何らかの組合せの各々は、ネットワーク接続環境の実装を含むことができる。プログラム・モジュール８１８は、一般に、本明細書に記載される、本出願の様々な実施形態の機能または方法あるいはその両方を実行する。

【０１２１】

当業者によって理解されるように、本出願の態様は、システム、方法、またはコンピュータ・プログラム製品として具体化することができる。したがって、本出願の態様は、全体的にハードウェアの実施形態、全体的にソフトウェアの実施形態（ファームウェア、常駐ソフトウェア、マイクロコードなどを含む）、または、本明細書において、一般的に、すべてが「回路」、「モジュール」、または「システム」と呼ばれることのある、ソフトウェアおよびハードウェア態様を組み合わせた実施形態の形をとることができる。さらに、本出願の態様は、コンピュータ可読プログラム・コードがその上で具体化されている１つまた複数のコンピュータ可読媒体において具体化されたコンピュータ・プログラム製品の形をとることもできる。

【０１２２】
コンピュータ・システム／サーバ８０２はまた、キーボード、ポインティング・デバイス、ディスプレイ８２２などの１つまたは複数の外部デバイス８２０、ユーザがコンピュータ・システム／サーバ８０２と対話することを可能にする１つまたは複数のデバイス、またはコンピュータ・システム／サーバ８０２が１つまたは複数の他の演算デバイスと通信することを可能にする任意のデバイス（例えば、ネットワーク・カード、モデムなど）、あるいはその組合せと通信することができる。このような通信は、Ｉ／Ｏインターフェース８２４を介して行うことができる。さらに、コンピュータ・システム／サーバ８０２は、ネットワーク・アダプタ８２６を介して、ローカル・エリア・ネットワーク（ＬＡＮ）、一般的な広域ネットワーク（ＷＡＮ）、またはパブリック・ネットワーク（例えば、インターネット）、あるいはその組合せなどの１つまたは複数のネットワークと通信することができる。図示されているように、ネットワーク・アダプタ８２６は、バスを介してコンピュータ・システム／サーバ８０２のその他のコンポーネントと通信する。図示されていないが、その他のハードウェアまたはソフトウェア・コンポーネントあるいはその両方をコンピュータ・システム／サーバ８０２と共に使用してもよいことを理解されたい。例として、マイクロコード、デバイス・ドライバ、冗長処理ユニット、外部ディスク・ドライブ・アレイ、ＲＡＩＤシステム、テープ・ドライブ、およびデータ・アーカイブ・ストレージ・システムなどを含むが、これらに限定されない。
【０１２３】
システム、方法、および非一過性コンピュータ可読媒体のうちの少なくとも１つの例示的な実施形態が、添付の図面に示され、上記の詳細な説明において記述されているが、本出願は、開示された実施形態に限定されるものではなく、以下の特許請求の範囲によって記述および定義される多数の再構成、変更、および置換が可能であることを理解されたい。例えば、様々な図のシステムの能力は、本明細書に記載のモジュールもしくはコンポーネントのうちの１つまたは複数によって、または分散型アーキテクチャにおいて実行することができ、トランスミッタ、レシーバ、またはこれら両方のペアを含むことができる。例えば、個々のモジュールによって実行される機能性のすべてまたは一部は、これらのモジュールのうちの１つまたは複数によって実行することができる。さらに、本明細書に記載の機能性は、様々な時点において、かつモジュールまたはコンポーネントの内部または外部の様々なイベントとの関係において、実行することもできる。また、様々なモジュール間で送信される情報は、データ・ネットワーク、インターネット、音声ネットワーク、インターネット・プロトコル・ネットワーク、無線デバイス、有線デバイスのうちの少なくとも１つを介して、または複数のプロトコルを介して、あるいはその組合せを介して、モジュール間で送信することができる。また、モジュールのうちのいずれかによって送受信されるメッセージは、直接的に、またはその他のモジュールのうちの１つまたは複数を介して、あるいはその両方によって、送受信することもできる。
【０１２４】
当業者は、「システム」が、パーソナル・コンピュータ、サーバ、コンソール、パーソナル・デジタル・アシスタント（ＰＤＡ）、携帯電話、タブレット演算デバイス、スマートフォン、または任意の他の適切な演算デバイス、あるいはデバイスの組合せとして具体化され得ることを理解するだろう。上記の機能を「システム」によって実行されるものとして提示することは、本出願の範囲を何らかの方法によって限定することを意図したものではなく、多数の実施形態の一例を提示することを意図したものである。実際に、本明細書に開示される方法、システム、および装置は、演算技術と一貫性を有する、局所化され分散された形態で実装することができる。
【０１２５】
本明細書に記載されたシステム特徴のいくつかは、その実装の独立性をより具体的に強調するために、モジュールとして提示されていることに留意されたい。例えば、モジュールは、カスタム超大規模集積（ＶＬＳＩ）回路またはゲート・アレイ、論理チップ、トランジスタなどの市販の半導体、あるいはその他の個別のコンポーネントを含むハードウェ

ア回路として実装されてよい。モジュールは、フィールド・プログラマブル・ゲート・アレイ、プログラマブル・アレイ・ロジック、プログラマブル・ロジック・デバイス、グラフィクス処理ユニットなどのプログラム可能なハードウェア・デバイスにおいて実装されていてもよい。

【０１２６】

モジュールは、様々なタイプのプロセッサによって実行するために、ソフトウェアに少なくとも部分的に実装されていてもよい。例えば、実行可能コードの識別されたユニットは、例えばオブジェクト、手順、または関数として編成され得る、コンピュータ命令の１つまたは複数の物理的または論理的ブロックを含むことができる。それにもかかわらず、識別されたモジュールの実行可能コードは、物理的に一緒に配置する必要はなく、異なる位置に格納された異種の命令を含むことができ、これらの命令は、論理的に一緒に結合されたときに、モジュールを含み、モジュールについて記述されている目的を実現する。さらに、モジュールは、例えば、ハード・ディスク・ドライブ、フラッシュ・デバイス、ランダム・アクセス・メモリ（ＲＡＭ）、テープ、またはデータを格納するために使用される任意の他のそのような媒体であり得る、コンピュータ可読媒体上に格納されてもよい。

【０１２７】

実際に、実行可能コードのモジュールは、単一の命令または多数の命令であり得、いくつかの異なるコード・セグメントにわたって、異なるプログラム間において、およびいくつかのメモリ・デバイスにまたがって、分散されていてもよい。同様に、動作データが、本明細書においてモジュール内で識別および図示されてよく、任意の適切な形態で具体化され、任意の適切なタイプのデータ構造内で編成されてよい。動作データは、単一のデータセットとして収集されてもよく、または異なるストレージ・デバイスを含む、異なる位置にわたって分散されていてもよく、少なくとも部分的に、システムまたはネットワーク上の単なる電子信号として存在していてもよい。

【０１２８】

本明細書の図において概略的に説明され図示されているように、本出願の構成要素は、様々な異なる構成で配置および設計され得ることが容易に理解されよう。したがって、実施形態の詳細な説明は、特許請求される本出願の範囲を限定することを意図したものではなく、本出願の選択された実施形態を表すものに過ぎない。

【０１２９】

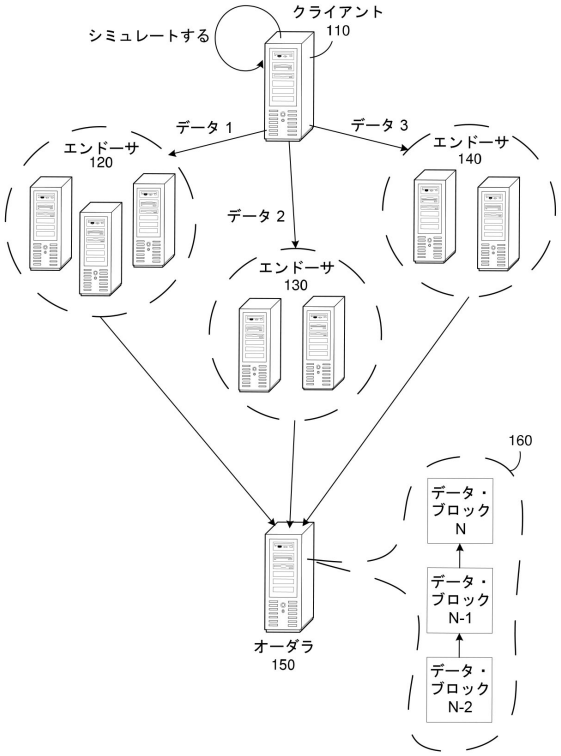
当業者は、上記の内容が、異なる順序におけるステップにより、または開示されているものとは異なる構成におけるハードウェア要素により、あるいはその両方により、実施され得ることを容易に理解するだろう。したがって、本出願は、これらの好ましい実施形態に基づいて記載されているが、特定の変更、変形、および代替構成が明白であることが当業者には明らかであろう。

【０１３０】

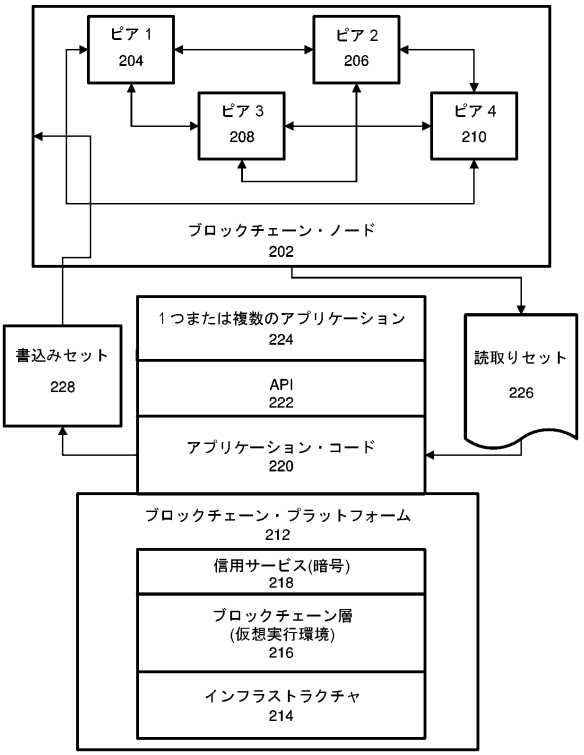
本出願の好適な実施形態について説明したが、説明した実施形態は、例示的なものに過ぎず、本出願の範囲は、添付の特許請求の範囲の等価物およびこれに対する変更（例えば、プロトコル、ハードウェア・デバイス、ソフトウェア・プラットフォームなど）の完全な範囲で考えたときに、添付の特許請求の範囲によってのみ定義されるべきであることを理解されたい。

【図面】

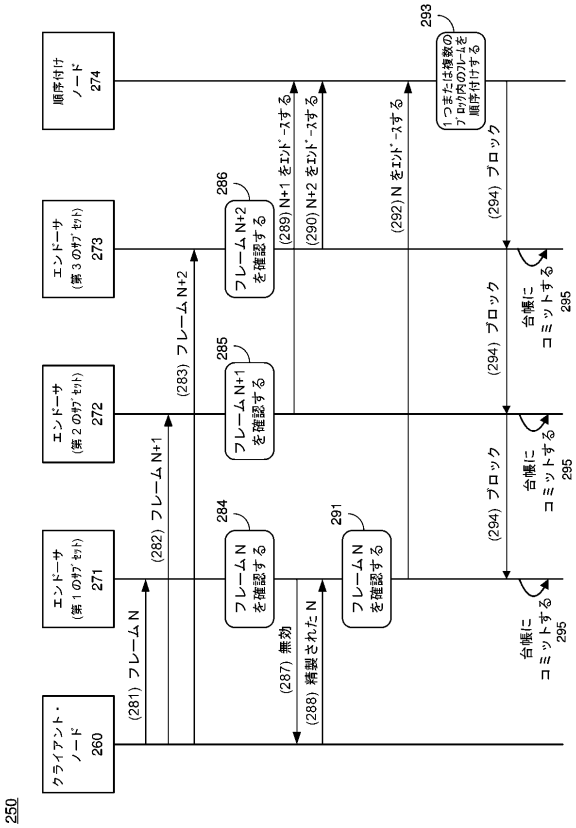
【図１】



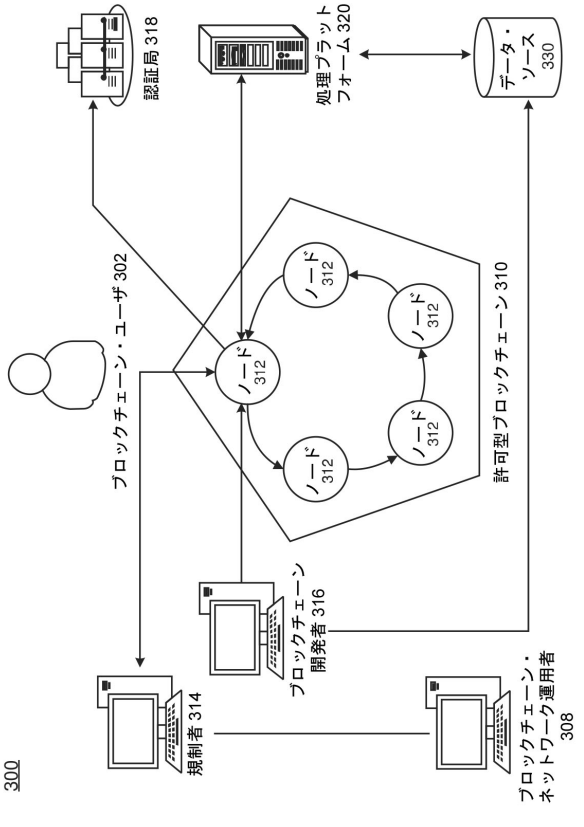
【図２Ａ】



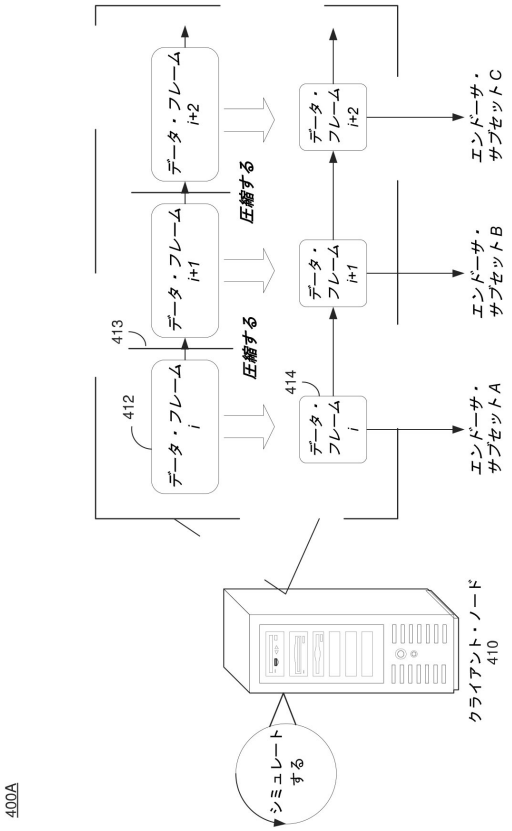
【図２Ｂ】



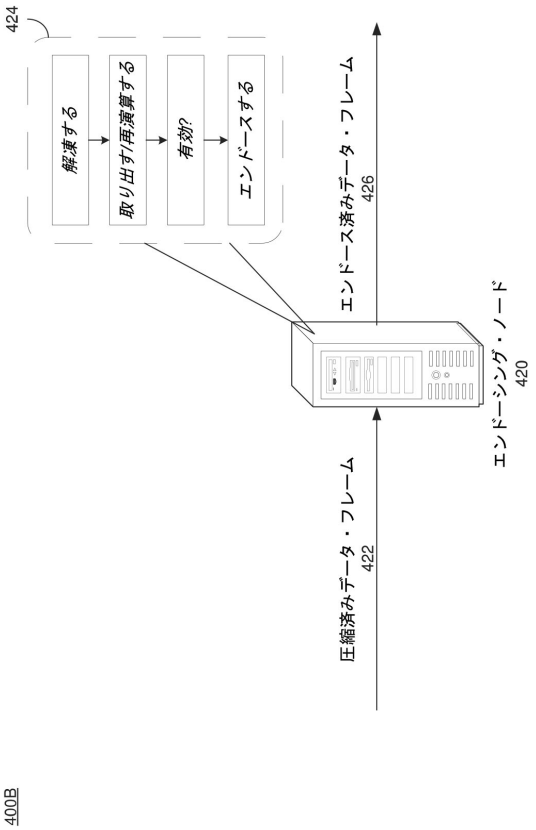
【図３】



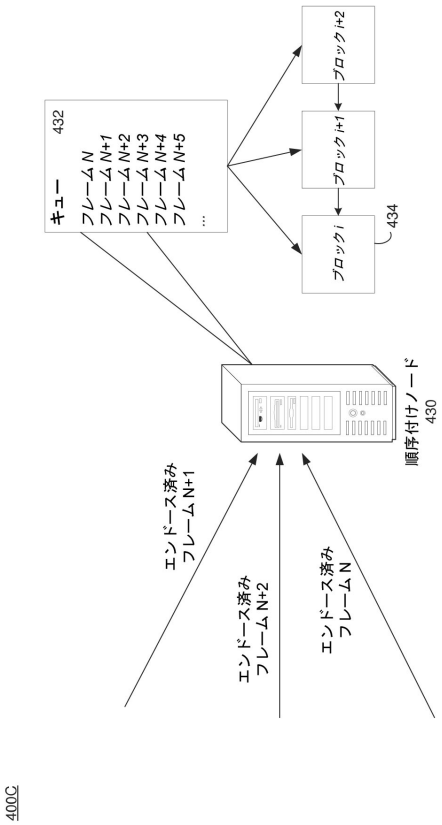
【図４Ａ】



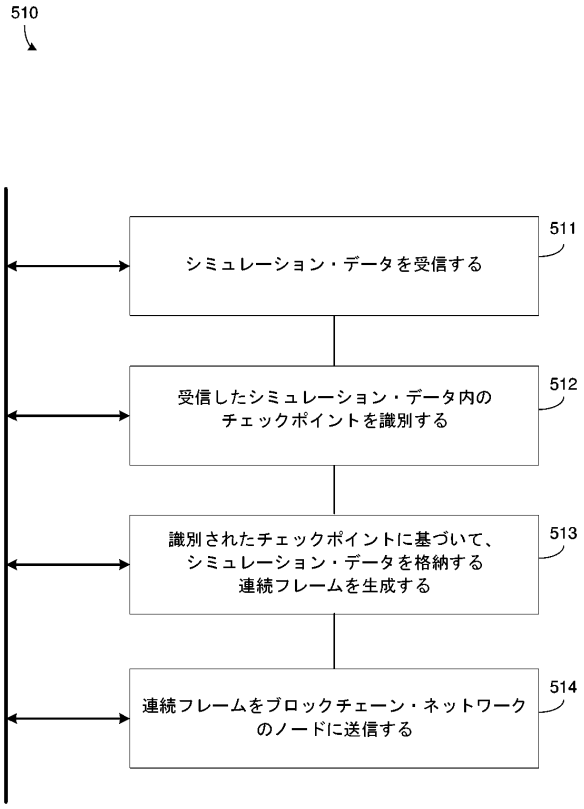
【図４Ｂ】



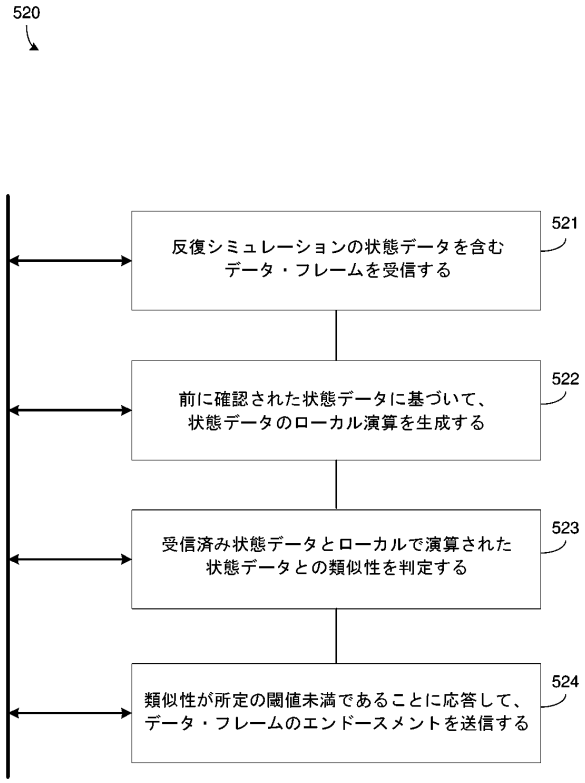
【図４Ｃ】



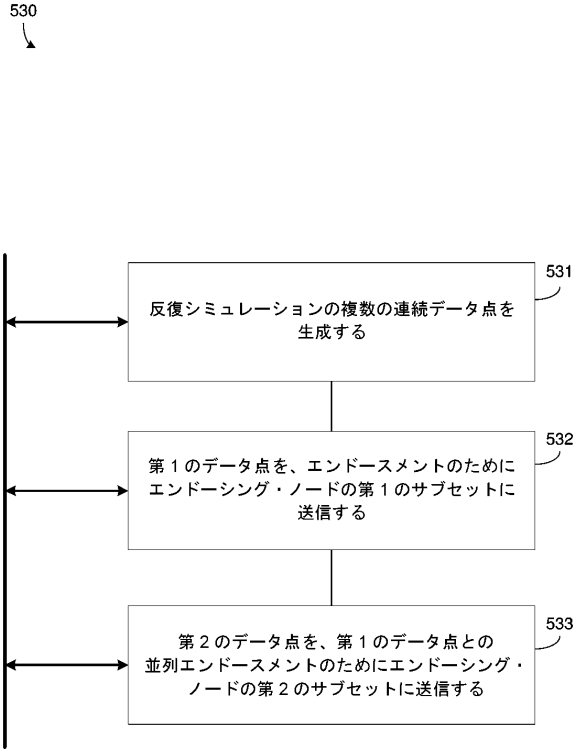
【図５Ａ】



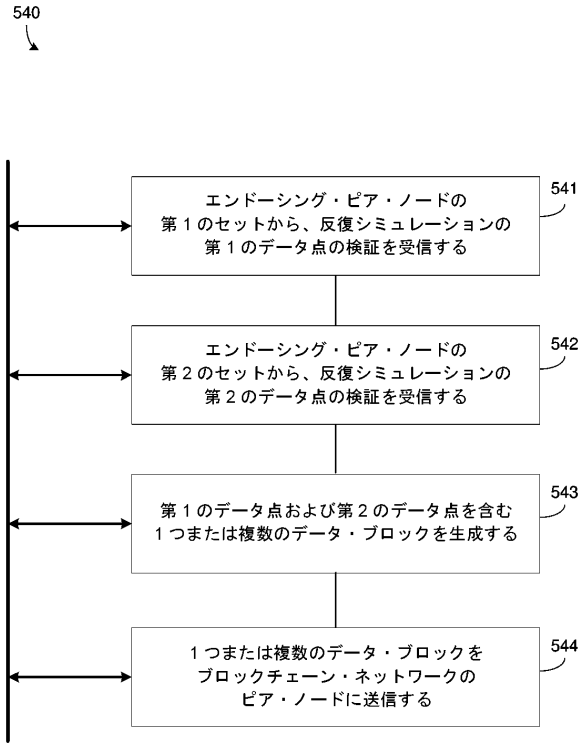
【図５Ｂ】



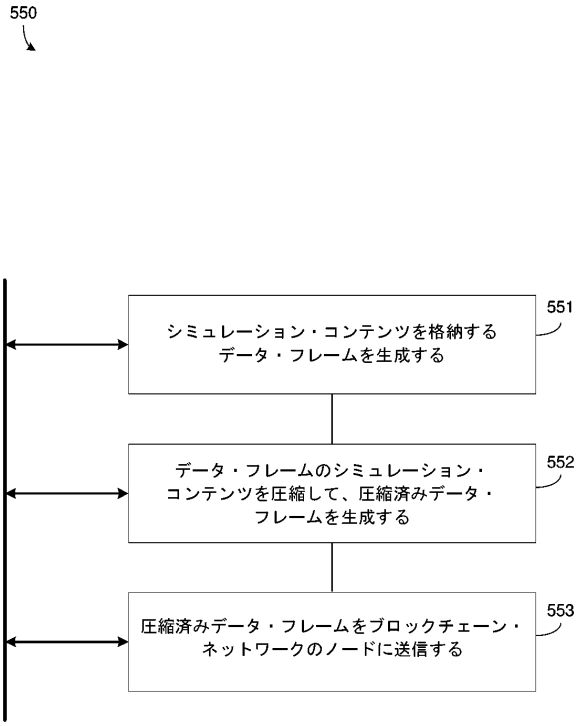
【図５Ｃ】



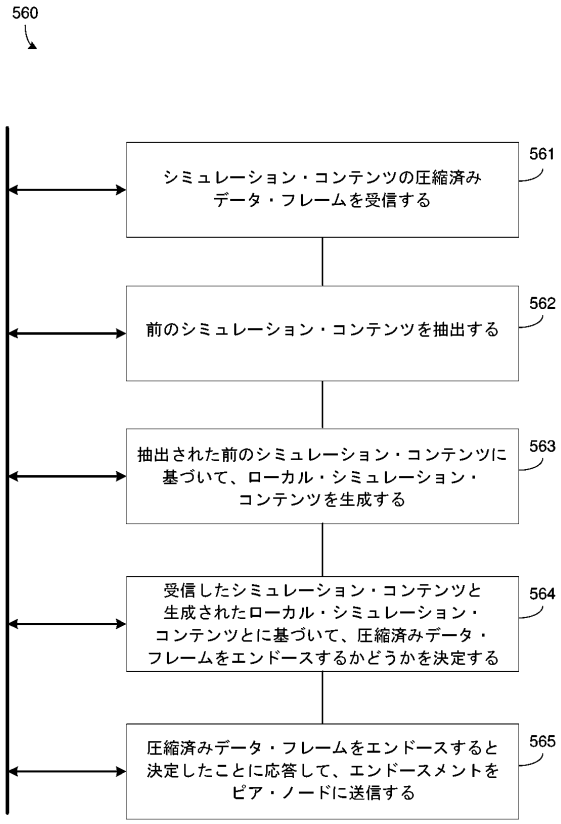
【図５Ｄ】



【図５Ｅ】

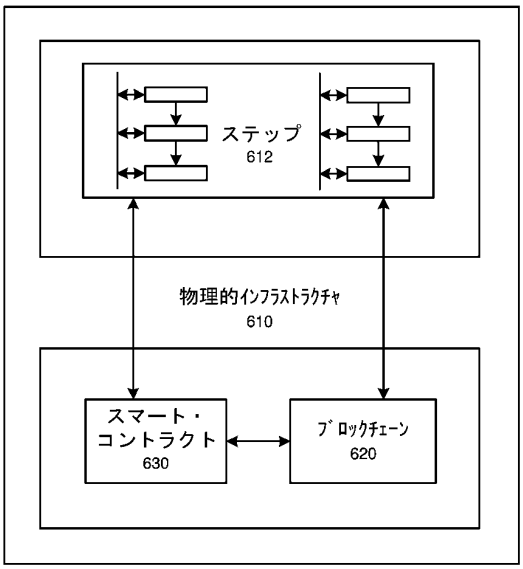


【図５Ｆ】



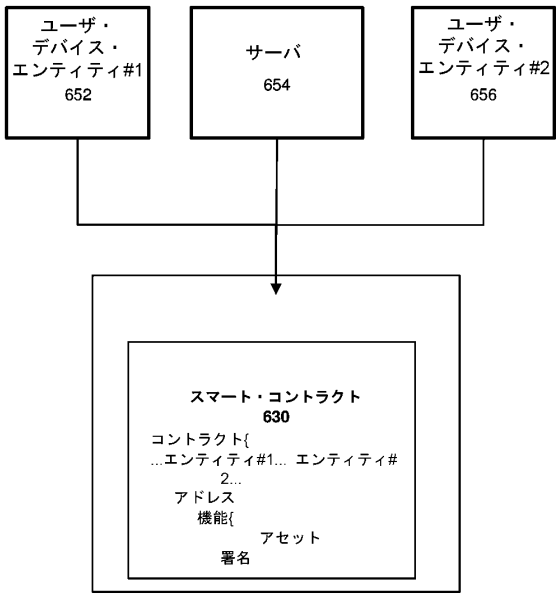
【図６Ａ】

600



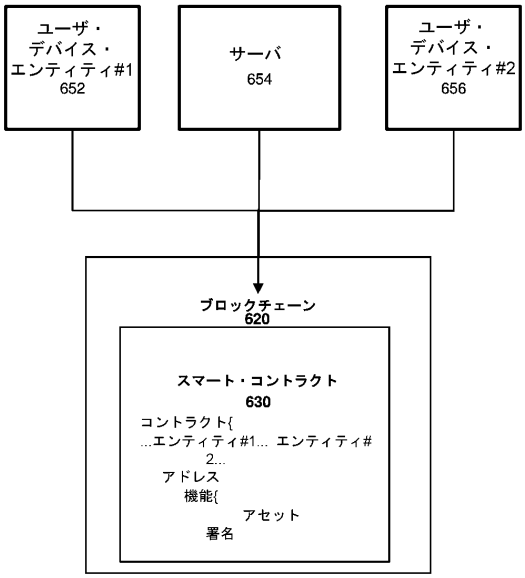
【図６Ｂ】

650

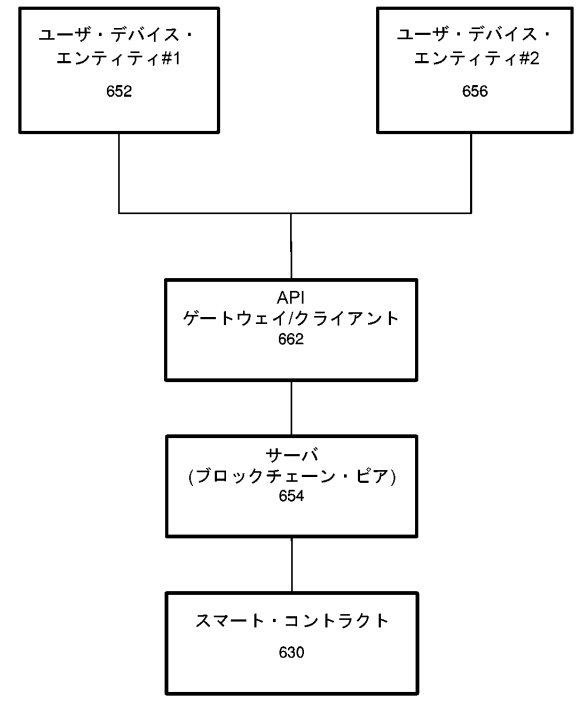


【図６Ｃ】

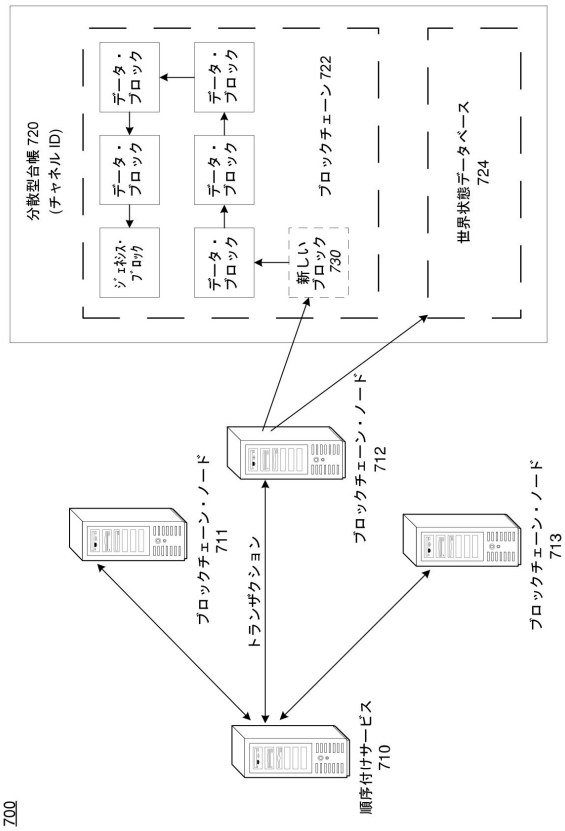
650



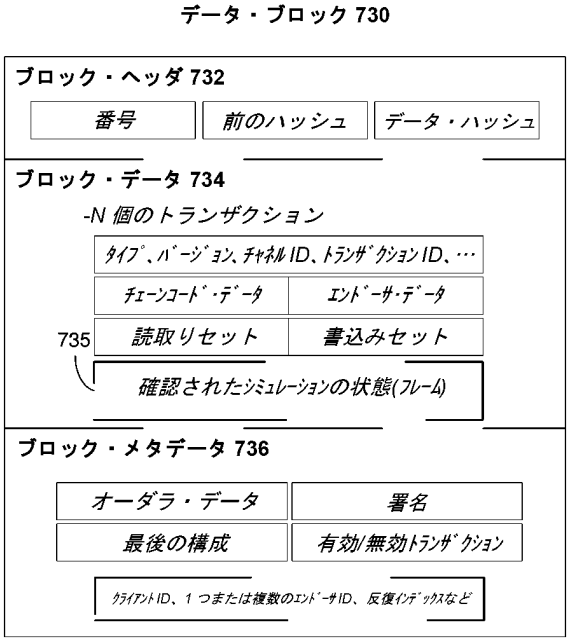
【図６Ｄ】



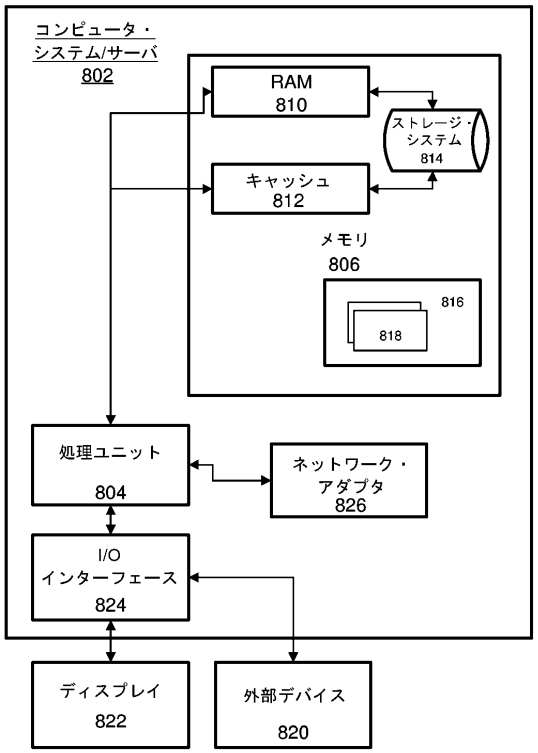
【図７Ａ】



【図７Ｂ】



【図８】



フロントページの続き

(72)発明者　ラマン、ラヴィ キラン
アメリカ合衆国１２６０１ ニューヨーク州ポキプシー エムエス・ピー３８６ サウス・ロード２４５５
(72)発明者　ヴァーシュニー、クシュ、ラージュ
アメリカ合衆国１０５９８ ニューヨーク州ヨークタウン・ハイツ キッチャワン・ロード１１０１
(72)発明者　バクリン、ローマン
アメリカ合衆国１０５９８ ニューヨーク州ヨークタウン・ハイツ キッチャワン・ロード１１０１
(72)発明者　ハインド、マイケル
アメリカ合衆国１０５９８ ニューヨーク州ヨークタウン・ハイツ キッチャワン・ロード１１０１
(72)発明者　レミー、セコウ、ライオネル
ケニア００１００ ナイロビ ランガタ・ロード カトリック・ユニバーシティ・オブ・イーストアフリカ
(72)発明者　ピッサダキ、エレフテリア
アメリカ合衆国１２６０１ ニューヨーク州ポキプシー エムエス・ピー３８６ サウス・ロード２４５５
(72)発明者　ボーア、ネルソン、キビチ
ケニア００２００ ナイロビ キリマニ レナーナ・ロード １／３０１ ジ・アトリウム・エル・アール

審査官　吉田 歩
(56)参考文献　特開２００１－１４２３８５（ＪＰ，Ａ）
米国特許出願公開第２０１８／０２０５５５２（ＵＳ，Ａ１）
特開２００８－２８２３０８（ＪＰ，Ａ）
特開平１０－１０５０５４（ＪＰ，Ａ）
特開２００１－２０２３９１（ＪＰ，Ａ）
国際公開第２０１８／０２０９４４（WO，Ａ１）
(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　２１／６４