(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7379371号
(P7379371)

(45)発行日　令和5年11月14日(2023.11.14)

(24)登録日　令和5年11月6日(2023.11.6)

(51)国際特許分類
H04L 9/32 (2006.01)

FI
H04L 9/32 200Z

請求項の数　8　（全20頁）

(21)出願番号　特願2020-557911(P2020-557911)
(86)(22)出願日　平成31年4月24日(2019.4.24)
(65)公表番号　特表2021-522702(P2021-522702A)
(43)公表日　令和3年8月30日(2021.8.30)
(86)国際出願番号　PCT/IB2019/053380
(87)国際公開番号　WO2019/207501
(87)国際公開日　令和1年10月31日(2019.10.31)
審査請求日　令和4年3月25日(2022.3.25)
(31)優先権主張番号　1806907.0
(32)優先日　平成30年4月27日(2018.4.27)
(33)優先権主張国・地域又は機関
英国(GB)
(31)優先権主張番号　1806909.6
(32)優先日　平成30年4月27日(2018.4.27)
最終頁に続く

(73)特許権者　318001991
エヌチェーン ライセンシング アーゲー
スイス・６３００・ツーク・グラーフェナウヴェーク・６
(74)代理人　100107766
弁理士　伊東 忠重
(74)代理人　100070150
弁理士　伊東 忠彦
(74)代理人　100135079
弁理士　宮崎 修
(72)発明者　クラマー，ディーン
イギリス国 シーエフ１０ ２エイチエイチ カーディフ チャーチル ウェイ チャーチル ハウス ７ス フロア アーカート－ダイクス アンド ロード エルエルピー
最終頁に続く

(54)【発明の名称】　ブロックチェーンネットワークの分割

(57)【特許請求の範囲】
【請求項１】
ブロックチェーンネットワークに関連するトランザクションを割り当てるためのコンピュータ実施方法であり、前記ブロックチェーンネットワークは複数のシャードに分割され、各シャードは少なくとも1つのノードを含み、前記ブロックチェーンネットワーク内の各ノードは前記複数のシャードの中の少なくとも1つのシャードに関連付けられる、方法であって、
所与のトランザクションのインプットを識別するステップと、
前記所与のトランザクションの前記インプットに対応する前のトランザクションのアウトプットを識別するステップと、
前記前のトランザクションが前記所与のトランザクションの親トランザクションであると決定するステップであり、前記所与のトランザクションは子トランザクションである、ステップと、
前記複数のシャードの中の或るシャードを識別するステップと、
前記親トランザクション及び前記子トランザクションを前記複数のシャードの中の前記識別されたシャードに割り当てるステップと、
前記親トランザクション及び前記子トランザクションを決定されたシャード内の少なくとも1つのノードに分配するステップと
を含む方法。
【請求項２】

前記親トランザクションは、子ブロックチェーントランザクションの複数のインプットのうち、インデックスを有するインプットを使用して識別され、使用される前記インプットは、前記インデックスに基づいて選択される、請求項１に記載の方法。
【請求項３】
親トランザクションは、子ブロックチェーントランザクションの複数のインプットのうち最大のサブセットのインプットを使用して識別される、請求項１又は２に記載の方法。
【請求項４】
ノードのシャードメンバーシップ情報の要求を他のノードに通信するステップを更に含む、請求項１乃至３のうちいずれか１項に記載の方法。
【請求項５】
ノードのシャードメンバーシップ情報を他のノードに通信するステップを更に含む、請求項１乃至４のうちいずれか１項に記載の方法。
【請求項６】
前記通信は、修正addrメッセージを使用して実行される、請求項４又は５に記載の方法。
【請求項７】
システムであって、
プロセッサと、
前記プロセッサによる実行の結果として、当該システムに請求項１乃至６のうちいずれか１項に記載のコンピュータ実施方法を実行させる実行可能命令を含むメモリと
を含むシステム。
【請求項８】
コンピュータシステムのプロセッサにより実行された結果として、前記コンピュータシステムに請求項１乃至６のうちいずれか１項に記載のコンピュータ実施方法を少なくとも実行させる実行可能命令を記憶した非一時的なコンピュータ読み取り可能記憶媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本開示は、ブロックチェーンネットワークを分割するための方法及び分割されたブロックチェーンネットワークのトランザクションを承認するための方法に関し、排他的ではないが、特に、ビットコインブロックチェーンの未使用トランザクションアウトプット(unspent transaction output, UTXO)セットを分割するための方法及びビットコインブロックチェーンの分割されたUTXOセットのトランザクションを承認するための方法に関する。
【背景技術】
【０００２】
この文献では、全ての形式の電子的なコンピュータに基づく分散型の台帳を含む「ブロックチェーン(blockchain)」という用語を使用する。これらは、コンセンサスに基づくブロックチェーン及びトランザクションチェーン技術、プライベート型(permissioned)及びパブリック型(un-permissioned)台帳、共有台帳及びこれらの変形を含む。ブロックチェーン技術の最も広く知られているアプリケーションはビットコイン台帳であるが、他のブロックチェーンの実装も提案されて開発されている。ここでは、便宜上及び説明上の目的でビットコインが参照されることがあるが、本開示は、ビットコインブロックチェーンでの使用に限定されず、代替のブロックチェーンの実装及びプロトコルも、本開示の範囲内に入る点に留意すべきである。「ユーザ」という用語は、ここでは人間又はプロセッサに基づくリソースを示すことがある。「ビットコイン」という用語は、ここではビットコインプロトコルから派生するか或いはこれに基づくいずれかのバージョン又はバリエーションを含むように使用される。
【０００３】
ブロックチェーンは、ピアツーピアの電子台帳であり、これは、結果としてトランザク

ションで構成される、ブロックで構成されるコンピュータに基づく非集中的な分散型システムとして実装される。各トランザクションは、ブロックチェーンシステム内の参加者の間のデジタル資産の制御の移転を符号化するデータ構造であり、少なくとも1つのインプット及び少なくとも1つのアウトプットを含む。各ブロックは、前のブロックのハッシュを含み、それにより、ブロックチェーンの開始以降にブロックチェーンに書き込まれた全てのトランザクションの永続的で修正不可能なレコードを作成するように、ブロックが一緒につながれる。トランザクションは、トランザクションのインプット及びアウトプットに埋め込まれたスクリプトとして知られる小さいプログラムを含み、これは、トランザクションのアウトプットがどのように誰によってアクセスできるかを指定する。ビットコインプラットフォームでは、これらのスクリプトはスタックに基づくスクリプト言語を使用して記述される。

【０００４】

トランザクションがブロックチェーンに書き込まれるために、これは「承認され(validated)」なければならない。ネットワークノード(マイナー)は、各トランザクションが有効であることを確保するために作業を実行し、無効なトランザクションはネットワークから拒否される。ノードにインストールされたソフトウェアクライアントは、ロック及びロック解除スクリプトを実行することにより、未使用トランザクション(unspent transaction, UTXO)に対してこの承認動作を実行する。ロック及びロック解除スクリプトの実行がTRUEに評価された場合、トランザクションは有効であり、トランザクションはブロックチェーンに書き込まれる。したがって、トランザクションがブロックチェーンに書き込まれるために、これは、i)トランザクションを受信した最初のノードにより承認されなければならず、トランザクションが承認された場合、ノードはそれをネットワーク内の他のノードに中継し、ii)マイナーにより構築された新たなブロックに追加されなければならず、且つ、iii)マイニングされなければならず、すなわち、過去のトランザクションの公開台帳に追加されなければならない。

【０００５】

ブロックチェーン技術は、暗号通貨の実装の使用にとって最も広く知られているが、デジタル起業家は、新たなシステムを実装するために、ビットコインが基礎となる暗号セキュリティシステムと、ブロックチェーンに記憶できるデータとの双方の使用を調査し始めている。ブロックチェーンが、暗号通貨の領域に限定されない自動化タスク及びプロセスに使用され得る場合には、非常に有利になる。このような解決策は、ブロックチェーンの利点(例えば、イベントの永続的な改ざん防止記録、分散処理等)を利用することができる一方で、これらの用途においてより多目的になることができる。

【０００６】

上記のように、ブロックチェーンネットワーク、例えば、ビットコインブロックチェーンネットワークは、安全な分散型コンピューティングシステムである。システムのフルノード(full node)は、全体のブロックチェーンのコピーを保持及び管理し、トランザクションを送信及び受信し、これらを承認し、共有分散型コンセンサスプロトコルに基づいてブロックをブロックチェーンに追加する。この手法は安全であるが、各トランザクションが全てのフルノードにより承認されて記憶されるという事実に関連するスケーリングの欠陥を有する。承認に関して、これは、各トランザクションがマイナーに向かって伝搬できる前に承認される必要があるので、トランザクションのネットワーク伝搬の遅延を引き起こす。さらに、承認に起因する遅延は、ネットワークを、シビル(Sybil)攻撃のような「二重支出(double-spend)」に関連する攻撃の影響を受けやすくする。

【発明の概要】

【０００７】

本開示は、シャード(shard)されたブロックチェーンネットワーク上でトランザクションを割り当てる及び/又は承認するのための関連技術又はプロトコルと共に、シャーディング(sharding)としても知られる水平分割の使用を通じて、ブロックチェーンネットワークのスケーラビリティ、速度、信頼性及びセキュリティを改善することを目的とする。以下

に、
・シャードされたブロックチェーンのネットワーク構造、及び
・シャードされたUTXO及びメモリプール(mempool)構造
について開示する。
【０００８】
当該技術分野における分割は、2つの特定の次元、すなわち、水平次元及び垂直次元を考慮する。シャードとして知られる水平分割データベースの分割セクションでは、特定のデータベーススキーマの複数のインスタンスが事実上存在し、データはこれらのインスタンスのそれぞれの間に広がり、インスタンスの冗長性を割り引いている。しかし、垂直分割は、特定のデータベーススキーマを複数のノードの間で分割することであり、それにより、特定のオブジェクトの属性は正規化を使用して分散される。
【０００９】
ブロックチェーンネットワークに関与することを望む異なる関係者は、小さい低電力マシンからサーバーファームまでの範囲の、様々なコンピューティングリソースを保有できる。したがって、参加者は、コンピューティングリソースによって、ブロックチェーンネットワークにおける所定のレベルの関与に制限される。
【００１０】
ビットコインでは、ブロックチェーン自体は、ブロックの作成時にマイニングされたコインの所有権の特定の変化を示す連結されたトランザクションのセットである。トランザクションの承認中に、必要とされるチェックの1つは、二重支出が存在していないことをチェックすることである。二重支出は、トランザクションアウトプットが既にビットコインメモリプールにあるか或いはブロックチェーンに対して確認されているトランザクションインプットで参照されたときに生じる。メモリプールは、各フルノードが自身のために維持管理するビットコイントランザクションのためのメモリプール又は領域への参照であることが知られている。従来、トランザクションがノードによって検証された後に、ブロックに挿入されるまでメモリプール内で待機する。トランザクションの承認を、トランザクションインプットをチェックする点でより効率的にするために、全体のブロックチェーンを承認する代わりに、ネットワークの現在の状態が、UTXOセットとして知られる独立した構造内に保持される。この構造は、トランザクションにより既に使用されることになっている各トランザクションアウトプットを含み、これは、コインベース及び標準トランザクションを含むことができる。
【００１１】
本開示の一態様によれば、ブロックチェーンネットワークをシャードに分割するコンピュータ実施方法が提供される。当該方法は、ブロックチェーントランザクションのトランザクションIDを識別するステップと、トランザクションIDに基づいてトランザクションをシャードに割り当てるステップとを含む。
【００１２】
ブロックチェーンネットワークをシャードに分割することは、ユーザがブロックチェーンネットワークとの関与の自分のレベルを選択することを可能にする。各ユーザは、1つ以上のシャードのメンバーになることを選択できる。全てのシャードよりも少ないメンバーであるユーザは、ユーザがメンバーであるシャードに割り当てられたトランザクションの全てを記憶するために、より少ない記憶空間を必要とする。トランザクションIDに基づいてトランザクションをシャードに割り当てることは、結果としてのシャードのサイズがほぼ等しくなるという利点を提供し、それにより、より小さいシャードのメンバーに対してより大きいシャードのメンバーに過度の負担をかけることを回避し、同時に、トランザクション及び関連する検証が過度の遅延なく正確に実行されることを可能にする。
【００１３】
ここで言及されるユーザは、1つ以上のノード又は計算デバイスに関連付けられてもよく、これらのノードはまた、分割されたブロックチェーンネットワーク内のクライアントエンティティとも呼ばれてもよい。以下、ユーザへの言及はまた、ユーザに関連付けられ

たノード又はエンティティ(シャード又は分割されたブロックチェーンネットワークの一部であるノード又はエンティティを所有又は制御してもよい)への言及であると理解されてもよい。各ノードは、分割されたブロックチェーンネットワーク内の少なくとも1つ以上の他のノードと通信可能に結合されてもよい。

【００１４】

本開示の他の態様(以下に説明する)と共に、これに関連してここで説明するこれらの利点は、ノードの構造と、シャードされたブロックチェーンネットワークの結果のネットワークトポロジー及びアーキテクチャと、ネットワークのノードに関連するプロトコルに起因する。このようなシャードされたネットワークにおいてUTXOを受信、記憶及び/又は承認することは、異なるシャードに属するノードとの通信に関連するルール及びプロトコルに基づいて、各シャード内のノードについての承認技術と共に、通信、データ記憶、データ共有のための、説明すると共に特許請求の範囲に記載される方法、ルール又はプロトコルを使用して実行される。

【００１５】

これらの特定の構造、データフローの方法、トランザクション割り当て及び承認プロトコルについて、本開示の様々な実施形態に関して以下に更に説明する。有利には、シャードされたブロックチェーンネットワーク内のトランザクションの割り当て及びこのような割り当てられたトランザクションの承認のためにここに記載されるシャードされたネットワーク構造又はアーキテクチャ及び関連する方法は、データフロー、データ記憶及びUTXO承認チェックのための新規な技術を可能にする。さらに、これらの技術は、構造及びデータ通信/承認プロトコルの観点から、ビットコインブロックチェーンにおけるシビル攻撃のような二重支出攻撃を有利に防止する。

【００１６】

当該方法は、トランザクションIDを使用して動作を実行するステップを更に含んでもよい。トランザクションをシャードに割り当てるステップは、動作の結果に基づいてもよい。

【００１７】

これは、シャードの配置が動作の選択に依存して調整できるという利点を提供する。

【００１８】

動作は、モジュロ演算を含んでもよい。

【００１９】

これは、所望の数の等しいサイズのシャードがより容易に生成できるという利点を提供する。

【００２０】

本開示によれば、ブロックチェーンネットワークをシャードに分割する更なるコンピュータ実施方法が提供される。当該方法は、親ブロックチェーントランザクションを識別するステップであり、親トランザクションは、子ブロックチェーントランザクションのインプットに対応するアウトプットにより定義される、ステップと、親トランザクション及び子トランザクションを同じシャードに割り当てるステップとを含む。

【００２１】

ブロックチェーンネットワークをシャードに分割することは、ユーザがブロックチェーンネットワークとの関与の自分のレベルを選択することを可能にする。各ユーザは、1つ以上のシャードのメンバーになることを選択できる。全てのシャードよりも少ないメンバーであるユーザは、ユーザがメンバーであるシャードに割り当てられたトランザクションの全てを記憶するために、より少ない記憶空間を必要とする。子ブロックチェーントランザクションのインプットに対応するアウトプットにより定義される親ブロックチェーントランザクションを識別し、親トランザクション及び子トランザクションを同じシャードに割り当てることに基づいて、トランザクションをシャードに割り当てることは、特定のシャードのメンバーであるユーザにより実行される承認動作が、異なるシャードのメンバーであるユーザに/ユーザから送信される少ない情報を必要としつつ実行され得るという利点を提供する。この理由は、承認されている子トランザクションが同じシャードのメンバー

である親トランザクションを常に有するためである。

【００２２】

本開示によれば、ブロックチェーンネットワークに関連するトランザクションを割り当てるためのコンピュータ実施方法に関し、ブロックチェーンネットワークは複数のシャードに分割され、各シャードは少なくとも1つのノードを含み、ブロックチェーンネットワーク内の各ノードは、複数のシャードの中の少なくとも1つのシャードに関連付けられ、当該方法は、所与のトランザクションのインプットを識別するステップと、所与のトランザクションのインプットに対応する前のトランザクションのアウトプットを識別するステップと、前のトランザクションが所与のトランザクションの親トランザクションであると決定するステップであり、所与のトランザクションが子トランザクションである、ステップと、複数のシャードの中の或るシャードを識別するステップと、親及び子トランザクションを複数のシャードの中の識別されたシャードに割り当てるステップと、親及び子トランザクションを決定されたシャード内の少なくとも1つのノードに分配するステップとを含む。

【００２３】

親トランザクションは、子ブロックチェーントランザクションの複数のインプットのうち或るインプットを使用して識別されてもよい。使用されるインプットは、そのインデックスに基づいて選択されてもよい。インデックスは1でもよく、その場合、使用されるインプットは、複数のインプットのうち最初のインプットである。

【００２４】

これは、複数のインプットを有する子トランザクションがシャードに割り当てられることを可能にするという利点を提供する。

【００２５】

親トランザクションは、子ブロックチェーントランザクションの複数のインプットのうち最大のサブセットのインプットを使用して識別されてもよい。例えば、子トランザクションが5つのインプットを有し、5つのうち2つが前のトランザクションの2つのアウトプットを参照し、残りの3つのインプットのそれぞれが3つの異なる前のトランザクションを参照する場合、親トランザクションは、2つのインプットが最大のサブセットのインプットであるので、2つのインプットが共に参照する前のトランザクションとして定義される。

【００２６】

これは、複数のインプットを有する子トランザクションについて、異なるシャードのメンバーであるユーザから必要とされる情報の量が低減されるという利点を提供する。

【００２７】

いくつかの実施形態では、当該方法は、ノードのシャードメンバーシップ情報を、ノードに関連するシャード内の全ての他のノード及び/又はネットワーク内の1つ以上の他のノードに通信又はブロードキャストするステップを含む。当該方法はまた、通信が修正addrメッセージを使用して実行されることを含んでもよく、修正addrメッセージは、ノードが関連付けられている1つ以上のシャードの指示を含む。

【００２８】

本開示によれば、ブロックチェーントランザクションを承認するためのコンピュータ実施方法も提供される。当該方法は、少なくとも1つのUTXOを含む少なくとも1つのシャードのメンバーノードからトランザクションの少なくとも1つのそれぞれのインプットにより参照される少なくとも1つのUTXOを要求するステップと、少なくとも1つのノードから少なくとも1つのUTXOの有効性データを取得するステップと、有効性データを使用して少なくとも1つのインプットに対して承認チェックを実行するステップとを含む。

【００２９】

この方法は、ブロックチェーントランザクションの承認がシャードされたブロックチェーンネットワークで行われることを可能にする。この方法により提供される利点は、各ユーザが1つ以上のシャードのメンバーになることを選択でき、全てのシャードよりも少ないメンバーであるユーザが、当該ユーザがメンバーであるシャードに割り当てられたトラ

ンザクションの全てを承認するために、より少ない計算能力を必要とすることである。
【００３０】
本開示によれば、ブロックチェーントランザクションを承認するためのコンピュータ実施方法が更に提供される。当該方法は、トランザクションの少なくとも1つのそれぞれのインプットにより参照される少なくとも1つのUTXOを含む少なくとも1つのシャードを識別するステップと、トランザクションを少なくとも1つのシャードの少なくとも1つのメンバーノードに送信するステップと、UTXOの有効性データを使用して少なくとも1つのインプットに対して承認チェックを実行するステップとを含む。
【００３１】
この方法は、ブロックチェーントランザクションの承認がシャードされたブロックチェーンネットワークで行われることを可能にする。この方法により提供される利点は、各ユーザが1つ以上のシャードのメンバーになることを選択でき、全てのシャードよりも少ないメンバーであるユーザが、当該ユーザがメンバーであるシャードに割り当てられたトランザクションの全てを承認するために、より少ない計算能力を必要とすることである。
【００３２】
上記の方法のいずれかは、ノードのシャードメンバーシップ情報の要求を他のノードに通信するステップを更に含んでもよい。
【００３３】
これは、シャードメンバーシップ情報を求めるノードが、より容易にその情報を見つけるためのメカニズムを備えるという利点を提供する。
【００３４】
上記の方法のいずれかは、ノードのシャードメンバーシップ情報を他のノードに通信するステップを更に含んでもよい。
【００３５】
これは、ノードのシャードメンバーシップ情報がノードの間で転送されるメカニズムを提供し、それにより、承認動作を実行するノードが失敗する可能性を減少させるという利点を提供する。
【００３６】
通信は、修正addrメッセージを使用して実行されてもよい。
【００３７】
これは、ノードの間でシャードメンバーシップ情報を交換するためのより安全なメカニズムを提供するという利点を提供する。
【００３８】
本開示はまた、
プロセッサと、
プロセッサによる実行の結果として、システムにここに記載のコンピュータ実施方法のいずれかの実施形態を実行させる実行可能命令を含むメモリと
を含むシステムを提供する。
【００３９】
本開示はまた、コンピュータシステムのプロセッサにより実行された結果として、コンピュータシステムにここに記載のコンピュータ実施方法の実施形態を少なくとも実行させる実行可能命令を記憶した非一時的なコンピュータ読み取り可能記憶媒体を提供する。
【００４０】
以下に、本開示の好ましい実施形態について、添付の図面を参照して、限定的な意味ではなく、一般的に説明する。
【図面の簡単な説明】
【００４１】
【図１ａ】本開示の第1の実施形態による、従来のブロックチェーンネットワーク(図１ａ)とシャードされたブロックチェーンネットワーク(図１ｂ)との比較を示す。
【図１ｂ】本開示の第1の実施形態による、従来のブロックチェーンネットワーク(図１ａ)

とシャードされたブロックチェーンネットワーク(図１ｂ)との比較を示す。
【図２】本開示の第2の実施形態に関連して使用されるノード使用の回転を示す。
【図３】本開示の第3の実施形態に従ってトランザクションをシャードに割り当てる方法を示す。
【図４】第4の実施形態に従ってトランザクションをシャードに割り当てる方法を示す。
【図５】従来技術のUTXOデータ構造を示す。
【図６】本開示の第5の実施形態を示す。
【図７ａ】本開示の第6の実施形態を示す。
【図７ｂ】本開示の第6の実施形態を示す。
【図８ａ】本開示の第7の実施形態を示す。
【図８ｂ】本開示の第7の実施形態を示す。
【図９】本開示の様々な実施形態が実装できる計算環境を示す概略図である。
【発明を実施するための形態】
【００４２】
現在のブロックチェーンネットワークでは、異なるノードがほとんど構造化されていない様式でピアツーピアで接続されている(ノード発見を支援するためのビットコインクライアント内の多数のハードコードされたネットワークシードを除く)。これらのノードは、有効なトランザクション、ブロック及び他のノードに関する情報を共有するために通信する。
【００４３】
［シャードされたネットワークの構造］
本開示の第1の実施形態は図１ｂに見られ、これは、本開示による、シャードされたブロックチェーンネットワークの構造を示す。他方、図１ａは、既存の、すなわち、従来技術のブロックチェーンネットワークの構造を示す。
【００４４】
本開示によれば、関係者がブロックチェーンネットワークにおける所定のレベルの関与に参加するために、高価で強力な計算リソースを有することへの依存を低減するために、関係者は、シャードされたブロックチェーンネットワークのいずれかの数のシャードのメンバーになることが許容されてもよい。これは、趣味を楽しむ人を含む小規模の関係者が、図１ｂに示すネットワークの単一のシャードのメンバーになることを選択でき、金融機関のような大規模な関係者が、図１ｂのシャードされたブロックチェーンネットワークの多く或いは全てのシャードのメンバーになることを選択できることを意味する。この手法は、トランザクション履歴のセキュリティを必要とし得るエンティティ又は関係者に対応し、例えば、より大きいセキュリティを望むか或いは必要とする関係者が、ブロックチェーン内の全てのトランザクションを承認して記憶できるようにする一方で、同じ(より大きい)レベルを望まないか或いは必要としない可能性があるか、或いはより軽い関与を望む可能性がある他の関係者又はエンティティも、図１ｂの同じシャードされたブロックチェーンネットワークに参加し、ブロックチェーンのサブセットのみを記憶し得ることを確保する。
【００４５】
図１ｂに見られるように、特定のノードは、1つ以上のシャードグループのメンバーとなることができる。これは、この図面に示す網掛け線から分かり、網掛けされた領域内のノードは、シャード2とシャード3との双方のメンバーである。通信のために、現在のビットコインネットワークとビットコインSV(Bitcoin SV, BSV)クライアントでは、利用可能なピアのリスト、すなわち、ネットワーク内のノードが、接続可能なノード、配布可能なノード及び受信可能なノードに関する情報を保持する。第1の実施形態によるシャードされたブロックチェーンでは、各ノードがどのシャードのメンバーであるかを含む更なる情報が保持される。いくつかの実装では、ネットワークの間のトランザクション伝搬を処理するために、図１ｂにおけるシャードされたネットワークに示す各ノードは、異なるシャード宛のこれらのトランザクションを伝搬するために、各シャードからの少なくとも単一のノードと通信し得るように配置又は構成される。いくつかの実装では、各ノードによ

り保持される情報は、接続可能なノード、配布可能なノード、受信可能なノードと、図１ｂに見られるシャードされたネットワークにおいて属するシャードを示すための、データ構造の形式でもよい。識別子、エンティティアソシエーション等のようなノードに関連する他の詳細も保持されてもよい。このデータ構造は、各ノードに関連付けられたメモリ内に保持されてもよく、或いは、例えば、シャードに関連付けられたメモリ内に保持されてもよい。

【００４６】

図１ｂに見られるようなシャードされたネットワークに関する第2の実施形態では、ノードが単一のシャード内の複数の他のノードと通信する技術について説明する。この技術は、ブロックチェーンネットワーク内の「シビル型(Sybil style)」攻撃を有利に防止する。

【００４７】

シビル攻撃は、単一の敵対者又は悪意のあるエンティティが、ネットワークに知られていないネットワーク上の複数のノードを制御している可能性がある攻撃である。例えば、敵対者は、複数のコンピュータ及びIPアドレスを作成してもよく、また、複数のアカウント/ノードが全て存在すると装うことを試みて、複数のアカウント/ノードを作成することができる。このような攻撃の現れは、以下の例示的な実装により見られる可能性がある。攻撃者が制御するクライアントでネットワークを埋めることを試みる場合、ノードは攻撃者のノードのみに接続する可能性が非常に高くなる。例えば、攻撃者は、ノードについてのブロック及びトランザクションを中継するのを拒否でき、事実上、その特定のノードをネットワークから切断する。これはまた、作成したブロックを攻撃者が中継することにより現れることも可能であり、事実上、ノード又はエンティティを別のネットワークに配置し、それにより、ノード及びそのノード又はエンティティに関連するトランザクションを分離し、これは、二重支出攻撃にとって開放されることを表す。したがって、シビル攻撃は、既存のブロックチェーンネットワークにとって問題である。

【００４８】

図１ｂに見られるようなシャードされたブロックチェーンネットワークにおいてシビル攻撃を防止するために、第2の実施形態によれば、ノードは、単一のシャード内の複数又は全ての他のノードと通信するように構成される。上記のように、シビル攻撃は、特定のノードから送信されたトランザクションを事実上無視でき、ネットワークを通じたこれらの伝搬を更に防止する。したがって、本開示の第2の実施形態では、図２に見られるように、所与のシャード内のノードが、他のシャード内のノードに関する情報を交換し、これらの使用を回転させることができる技術を提供する。

【００４９】

第2の実施形態によれば、シャードされたネットワーク内の各ノードは、全てのトランザクションを互いにブロードキャストできる。所与のノードがトランザクションに関連するシャードのメンバーではない場合(この関連付けについては、第3及び第4の実施形態を参照して以下に説明する)、完全なトランザクション承認を行う代わりに、先に伝搬する前に基本的なトランザクションレベルのチェックを実行する。いくつかの実装では、第2の実施形態に関連して上記に説明したプロトコル及びルールは、ここで説明する本開示の他の実施形態のうち1つ以上又は全てに関連し、これらの一部であると考えられる点に留意する。

【００５０】

異なる或いは特定の時間/インスタンスにおいて、他のノードに関する詳細もまた、特定のシャード内のノードの間で共有できる。これは、本開示の第2の実施形態に従って、addrプロトコルメッセージの修正バージョンを使用して実行される。ビットコインプロトコルの一部として現在存在するaddrメッセージの実装は、1つ以上のIPアドレス及びポートを列挙又は識別するために使用される。例えば、getaddr要求は、(例えば、ブートストラッピングのために)一束の既知のアクティブなピアを含むaddrメッセージを取得するために使用されてもよい。addrメッセージは、1つのアドレスのみをしばしば含むが、場合によ

っては更に多くを含み、いくつかの例では、1000個までを含む。いくつかの例では、全てのノードは、周期的に、すなわち、24時間毎に、自分のIPアドレスを含むaddrをブロードキャストする。次いで、ノードは、これらのメッセージを自分のピアに中継してもよく、新たなものである場合、中継されたアドレスを記憶できる。このように、ネットワーク内のノードは、ネットワークに接続した時点或いはその後に、どのIPがネットワークに接続されているかを合理的に明確に把握してもよい。ほとんどの場合、IPアドレスは、初期のaddrブロードキャストのため、全員のアドレスデータベースに追加される。

【００５１】

本開示による修正addrプロトコルの実装は、上記に加えて、特定のノードがどのシャードに属するかに関する更なる情報を送信できる。例えば、修正addrプロトコルでは、図１ｂのようなシャードされたネットワーク内のノードがネットワーク内の特定のシャードに参加するとき、addrメッセージの一部としてブロードキャストされるものはまた、それがメンバーである1つ以上のシャードを識別するフィールドを含んでもよい。したがって、この情報はまた、図１ｂのネットワーク内のピアからのgetaddr要求に応じて返信される。第1の実施形態で説明したように、このような情報は、各ノード及び/又はノードが関連する各シャードに関連するデータ構造に基づいてもよい。いくつかの実施形態では、修正addrプロトコルはまた、ノードがメンバーであるシャードの状態及び/又はノード自体の状態を含んでもよい。例えば、各メンバーのシャード内のノードの数の詳細が識別されてもよく、或いは、特定のシャードがアクティブであるか否か又は所与のシャード内のアクティブなノードの数も識別されてもよい。

【００５２】

［シャードされたネットワークにおけるシャードへのトランザクションの割り当て］

上記のように、シャードされたブロックチェーンネットワークでは、トランザクションは全てのノードにより承認されて記憶されるのではなく、その代わりに、1つ以上の特定のシャードに割り当てられる。したがって、トランザクションを異なるシャードに割り当てるための方策が必要である。以下に、2つの可能な実施形態について説明し、これらは、本開示の第3の実施形態による「トランザクションIDベース」のシャード及び本開示の第4の実施形態による「インプットベース」のシャードと呼ばれる。

【００５３】

いくつかの実装では、既存のビットコインプロトコルがいずれかの方式を開始するために分岐を受ける可能性がある。所与のシャード内のノードがその後にトランザクションを受信したとき、ノードは、正しいシャードに送信されていることをチェックしてもよい。この手法は、シャードの間のトランザクションのバランスを提供する。

【００５４】

いずれかのシャーディング方法は、ブロックチェーンに遡及して、また、いずれかの程度まで適用されてもよい。すなわち、いずれかの方法は、シャードされたネットワークがブロックチェーン内の最初のブロック(ビットコインブロックチェーンの場合は、いわゆるジェネシスブロック)の時点から将来の任意に選択されたブロック番号に至るまで存在すると定義されるように適用されてもよい。

【００５５】

以下に説明するシャーディング方法は、順に、また、何れかの順序で、複数回適用されてもよい。例えば、最初にトランザクションIDシャーディングが実行されてもよく、インプットベースのシャーディングが後日実行されてもよい。さらに、上記のように、いずれかの方法が遡及して適用されてもよく、さらに、いずれかの方法がその後に適用されてもよい。シャードの数nは、シャード方法が適用される毎に選択されてもよく、ノードの数を増加することによりプロトコルをスケーリングすることを可能にする。ノードの数は、ネットワーク上の全ノードの数、ブロックチェーンのサイズ及び/又は他の特性に基づいて選択されてもよい。以下に説明するシャーディング方法の双方について、シャーディングが行われたときにトランザクションが各ノードにより記憶される方式についても説明する。

【００５６】

［トランザクションIDベースのシャード分配］
水平分割のブロックチェーンでは、各シャードはネットワーク上の全てのトランザクションを含まず処理しないので、トランザクションを異なるシャードに割り当てるための方策が必要となる。さらに、いずれかのシャーディング方法は、更なるシャーディングを実行可能である必要がある。本開示の第3の実施形態では、図３を用いて説明するように、シャードの間のトランザクション分配は、トランザクションID(txid)に基づいて処理される。

【００５７】
ステップ302において、所与のトランザクションについてのトランザクションIDが作成され、txidとして示される。いくつかの実装では、このtxidは、SHA256関数をトランザクションデータに適用した結果として取得される。

【００５８】
ステップ304において、このトランザクションIDを使用して、txid及びシャードされたネットワーク内の利用可能なシャードの数に基づいて動作が実行される。いくつかの実装では、ブロックチェーンネットワーク上で現在アクティブなシャードの数のモジュロがトランザクションIDに適用され、すなわち、シャード番号=txid　mod　nであり、nは(所与の或いはアクティブな)シャードの数である。

【００５９】
ステップ306において、ステップ304の結果は、所与のトランザクションが割り当てられるシャードに対応する。

【００６０】
ステップ308において、ステップ306において割り当てられると、トランザクションは、識別されたシャードに分配され、すなわち、トランザクションは、ステップ306において識別されたシャードに含まれるノードに分配される。

【００６１】
したがって、所与のシャード内のノードがトランザクションを受信したとき、ノードは、正しいシャードに送信されていることを容易にチェックできる。いくつかの実施形態では、このようなチェックは、第1の実施形態において上記に説明したように、ノードに関連する情報を含む、各ノードに関連するデータ構造に基づいて実現されてもよい。有利には、この手法は、シャードの間のトランザクションの均等なバランスを提供する。

【００６２】
ネットワーク上のシャードカウントは、
・ネットワーク上の全ノードの数、及び
・ブロックチェーンのサイズ
を含む多数のパラメータに基づいて任意に選択できる。

【００６３】
［インプットベースのシャード分配］
本開示の第4の実施形態によるシャーディング方法について、図４のフロー図を用いて説明する。

【００６４】
この実施形態では、ステップ402において、所与のトランザクションのインプットが識別される。いくつかの実装では、これは、トランザクションの最初のインプットである。

【００６５】
ステップ404において、ステップ402のインプットが参照する前のトランザクションのアウトプットが識別される。

【００６６】
ステップ406において、ステップ402及び404の結果、すなわち、対応するインプットと、前のトランザクションからのアウトプットとの双方が、図１ｂに見られるようなシャードされたネットワーク内の同じシャードに割り当てられる。いくつかの実装では、このステップは、双方のトランザクションを割り当てるためのシャードを識別することを含む

。一例では、これは、既に割り当てられている場合、前のトランザクションに関連するシャードでもよい。他の例では、上記のように、所与のトランザクション又は前のトランザクションのいずれかについての修正addrブロードキャスト又はgetaddr要求に対する応答は、シャードを識別するために使用されてもよい。他の例では、シャードは、双方のトランザクションが同じシャードに割り当てられる限り、双方のトランザクションに対してランダムに或いは指定の通り、すなわち、ローテーションに基づいて選択されてもよい。これは、例えば、親トランザクションが識別されない場合、すなわち、受信するものがコインベース又は最初のトランザクションである場合に適用されてもよい。

【００６７】

ステップ408は、ステップ402～406における上記のプロセスが、第1のインプットにより連結されたトランザクションのチェーンを生成するように繰り返されることを示す。

【００６８】

後続のトランザクションの第1のインプットによりアウトプットが参照されるトランザクションは、この文脈では「親」トランザクションと呼ばれ、後続のトランザクションは「子」トランザクションと呼ばれる。

【００６９】

ステップ402において親を定義するための第1のインプットの使用は、当該方法に必須ではなく、複数のインプットが所与のトランザクションに存在する場合、当該方法を実行するためにいずれかのインプットが選択されてもよい点に留意すべきである。例えば、前のトランザクションは、子トランザクションの特定の数のインプットが親と同じシャード内のアウトプットを参照する場合、子トランザクションの親として定義されてもよい。インプットの数は、子トランザクションのインプットの過半数と定義してもよい。

【００７０】

第3及び第4の実施形態の上記の2つのシャーディング方法は、いずれかの順序で順に実行されてもよく、2つの方法は、所望に応じて複数回実行されてもよい点に留意すべきである。例えば、ブロックチェーンネットワークは、第4の実施形態のインプットベースの分配に従って分岐されてもよく、その後に、結果の分岐のうち1つ以上は、第3の実施形態のトランザクションIDベースの分配に従ってシャードされてもよい。したがって、ステップ406において上記に説明したいくつかの実装では、割り当てられるシャードは、インプットの数又は考慮されるべき実際の第1のインプット若しくは他の指定の特定のインプットのいずれかに基づいて、識別された親のものと同じになる。

【００７１】

［UTXOセット/メモリプールのシャーディング］

現在、ビットコインネットワークでは、全てのノードが自分のUTXOセットを維持管理し、これは、承認中にチェック及び更新される。UTXOセットの例が図５に示されている。

【００７２】

本開示の第5の実施形態によれば、シャードされたブロックチェーン(図１ｂに示す)において、1つ以上のシャードの各メンバーノードは、そのノードがメンバーである各シャードに関連するトランザクションに関するUTXOセットを有する。これは図６に更に示されており、図６は、1つより多くのシャードのメンバーであるノードを示す。これらは、この図では、重複する区別して網掛けされた領域により見られる。いくつかの実装では、第5の実施形態に関して、このようなUTXOセット(以下、シャードされたUTXOと呼ぶ)は、ここに説明する本開示の他の実施形態の1つ以上又は全ての一部に関し、これらの一部であると考えられてもよいことが理解される。

【００７３】

［トランザクション承認］

承認するトランザクションについて、ビットコインネットワークにおいてUTXOセットがチェックされて更新される必要がある。本開示は、UTXOセットがシャードされるとき、シャードされたブロックチェーンの承認を実装するためのこのプロセスの新たなバージョンを提供する。上記のように、図１ｂのようなシャードされたブロックチェーン上の各

ノードは、メンバーであるシャードの情報を含む、ネットワーク上のノードのリストに関連付けられるか或いはリストを維持管理する。これは、第1の実施形態に関連して上記に説明されている。

【００７４】

本開示によるトランザクション承認について、UTXOセットをチェックするために使用できる2つの方法について以下に説明する。これらは、それぞれ、第6の実施形態によるトランザクションシャード承認及び第7の実施形態によるUTXOシャード承認と呼ばれる。

【００７５】

［トランザクションシャード承認］

第6の実施形態では、トランザクション承認は、トランザクションが割り当てられたシャードにより実行される。第3の実施形態に関して上記に説明したように、トランザクションは、トランザクションIDに適用されるモジュロ関数の結果を使用してシャードに分配される。トランザクションは、異なるシャードからのインプットを有する可能性があるので、承認ノードは、UTXOチェックのために他のシャードと通信する。

【００７６】

図７ａを参照して、異なるシャード内のノードの間で実行されるUTXOセットのチェックについて説明する。このプロセスについて、図７ｂに関連しても説明する。

【００７７】

第6の実施形態によれば、シャード4内のノードは、UTXOを取得することが知られているシャード1内のノードに対して要求を行う。これは、ステップ702において見られる。シャード番号は例示のみのために指定されており、いずれかの所与のシャードに関連するいずれかの所与のノードがこの要求を実行してもよい。

【００７８】

次いで、ステップ704において、受信された応答の有効性が評価される。どのノードもUTXOを有さない場合、ヌル応答が与えられる。この場合、問題のトランザクションはステップ706において無効とみなされる。この場合、トランザクションの更なる伝搬は行われない。いくつかの場合、スクリプトエラー又はUTXOが利用できないといういずれかの指示が存在する場合にも、トランザクションは無効とみなされる。

【００７９】

所与のトランザクションのUTXOが受信された場合、ステップ708において、トランザクションインプットは有効であるとみなされる。背景技術の段落で説明したように、ノードにインストールされたソフトウェアクライアント又はプログラム又はアプリケーションが、そのロック及びロック解除スクリプトを実行することにより、UTXOに対してこの承認を実行してもよいことが知られている。いくつかの実装では、これはトランザクションの有効性データと呼ばれる。ロック及びロック解除スクリプトの実行がTRUEに評価された場合、トランザクションは有効であり、トランザクションはブロックチェーンに書き込まれる。さらに、上記のように、有効性チェックの1つは、二重支出が存在していないことをチェックすることである。いくつかの実装では、ノードがトランザクションを受信したとき、トランザクションがノードに関連付けられたデータ構造、又は関連するシャードのデータ構造で使用するUTXOを検索する。

【００８０】

次いで、ステップ710において、問題のトランザクションが、シャード4又はシャード4のメモリプール上のノードに追加される。

【００８１】

次いで、ステップ712において、トランザクションは、シャード4内の他のノードに伝搬される。

【００８２】

［UTXOシャードベースの承認］

第7の実施形態では、トランザクションは、所与のトランザクションのUTXOを含むシャード(図１ｂに見られるようなシャードされたネットワークのシャード)に伝搬される。

【００８３】
図８ａは、シャード4内のノードにより作成された支出トランザクション(Tx)が、そのトランザクションのUTXOを含むシャードのそれぞれに伝搬されることを示す。この実施形態では、ノードはトランザクションをシャード1及び2の双方に送信する。当該プロセスを図８ｂに更に示す。
【００８４】
ステップ802において、シャード内のノードが所与のトランザクションを受信したとき、この実施形態では、同じシャード内にあるインプットに基づいてトランザクションを承認するように進む。
【００８５】
したがって、ステップ804において、所与のトランザクションのインプットが同じシャードに関連するか否かがチェックされる。トランザクションは、第4の実施形態に従って上記のようにシャードに割り当てられてもよい。上記のように、ノードが1つより多くのシャードのメンバーである場合、このステップにおける「同じ」シャードのチェックは、このようなシャードのいずれかに適用される。
【００８６】
ステップ806bに見られるように、異なるシャード内のUTXOに関連するインプットは承認されない。いくつかの実装では、各インプットの承認は、承認がビットコインネットワークで現在実行され得るものとほとんど同じ方法で実行されてもよい。そうでない場合、ステップ806aにおいて、ノードはトランザクションを承認するように進む。
【００８７】
ステップ806aに続いて、ステップ808において、所与のトランザクションに関連するインプットの有効性がチェックされる。上記のように、また、図７ｂのステップ706及び708においても、未使用トランザクション(UTXO)に対する承認は、そのロック及びロック解除スクリプトを実行することによるものでもよい。いくつかの実装では、これはトランザクションの有効性データと呼ばれる。ロック及びロック解除スクリプトの実行がTRUEに評価された場合、トランザクションは有効であり、トランザクションはブロックチェーンに書き込まれる。さらに、また、上記のように、有効性チェックの1つは、二重支出が存在していないことをチェックすることである。いくつかの実装では、ノードがトランザクションを受信したとき、トランザクションがノードに関連するデータ構造又はノードのシャードに関連するデータ構造において使用するUTXOを検索する。
【００８８】
UTXOが存在しない場合、値がUTXOよりも大きい場合又はスクリプトエラーが存在する場合のように、インプットが無効である場合、ステップ810bにおいて見られるように、所与のトランザクションが破棄される。この場合、所与のトランザクションは、同じシャード内の他のノードに伝搬されない。
【００８９】
インプットが有効であるとみなされた場合、ステップ810aにおいて、トランザクションが有効であると識別される。
【００９０】
次いで、ステップ812において、トランザクションがノードのメモリプールに追加される。
【００９１】
ステップ814において、トランザクションは、ノードに関連するシャード内の他のノードに伝搬される。
【００９２】
次に図９を参照すると、本開示の少なくとも1つの実施形態を実施するために使用され得る計算デバイス2600の簡略化したブロック図が例として提供される。様々な実施形態では、計算デバイス2600は、図１ｂに見られるようなシャードされたブロックチェーンネットワークの1つ以上のシャード内のノード又はノードの組み合わせ、及び/又は単独で

或いは1つ以上のこのようなノード又はシステムに通信可能に結合されたとき上記に図示及び説明するコンピュータ実施システム、方法又はプロトコルのいずれかを実装するために使用されてもよい。

【００９３】

例えば、計算デバイス2600は、データサーバ、ウェブサーバ、ポータブル計算デバイス、パーソナルコンピュータ又はいずれかの電子計算デバイスとして使用するために構成されてもよい。図９に示すように、計算デバイス2600は、1つ以上のレベルのキャッシュメモリを有する1つ以上のプロセッサと、メインメモリ2608及び永続ストレージ2610を含む記憶サブシステム2606と通信するように構成できるメモリコントローラ(併せて2602とラベル付けされる)とを含んでもよい。メインメモリ2608は、図示のように、ダイナミックランダムアクセスメモリ(DRAM)2618及び読み取り専用メモリ(ROM)2620を含んでもよい。記憶サブシステム2606及びキャッシュメモリ2602は、本開示に記載のトランザクション及びブロックに関連する詳細のような情報の記憶のために使用されてもよい。プロセッサ2602は、本開示に記載のいずれかの実施形態のステップ又は機能を提供するために利用されてもよい。

【００９４】

プロセッサ2602はまた、1つ以上のユーザインタフェース入力デバイス2612、1つ以上のユーザインタフェース出力デバイス2614及びネットワークインタフェースサブシステム2616と通信できる。

【００９５】

バスサブシステム2604は、計算デバイス2600の様々な構成要素及びサブシステムが、意図したように相互に通信することを可能にする機構を提供してもよい。バスサブシステム2604は、単一のバスとして概略的に示されているが、バスサブシステムの代替実施形態は、複数のバスを利用してもよい。

【００９６】

ネットワークインタフェースサブシステム2616は、他の計算デバイス及びネットワークへのインタフェースを提供してもよい。ネットワークインタフェースサブシステム2616は、他のシステムからデータを受信し、計算デバイス2600から他のシステムにデータを送信するためのインタフェースとして機能してもよい。例えば、ネットワークインタフェースサブシステム2616は、データ技術者が、データセンタのような遠隔地にいる間に、データをデバイスに送信し、デバイスからデータを受信することができるように、デバイスをネットワークに接続することを可能にしてもよい。

【００９７】

ユーザインタフェース入力デバイス2612は、キーボードのような1つ以上のユーザ入力デバイスと、統合マウス、トラックボール、タッチパッド又はグラフィックスタブレットのようなポインティングデバイスと、スキャナと、バーコードスキャナと、ディスプレイに組み込まれたタッチスクリーンと、音声認識システム、マイクロフォンのようなオーディオ入力デバイスと、他のタイプの入力デバイスとを含んでもよい。一般的に、「入力デバイス」という用語の使用は、情報を計算デバイス2600に入力するための全ての可能な種類のデバイス及び機構を含むことを意図する。

【００９８】

1つ以上のユーザインタフェース出力デバイス2614は、表示サブシステム、プリンタ、又はオーディオ出力デバイスのような非視覚的ディスプレイを含んでもよい。表示サブシステムは、陰極線管(CRT)、液晶ディスプレイ(LCD)、発光ダイオード(LED)ディスプレイのようなフラットパネルデバイス、又は投影若しくは他の表示デバイスでもよい。一般的に、「出力デバイス」という用語の使用は、計算デバイス2600から情報を出力するための全ての可能な種類のデバイス及び機構を含むことを意図する。1つ以上のユーザインタフェース出力デバイス2614は、例えば、ユーザ相互作用が適切になり得るときに、ここに記載のプロセス及びその変形を実行するアプリケーションとのユーザ相互作用を実現するように、ユーザインタフェースを提示するために使用されてもよい。

【００９９】
記憶サブシステム2606は、本開示の少なくとも1つの実施形態の機能を提供し得る基本的なプログラミング及びデータ構成を記憶するためのコンピュータ読み取り可能記憶媒体を提供してもよい。アプリケーション(プログラム、コードモジュール、命令)は、1つ以上のプロセッサにより実行されたとき、本開示の1つ以上の実施形態の機能を提供してもよく、記憶サブシステム2606に記憶されてもよい。これらのアプリケーションモジュール又は命令は、1つ以上のプロセッサ2602により実行されてもよい。記憶サブシステム2606は、本開示に従って使用されるデータを記憶するためのリポジトリを更に提供してもよい。例えば、メインメモリ2608及びキャッシュメモリ2602は、プログラム及びデータのための揮発性ストレージを提供できる。永続ストレージ2610は、プログラム及びデータのための永続的な(不揮発性)ストレージを提供でき、フラッシュメモリと、1つ以上のソリッドステートドライブと、1つ以上の磁気ハードディスクドライブと、関連する取り外し可能媒体を有する1つ以上のフロッピーディスクドライブと、関連する取り外し可能媒体を有する1つ以上の光ドライブ(例えば、CD-ROM又はDVD又はブルーレイ)ドライブと、他の同様の記憶媒体とを含んでもよい。このようなプログラム及びデータは、本開示に記載の1つ以上の実施形態のステップを実行するためのプログラムと、本開示に記載のトランザクション及びブロックに関連するデータとを含むことができる。

【０１００】
計算デバイス2600は、ポータブルコンピュータデバイス、タブレットコンピュータ、ワークステーション、又は以下に記載のいずれかの他のデバイスを含む、様々な種類のものでもよい。さらに、計算デバイス2600は、1つ以上のポート(例えば、USB、ヘッドフォンジャック、ライトニングコネクタ等)を通じて計算デバイス2600に接続され得る他のデバイスを含んでもよい。計算デバイス2600に接続され得るデバイスは、光ファイバコネクタを受け入れるように構成された複数のポートを含んでもよい。したがって、このデバイスは、光信号を電気信号に変換するように構成されてもよく、電気信号は、処理のために当該デバイスを計算デバイス2600に接続するポートを通じて送信されてもよい。コンピュータ及びネットワークの絶えず変化する性質のため、図９に示す例示的な計算デバイス2600の説明は、デバイスの好ましい実施形態を説明する目的のための特定の例としてのみ意図されている。図９に示すシステムよりも多くの構成要素又は少ない構成要素を有する多くの他の構成が可能である。

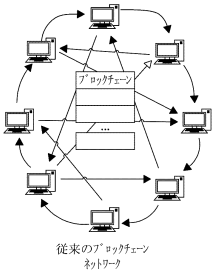
【０１０１】
上記の実施形態は、本発明を限定するものではなく例示するものであり、当業者は、添付の特許請求の範囲により定義される本開示の範囲から逸脱することなく、多くの代替実施形態を設計できる点に留意すべきである。特許請求の範囲において、括弧内に付したいずれかの参照符号は、特許請求の範囲を限定するものと解釈されないものとする。「含む(comprising)」及び「含む(comprises)」等の用語は、全体としていずれかの請求項又は明細書に列挙されたもの以外の要素又はステップの存在を除外しない。本明細書において、「含む(comprises)」とは、「含む(includes)又はからなる(consists of)」を意味し、「含む(comprising)」とは、「含む(comprising)又はからなる(consisting of)」を意味する。要素の単数形の参照は、このような要素の複数形の参照を除外するものではなく、逆も同様である。本発明は、いくつかの別個の要素を含むハードウェアにより、且つ、適切にプログラムされたコンピュータにより実装されてもよい。いくつかの手段を列挙するデバイスの請求項において、これらの手段のいくつかは、1つの同じハードウェアアイテムにより具体化されてもよい。特定の手段が相互に異なる従属項に記載されているという単なる事実は、これらの手段の組み合わせが有利に利用できないことを示すものではない。

【０１０２】
上記の説明は、例示的であり限定的ではないことを意図するものであることが理解されるべきである。多くの他の実装は、上記の説明を読んで理解することにより、当業者に明らかになる。本開示は、特定の例示的な実装を参照して記載されているが、本開示は、記

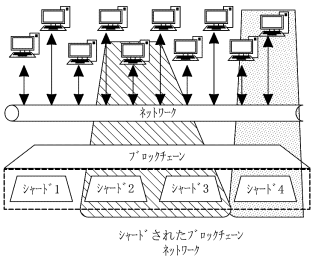
載の実装に限定されず、添付の特許請求の範囲の範囲内の修正及び変更を用いて実施可能であることが認識される。したがって、明細書及び図面は、限定的な意味ではなく、例示的な意味で考えられるべきである。したがって、本開示の範囲は、添付の特許請求の範囲が権利を有する均等物の全範囲と共に、添付の特許請求の範囲を参照して決定されるべきである。

【図面】

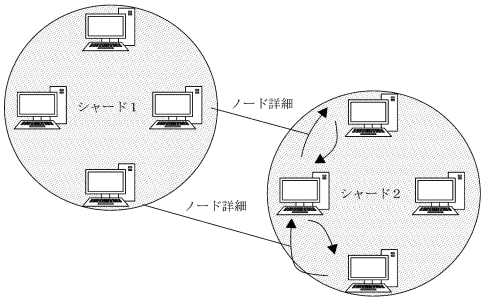
【図１ａ】



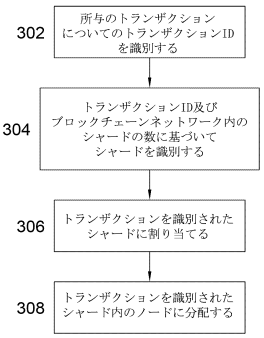
【図１ｂ】



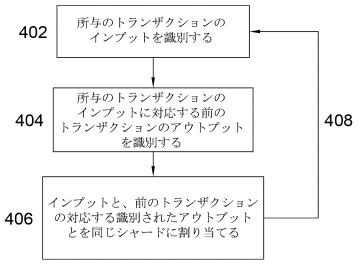
【図２】



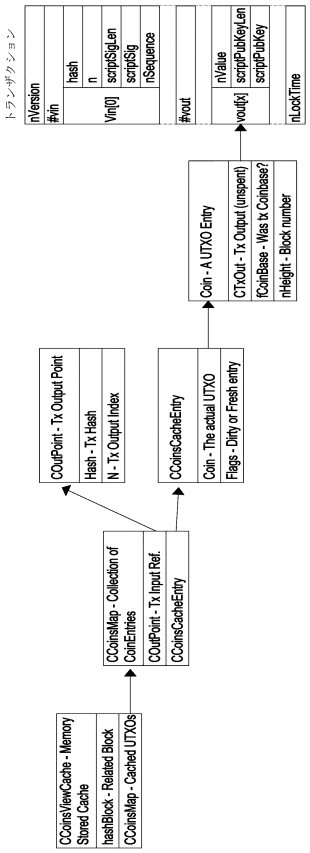
【図３】



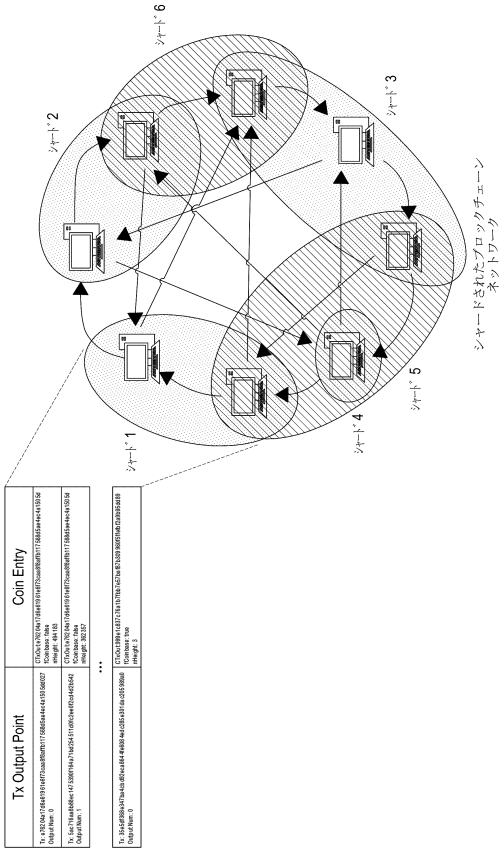
【図４】


402
所与のトランザクションの
インプットを識別する
404
所与のトランザクションの
インプットに対応する前の
トランザクションのアウトプット
を識別する
406
インプットと、前のトランザクション
の対応する識別されたアウトプット
とを同じシャードに割り当てる
408

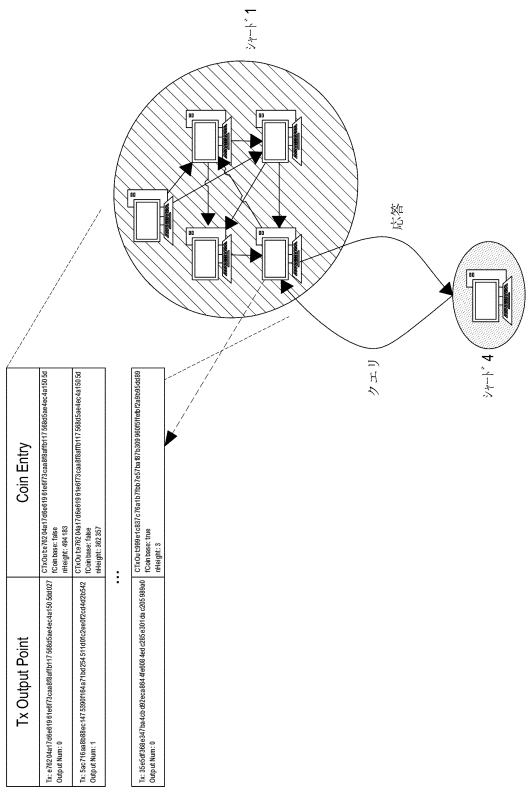
【図５】


トランザクション
nVersion
#vin
hash
n
Vin[]
scriptSigLen
scriptSig
nSequence
#vout
nValue
vout[x]
scriptPubKeyLen
scriptPubKey
nLockTime
Coin - A UTXO Entry
CTxOut - Tx Output (unspent)
fCoinBase - Was tx Coinbase?
nHeight - Block number
COutPoint - Tx Output Point
Hash - Tx Hash
N - Tx Output Index
CCoinsCacheEntry
Coin - The actual UTXO
Flags - Dirty or Fresh entry
CCoinsMap - Collection of CoinEntries
COutPoint - Tx Input Ref.
CCoinsCacheEntry
CCoinsViewCache - Memory Stored Cache
hashBlock - Related Block
CCoinsMap - Cached UTXOs

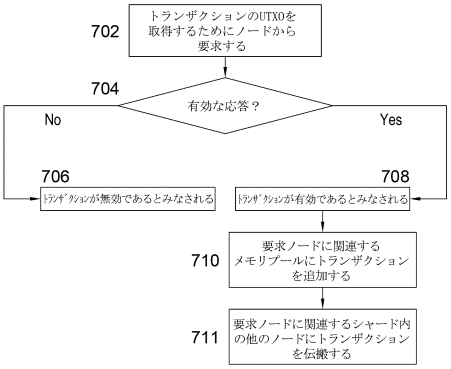
【図６】


シャード 6
シャード 2
シャード 3
シャード 1
シャード 4
シャード 5
シャードされたブロックチェーン
ネットワーク
Tx Output Point
Coin Entry

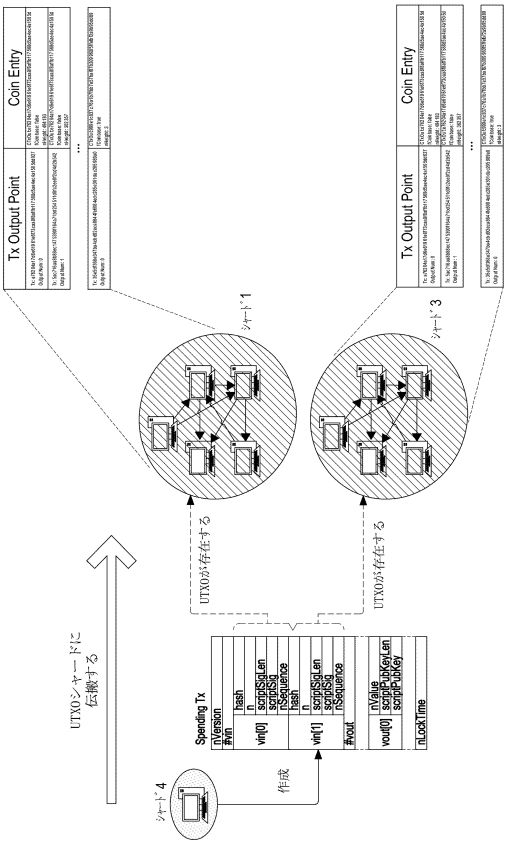
【図７ａ】


シャード 1
応答
クエリ
シャード 4
Tx Output Point
Coin Entry

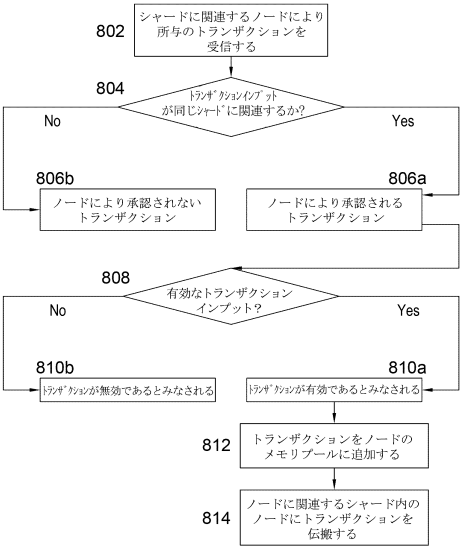
【図７ｂ】



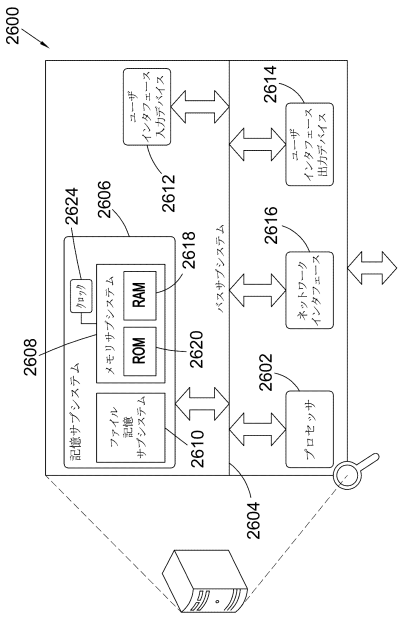
【図８ａ】



【図８ｂ】



【図９】



フロントページの続き

(33)優先権主張国・地域又は機関
英国(GB)
(31)優先権主張番号 1806911.2
(32)優先日 平成30年4月27日(2018.4.27)
(33)優先権主張国・地域又は機関
英国(GB)
(31)優先権主張番号 1806914.6
(32)優先日 平成30年4月27日(2018.4.27)
(33)優先権主張国・地域又は機関
英国(GB)
(31)優先権主張番号 1806930.2
(32)優先日 平成30年4月27日(2018.4.27)
(33)優先権主張国・地域又は機関
英国(GB)
内
(72)発明者 シーウェル，マーティン
イギリス国 シーエフ１０ ２エイチエイチ カーディフ チャーチル ウェイ チャーチル ハウス ７ス フロア アーカート－ダイクス アンド ロード エルエルピー 内
(72)発明者 アマール，バセム
イギリス国 シーエフ１０ ２エイチエイチ カーディフ チャーチル ウェイ チャーチル ハウス ７ス フロア アーカート－ダイクス アンド ロード エルエルピー 内
審査官 行田 悦資
(56)参考文献 特開２０１８－０６７１０８（ＪＰ，Ａ）
中国特許出願公開第１０７０１８１２５（ＣＮ，Ａ）
米国特許出願公開第２０１８／００１９８６７（ＵＳ，Ａ１）
特表２０２０－５２１２５２（ＪＰ，Ａ）
特許第６７４５００４（ＪＰ，Ｂ１）
KOKORIS-KOGIAS, E. et al，OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding，Cryptology ePrintArchive，2018年02月21日，pp.1-16，[online], [令和5年4月3日検索], インターネット〈https://eprint.iacr.org/2017/406〉
LUU, L. et al，A Secure Sharding Protocol For Open Blockchains，CCS '16: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security，2016年10月，pp.17-30，〈doi:10.1145/2976749.2978389〉
DANEZIS, G. et al，Centrally Banked Cryptocurrencies，Cryptology ePrint Archive，2015年12月18日，pp.1-14，[online], [令和5年4月3日検索], インターネット〈https://eprint.iacr.org/2015/502〉
AL-BASSAM, M. et al.，Chainspace: A Sharded Smart Contracts Platform，arXiv:1708.03778v1，2017年08月12日，pp.1-16，〈doi:10.48550/arXiv.1708.03778〉
加嵜 長門 ほか，ブロックチェーンアプリケーション開発の教科書，第1版，株式会社マイナビ出版，2018年01月31日，pp.294-296
(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｈ０４Ｌ ９／３２