

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6726367号

(P6726367)

(45) 発行日 令和2年7月22日(2020.7.22)

(24) 登録日 令和2年6月30日(2020.6.30)

(51) Int.Cl.

F I

G06F 11/18 (2006.01)

G06F 11/18 630

H04L 9/32 (2006.01)

H04L 9/00 675Z

G06F 21/64 (2013.01)

G06F 21/64

請求項の数 27 (全 58 頁)

(21) 出願番号 特願2019-543000 (P2019-543000)
 (86) (22) 出願日 平成30年12月13日(2018.12.13)
 (65) 公表番号 特表2020-513170 (P2020-513170A)
 (43) 公表日 令和2年4月30日(2020.4.30)
 (86) 国際出願番号 PCT/CN2018/120873
 (87) 国際公開番号 WO2019/072296
 (87) 国際公開日 平成31年4月18日(2019.4.18)
 審査請求日 令和1年10月8日(2019.10.8)

早期審査対象出願

(73) 特許権者 510330264
 アリババ・グループ・ホールディング・リ
 ミテッド
 ALIBABA GROUP HOLDI
 NG LIMITED
 英国領、ケイマン諸島、グランド・ケイマ
 ン、ジョージ・タウン、ワン・キャピタル
 ・プレイス、フォース・フロア、ビー・オ
 ー、ボックス 847

(74) 代理人 100188558

弁理士 飯田 雅人

(74) 代理人 100205785

弁理士 ▲高▼橋 史生

最終頁に続く

(54) 【発明の名称】 分散システムにおける1次ノードの変更を行うこと

(57) 【特許請求の範囲】

【請求項1】

複数のネットワークノードを含むブロックチェーンネットワークにおける1次ノードの変更を行うコンピュータ実行方法であって、前記複数のネットワークノードが、前記1次ノードおよび1つまたは複数のバックアップノードを含み、前記方法が、

エポック変更が行われる必要があるとバックアップノードによって判断するステップであって、前記エポック変更が、現在の1次ノードによる現在のエポックから、新しい1次ノードによる新しいエポックへの変更を引き起こし、前記現在のエポックが、前記1次ノードを使用して、前記複数のネットワークノード間の合意を入手するためのコンセンサス処理を含み、前記コンセンサス処理が、3つのフェーズを含む、ステップと、

10

前記現在のエポックにおける前記コンセンサス処理の前記3つのフェーズのそれぞれと関連付けられた前記バックアップノードの個々の重みを前記バックアップノードによって判断するステップであって、前記重みが、前記新しい1次ノードになるための前記バックアップノードの資格の測定基準である、ステップと、

前記現在のエポックにおける前記3つのフェーズのそれぞれと関連付けられた前記バックアップノードの前記個々の重みに基づいて、前記バックアップノードに対する加重和を前記バックアップノードによって判断するステップと、

前記加重和が第1の所定の閾値に達したとの判断に応答して、前記バックアップノードによって、前記ネットワークノード以外の前記複数のネットワークノードにEPOCH_CHANGEメッセージを送るステップであって、前記EPOCH_CHANGEメッセージが、前記現在の1次ノ

20

ードによる前記現在のエポックから、前記バックアップノードが前記新しい1次ノードである前記新しいエポックへの変更のための要求を示し、前記EPOCH_CHANGEメッセージが、前記バックアップノードの前記加重和を含む、ステップと、

前記バックアップノードによって、前記バックアップノード以外の前記複数のネットワークノードのうちの少なくとも1つから少なくとも1つのNEW_EPOCHメッセージを受け取るステップであって、前記NEW_EPOCHメッセージが、前記バックアップノードが前記新しい1次ノードになることについての承認を示す、ステップと、

前記少なくとも1つのNEW_EPOCHメッセージが有効であるかどうかを前記バックアップノードによって検証するステップと、

前記少なくとも1つのNEW_EPOCHメッセージからの有効なNEW_EPOCHメッセージの数が第2の所定の閾値を超過するかどうかを前記バックアップノードによって判断するステップと、

10

有効なNEW_EPOCHメッセージの前記数が前記第2の所定の閾値を超過するとの判断に応答して、前記バックアップノードが前記新しいエポックにおける前記新しい1次ノードであると前記バックアップノードによって判断するステップとを含む、

前記現在のエポックにおける前記コンセンサス処理の前記3つのフェーズのそれぞれと関連付けられた前記バックアップノードの個々の重みを判断するステップが、

前記現在のエポックにおける前記コンセンサス処理の第3のフェーズにおいて定足数検証に失敗したとの判断に応答して、前記コンセンサス処理の前記第3のフェーズに対する前記バックアップノードの重みが第3の値であると判断するステップと、

20

前記現在のエポックにおける前記コンセンサス処理の前記第3のフェーズにおいて定足数検証に成功したとの判断に応答して、前記コンセンサス処理の前記第3のフェーズに対する前記バックアップノードの前記重みが第4の値であると判断するステップであって、前記第3の値が前記第4の値より小さい、ステップとを含む、方法。

【請求項2】

前記現在のエポックにおける前記コンセンサス処理の前記3つのフェーズのそれぞれと関連付けられた前記バックアップノードの個々の重みを判断するステップが、前記コンセンサス処理の第1のフェーズに対する前記バックアップノードの重みが第1の値であると判断するステップを含む、請求項1に記載のコンピュータ実行方法。

30

【請求項3】

前記現在のエポックにおける前記コンセンサス処理の前記3つのフェーズのそれぞれと関連付けられた前記バックアップノードの個々の重みを判断するステップが、

前記現在のエポックにおける前記コンセンサス処理の第2のフェーズにおいて定足数検証に失敗したとの判断に応答して、前記コンセンサス処理の前記第2のフェーズに対する前記バックアップノードの重みが第1の値であると判断するステップと、

前記現在のエポックにおける前記コンセンサス処理の前記第2のフェーズにおいて定足数検証に成功したとの判断に応答して、前記コンセンサス処理の前記第2のフェーズに対する前記バックアップノードの前記重みが第2の値であると判断するステップであって、前記第1の値が前記第2の値より小さい、ステップとを含む、請求項1に記載のコンピュータ実行方法。

40

【請求項4】

前記ネットワークノードに対する前記第2のフェーズにおける前記定足数検証が、他のネットワークノードから所定の数のECHOメッセージを受け取るステップを含む、請求項3に記載のコンピュータ実行方法。

【請求項5】

前記ネットワークノードに対する前記第3のフェーズにおける前記定足数検証が、他のネットワークノードから所定の数の受け入れメッセージを受け取るステップを含み、他のネットワークノードからの前記受け入れメッセージのそれぞれが、前記他のネットワーク

50

ノードのそれぞれが所定の数のECHOメッセージを受け入れたことを示す、請求項1に記載のコンピュータ実行方法。

【請求項6】

前記EPOCH_CHANGEメッセージが、前記複数のネットワークノードからのネットワークノードのセットと関連付けられた署名のセットをさらに含み、前記NEW_EPOCHメッセージが、前記EPOCH_CHANGEメッセージのダイジェストを含む、請求項1に記載のコンピュータ実行方法。

【請求項7】

前記少なくとも1つの有効なNEW_EPOCHメッセージが有効であるかどうかを検証するステップが、前記少なくとも1つのNEW_EPOCHメッセージ内の前記EPOCH_CHANGEメッセージの前記ダイジェストが有効であるかどうかを検証するステップを含み、前記少なくとも1つのNEW_EPOCHメッセージ内の前記EPOCH_CHANGEメッセージの前記ダイジェストが有効であるかどうかを検証するステップが、前記EPOCH_CHANGEメッセージ内の署名の前記セットが有効であるかどうかを検証するステップを含む、請求項6に記載のコンピュータ実行方法。

10

【請求項8】

エポック変更が行われる必要があると判断するステップが、所定の期間内に古いエポックにおいて前記コンセンサスが実現されなかったとの判断に応答して、エポック変更が行われる必要があると判断するステップを含む、請求項1に記載のコンピュータ実行方法。

【請求項9】

前記新しい1次ノードによる前記新しいエポックにおいて動作させるステップをさらに含み、前記新しいエポックが、前記新しい1次ノードを使用して、前記複数のネットワークノード間のコンセンサスを実現するためのコンセンサス処理を含む、請求項1に記載のコンピュータ実行方法。

20

【請求項10】

複数のネットワークノードを含むブロックチェーンネットワークのバックアップノードによって、エポック変更が行われる必要があると判断することであって、前記複数のネットワークノードが、1次ノード、および前記バックアップノードを含む1つまたは複数のバックアップノードを含み、前記エポック変更が、現在の1次ノードによる現在のエポックから、新しい1次ノードによる新しいエポックへの変更を引き起こし、前記現在のエポックが、前記1次ノードを使用して、前記複数のネットワークノード間のコンセンサスを実現するためのコンセンサス処理を含む、前記コンセンサス処理が、3つのフェーズを含む、判断することと、

30

前記現在のエポックにおける前記コンセンサス処理の前記3つのフェーズのそれぞれと関連付けられた前記バックアップノードの個々の重みを前記バックアップノードによって判断することであって、前記重みが、前記新しい1次ノードになるための前記バックアップノードの資格の測定基準である、判断することと、

前記現在のエポックにおける前記3つのフェーズのそれぞれと関連付けられた前記バックアップノードの前記個々の重みに基づいて、前記バックアップノードに対する加重和を前記バックアップノードによって判断することと、

前記加重和が第1の所定の閾値に達したとの判断に応答して、前記バックアップノードによって、前記ネットワークノード以外の前記複数のネットワークノードにEPOCH_CHANGEメッセージを送ることであって、前記EPOCH_CHANGEメッセージが、前記現在の1次ノードによる前記現在のエポックから、前記バックアップノードが前記新しい1次ノードである前記新しいエポックへの変更のための要求を示し、前記EPOCH_CHANGEメッセージが、前記バックアップノードの前記加重和を含む、送ることと、

40

前記バックアップノードによって、前記バックアップノード以外の前記複数のネットワークノードのうちの少なくとも1つから少なくとも1つのNEW_EPOCHメッセージを受け取ることであって、前記NEW_EPOCHメッセージが、前記バックアップノードが前記新しい1次ノードになることについての承認を示す、受け取ることと、

前記少なくとも1つのNEW_EPOCHメッセージが有効であるかどうかを前記バックアップノ

50

ードによって検証することと、

前記少なくとも1つのNEW_EPOCHメッセージからの有効なNEW_EPOCHメッセージの数が第2の所定の閾値を超過するかどうかを前記バックアップノードによって判断することと、

有効なNEW_EPOCHメッセージの前記数が前記第2の所定の閾値を超過するとの判断に応答して、前記バックアップノードが前記新しいエポックにおける前記新しい1次ノードであると前記バックアップノードによって判断することと

を含み、

前記現在のエポックにおける前記コンセンサス処理の前記3つのフェーズのそれぞれと関連付けられた前記バックアップノードの個々の重みを判断することが、

前記現在のエポックにおける前記コンセンサス処理の第3のフェーズにおいて定足数検証に失敗したとの判断に応答して、前記コンセンサス処理の前記第3のフェーズに対する前記バックアップノードの重みが第3の値であると判断することと、

前記現在のエポックにおける前記コンセンサス処理の前記第3のフェーズにおいて定足数検証に成功したとの判断に応答して、前記コンセンサス処理の前記第3のフェーズに対する前記バックアップノードの前記重みが第4の値であると判断することであって、前記第3の値が前記第4の値より小さい、判断することと

を含む、動作を行うようにコンピュータシステムによって実行可能な1つまたは複数の命令を格納する、非一時的コンピュータ可読媒体。

【請求項 1 1】

前記現在のエポックにおける前記コンセンサス処理の前記3つのフェーズのそれぞれと関連付けられた前記バックアップノードの個々の重みを判断することが、前記コンセンサス処理の第1のフェーズに対する前記バックアップノードの重みが第1の値であると判断することを含む、請求項10に記載の非一時的コンピュータ可読媒体。

【請求項 1 2】

前記現在のエポックにおける前記コンセンサス処理の前記3つのフェーズのそれぞれと関連付けられた前記バックアップノードの個々の重みを判断することが、

前記現在のエポックにおける前記コンセンサス処理の第2のフェーズにおいて定足数検証に失敗したとの判断に応答して、前記コンセンサス処理の前記第2のフェーズに対する前記バックアップノードの重みが第1の値であると判断することと、

前記現在のエポックにおける前記コンセンサス処理の前記第2のフェーズにおいて定足数検証に成功したとの判断に応答して、前記コンセンサス処理の前記第2のフェーズに対する前記バックアップノードの前記重みが第2の値であると判断することであって、前記第1の値が前記第2の値より小さい、判断することと

を含む、請求項10に記載の非一時的コンピュータ可読媒体。

【請求項 1 3】

前記ネットワークノードに対する前記第2のフェーズにおける前記定足数検証が、他のネットワークノードから所定の数のECHOメッセージを受け取ることを含む、請求項12に記載の非一時的コンピュータ可読媒体。

【請求項 1 4】

前記ネットワークノードに対する前記第3のフェーズにおける前記定足数検証が、他のネットワークノードから所定の数の受け入れメッセージを受け取ることを含む、他のネットワークノードからの前記受け入れメッセージのそれぞれが、前記他のネットワークノードのそれぞれが所定の数のECHOメッセージを受け入れたことを示す、請求項10に記載の非一時的コンピュータ可読媒体。

【請求項 1 5】

前記EPOCH_CHANGEメッセージが、前記複数のネットワークノードからのネットワークノードのセットと関連付けられた署名のセットをさらに含み、前記NEW_EPOCHメッセージが、前記EPOCH_CHANGEメッセージのダイジェストを含む、請求項10に記載の非一時的コンピュータ可読媒体。

【請求項 1 6】

前記少なくとも1つの有効なNEW_EPOCHメッセージが有効であるかどうかを検証することが、前記少なくとも1つのNEW_EPOCHメッセージ内の前記EPOCH_CHANGEメッセージの前記ダイジェストが有効であるかどうかを検証することを含み、前記少なくとも1つのNEW_EPOCHメッセージ内の前記EPOCH_CHANGEメッセージの前記ダイジェストが有効であるかどうかを検証することが、前記EPOCH_CHANGEメッセージ内の署名の前記セットが有効であるかどうかを検証することを含む、請求項15に記載の非一時的コンピュータ可読媒体。

【請求項17】

エポック変更が行われる必要があると判断することが、所定の期間内に古いエポックにおいて前記コンセンサスが実現されなかったとの判断に応答して、エポック変更が行われる必要があると判断することを含む、請求項10に記載の非一時的コンピュータ可読媒体。

【請求項18】

前記動作が、前記新しい1次ノードによる前記新しいエポックにおいて動作させることであって、前記新しいエポックが、前記新しい1次ノードを使用して、前記複数のネットワークノード間のコンセンサスを実現するためのコンセンサス処理を含む、動作させることと
をさらに含む、請求項10に記載の非一時的コンピュータ可読媒体。

【請求項19】

1つまたは複数のコンピュータと、

前記1つまたは複数のコンピュータに相互利用可能に連結され、1つまたは複数のコンピュータによって実行されると、

複数のネットワークノードを含むブロックチェーンネットワークのバックアップノードによって、エポック変更が行われる必要があると判断することであって、前記複数のネットワークノードが、1次ノード、および前記バックアップノードを含む1つまたは複数のバックアップノードを含み、前記エポック変更が、現在の1次ノードによる現在のエポックから、新しい1次ノードによる新しいエポックへの変更を引き起こし、前記現在のエポックが、前記1次ノードを使用して、前記複数のネットワークノード間のコンセンサスを実現するためのコンセンサス処理を含み、前記コンセンサス処理が、3つのフェーズを含む、判断すること、

前記現在のエポックにおける前記コンセンサス処理の前記3つのフェーズのそれぞれと関連付けられた前記バックアップノードの個々の重みを前記バックアップノードによって判断することであって、前記重みが、前記新しい1次ノードになるための前記バックアップノードの資格の測定基準である、判断すること、

前記現在のエポックにおける前記3つのフェーズのそれぞれと関連付けられた前記バックアップノードの前記個々の重みに基づいて、前記バックアップノードに対する加重和を前記バックアップノードによって判断すること、

前記加重和が第1の所定の閾値に達したとの判断に応答して、前記バックアップノードによって、前記ネットワークノード以外の前記複数のネットワークノードにEPOCH_CHANGEメッセージを送ることであって、前記EPOCH_CHANGEメッセージが、前記現在の1次ノードによる前記現在のエポックから、前記バックアップノードが前記新しい1次ノードである前記新しいエポックへの変更のための要求を示し、前記EPOCH_CHANGEメッセージが、前記バックアップノードの前記加重和を含む、送ること、

前記バックアップノードによって、前記バックアップノード以外の前記複数のネットワークノードのうちの少なくとも1つから少なくとも1つのNEW_EPOCHメッセージを受け取ることであって、前記NEW_EPOCHメッセージが、前記バックアップノードが前記新しい1次ノードになることについての承認を示す、受け取ること、

前記少なくとも1つのNEW_EPOCHメッセージが有効であるかどうかを前記バックアップノードによって検証すること、

前記少なくとも1つのNEW_EPOCHメッセージからの有効なNEW_EPOCHメッセージの数が第2の所定の閾値を超過するかどうかを前記バックアップノードによって判断すること、ならびに

10

20

30

40

50

有効なNEW_EPOCHメッセージの前記数が前記第2の所定の閾値を超過するとの判断に
応じて、前記バックアップノードが前記新しいエポックにおける前記新しい1次ノードで
あると前記バックアップノードによって判断すること
を含み、

前記現在のエポックにおける前記コンセンサス処理の前記3つのフェーズのそれぞれ
と関連付けられた前記バックアップノードの個々の重みを判断することが、

前記現在のエポックにおける前記コンセンサス処理の第3のフェーズにおいて定足
数検証に失敗したとの判断に应诉して、前記コンセンサス処理の前記第3のフェーズに対
する前記バックアップノードの重みが第3の値であると判断することと、

前記現在のエポックにおける前記コンセンサス処理の前記第3のフェーズにおいて
定足数検証に成功したとの判断に应诉して、前記コンセンサス処理の前記第3のフェーズ
に対する前記バックアップノードの前記重みが第4の値であると判断することであって、
前記第3の値が前記第4の値より小さい、判断することと

を含む1つまたは複数の動作を行う1つまたは複数の命令を格納する有形の非一時的機械可
読媒体を有する1つまたは複数のコンピュータメモリデバイスと
を含む、システム。

【請求項 2 0】

前記現在のエポックにおける前記コンセンサス処理の前記3つのフェーズのそれぞれと
関連付けられた前記バックアップノードの個々の重みを判断することが、前記コンセンサ
ス処理の第1のフェーズに対する前記バックアップノードの重みが第1の値であると判断す
ることを含む、請求項19に記載のシステム。

【請求項 2 1】

前記現在のエポックにおける前記コンセンサス処理の前記3つのフェーズのそれぞれと
関連付けられた前記バックアップノードの個々の重みを判断することが、

前記現在のエポックにおける前記コンセンサス処理の第2のフェーズにおいて定足数検
証に失敗したとの判断に应诉して、前記コンセンサス処理の前記第2のフェーズに対す
る前記バックアップノードの重みが第1の値であると判断することと、

前記現在のエポックにおける前記コンセンサス処理の前記第2のフェーズにおいて定足
数検証に成功したとの判断に应诉して、前記コンセンサス処理の前記第2のフェーズに対
する前記バックアップノードの前記重みが第2の値であると判断することであって、前記
第1の値が前記第2の値より小さい、判断することと

を含む、請求項19に記載のシステム。

【請求項 2 2】

前記ネットワークノードに対する前記第2のフェーズにおける前記定足数検証が、他の
ネットワークノードから所定の数のECHOメッセージを受け取ることを含む、請求項21に記
載のシステム。

【請求項 2 3】

前記ネットワークノードに対する前記第3のフェーズにおける前記定足数検証が、他の
ネットワークノードから所定の数の受け入れメッセージを受け取ることを含む、他のネッ
トワークノードからの前記受け入れメッセージのそれぞれが、前記他のネットワークノー
ドのそれぞれが所定の数のECHOメッセージを受け入れたことを示す、請求項19に記載のシ
ステム。

【請求項 2 4】

前記EPOCH_CHANGEメッセージが、前記複数のネットワークノードからのネットワークノ
ードのセットと関連付けられた署名のセットをさらに含む、前記NEW_EPOCHメッセージが
、前記EPOCH_CHANGEメッセージのダイジェストを含む、請求項19に記載のシステム。

【請求項 2 5】

前記少なくとも1つの有効なNEW_EPOCHメッセージが有効であるかどうかを検証すること
が、前記少なくとも1つのNEW_EPOCHメッセージ内の前記EPOCH_CHANGEメッセージの前記ダ
イジェストが有効であるかどうかを検証することを含む、前記少なくとも1つのNEW_EPOCH

10

20

30

40

50

メッセージ内の前記EPOCH_CHANGEメッセージの前記ダイジェストが有効であるかどうかを検証することが、前記EPOCH_CHANGEメッセージ内の署名の前記セットが有効であるかどうかを検証することを含む、請求項24に記載のシステム。

【請求項26】

エポック変更が行われる必要があると判断することが、所定の期間内に古いエポックにおいて前記コンセンサスが実現されなかったとの判断に応答して、エポック変更が行われる必要があると判断することを含む、請求項19に記載のシステム。

【請求項27】

前記動作が、

前記新しい1次ノードによる前記新しいエポックにおいて動作させることであって、前記新しいエポックが、前記新しい1次ノードを使用して、前記複数のネットワークノード間のコンセンサスを実現するためのコンセンサス処理を含む、動作させることをさらに含む、請求項19に記載のシステム。

【発明の詳細な説明】

【技術分野】

【0001】

関連出願の相互参照

この出願は、2018年12月13日に提出されたPCT出願番号PCT/CN2018/120873の継続であり、その全体が参照によってここに組み込まれる。

【背景技術】

【0002】

分散台帳システム(DLS: Distributed ledger system)は、コンセンサスネットワークおよび/またはブロックチェーンネットワークと呼ばれることも可能であり、参加エンティティが安全かつ不変にデータを格納することを可能にする。DLSは、一般に、いずれかの特定のユーザ事例を参照することなく、ブロックチェーンネットワークと呼ばれる。ブロックチェーンネットワークの例は、公開ブロックチェーンネットワーク、プライベートブロックチェーンネットワーク、およびコンソーシアムブロックチェーンネットワークを含むことができる。公開ブロックチェーンネットワークは、全てのエンティティがDLSを使用してコンセンサス処理に参加することに対して開かれている。プライベートブロックチェーンネットワークは、読み書き許可を中心に制御する特定のエンティティに提供される。コンソーシアムブロックチェーンネットワークは、コンセンサス処理を制御するエンティティの選択グループに提供され、アクセス制御層を含む。

【0003】

コンセンサスメカニズムは、分散ブロックチェーンシステムの主要な構成要素である。コンセンサスメカニズムは、分散処理またはシステムの間の一データ値に対する合意を実現するために使用されるコンピュータサイエンスにおける処理である。コンセンサスメカニズムは、複数の信頼できないノードを含むネットワークにおける信頼性を実現するように設計される。コンセンサス問題として知られるこの問題を解決することは、分散コンピューティングおよびマルチエージェントシステムにおいて重要である。

【0004】

ブロックチェーンは、ノード間の合意に達するためのコンセンサスメカニズムに依存する。ブロックチェーンは、ピアツーピア(P2P)ネットワーク上の分散コンピュータによって管理される非集中型データベースである。各ピアは、単一障害点(SPOF)を防ぐために台帳のコピーを維持する。更新および有効化は、全てのコピーに同時に反映される。

ブロックチェーンシステムのネットワークノード間のコンセンサスを行うために、いくつかの既存の技法が使用されることが可能であるが、コンセンサスを行うためのより効率的な解決策が有利になるであろう。

【発明の概要】

【課題を解決するための手段】

【0005】

本明細書の実装形態は、分散システム(例えばブロックチェーンネットワーク)におけるコンセンサス問題を解決するためのコンピュータ実行方法を含む。より詳細には、本明細書の実装形態は、分散システムにおける1次ノードの変更を行うことを対象とする。

【0006】

いくつかの実装形態において、アクションは、エポック変更が行われる必要があるとブロックチェーンネットワークのバックアップノードによって判断することであって、エポック変更が、現在の1次ノードによる現在のエポックから、新しい1次ノードによる新しいエポックへの変更を引き起こし、現在のエポックが、1次ノードを使用して、いくつかのネットワークノード間のコンセンサスを実現するためのコンセンサス処理を含み、コンセンサス処理が、3つのフェーズを含む、判断することと、現在のエポックにおけるコンセンサス処理の3つのフェーズのそれぞれと関連付けられたバックアップノードの個々の重みをバックアップノードによって判断することと、現在のエポックにおける3つのフェーズのそれぞれと関連付けられたバックアップノードの個々の重みに基づいて、バックアップノードによってバックアップノードに対する加重和を判断することと、加重和が第1の所定の閾値に達したとの判断にตอบสนองして、バックアップノードによって、このネットワークノード以外のいくつかのネットワークノードにEPOCH_CHANGEメッセージを送ることであって、EPOCH_CHANGEメッセージが、現在の1次ノードによる現在のエポックから、バックアップノードが新しい1次ノードである新しいエポックへの変更のための要求を示し、EPOCH_CHANGEが、バックアップノードの加重和を含む、送ることと、バックアップノードによって、このバックアップノード以外のいくつかのネットワークノードのうちの少なくとも1つから少なくとも1つのNEW_EPOCHメッセージを受け取ることであって、NEW_EPOCHメッセージが、バックアップノードが新しい1次ノードになることについての承認を示す、受け取ることと、少なくとも1つのNEW_EPOCHメッセージが有効であるかどうかをバックアップノードによって検証することと、少なくとも1つのNEW_EPOCHメッセージからの有効なNEW_EPOCHメッセージの数が第2の所定の閾値を超過するかどうかをバックアップノードによって判断することと、有効なNEW_EPOCHメッセージの数が第2の所定の閾値を超過するとの判断にตอบสนองして、バックアップノードが新しいエポックにおける新しい1次ノードであるとバックアップノードによって判断することと、を含む。

【0007】

他の実装形態は、コンピュータストレージデバイス上でエンコードされた方法のアクションを行うように構成された、対応するシステム、装置、およびコンピュータプログラムを含む。

【0008】

これらおよび他の実装形態は、それぞれ任意選択で、以下の特徴の1つまたは複数を含む。

【0009】

以下の特徴のいずれかと組み合わせ可能であり、コンセンサス処理の第1のフェーズに対するバックアップノードの重みが第1の値であるとバックアップノードが判断する、第1の特徴。

【0010】

以下の特徴のいずれかと組み合わせ可能であり、現在のエポックにおけるコンセンサス処理の第2のフェーズにおいて定足数検証に失敗したとの判断にตอบสนองして、コンセンサス処理の第2のフェーズに対するバックアップノードの重みが第1の値であるとバックアップノードが判断し、現在のエポックにおけるコンセンサス処理の第2のフェーズにおいて定足数検証に成功したとの判断にตอบสนองして、コンセンサス処理の第2のフェーズに対するバックアップノードの重みが、第1の値より大きい第2の値であるとバックアップノードが判断する、第2の特徴。

【0011】

以下の特徴のいずれかと組み合わせ可能であり、ネットワークノードに対する第2のフェーズにおける定足数検証が、他のネットワークノードから所定の数のECHOメッセージを

受け取ることを含む、第3の特徴。

【0012】

以下の特徴のいずれかと組み合わせ可能であり、現在のエポックにおけるコンセンサス処理の第3のフェーズにおいて定足数検証に失敗したとの判断に応答して、コンセンサス処理の第3のフェーズに対するバックアップノードの重みが第3の値であるとバックアップノードが判断し、現在のエポックにおけるコンセンサス処理の第3のフェーズにおいて定足数検証に成功したとの判断に応答して、コンセンサス処理の第3のフェーズに対するバックアップノードの重みが、第3の値より大きい第4の値であるとバックアップノードが判断する、第4の特徴。

【0013】

以下の特徴のいずれかと組み合わせ可能であり、ネットワークノードに対する第3のフェーズにおける定足数検証が、他のネットワークノードから所定の数の受け入れメッセージを受け取ることを含む、他のネットワークノードからの受け入れメッセージのそれぞれが、他のネットワークノードのそれぞれが所定の数のECHOメッセージを受け入れたことを示す、第5の特徴。

【0014】

以下の特徴のいずれかと組み合わせ可能であり、EPOCH_CHANGEメッセージが、いくつかのネットワークノードからのネットワークノードのセットと関連付けられた署名のセットをさらに含み、NEW_EPOCHメッセージが、EPOCH_CHANGEメッセージのダイジェストを含む、第6の特徴。

【0015】

以下の特徴のいずれかと組み合わせ可能であり、少なくとも1つの有効なNEW_EPOCHメッセージが有効であるかどうかを検証することが、少なくとも1つのNEW_EPOCHメッセージ内のEPOCH_CHANGEメッセージのダイジェストが有効であるかどうかを検証することを含む、第7の特徴。

【0016】

以下の特徴のいずれかと組み合わせ可能であり、少なくとも1つのNEW_EPOCHメッセージ内のEPOCH_CHANGEメッセージのダイジェストが有効であるかどうかを検証することが、EPOCH_CHANGEメッセージ内の署名のセットが有効であるかどうかを検証することを含む、第8の特徴。

【0017】

以下の特徴のいずれかと組み合わせ可能であり、所定の期間内に古いエポックにおいてコンセンサスが実現されなかったとの判断に応答して、エポック変更が行われる必要があるとバックアップノードが判断する、第9の特徴。

【0018】

以下の特徴のいずれかと組み合わせ可能であり、新しいエポックが、新しい1次ノードを使用して、いくつかのネットワークノード間のコンセンサスを実現するためのコンセンサス処理を含む、第10の特徴。

【0019】

いくつかの実装形態において、アクションは、いくつかのネットワークノードのうちの1つのネットワークノードによって、このネットワークノード以外のバックアップノードからEPOCH_CHANGEメッセージを受け取ることであって、EPOCH_CHANGEメッセージが、エポック変更が行われる必要があるという指示を含み、エポック変更が、現在の1次ノードによる現在のエポックから、新しい1次ノードによる新しいエポックへの変更を引き起こす、受け取ることと、EPOCH_CHANGEメッセージが有効であるかどうかをネットワークノードによって検証することと、EPOCH_CHANGEメッセージが有効であることの検証に応答して、他のネットワークノードにNEW_EPOCHメッセージをネットワークノードによって送ることと、NEW_EPOCHメッセージが、EPOCH_CHANGEメッセージのダイジェストを含む、送ることと、ネットワークノードによって、このネットワークノード以外のいくつかのネットワークノードのうちの少なくとも1つから少なくとも1つのNEW_EPOCHメッセージを受け

10

20

30

40

50

取ることと、少なくとも1つのNEW_EPOCHメッセージが有効であるかどうかをネットワークノードによって検証することと、少なくとも1つのNEW_EPOCHメッセージからの有効なNEW_EPOCHメッセージの数が所定の閾値を超過するかどうかをバックアップノードによって判断することと、有効なNEW_EPOCHメッセージの数が所定の閾値を超過するとの判断にตอบสนองして、バックアップノードが新しいエポックにおける新しい1次ノードであるとネットワークノードによって判断することと、を含む。

【0020】

他の実装形態は、コンピュータストレージデバイス上でエンコードされた方法のアクションを行うように構成された、対応するシステム、装置、およびコンピュータプログラムを含む。

10

【0021】

これらおよび他の実装形態は、それぞれ任意選択で、以下の特徴の1つまたは複数を含むことができる。

【0022】

以下の特徴のいずれかと組み合わせ可能であり、EPOCH_CHANGEメッセージが、バックアップノードと関連付けられた加重和、およびいくつかのネットワークノードからのネットワークノードのセットと関連付けられた署名のセットを含む、第1の特徴。

【0023】

以下の特徴のいずれかと組み合わせ可能であり、EPOCH_CHANGEメッセージが有効であるかどうかを検証することが、EPOCH_CHANGEメッセージ内の加重和が有効であるかどうかを検証することを含み、EPOCH_CHANGEメッセージ内の加重和が有効であるかどうかを検証することが、署名のセットが有効であるかどうかを検証することを含む、第2の特徴。

20

【0024】

以下の特徴のいずれかと組み合わせ可能であり、少なくとも1つのNEW_EPOCHメッセージが有効であるかどうかを検証することが、少なくとも1つのNEW_EPOCHメッセージ内のEPOCH_CHANGEメッセージのダイジェストが有効であるかどうかを検証することを含み、少なくとも1つのNEW_EPOCHメッセージ内のEPOCH_CHANGEメッセージのダイジェストが有効であるかどうかを検証することが、EPOCH_CHANGEメッセージ内の署名のセットが有効であるかどうかを検証することを含む、第3の特徴。

【0025】

本明細書は、1つまたは複数のプロセッサに連結され、1つまたは複数のプロセッサによって実行されると、本明細書で提供される方法の実装形態による動作を1つまたは複数のプロセッサに行わせる、1つまたは複数の非一時的コンピュータ可読ストレージ媒体に格納された命令を有する、1つまたは複数の非一時的コンピュータ可読ストレージ媒体も提供する。

30

【0026】

本明細書は、本明細書で提供される方法を実行するためのシステムをさらに提供する。システムは、1つまたは複数のプロセッサと、1つまたは複数のプロセッサによって実行されると、本明細書で提供される方法の実装形態による動作を1つまたは複数のプロセッサに行わせる、コンピュータ可読ストレージ媒体に格納された命令を有する、1つまたは複数のプロセッサに連結されたコンピュータ可読ストレージ媒体と、を含む。

40

【0027】

本明細書は、分散システムにおけるネットワークノード間のコンセンサスを実現するため、分散システムにおける1次ノードの変更を行うため、および分散システムにおけるネットワークノードに対する回復処理を行うため、の技法を含む改良されたコンセンサスメカニズムを開示する。説明されるコンセンサスメカニズムは、様々な用途において様々な長所を実現することができる。

【0028】

例えば、下記で論じられるようなコンセンサス処理は、ブロックチェーンシステムの動作を改善し、ネットワークのボトルネックを緩和させるのに役立つ多くの特徴を含む。例

50

えば、説明されるコンセンサス処理は、ECコードに従って取引要求をいくつかの消去コード(EC)ブロックにコンバートすること、およびネットワークノードのそれぞれにECブロックのうちの1つを送ること、を含む。ECブロックは、元の取引要求よりサイズが小さい。したがって、完全な取引要求の代わりにECブロックをネットワークノードに送ることは、ブロックチェーンネットワークのネットワークノード間で送信されるデータブロックのサイズを低減させ、このことにより、ネットワーク帯域幅を節約し、ネットワーク負荷を低減させる。これは、ネットワークノードのメモリ空間との間で読み書きされるデータのサイズをさらに低減させ、このことにより、ネットワークノードのメモリ空間に対する負担を低減させ、全体的なブロックチェーンシステムの効率性を改善する。

【0029】

10

さらに、本明細書は、コンセンサス処理の複数のフェーズに個々の重みを割り当てること、複数のフェーズの個々の重みに基づいて加重和を判断すること、および加重和に基づいて新しい1次ノードを判断することを含むエポック変更処理を説明する。ラウンドロビン方法の代わりに加重和に基づくエポック変更処理は、故障していない新しい1次ノードをタイムリに選ぶことを容易にすることができる。ラウンドロビン方法とは異なり、本明細書におけるエポック変更処理は、加重和に応じて新しい1次ノードを選択し、これは、故障していない新しい1次ノードを見つける際のレイテンシまたは遅延を低減させることができる。これは、ブロックチェーンサービスを提供する際に全体的なブロックチェーンシステムの効率性をさらに改善することができる。

【0030】

20

さらに、本明細書は、新しい1次ノードとして使用されるネットワークノードによって状態要求メッセージを送ること、および他のネットワークノードから状態リプライメッセージを受け取ることなどの動作を含む回復処理を論じる。これらの動作は、故障しているネットワークノードの回復処理が、他の故障していないネットワークノード間のコンセンサス処理の正常動作に支障を来さないように行われる。これは、回復処理の複雑性を低減させることによって、故障しているネットワークノードを回復させるためのコンピューティングリソースおよびネットワークリソースを節約することを容易にする。

【0031】

本明細書による方法は、本明細書で説明される態様および特徴の任意の組合せを含むことができるということが認知される。すなわち、本明細書による方法は、本明細書で具体的に説明される態様および特徴の組合せに限定されるのではなく、提供される態様および特徴の任意の組合せも含む。

30

【0032】

本明細書の1つまたは複数の実装形態の詳細が、添付の図面および下記の説明において示される。本明細書の他の特徴および長所は、説明および図面から、また特許請求の範囲から明らかになるであろう。

【図面の簡単な説明】**【0033】**

【図1】本明細書の実装形態を実行するために使用されることが可能な環境の例を描写する図である。

40

【図2】本明細書の実装形態による概念アーキテクチャの例を描写する図である。

【図3】本明細書の実装形態に従って実行されることが可能なコンセンサス処理の例を描写する図である。

【図4】本明細書の実装形態に従って実行されることが可能なコンセンサス処理の例を描写する図である。

【図5】本明細書の実装形態によるハッシュ木の例を描写する図である。

【図6】本明細書の実装形態による分散システムのネットワークノード間で通信されるメッセージの例を描写する図である。

【図7】本明細書の実装形態による分散システムにおける1次ノードの変更を行う処理の例を描写する図である。

50

【図8】本明細書の実装形態による分散システムにおける1次ノードの変更を行う処理の例を描写する図である。

【図9】本明細書の実装形態による分散システムのネットワークノード間で通信されるメッセージの例を描写する図である。

【図10】本明細書の実装形態による分散システムにおけるネットワークノードの回復処理を行う処理の例を描写する図である。

【図11】本明細書の実装形態による分散システムにおけるネットワークノードの回復処理を行う処理の例を描写する図である。

【図12】本明細書の実装形態による分散システムのネットワークノード間で通信されるメッセージの例を描写する図である。

【図13】本明細書の実装形態によるコンセンサス装置のモジュールを示す図の例を描写する図である。

【図14】本明細書の実装形態によるコンセンサス装置のモジュールを示す図の例を描写する図である。

【図15】本明細書の実装形態による1次ノード変更装置のモジュールを示す図の例を描写する図である。

【図16】本明細書の実装形態による1次ノード変更装置のモジュールを示す図の例を描写する図である。

【図17】本明細書の実装形態による回復装置のモジュールを示す図の例を描写する図である。

【発明を実施するための形態】

【0034】

様々な図面における同様の参照符号は、同様の要素を示す。

【0035】

本明細書の実装形態は、分散システム(例えばブロックチェーンネットワーク)におけるコンセンサス問題に対処するためのコンピュータ実行方法を含む。より詳細には、本明細書の実装形態は、分散システムにおける1次ノードの変更を行うことを対象とする。

【0036】

いくつかの実装形態において、アクションは、エポック変更が行われる必要があるとブロックチェーンネットワークのバックアップノードによって判断することであって、エポック変更が、現在の1次ノードによる現在のエポックから、新しい1次ノードによる新しいエポックへの変更を引き起こし、現在のエポックが、1次ノードを使用して、いくつかのネットワークノード間のコンセンサスを実現するためのコンセンサス処理を含み、コンセンサス処理が、3つのフェーズを含む、判断することと、現在のエポックにおけるコンセンサス処理の3つのフェーズのそれぞれと関連付けられたバックアップノードの個々の重みをバックアップノードによって判断することと、現在のエポックにおける3つのフェーズのそれぞれと関連付けられたバックアップノードの個々の重みに基づいて、バックアップノードによってバックアップノードに対する加重和を判断することと、加重和が第1の所定の閾値に達したとの判断に応答して、バックアップノードによって、このネットワークノード以外のいくつかのネットワークノードにEPOCH_CHANGEメッセージを送ることであって、EPOCH_CHANGEメッセージが、現在の1次ノードによる現在のエポックから、バックアップノードが新しい1次ノードである新しいエポックへの変更のための要求を示し、EPOCH_CHANGEが、バックアップノードの加重和を含む、送ることと、バックアップノードによって、このバックアップノード以外のいくつかのネットワークノードのうちの少なくとも1つから少なくとも1つのNEW_EPOCHメッセージを受け取ることであって、NEW_EPOCHメッセージが、バックアップノードが新しい1次ノードになることについての承認を示す、受け取ることと、少なくとも1つのNEW_EPOCHメッセージが有効であるかどうかをバックアップノードによって検証することと、少なくとも1つのNEW_EPOCHメッセージからの有効なNEW_EPOCHメッセージの数が第2の所定の閾値を超過するかどうかをバックアップノードによって判断することと、有効なNEW_EPOCHメッセージの数が第2の所定の閾値を超過するとの

10

20

30

40

50

判断に応答して、バックアップノードが新しいエポックにおける新しい1次ノードであるとバックアップノードによって判断することと、を含む。

【0037】

いくつかの実装形態において、アクションは、いくつかのネットワークノードのうちの1つのネットワークノードによって、このネットワークノード以外のバックアップノードからEPOCH_CHANGEメッセージを受け取ることであって、EPOCH_CHANGEメッセージが、エポック変更が行われる必要があるという指示を含み、エポック変更が、現在の1次ノードによる現在のエポックから、新しい1次ノードによる新しいエポックへの変更を引き起こす、受け取ることと、EPOCH_CHANGEメッセージが有効であるかどうかをネットワークノードによって検証することと、EPOCH_CHANGEメッセージが有効であることの検証に応答して、他のネットワークノードにNEW_EPOCHメッセージをネットワークノードによって送ることと、NEW_EPOCHメッセージが、EPOCH_CHANGEメッセージのダイジェストを含む、送ることと、ネットワークノードによって、このネットワークノード以外のいくつかのネットワークノードのうちの少なくとも1つから少なくとも1つのNEW_EPOCHメッセージを受け取ることと、少なくとも1つのNEW_EPOCHメッセージが有効であるかどうかをネットワークノードによって検証することと、少なくとも1つのNEW_EPOCHメッセージからの有効なNEW_EPOCHメッセージの数が所定の閾値を超過するかどうかをバックアップノードによって判断することと、有効なNEW_EPOCHメッセージの数が所定の閾値を超過するとの判断に応答して、バックアップノードが新しいエポックにおける新しい1次ノードであるとネットワークノードによって判断することと、を含む。

【0038】

本明細書の実装形態に対するさらなる背景を示すために、また上記で紹介されたように、(例えばピアツーピアノードで構成される)コンセンサスネットワークまたはブロックチェーンネットワークと呼ばれることも可能な分散台帳システム(DLS)は、参加エンティティが安全かつ不変に取引を行うこと、およびデータを格納することを可能にする。用語ブロックチェーンは、任意の特定のユースケースを参照することなく、全体的にDLSを指すために本明細書で使用される。上記で紹介されたように、ブロックチェーンネットワークは、公開ブロックチェーンネットワーク、プライベートブロックチェーンネットワーク、またはコンソーシアムブロックチェーンネットワークとして提供されることが可能である。

【0039】

ブロックチェーンは、チェーンに格納された先の全ての取引との一貫性について、将来の取引が検証されるのを可能にする方式で取引を格納するデータ構造である。ブロックチェーンは、1つまたは複数のブロックを含む。チェーン内の各ブロックは、前のブロックの暗号的ハッシュを含めることによって、チェーン内のブロックの直前の先のブロックにリンクされる。各ブロックは、タイムスタンプ、ブロック自体の暗号的ハッシュ、および1つまたは複数の取引も含む。ブロックチェーンネットワークのノードによって既に検証された取引は、ハッシュされ、マークル木にエンコードされる。マークル木は、マークル木の葉ノードにあるデータがハッシュされるデータ構造であり、マークル木の各枝における全てのハッシュが、枝のルートにおいて結合される。この処理は、マークル木に沿って、マークル木の中の全てのデータを表すハッシュを格納する木全体のルートまで続く。木に格納された取引のものであるとされるハッシュは、ハッシュが木の構造と一致するものであるかどうかを判断することによって迅速に検証されることが可能である。

【0040】

ブロックチェーンが取引を格納するためのデータ構造である一方で、ブロックチェーンネットワークは、1つまたは複数のブロックチェーン構造を管理、更新、および維持するコンピューティングノードのネットワークである。上記で紹介されたように、ブロックチェーンネットワークは、公開ブロックチェーンネットワーク、プライベートブロックチェーンネットワーク、またはコンソーシアムブロックチェーンネットワークとして提供されることが可能である。

【0041】

公開ブロックチェーンネットワークにおいて、コンセンサス処理は、コンセンサスネットワークのノードによって制御される。例えば、何百、何千、何百万ものエンティティでさえ、公開ブロックチェーンネットワークにおいて連携することができ、これらのそれぞれが、公開ブロックチェーンネットワーク内で少なくとも1つのノードを動作させる。したがって、公開ブロックチェーンネットワークは、参加エンティティに対するパブリックネットワークと見なされることが可能である。いくつかの例において、エンティティ(ノード)の大多数は、ブロックが有効になり、ブロックチェーンネットワークのブロックチェーン(分散台帳)に追加されるように、あらゆるブロックに署名しなければならない。公開ブロックチェーンネットワークの例は、ブロックチェーンと呼ばれる分散台帳を活用する特定のピアツーピアの支払ネットワークを含む。上述のように、用語ブロックチェーンは、それでも、いずれの特定のブロックチェーンネットワークに対する特定の参照もなく、分散台帳を一般に指すために使用される。

10

【0042】

一般に、公開ブロックチェーンネットワークは、公開取引をサポートする。公開取引は、公開ブロックチェーンネットワーク内のノードの全てによって共有され、グローバルブロックチェーンに格納される。グローバルブロックチェーンは、全てのノードにわたって複製されるブロックチェーンである。すなわち、全てのノードは、グローバルブロックチェーンに対して完璧な状態のコンセンサスになる。コンセンサス(例えばブロックチェーンへのブロックの追加に対する合意)を実現するために、コンセンサスプロトコルが、公開ブロックチェーンネットワーク内で実行される。コンセンサスプロトコルの例は、プルーフオブワーク(POW: proof-of-work)、プルーフオブステーク(POS: proof-of-stake)、およびプルーフオブオーソリティ(POA: proof-of-authority)を限定することなく含む。POWは、非限定的な例として本明細書でさらに参照される。

20

【0043】

一般に、プライベートブロックチェーンネットワークは、読み書き許可を中心に制御する特定のエンティティに提供される。エンティティは、どのノードがブロックチェーンネットワークに参加できるかを制御し、したがって、プライベートブロックチェーンネットワークは、一般に、誰がネットワークに参加することを許されているかに対して、および彼らの参加レベル(例えば一定の取引においてのみ)に対して制約を課す、許可制ネットワーク(permissioned network)と呼ばれる。様々なタイプのアクセス制御メカニズムが使用されることが可能である(例えば、既存の参加者が新しいエンティティの追加を採決し、監督機関が入会を制御することができる)。

30

【0044】

一般に、コンソーシアムブロックチェーンネットワークは、参加エンティティ間でプライベートなものである。コンソーシアムブロックチェーンネットワークにおいて、コンセンサス処理は、権限付与された一組のノードによって制御され、1つまたは複数のノードは、個々のエンティティ(例えば、金融機関、保険会社)によって運用される。例えば、10(10)個のエンティティ(例えば、金融機関、保険会社)のコンソーシアムが、コンソーシアムブロックチェーンネットワークを運用することができ、エンティティのそれぞれは、コンソーシアムブロックチェーンネットワーク内で少なくとも1つのノードを運用する。したがって、コンソーシアムブロックチェーンネットワークは、参加エンティティに対するプライベートネットワークと見なされることが可能である。いくつかの例において、各エンティティ(ノード)は、ブロックが有効になり、ブロックチェーンに追加されるように、あらゆるブロックに署名しなければならない。いくつかの例において、エンティティ(ノード)の少なくともサブセット(例えば少なくとも7つのエンティティ)は、ブロックが有効になり、ブロックチェーンに追加されるように、あらゆるブロックに署名しなければならない。

40

【0045】

本明細書の実装形態は、コンソーシアムブロックチェーンネットワークを参照しながら

50

本明細書でさらに詳細に説明される。しかし、本明細書の実装形態は、任意の妥当なタイプのブロックチェーンネットワーク内で実現されることが可能であるということが意図される。

【0046】

本明細書の実装形態は、上記の背景を考慮して本明細書でさらに詳細に説明される。より具体的には、また上記で紹介されたように、本明細書の実装形態は、分散システムにおけるネットワークノードに対して回復処理を行うことを対象とする。

【0047】

ブロックチェーンは、公開またはプライベートのピアツーピアネットワークに取引を記録する、改ざん耐性のある共有デジタル台帳である。台帳は、ネットワーク内の全てのメンバノードに配布され、ネットワーク内で発生する資産取引の履歴は、ブロック内に永久に記録される。

10

【0048】

コンセンサスメカニズムは、分散ブロックチェーンネットワーク内の全てのネットワークノードが、同じ順序で取引を実行し、その後、同じ台帳に書き込むことを保証する。コンセンサスモデルが対処することを目指す1つの問題は、ビザンチン障害を克服することである。ビザンチン障害において、分散ブロックチェーンネットワークのサーバまたはネットワークノードなどの構成要素は、障害検出システムには故障したようにも、機能しているようにも矛盾して見えることがあり、異なる観察者に異なる症状を示す。他のネットワークノードは、どのネットワークノードが最初に故障したのかについてのコンセンサスに最初に達する必要があるので、構成要素が故障していると宣言して、構成要素をネットワークから切り離すのは他のネットワークノードには困難である。

20

【0049】

分散システムの背景において、ビザンチンフォールトトレランス(BFT: Byzantine fault tolerance)は、システムの悪意のある構成要素(すなわちブロックチェーンネットワークのネットワークノード)が故障しているか、正しくない情報を他のピアに伝搬させているにもかかわらず、要望通りに機能し、十分なコンセンサスに正しく達するようにする分散コンピュータネットワークの能力である。目的は、ネットワークの正しい機能、およびシステムにおけるまともなノードによって達せられる正しいコンセンサスを、これらの悪意のあるノードが脅かす影響を緩和させることによって壊滅的なシステム障害から守ることである。

30

【0050】

しかし、既存のBFTメカニズムは、多くの態様において非効率的であることがわかってきた。例えば、既存のBFTメカニズムは、ビザンチン障害を克服しようとするときの分散ブロックチェーンネットワークに対する実行の複雑性を増加させ、その結果、分散ブロックチェーンネットワークのネットワークノード間の通信に対するレイテンシが増加する。プラクティカルビザンチンフォールトトレランス(PBFT: Practical Byzantine Fault Tolerance)は、既存のBFTコンセンサスメカニズムを改良することを目指す最適化のうちの1つである。PBFTモデルは、独立ノードの故障、および特定の独立ノードによって伝搬される操作されたメッセージがあるという想定を通じて、ビザンチン障害(悪意のあるノード)に耐性のある実用的なビザンチン状態機械の複製を行うことに焦点を当てる。

40

【0051】

PBFTモデルにおいて、ノードの全ては順番に並べられ、一方のノードが1次ノード(リーダー)であり、他方がバックアップノードと呼ばれる。システムにおけるノードの全てが互いに通信し、目標は、まともなノードの大多数がシステムの状態の合意に至ることである。ノードは互いに通信して、メッセージが特定のピアノードから来たことを証明しなければならないだけでなく、メッセージが送信中に修正されなかったことを検証する必要がある。

【0052】

PBFTモデルが機能するためには、脆弱性の所与のウィンドウの中で、ネットワーク内の

50

悪意のあるノードの量が、システムにおける全ノードの1/3以上になることができないということが前提となる。システムにおけるノードが多ければ多いほど、全ノードの1/3に近い数が悪意のあるものである可能性が数学的にますます低くなる。アルゴリズムは、多くても $(n-1)/3$ 個のノードが同時に悪意のあるものであるか故障している間、生存性(liveness)と安全性の両方を効果的にもたらし、ここで、 n は、全ノードを表す。

【0053】

(ビューと呼ばれる)PBFTコンセンサスのそれぞれのラウンドは、4つのフェーズを含む。

- (1)クライアントは、サービス動作を起動させるためにリーダーノードに要求を送る。
- (2)リーダーノードは、バックアップノードに要求をマルチキャストする。
- (3)ノードは、要求を実行し、その後、クライアントにリプライを送る。
- (4)クライアントは、同じ結果を有する、異なるノードからの $f+1$ 個のリプライを待つ(f は、故障している可能性のあるノードの最大数を表す)。

最終結果は、全てのまともなノードが、記録の順序について合意に至り、これらのノードが、合意を受け入れるか、拒絶するというものである。リーダーノードは、全てのビューの中で、ラウンドロビン方式で変更され、リーダーノードが要求をマルチキャストすることなく指定の時間が過ぎた場合は、ビュー変更と呼ばれるプロトコルで置き換えられることさえ可能である。まともなノードの大多数は、リーダーが故障しているかどうかを判定し、このリーダーを、代わりとして一列に並んだ次のリーダーを用いて除去することもできる。

【0054】

しかし、PBFTコンセンサスメカニズムにはいくつかの限界がある。例えば、PBFTモデルは、ノード間で要求される手に負えない量の通信のため、比較的小さいコンセンサスグループサイズの、このモデルの伝統的な形ではうまく機能することができる。ネットワークノード間で送信される巨大なブロックデータは、ネットワーク負荷問題を引き起こし、ネットワークのボトルネックをもたらす。さらに、PBFTモデルにおける認証メッセージのフォーマットとして方法認証コード(MAC: method authentication code)を使用することは、暗号通貨ネットワークなどの大きいコンセンサスグループにおけるノード間で必要な通信量によって、またMACによって、非効率になる可能性がある。サードパーティへのメッセージの信憑性を証明することが本来的にできなくなる可能性がある。

【0055】

さらに、PBFTによって使用されるラウンドロビン方法を使用してリーダーノードを変更するときに連続して悪意のあるノードに遭遇することは、まともなリーダーノードを見つける際にレイテンシまたは遅延をもたらすことによってブロックチェーンサービスに影響を及ぼす。例えば、新しいリーダーノードとして第1のネットワークノードを選択するとき、第1のネットワークノードは、悪意のあるノードである可能性があり、したがって新しいリーダーノードとして選択されることは不可能である。ラウンドロビン方法においては、一列に並んだ第2のネットワークノードが、新しいリーダーノードとして選択されることが可能である。しかし、第2のネットワークノードも悪意のあるノードである場合、一列に並んだ別のネットワークノードが、リーダーノードとして適切であるかどうかについて検証されることになる。この処理は、まともな新しいリーダーノードが識別されるまで続く。リーダーノードのこのような頻繁な変更は、ブロックチェーンサービスにおけるかなりのレイテンシをもたらす。

【0056】

さらに、ブロックチェーンネットワーク内のネットワークノードは、ビザンチン障害またはクラッシュ障害にいつでも遭遇する可能性がある。例えば、ネットワークノードは、悪意のあるサイバー攻撃者によって危険にさらされ、不適当にふるまう可能性がある。危険にさらされるネットワークノードが、即座に回復されない場合、悪意のあるサイバー攻撃者は、検出されることなくネットワークノードの1/3超を壊すことによって、ブロックチェーンネットワークおよびサービスを危険にさらす可能性がある。

【0057】

既存のBFTコンセンサスメカニズムおよびPBFTコンセンサスメカニズムと関連付けられた上述の問題および懸念に対処するために、本明細書は、分散システムにおけるネットワークノード間のコンセンサスを実現するため、分散システムにおける1次ノードの変更を行うため、および分散システムにおけるネットワークノードに対する回復処理を行うための技法を含む改良されたコンセンサスメカニズムを開示する。説明されるコンセンサスメカニズムは、様々な用途における様々な長所を実現することができる。

【0058】

例えば、下記で論じられるようなコンセンサス処理は、ブロックチェーンシステムの動作を改善し、ネットワークのボトルネックを緩和させるのに役立つ多くの特徴を含む。例えば、説明されるコンセンサス処理は、ECコードに従って取引要求をいくつかの消去コード(EC)ブロックにコンバートすること、およびネットワークノードのそれぞれにECブロックのうちの1つを送ることを含む。ECブロックは、元の取引要求よりサイズが小さい。したがって、完全な取引要求の代わりにECブロックをネットワークノードに送ることは、ブロックチェーンネットワークのネットワークノード間で送信されるデータブロックのサイズを低減させ、このことにより、ネットワーク帯域幅を節約し、ネットワーク負荷を低減させる。これは、ネットワークノードのメモリ空間との間で読み書きされるデータのサイズをさらに低減させ、このことにより、ネットワークノードのメモリ空間に対する負担を低減させ、全体的なブロックチェーンシステムの効率性を改善する。

【0059】

さらに、本明細書は、コンセンサス処理の複数のフェーズに個々の重みを割り当てること、複数のフェーズの個々の重みに基づいて加重和を判断すること、および加重和に基づいて新しい1次ノードを判断することを含むエポック変更処理を説明する。ラウンドロビン方法の代わりに加重和に基づくエポック変更処理は、故障していない新しい1次ノードをタイムリに選ぶことを容易にすることができる。1次ノードは、1次ノードを含むいくつかのネットワークノード間のコンセンサス処理のラウンドを始める権限を有するリーダーノードであることが可能である。ブロックチェーンネットワークの他のネットワークノードは、バックアップノードと呼ばれることが可能である。エポック変更処理は、新しい1次ノードに対して一列に並んだ複数のネットワークノードが故障しているときに、1次ノードの頻繁な変更を引き起こすラウンドロビン方法の問題に対処するのに役立つ可能性がある。ラウンドロビン方式とは異なり、本明細書におけるエポック変更処理は、加重和に応じて新しい1次ノードを選択し、これは、故障していない新しい1次ノードを見つける際のレイテンシまたは遅延を低減させることができる。これは、ブロックチェーンサービスを提供する際の全体的なブロックチェーンシステムの効率性をさらに改善することができる。

【0060】

さらに、本明細書は、新しい1次ノードとして使用されるネットワークノードによって状態要求メッセージを送ること、および他のネットワークノードから状態リプライメッセージを受け取ることなどの動作を含む回復処理を論じる。これらの動作は、故障しているネットワークノードの回復処理が、他の故障していないネットワークノード間のコンセンサス処理の正常動作に支障を来さないように行われる。これは、回復処理の複雑性を低減させることによって故障しているネットワークノードを回復させるためのコンピューティングリソースおよびネットワークリソースを節約することを容易にする。

【0061】

図1は、本明細書の実装形態を実行するために使用されることが可能な環境100の例を描写する。いくつかの例において、環境100は、コンソーシアムブロックチェーンネットワーク102にエンティティが参加することを可能にする。環境100は、コンピューティングデバイスまたはシステム106、108、およびネットワーク110を含む。いくつかの例において、ネットワーク110は、ローカルエリアネットワーク(LAN)、広域ネットワーク(WAN)、インターネット、またはこれらの組合せを含み、ウェブサイト、ユーザデバイス(例えばコ

ンピューティングデバイス)、およびバックエンドシステムを接続する。いくつかの例において、ネットワーク110は、有線および/またはワイヤレス通信リンクでアクセスされることが可能である。いくつかの例において、ネットワーク110は、コンソーシアムブロックチェーンネットワーク102との通信、およびコンソーシアムブロックチェーンネットワーク102内での通信を可能にする。全体的に、ネットワーク110は、1つまたは複数の通信ネットワークを表す。いくつかのケースにおいて、コンピューティングデバイス106、108は、クラウドコンピューティングシステム(図示せず)のノードであることが可能であり、また各コンピューティングデバイス106、108は、ネットワークによって相互接続され、分散処理システムとして機能する複数のコンピュータを含む別々のクラウドコンピューティングシステムであることが可能である。

10

【0062】

描写された例において、コンピューティングシステム106、108は、それぞれ、コンソーシアムブロックチェーンネットワーク102にノードとして参加することを可能にする任意の妥当なコンピューティングシステムを含むことができる。コンピューティングデバイスの例は、サーバ、デスクトップコンピュータ、ラップトップコンピュータ、タブレットコンピューティングデバイス、およびスマートフォンを限定することなく含む。いくつかの例において、コンピューティングシステム106、108は、コンソーシアムブロックチェーンネットワーク102と対話するために1つまたは複数のコンピュータ実行サービスをホストする。例えば、コンピューティングシステム106は、1つまたは複数の他のエンティティ(例えば他のユーザ)との第1のエンティティの取引を管理するために第1のエンティティが使用する取引管理システムなどの、第1のエンティティ(例えばユーザA)のコンピュータ実行サービスをホストすることができる。コンピューティングシステム108は、1つまたは複数の他のエンティティ(例えば他のユーザ)との第2のエンティティの取引を管理するために第2のエンティティが使用する取引管理システムなどの、第2のエンティティ(例えばユーザB)のコンピュータ実行サービスをホストすることができる。図1の例において、コンソーシアムブロックチェーンネットワーク102は、ノードのピアツーピアネットワークとして表され、コンピューティングシステム106、108は、コンソーシアムブロックチェーンネットワーク102に参加する第1のエンティティおよび第2のエンティティのそれぞれのノードを提供する。

20

【0063】

図2は、本明細書の実装形態による概念アーキテクチャ200の例を描写する。概念アーキテクチャ200の例は、参加者A、参加者B、および参加者Cにそれぞれ対応する参加者システム202、204、206を含む。各参加者(例えば、ユーザ、企業)は、複数のノード214を含むピアツーピアネットワークとして提供されるブロックチェーンネットワーク212に参加し、複数のノード214のうちの少なくともいくつかは、ブロックチェーン216に情報を不変に記録する。単一のブロックチェーン216が、ブロックチェーンネットワーク212内に概略的に描写されているが、本明細書でさらに詳細に説明されるように、ブロックチェーンネットワーク212の全体にわたってブロックチェーン216の複数のコピーがあり、維持される。

30

【0064】

描写された例において、各参加者システム202、204、206は、参加者A、参加者B、および参加者Cのそれぞれによって、またはこれらの代わりに提供され、ブロックチェーンネットワーク内の個々のノード214として機能する。本明細書で使用されるように、ノードは、一般に、ブロックチェーンネットワーク212に接続され、個々の参加者がブロックチェーンネットワークに参加することを可能にする、個々のシステム(例えば、コンピュータ、サーバ)を指す。図2の例において、参加者は、各ノード214に対応する。しかし参加者は、ブロックチェーンネットワーク212内で複数のノード214を動作させることができ、および/または複数の参加者は、ノード214を共有することができるということが意図される。いくつかの例において、参加者システム202、204、206は、プロトコル(例えばハイパーテキスト転送プロトコルセキュア(HTTPS))を使用して、および/またはリモートプロシージャコール(RPC)を使用して、ブロックチェーンネットワーク212と通信するか、ブロッ

40

50

クチェーンネットワーク212を介して通信する。

【0065】

ノード214は、ブロックチェーンネットワーク212内で様々な度合いの参加を行うことができる。例えば、ノード214の中には、(例えばブロックチェーン216にブロックを追加する監視ノード(minder node)として)コンセンサス処理に参加できるものもある一方で、他のノード214は、コンセンサス処理には参加しない。別の例として、ノード214の中には、ブロックチェーン216の完全コピーを格納するものもある一方で、他のノード214は、ブロックチェーン216の一部のコピーしか格納しない。例えば、データアクセス権は、個々の参加者の個々のシステムに個々の参加者が格納するブロックチェーンデータを制限することができる。図2の例において、参加者システム202、204、206は、ブロックチェーン216の個々の完全コピー216'、216''、216'''を格納する。

10

【0066】

ブロックチェーン(例えば図2のブロックチェーン216)はブロックのチェーンで構成され、各ブロックがデータを格納する。データの例は、2つ以上の参加者の間の取引を表す取引データを含む。非限定的な例として取引が本明細書で使用されるが、任意の妥当なデータ(例えば、文書、画像、ビデオ、オーディオ)がブロックチェーンに格納されることが可能であるということが意図される。取引の例は、何らかの価値(例えば、資産、製品、サービス、および通貨)の交換を限定することなく含むことができる。取引データは、ブロックチェーン内に不変に格納される。すなわち、取引データが変更されることは不可能である。

20

【0067】

ブロックに格納する前に、取引データはハッシュされる。ハッシングは、(文字列データとして提供される)取引データを(同様に文字列データとして提供される)固定長ハッシュ値に変換する処理である。ハッシュ値をアンハッシュして取引データを入手することは不可能である。ハッシングは、取引データのわずかな変更でさえ、完全に異なるハッシュ値を生じることを保証する。さらに、また上述のように、ハッシュ値は固定長のものである。すなわち、取引データのサイズに関係なく、ハッシュ値の長さは固定される。ハッシングは、ハッシュ関数によって取引データを処理し、ハッシュ値を生成することを含む。ハッシュ関数の例は、256ビットハッシュ値を出力するセキュアハッシュアルゴリズム(SHA)-256を限定することなく含む。

30

【0068】

複数の取引の取引データはハッシュされ、ブロックに格納される。例えば、2つの取引のハッシュ値が提供され、このハッシュ値自体がハッシュされて別のハッシュをもたらす。この処理は、全ての取引がブロックに格納されるために、単一のハッシュ値が提供されるまで繰り返される。このハッシュ値は、マークルルートハッシュと呼ばれ、ブロックのヘッダに格納される。取引のいずれかの変更は、取引のハッシュ値の変更、および最終的にマークルルートハッシュの変更を生じることになる。

【0069】

ブロックは、コンセンサスプロトコルを通じてブロックチェーンに追加される。ブロックチェーンネットワーク内の複数のノードは、コンセンサスプロトコルに参加し、ブロックをブロックチェーンに追加することを競う。このようなノードは、マイナー(miner)(または監視ノード)と呼ばれる。上記で紹介されたPOWは、非限定的な例として使用される。

40

【0070】

マイナーノードは、コンセンサス処理を実行して、ブロックチェーンに取引を追加する。複数のマイナーノードがコンセンサス処理に参加するが、ただ1つのマイナーノードしか、ブロックチェーンにブロックを書き込むことができない。すなわち、マイナーノードは、コンセンサス処理において、自分のブロックをブロックチェーンに追加することを競う。さらに詳細には、マイナーノードは、(例えば、もしあれば、ブロックに含まれることが可能な取引数に対する所定の限界まで)取引プールから未決の取引を定期的に収集する。取引プールは、ブロックチェーンネットワークの参加者からの取引メッセージを含む

50

。マイナーノードはブロックを構築し、ブロックに取引を追加する。ブロックに取引を追加する前に、マイナーノードは、取引のいずれかがブロックチェーンのブロックに既に含まれているかどうかをチェックする。取引が別のブロックに既に含まれている場合、取引は廃棄される。

【0071】

マイナーノードはブロックヘッダを生成し、ブロック内の取引の全てをハッシュし、単一のハッシュ値がブロック内の全ての取引に提供されるまでハッシュ値をさらに生成するために2つ1組でハッシュ値を組み合わせる(マークルルートハッシュ)。このハッシュはブロックヘッダに追加される。マイナーは、ブロックチェーン内の最も新しいブロックのハッシュ値(すなわちブロックチェーンに追加される最後のブロック)も判断する。マイナーノードは、ノンス値およびタイムスタンプもブロックヘッダに追加する。マイニング処理において、マイナーノードは、必要なパラメータを満たすハッシュ値を見つけようと試みる。マイナーノードは、必要なパラメータを満たすハッシュ値を見つけるまでノンス値を変更し続ける。

10

【0072】

ブロックチェーンネットワーク内の全てのマイナーは、必要なパラメータを満たすハッシュ値を見つけようと試み、このようにして、互いに競う。最終的に、マイナーノードのうちの1つが、必要なパラメータを満たすハッシュ値を見つけ、このことを、ブロックチェーンネットワーク内の他の全てのマイナーノードに知らせる。他のマイナーノードはハッシュ値を検証し、正しいと判断されると、ブロック内の各取引を検証し、ブロックを受け入れ、ブロックチェーンのこれらのコピーにブロックを付加する。このように、ブロックチェーンのグローバル状態は、ブロックチェーンネットワーク内の全てのマイナーノードの全体にわたって一致している。上述の処理は、POWコンセンサスプロトコルである。

20

【0073】

図2を参照しながら非限定的な例が提供される。この例において、参加者Aは、ある程度の資金を参加者Bに送ることを望む。参加者Aは、(例えば、Fromフィールド、Toフィールド、および値フィールドを含む)取引メッセージを生成し、ブロックチェーンネットワークに取引メッセージを送り、ブロックチェーンネットワークは、取引メッセージを取引プールに追加する。ブロックチェーンネットワーク内の各マイナーノードはブロックを作り出し、(例えば、もしあれば、ブロックに追加されることが可能な取引数に対する所定の限度まで)取引プールから全ての取引を入手し、ブロックに取引を追加する。このように、参加者Aによって公表された取引は、マイナーノードのブロックに追加される。

30

【0074】

いくつかのブロックチェーンネットワークにおいて、取引のプライバシーを維持するために暗号法が実行される。例えば、ブロックチェーンネットワーク内の他のノードが取引の詳細を見分けられないように2つのノードが取引をプライベートに保つことを望む場合、ノードは、取引データを暗号化することができる。暗号化方法の例は、対称暗号化および非対称暗号化を限定することなく含む。対称暗号化は、暗号化(平文から暗号文を生成すること)、および複合(暗号文から平文を生成すること)の両方のために単一キーを使用する暗号化処理を指す。対称暗号化において、複数のノードに対して同じキーが利用できるもので、各ノードは、取引データを暗号化すること/復号することができる。

40

【0075】

非対称暗号化は、それぞれが秘密鍵および公開鍵を含むキーのペアを使用し、秘密鍵は個々のノードにだけ知られており、公開鍵は、ブロックチェーンネットワーク内のいずれかまたは他の全てのノードに知られている。ノードは、別のノードの公開鍵を使用してデータを暗号化することができ、暗号化されたデータは、他のノードの秘密鍵を使用して復号されることが可能である。例えば、また図2を再度参照すると、参加者Aは、参加者Bの公開鍵を使用してデータを暗号化し、暗号化されたデータを参加者Bに送ることができる。参加者Bは、参加者Bの秘密鍵を使用して暗号化されたデータ(暗号文)を復号し、元のデータ(平文)を抽出することができる。ノードの公開鍵で暗号化されたメッセージは、この

50

ノードの秘密鍵を使用して復号されること以外は不可能である。

【0076】

非対称暗号化は、取引の参加者が、取引の他の参加者ならびに取引の有効性を確認することを可能にするデジタル署名を行うために使用される。例えば、ノードはメッセージにデジタル的に署名することができ、別のノードは、参加者Aのデジタル署名に基づいてノードによってメッセージが送られたことを確認することができる。デジタル署名は、移送中にメッセージが改ざんされていないことを保証するために使用されることも可能である。例えば、また図2を再度参照すると、参加者Aは、メッセージを参加者Bに送ることになる。参加者Aはメッセージのハッシュを生成し、次に、参加者Aの秘密鍵を使用してハッシュを暗号化し、暗号化されたハッシュとしてデジタル署名を行う。参加者Aはデジタル署名をメッセージに付加し、デジタル署名付きのメッセージを参加者Bに送る。参加者Bは、参加者Aの公開鍵を使用してデジタル署名を復号し、ハッシュを抽出する。参加者Bはメッセージをハッシュし、ハッシュを比較する。ハッシュが同じ場合、参加者Bは、メッセージが実際に参加者Aからのものであったこと、および改ざんされていないことを確認することができる。

10

【0077】

図3は、本明細書の実装形態に従って実行されることが可能な分散システム(例えばブロックチェーンネットワーク102および212)のネットワークノード(例えばノード214)間のコンセンサスを実現するための処理300の例を描写する。具体的には、図3は、本明細書による通常のケースにおけるコンセンサスを実現する方法300の例示的实施形態を提示する図を示す。図3において示されるように、コンセンサス処理300は、下記で論じられるような3つのフェーズ、すなわちステージ310、320、および330を含む。

20

【0078】

コンセンサス処理300の第1のフェーズ310において、ブロックチェーンネットワークの1次ノード(またはリーダーノード)は、他のネットワークノード(すなわちバックアップノード)に第1のメッセージを送る。第1のメッセージは、1次ノードがコンセンサス処理を始めていることを示す。例えば、図3において示されるように、1次ノード R_0 は、ブロックチェーンネットワーク内の他のネットワークノード R_1 、 R_2 、および R_3 にINITIALメッセージを送る。処理300は、例証のためだけに4つのネットワークノード R_0 、 R_1 、 R_2 、および R_3 を含むものとして示されるが、処理300は、任意の適切な数のネットワークノードを含むことができるということに留意されたい。第1のフェーズ、およびINITIALメッセージのフォーマットは、図4～図6を参照しながら、より詳細に下記で論じられることになる。

30

【0079】

コンセンサス処理300の第2のフェーズ320において、バックアップノードのそれぞれは、1次ノードによって送られる第1のメッセージを受け取り、第1のメッセージに応答して第2のメッセージを用意し、他のネットワークノードに第2のメッセージをマルチキャストする。第2のメッセージは、バックアップノードが1次ノードから第1のメッセージを受け取ったこと、および第1のメッセージに応答してリプライを送っていることを示す。例えば、図3において示されるように、バックアップノード R_1 は、1次ノード R_0 によって送られているINITIALメッセージを受け取り、第2のメッセージの例としてECHOメッセージで1次ノード R_0 にリプライする。一方で、バックアップノード R_1 は、バックアップノード R_2 および R_3 などの他のバックアップノードにもECHOメッセージをマルチキャストする。同様に、バックアップノード R_2 および R_3 は、それぞれ、1次ノード R_0 を含む他のネットワークノードにECHOメッセージをマルチキャストする。

40

【0080】

例えば1次ノードまたはバックアップノードなどのネットワークノードが、他のネットワークノードからECHOメッセージを受け取るとき、ネットワークノードは、ECHOメッセージ内の情報を検証することができる。第2のフェーズおよびECHOメッセージのフォーマットは、図4～図6を参照しながら、より詳細に下記で論じられることになる。

【0081】

50

コンセンサス処理300の第3のフェーズ330において、ネットワークノードのそれぞれは、他のネットワークノードに第3のメッセージをマルチキャストする。第3のメッセージは、ネットワークノードが所定の数の第2のメッセージを受け入れたことを示す。いくつかの実装形態において、第3のメッセージは、ネットワークノードが、取引を実行する用意ができたということを示すことができる。いくつかの実装形態において、第3のメッセージは、ネットワークノードにおいて取引が成功裏に再構築されたことを示すことができる。例えば、図3において示されるように、1次ノード R_0 は、バックアップノード R_1 、 R_2 、および R_3 にACCEPTメッセージをマルチキャストする。同様に、バックアップノード R_1 、 R_2 、および R_3 は、それぞれ、他のネットワークノードにACCEPTメッセージをマルチキャストする。本明細書のいくつかの実装形態において、ACCEPTメッセージをマルチキャストする前に、ネットワークノードは、消去コード(EC)に従ってACCEPTが送られるかどうか、およびECHOメッセージ内の情報が、第2のフェーズにおいて受け取られたものであるかどうかを判断する。第3のフェーズ、ECコード、およびACCEPTメッセージのフォーマットは、図4～図6を参照しながら、より詳細に下記で論じられることになる。

【0082】

ネットワークノードが他のネットワークノードから十分なACCEPTメッセージを受け取ると、ネットワークノードは、コンセンサスが実現されたと判断する。例えば、1次ノード R_0 またはバックアップノード R_1 、 R_2 、または R_3 が、定足数(例えば $2f+1$ 、ここで f は、故障しているネットワークノードの数を表す)のACCEPTメッセージを受け取ると、ネットワークノード間でコンセンサスが自動的に実現される。

【0083】

図4は、本明細書の実装形態に従って実行されることが可能な分散システム(例えばブロックチェーンネットワーク102または212)のネットワークノード(例えばノード214またはノード R_0 、 R_1 、 R_2 、および R_3)間でコンセンサスを実現するための処理400の例を描写する。いくつかの実装形態において、処理400は、1つまたは複数のコンピューティングデバイスを使用して実行される1つまたは複数のコンピュータ実行可能プログラムを使用して行われることが可能である。表現の明瞭さのために、続く説明は、全体的に、本説明における他の図の背景において方法400を説明する。方法400は、例えば、任意の適切なシステム、環境、ソフトウェア、およびハードウェア、またはシステム、環境、ソフトウェア、およびハードウェアの組合せによって必要に応じて行われることが可能であるということが理解されよう。いくつかの実装形態において、方法400の様々なステップは、並行に、組み合わせ、ループさせて、または任意の順序で実行されることが可能である。

【0084】

初めに、処理400は、図1～図3において示されたようなシステム100～システム300と共に実行されることが可能である。本明細書のいくつかの実装形態において、ブロックチェーンネットワーク102および/または212は、1次ノード404および1つまたは複数のバックアップノード406を含む。ブロックチェーンネットワーク102および/または212は、ブロックチェーンサービスを提供するためにネットワーク110を介してクライアントノード402などの、コンピューティングシステム106および/または108と通信する。クライアントノード402、1次ノード404、およびバックアップノード406のそれぞれは、本明細書で論じられる処理を行うように構成された専用コンピュータまたは他のデータ処理装置であることが可能である。例えば、クライアントノード402は、ブロックチェーンネットワークと相互作用するクライアント端末またはクライアントデバイスと呼ばれることも可能である。クライアントノード402は、ブロックチェーンネットワークにアクセスするため、およびブロックチェーンネットワークと通信するためにブロックチェーンネットワークと接続状態にある、例えばクライアントアプリケーションまたはクライアントソフトウェア開発キット(SDK)をインストールすることができる。1次ノード404および1つまたは複数のバックアップノード406は、コンセンサスを実現し、ブロックチェーンネットワークに情報を不変に記録するコンセンサスノードまたはネットワークノードと呼ばれることも可能である。

【0085】

処理400は、408において始まり、ここで、クライアントノード402は、取引要求を生成する。本明細書のいくつかの実装形態において、取引要求は、ブロックチェーンネットワーク102および/または212からのブロックチェーンサービスを要求する要求を含むことができる。

【0086】

410において、クライアントノード402は、ブロックチェーンネットワーク102および/または212の1次ノード404に取引要求をマルチキャストする。本明細書のいくつかの実装形態において、1次ノード404は、クライアントノード402から取引要求を受け取った後に取引要求を追跡するために、取引要求にシーケンス番号を割り当てる。

【0087】

412において、1次ノード404は、クライアントノード402から取引要求を受け取った後に、いくつかのECブロックを生成する。本明細書のいくつかの実装形態において、1次ノード404は、取引要求を使用してECコードに従っていくつかのECブロックを生成する。例えば、図5を参照すると、1次ノード404は、取引要求502に対してECコード504を適用し、ECコード504を使用して取引要求502をECメッセージ506に変換する。ECコード504は、ビット消去の前提に基づく前方誤り訂正(FEC)コードである。ECコード504は、元の取引要求502がECメッセージ506の一部または断片から回復されることが可能になるように、元の取引要求502をより長いECメッセージ506に変換する。

【0088】

本明細書のいくつかの実装形態において、ECコード504は、Tornadoコードまたは低密度パリティ検査符号などの、ほぼ最適な消去コードである。本明細書の代替実装形態において、ECコード504は、ファウンテンコード、オンラインコード、Luby変換(LT)コード、またはラプターコードなどの、ほぼ最適なファウンテンコードである。本明細書の代替実装形態において、ECコード504は、パリティコード、Parchiveコード、リードソロモンコード、または再生成コードなどの、最適な消去コードである。本明細書のいくつかの実装形態において、ECコード504は、任意の適切なタイプの消去コードであることが可能である。

【0089】

取引要求502をECメッセージ506に変換した後、1次ノード404は、ECメッセージ506を使用して、いくつかのECブロック508を生成する。例えば、図5において示されるように、1次ノード404は、ECメッセージ506を分割することによって、ECブロックA、ECブロックB、ECブロックC、およびECブロックDという4つのECブロック508を生成する。ECブロック508は、例証のために4つのブロックを含むものとして図5において示されているが、ECブロック508は、任意の適切な数のECブロック508を含むものとして生成されることが可能であるということに留意されたい。ECブロック508は、INITIALメッセージの中で個々のバックアップノード406に送られることになる。

【0090】

本明細書のいくつかの実装形態において、ECブロック508は同じサイズを有する。しかし、代替実装形態において、ECブロック508は、互いに異なるサイズを有することができる。

【0091】

本明細書のいくつかの実装形態において、1次ノード404は、ECブロック508を使用してハッシュ木500(例えばマークル木)を生成する。ハッシュ木500は、データブロックのハッシュでラベル付けされたいくつかの葉ノード、および非葉ノードの子ノードのラベルの暗号学的ハッシュでラベル付けされたいくつかの非葉ノードを含む。例えば、図5に示されたように、ハッシュ木500は、4つの葉ノード510の個々のECブロック508の暗号学的ハッシュとして生成されたハッシュA、ハッシュB、ハッシュC、およびハッシュDという4つの葉ノード510、4つの非葉ノード512の個々の子ノード510の連鎖のハッシュとして生成された4つの非葉ノード512、および非葉ノード514の子ノード512のハッシュとして生成され、ハッシュ木500のルートハッシュである非葉ノード514を含むものとして構成される。

10

20

30

40

50

【0092】

ハッシュ木500は、大きいデータ構造の内容の効率的かつ安全な検証を可能にする。ハッシュ木500は、コンピュータ内で、またコンピュータ間で格納され、取り扱われ、移送される任意の種類のデータを検証するために使用されることが可能である。ハッシュ木500は、P2Pネットワーク内の他のピアから受け取られたデータブロックが無傷であり変更されずに受け取られることを保証すること、および他のピアが偽造ブロックを送っていないことをチェックすることにさえ役立つ可能性がある。ハッシュ木500を使用するデータブロックの検証が、コンセンサス処理400の以下のステップを参照しながら、より詳細に下記で論じられることになる。

【0093】

図4を再度参照すると、1次ノード404は、ECブロック508およびハッシュ木500を生成した後、第1のメッセージ(例えばINITIALメッセージ)を生成する。第1のメッセージは、1次ノードがコンセンサス処理を始めていることを示す。いくつかの実装形態において、第1のメッセージの例としてのINITIALメッセージは、ECブロック508およびハッシュ木500内の情報を使用して生成される。本明細書のいくつかの実装形態において、図6を参照すると、INITIALメッセージは、`<epoch, tx_root_hash, ec_block_hash, ec_block, seq, j>`というフォーマットを有し、ここで、「epoch」は、メッセージが送られているコンセンサスのラウンドを表し、「tx_root_hash」は、ハッシュ木500内のルートハッシュ514を表し、「ec_block_hash」は、ハッシュ木500内のハッシュ510および/または512を表し、「ec_block」は、ハッシュ木500内のECブロック508を表し、「seq」は、取引要求502と関連付けられたシーケンス番号を表し、「j」は、INITIALメッセージを生成して送るネットワークノードを表す。いくつかの実装形態において、INITIALメッセージは、例えば追加のフィールドまたは異なるフィールドを含めることによって、異なるフォーマットを有することができる。

【0094】

図4を再度参照すると、416において、コンセンサス処理の第1のフェーズにおいて、1次ノード404は、他のネットワークノード(例えばバックアップノード406)にINITIALメッセージをマルチキャストする。いくつかの実装形態において、バックアップノード406に送られるINITIALメッセージは、`<epoch, tx_root_hash, ec_block_hash, ec_block, seq, j>`というフォーマットを有する。例えば、1次ノード404は、第1のINITIALメッセージ`<epoch 1, Hash ABCD, {Hash B, Hash C, Hash D}, EC block A, 1, 0>`を第1のバックアップノード406に送ること、および第2のINITIALメッセージ`<epoch 1, Hash ABCD, {Hash A, Hash C, Hash D}, EC block B, 1, 0>`を第2のバックアップノード406に送ること、などを行うことができる。「ec_block」などのINITIALメッセージ内の情報は、ハッシュ木500を再構築するために「ec_block_hash」と共に使用されることが可能であるということに留意されたい。例えば、第1のINITIALメッセージ`<epoch 1, Hash ABCD, {Hash B, Hash C, Hash D}, EC block A, 1, 0>`において、「ECブロックA」というECブロック508は、「Hash A」という暗号的ハッシュ510を生成するためにハッシュされることが可能であり、「Hash A」は、ハッシュ木500を再構築するために「{Hash B, Hash C, Hash D}」という他のハッシュ510と共にさらに使用される。再構築されたハッシュ木500は、コンセンサス処理の以下のステップを参照しながら、より詳細に下記で論じられるような、ECHOメッセージを検証するために使用されることがになる。

【0095】

418において、バックアップノード406のそれぞれは、1次ノード404からINITIALメッセージを受け取った後、コンセンサス処理の第2のフェーズにおいて第2のメッセージ(例えばECHOメッセージ)を生成する。第2のメッセージは、バックアップノードが1次ノードから第1のメッセージを受け取ったことを示す。第2のメッセージは、第1のメッセージに応答してリプライとして送られる。本明細書のいくつかの実装形態において、ECHOメッセージは、INITIALメッセージまたはINITIALメッセージの一部、およびINITIALメッセージと関連付けられたバックアップノード406の署名を含むものとしてバックアップノード

ド406によって生成される。例えば、バックアップノード406は、秘密鍵を使用して、INITIALメッセージまたはINITIALメッセージのダイジェストに署名することによって署名を生成することができる。秘密鍵署名は、秘密鍵署名を含むECHOメッセージを認証するために、秘密鍵とペアになる公開鍵を使用して、他のネットワークノードによって使用されることが可能である。

【0096】

本明細書のいくつかの実装形態において、図6を参照すると、ECHOメッセージは、<epoch, tx_root_hash, ec_block_hash, ec_block, seq, sign_proof, j>というフォーマットを有し、ここで、「epoch」は、メッセージが送られているコンセンサスのラウンドを表し、「tx_root_hash」は、ハッシュ木500内のルートハッシュ514を表し、「ec_block_hash」は、ハッシュ木500内のハッシュ510および/または512を表し、「ec_block」は、個々のバックアップノード406によって受け取られるハッシュ木500内のECブロック508を表し、「seq」は、取引要求502と関連付けられたシーケンス番号を表し、「sign_proof」は、INITIALメッセージと関連付けられたバックアップノード406の署名を表し、「j」は、ECHOメッセージを生成して送るネットワークノードを表す。いくつかの実装形態において、ECHOメッセージは、例えば追加のフィールドまたは異なるフィールドを含めることによって、異なるフォーマットを有することができる。

10

【0097】

図4を再度参照すると、420において、バックアップノード406は、1次ノード404にECHOメッセージを送る。421において、バックアップノード406のそれぞれは、他のバックアップノード406にECHOメッセージを送る。423において、バックアップノード406のそれぞれは、他のバックアップノード406からECHOメッセージを受け取ることができる。

20

【0098】

422において、1次ノード404は、バックアップノード406によって送られたECHOメッセージを検証する。本明細書のいくつかの実装形態において、1次ノード404は、ハッシュ木500に従ってECHOメッセージが有効であるかどうかを検証する。例えば、1次ノード404は、第1のECHOメッセージ<epoch 1, Hash ABCD, {Hash B, Hash C, Hash D}, EC block A, 1, 1>を第1のバックアップノード406から受け取ることができる。1次ノード404は、「ECブロックA」というECブロック508をメッセージから取得し、「ECブロックA」をハッシュして、「ハッシュA」という暗号的ハッシュ510を生成する。1次ノード404は、メッセージ内の「{Hash B, Hash C, Hash D}」という他のハッシュ510と共に、「ハッシュA」という生成されたハッシュ510をさらに使用して、ハッシュ木500を再構築する。次に、1次ノード404は、再構築されたハッシュ木500のルートハッシュ514を判断し、このルートハッシュ514を、「Hash ABCD」などのECHOメッセージ内のルートハッシュ514と比較する。2つのルートハッシュ514がマッチする場合、1次ノード404は、ECHOメッセージが有効であると判断する。1次ノード404は、有効なECHOメッセージを格納すること、および無効であると判断されたECHOメッセージを廃棄することができる。

30

【0099】

424において、1次ノード404は、有効なECHOメッセージの数が所定の閾値を超過するかどうかを判断する。本明細書のいくつかの実装形態において、1次ノード404は、有効なECHOメッセージの数が、定足数 $n-f$ または $2f+1$ に達しているかどうかを判断し、ここで、 n は、ネットワークノードの総数であり、 f は、ネットワークが許容できる故障ノードの最大数である。

40

【0100】

426において、1次ノード404は、有効なECHOメッセージの数が定足数に達したとの判断に回答して取引要求502を再構築する。本明細書のいくつかの実装形態において、1次ノード404は、ECコードに従って有効なECHOメッセージの少なくともサブセットに基づいて取引要求502を再構築する。例えば、1次ノード404は、有効なECHOメッセージの定足数(例えば $n-f$ または $2f+1$)の範囲内の $n-2f$ 個または $f+1$ 個のECブロック508を取得し、取得したECブロック508を使用してECコード504に従って取引要求502を再構築することができる。

50

【0101】

428において、コンセンサス処理の第3のフェーズにおいて、1次ノード404は、取引要求502が成功裏に再構築されたとの判断に応答して、第3のメッセージ(例えばACCEPTメッセージ)を生成する。第3のメッセージは、ネットワークノードが所定の数の第2のメッセージを受け入れたことを示す。いくつかの実装形態において、第3のメッセージは、ネットワークノードが取引を実行する用意ができたということを示すことができる。いくつかの実装形態において、第3のメッセージは、ネットワークノードにおいて取引が成功裏に再構築されたことを示すことができる。例えば、ACCEPTメッセージは、取引要求502が成功裏に再構築されたことを他のネットワークノードに示すために使用されることが可能である。1次ノード404が、取引要求502を再構築するのに失敗した場合、1次ノード404は、ACCEPTメッセージを生成することができない。

10

【0102】

本明細書のいくつかの実装形態において、図6を参照すると、ACCEPTメッセージは、`<epoch, tx_root_hash, seq, sign_proofs, j>`というフォーマットを有し、ここで、「epoch」は、メッセージが送られているコンセンサスのラウンドを表し、「tx_root_hash」は、ハッシュ木500内のルートハッシュ514を表し、「seq」は、取引要求502と関連付けられたシーケンス番号を表し、「sign_proofs」は、有効なECHOメッセージ内の署名のセットを表し、「j」は、ACCEPTメッセージを生成して送るネットワークノードを表す。いくつかの実装形態において、ACCEPTメッセージは、例えば追加のフィールドまたは異なるフィールドを含めることによって、異なるフォーマットを有することが可能である。

20

【0103】

図4を再度参照すると、430において、1次ノード404は、バックアップノード406にACCEPTメッセージを送る。

【0104】

1次ノード404と同様に、バックアップノード406のそれぞれは、例えば1次ノード404としてステップ422～ステップ428と同様のステップを行うことによって、取引要求を再構築することができる。432において、バックアップノード406のそれぞれは、バックアップノード406によって取引要求502が成功裏に再構築したとの判断に応答して、ACCEPTメッセージを生成する。いくつかの実装形態において、1次ノード404およびバックアップノード406は、例えば図3において示されるように、ステップ422～ステップ428を並列に行うことができる。

30

【0105】

434において、バックアップノード406は、1次ノード404にACCEPTメッセージを送る。一方で、バックアップノード406のそれぞれは、他のバックアップノード406にACCEPTメッセージを送ることができる。

【0106】

436において、1次ノード404は、ACCEPTメッセージの数が所定の閾値を超過しているとの判断に応答して、取引要求502を実行する。本明細書のいくつかの実装形態において、1次ノード404は、受け取ったACCEPTメッセージが同一であるかどうか、および同一であるACCEPTメッセージの数が定足数(例えば $2f+1$)に達しているかどうかを判断する。同一のACCEPTメッセージの数が定足数に達していると、1次ノード404は、ネットワークノード全体の間でコンセンサスが実現されたと判断し、次に、取引要求502をローカルに実行する。本明細書のいくつかの実装形態において、同一であるACCEPTメッセージの数が定足数に達していないと1次ノード404が判断すると、1次ノード404は、ネットワークノード全体の間でコンセンサスが実現されなかったと判断し、その後、取引要求502を実行するのを控える。

40

【0107】

本明細書のいくつかの実装形態において、バックアップノード406のそれぞれは、取引要求502を実行する前に、436において上述のような、1次ノード404によって行われる同じ動作を行うことができる。バックアップノード406が受け取るACCEPTメッセージが所定の

50

閾値を超過しているとバックアップノード406が判断すると、バックアップノード406は、ネットワークノード間でコンセンサスが実現されたと判断し、取引要求502をローカルに実行する。本明細書のいくつかの実装形態において、同一であるACCEPTメッセージの数が定足数に達していないとバックアップノード406が判断すると、バックアップノード406は、ネットワークノード全体の間でコンセンサスが実現されなかったと判断し、その後、取引要求502を実行するのを控える。

【0108】

438において、1次ノード404は、取引要求502を実行した後、クライアントノード402に取引結果を送る。取引要求502をローカルに成功裏に実行したバックアップノード406は、取引要求502の個々の取引結果をクライアントノード402に送ることもできる。

10

【0109】

上述のようなコンセンサス処理は、ブロックチェーンシステム全体の動作を改善し、ネットワークのボトルネックを緩和させるのに役立つ多くの特徴を含む。例えば、本明細書におけるコンセンサス処理は、取引要求を使用してECコードに従っていくつかのECブロックを生成し、ネットワークノードのそれぞれにECブロックのうちの1つを送ることを含む。ECブロックは、元の取引要求よりサイズが小さい。したがって、取引要求の代わりにECブロックをネットワークノードに送ることは、ブロックチェーンネットワークのネットワークノード間で送信されるデータブロックのサイズを低減させ、このことにより、ネットワーク帯域幅を節約し、ネットワーク負荷を低減させる。これは、ネットワークノードのメモリ空間との間で読み書きされるデータのサイズをさらに低減させ、このことにより、ネットワークノードのメモリ空間に対する負担を低減させ、全体的なブロックチェーンシステムの効率性を改善する。

20

【0110】

コンセンサス処理の間、バックアップノードは、1次ノードからの要求を待っている。しかし、1次ノードは、ビザンチン障害またはクラッシュ障害に遭遇する可能性があり、その結果、1次ノードは、所定の時間ウィンドウの中で要求をブロードキャストすることができない。1次ノードが要求をマルチキャストすることなく指定の時間が過ぎると、実行すべき要求をバックアップノードが無期限に待つのを防ぐために新しい1次ノードが選ばれる必要がある可能性がある。

【0111】

30

図7は、本明細書の実装形態に従って実行されることが可能な分散システム(例えばブロックチェーンネットワーク102および212)の1次ノードの変更(例えばノード214または404)を行うための処理700の例を描写する。具体的には、図7は、本明細書による、1次ノードの変更を行う方法700の例示的实施形態を提示する図を示す。いくつかの実装形態において、1次ノードは、1次ノードがリーダーであるコンセンサス処理を含むエポックと関連付けられる。1次ノードの変更は、エポックの変更をもたらすことが可能である。

【0112】

いくつかの実装形態において、現在のエポックの1次ノードが変更される必要があるとの判断に応答して、ブロックチェーンネットワークのバックアップノードは、他のネットワークノードに第1のメッセージを送る。第1のメッセージは、新しいエポックにおける新しい1次ノードであることをバックアップノードが望んでいることを示す。例えば、図7において示されたように、バックアップノード R_0 は、現在の1次ノードが故障しており、エポック変更が行われる必要があるとのバックアップノード R_0 の判断に応答して、ブロックチェーンネットワーク内の他のネットワークノード R_1 、 R_2 、および R_3 にEPOCH_CHANGEメッセージを送る。EPOCH_CHANGEメッセージは、バックアップノード R_0 が新しい1次ノードとして使用されることを示す第1のメッセージの1つの例である。エポック変更は、現在の1次ノードによる現在のエポックから、新しい1次ノードによる新しいエポックへの変更を引き起こすことができる。処理700は、例証のためだけに4つのネットワークノードによって実行されるものとして示されているということに留意されたい。処理700は、任意の適切な数のネットワークノードによって実行されることが可能である。

40

50

【0113】

次に、ネットワークノードのそれぞれは、バックアップノードによって送られた第1のメッセージを受け取り、第1のメッセージに回答して第2のメッセージを用意し、他のネットワークノードに第2のメッセージをマルチキャストする。例えば、図7において示されたように、ネットワークノードR₁は、バックアップノードR₀によって送られたEPOCH_CHANGEメッセージを受け取り、バックアップノードR₀が新しい1次ノードになることができるという承認を示すNEW_EPOCHメッセージと共にバックアップノードR₀にリプライする。一方で、ネットワークノードR₁は、ネットワークノードR₂およびR₃などの他のネットワークノードにもNEW_EPOCHメッセージをマルチキャストする。同様に、ネットワークノードR₂およびR₃はそれぞれ、他のネットワークノードにNEW_EPOCHメッセージをマルチキャストする。

10

【0114】

上述のようなエポック変更処理、EPOCH_CHANGEメッセージのフォーマット、およびNEW_EPOCHメッセージのフォーマットは、図8～図9を参照しながら、より詳細に下記で論じられることになる。

【0115】

図8は、本明細書の実装形態に従って実行されることが可能な分散システム(例えばブロックチェーンネットワーク102または212)における1次ノードの変更を行うための処理800の例を描写する。いくつかの実装形態において、処理800の例は、1つまたは複数のコンピューティングデバイスを使用して実行される1つまたは複数のコンピュータ実行可能プログラムを使用して行われることが可能である。表現の明瞭さのために、続く説明は、全体的に、本説明における他の図の背景において方法800を説明する。方法800は、例えば、任意の適切なシステム、環境、ソフトウェア、およびハードウェア、またはシステム、環境、ソフトウェア、およびハードウェアの組合せによって必要に応じて行われることが可能であるということが理解されよう。いくつかの実装形態において、方法800の様々なステップは、並行に、組み合わせて、ループさせて、または任意の順序で実行されることが可能である。

20

【0116】

処理800は、806において始まり、ここで、バックアップノード802は、エポック変更が行われる必要があると判断する。本明細書で論じられるエポック変更は、現在の1次ノードによる現在のエポックから、新しい1次ノードによる新しいエポックへの変更を引き起こす。エポックの例は、図3～図6を参照しながら上述されたように、1次ノードを使用して、いくつかのネットワークノード間のコンセンサスを実現するためのコンセンサス処理(例えばコンセンサス処理300または400)を含むことができる。

30

【0117】

本明細書のいくつかの実装形態において、バックアップノード802は、現在の1次ノードからの要求を受け取ることなく指定の時間が過ぎた後、現在の1次ノードからの要求をバックアップノード802がまだ待っているとの判断に回答して、エポック変更が行われる必要があると判断する。例えば、現在の1次ノードは、ビザンチン障害またはクラッシュ障害に遭遇する可能性があり、その結果、現在の1次ノードは、所定の時間ウィンドウの中で要求をマルチキャストすることができない。したがって、実行すべき要求をバックアップノードが無期限に待つのを防ぐタイムアウトによってエポック変更が誘発される。本明細書で論じられるエポック変更処理は、生存性をもたらし、1次ノードが故障したときにシステムが進行するのを可能にすることによって、ネットワークレイテンシを低減させる。

40

【0118】

808において、バックアップノード802は、現在のエポックにおけるコンセンサス処理のフェーズのそれぞれと関連付けられたバックアップノード802の個々の重みを判断する。いくつかの実装形態において、コンセンサス処理は、図3～図6を参照しながら上述されたような3つのフェーズを含む。重みは、新しいエポックにおいて新しい1次ノードになるた

50

めのバックアップノード802の資格の測定基準である。

【0119】

本明細書のいくつかの実装形態において、バックアップノード802は、現在のエポックにおけるコンセンサス処理の第1のフェーズに対するバックアップノード802の重みが第1の値であると判断する。例えば、バックアップノード802は、コンセンサス処理の第1のフェーズ(例えばコンセンサス処理300の第1のフェーズ310)にバックアップノード802が入った場合、初期の重み10%を割り当てられることが可能である。本明細書の代替実装形態において、バックアップノード802は、現在のコンセンサス処理の第1のフェーズに対するバックアップノード802に任意の適切な重み値を割り当てることができる。

【0120】

本明細書のいくつかの実装形態において、バックアップノード802は、定足数検証処理に基づいて、現在のエポックにおけるコンセンサス処理の第2のフェーズ(例えばコンセンサス処理300の第2のフェーズ320)に対するバックアップノード802の重みを判断する。定足数検証処理は、コンセンサス処理の第2のフェーズにおいて所定の数(例えば $2f+1$)のECHOメッセージをバックアップノード802が他のネットワークノードから受け取ったかどうかを判断することによって行われる。

【0121】

本明細書のいくつかの実装形態において、バックアップノード802が定足数検証に失敗すると(例えば、所定の閾値より少ないいくつかのECHOメッセージをバックアップノード802が受け取る)、バックアップノード802は、コンセンサス処理の第2のフェーズに対するバックアップノード802の重みが第1の値であると判断することができる。バックアップノード802が定足数検証にパスすると(例えば、所定の閾値以上のいくつかのECHOメッセージをバックアップノード802が受け取る)、バックアップノード802は、コンセンサス処理の第2のフェーズに対するバックアップノード802の重みが第2の値であると判断することができる。本明細書のいくつかの実装形態において、第2の値は、第1の値より大きいものと判断される。例えば、バックアップノード802が定足数検証に失敗すると、バックアップノード802は、コンセンサス処理の第2のフェーズに対して、重み値ゼロを割り当てられることが可能である。バックアップノード802が定足数検証にパスすると、バックアップノード802は、コンセンサス処理の第2のフェーズに対するバックアップノード802に対して重み値45%を割り当てられることが可能である。しかし、本明細書の代替実装形態において、バックアップノード802は、現在のエポックにおけるコンセンサス処理の第2のフェーズに対するバックアップノード802に任意の適切な値を割り当てることができる。

【0122】

本明細書のいくつかの実装形態において、定足数検証は、コンセンサス処理の第2のフェーズ中にバックアップノード802が他のネットワークノードから受け取ったECHOメッセージが有効であるかどうかを検証することをさらに含む。例えば、バックアップノード802は、ECHOメッセージが有効であるかどうかを判断するための公開鍵を使用して、ECHOメッセージ内の秘密鍵署名を認証することができる。

【0123】

第2のフェーズに対して重みを判断することと同様に、いくつかの実装形態において、バックアップノード802は、定足数検証処理に基づいて、現在のエポックにおけるコンセンサス処理の第3のフェーズ(例えばコンセンサス処理300の第3のフェーズ330)に対するバックアップノード802の重みを判断する。定足数検証処理は、現在のエポックにおけるコンセンサス処理の第3のフェーズにおいて、所定の数(例えば $2f+1$)の受け入れメッセージをバックアップノード802が他のネットワークノードから受け取ったかどうかを判断することによって行われる。他のネットワークノードからの受け入れメッセージのそれぞれは、他のネットワークノードのそれぞれが所定の数(例えば $2f+1$)のECHOメッセージを受け入れたことを示す。受け入れメッセージは、例えば、コンセンサス処理300の第3のフェーズ330を参照しながら上述されたACCEPTメッセージであることが可能である。

【0124】

本明細書のいくつかの実装形態において、バックアップノード802が定足数検証に失敗すると(例えば、所定の閾値より少ないいくつかのACCEPTメッセージをバックアップノード802が受け取る)、バックアップノード802は、コンセンサス処理の第3のフェーズに対するバックアップノード802の重みが第1の値であると判断することができる。バックアップノード802が定足数検証にパスすると(例えば、所定の閾値以上のいくつかのACCEPTメッセージをバックアップノード802が受け取る)、バックアップノード802は、コンセンサス処理の第3のフェーズに対するバックアップノード802の重みが第2の値であると判断することができる。いくつかの実装形態において、第2の値は、第1の値より大きいものと判断される。例えば、バックアップノード802が定足数検証に失敗すると、バックアップノード802は、コンセンサス処理の第3のフェーズに対するバックアップノード802に対して重み値ゼロを割り当てられることが可能である。バックアップノード802が定足数検証にパスすると、バックアップノード802は、コンセンサス処理の第3のフェーズに対するバックアップノード802に対する重み値45%を割り当てられることが可能である。しかし、本明細書の代替実装形態において、バックアップノード802は、現在のエポックにおけるコンセンサス処理の第3のフェーズに対するバックアップノード802に任意の適切な値を割り当てることができる。

10

【0125】

810において、現在のエポックにおけるコンセンサス処理のフェーズに対してバックアップノード802の個々の重みを判断した後、バックアップノード802は、個々の重みに基づいてコンセンサス処理に対するバックアップノード802の加重和を判断する。本明細書のいくつかの実装形態において、加重和は、現在のエポックにおけるコンセンサス処理のフェーズのそれぞれと関連付けられたバックアップノードの個々の合計の合計である。例えば、第1のフェーズに対するバックアップノード802の第1の重み値が10%であり、第2のフェーズに対するバックアップノード802の第2の重み値が45%であり、第3のフェーズに対するバックアップノード802の第3の重み値が45%であるとバックアップノード802が判断した場合、バックアップノード802は加重和を100%であると判断する。別の例として、第1のフェーズに対するバックアップノード802の第1の重み値が10%であり、第2のフェーズに対するバックアップノード802の第2の重み値が45%であり、第3のフェーズに対するバックアップノード802の第3の重み値が0であるとバックアップノード802が判断すると、バックアップノード802は加重和が55%であると判断する。

20

30

【0126】

812において、バックアップノード802は、810において判断された加重和が所定の閾値に達しているか超過しているとバックアップノード802が判断すると、他のネットワークノード804にEPOCH_CHANGEメッセージを送る。例えば、バックアップノード802は、810において判断されたような加重和が100%に達すると、他のネットワークノード804にEPOCH_CHANGEメッセージを送ることができる。EPOCH_CHANGEメッセージは、現在の1次ノードによる現在のエポックから、バックアップノードが新しい1次ノードになる新しいエポックへの変更のための要求を示す。

【0127】

本明細書のいくつかの実装形態において、図9を参照すると、EPOCH_CHANGEメッセージは、<weight, epoch+1, ECHO{}, ACCEPT{}, j>というフォーマットを有し、ここで、「weight」は、コンセンサス処理に対する、810において以前に判断されたような、バックアップノード802の加重和を表し、「epoch+1」は、新しい1次ノードと関連付けられた新しいコンセンサスのラウンド(すなわち新しいエポック)を表し、「ECHO{}」は、コンセンサス処理の第2のフェーズ中にバックアップノード802が受け取ったECHOメッセージのセットを表し、「ACCEPT{}」は、コンセンサス処理の第3のフェーズ中にバックアップノード802が受け取ったACCEPTメッセージのセットを表し、「j」は、EPOCH_CHANGEメッセージを生成して送るネットワークノード(例えばバックアップノード802)を表す。いくつかの実装形態において、EPOCH_CHANGEメッセージは、例えば、追加のフィールドまたは異なるフィールドを含めることによって、異なるフォーマットを有することができる。

40

50

【0128】

図8を再度参照すると、814において、バックアップノード802以外のネットワークノード804は、バックアップノード802によって送られたEPOCH_CHANGEメッセージを検証する。いくつかの実装形態において、ネットワークノード804のそれぞれは、EPOCH_CHANGEメッセージ内の加重和が有効であるかどうかを検証することによって、EPOCH_CHANGEメッセージが有効であるかどうかを検証する。いくつかの実装形態において、EPOCH_CHANGEメッセージ内の加重和が有効であるかどうかを検証することは、EPOCH_CHANGEメッセージに含まれるECHOメッセージ内の署名のセットが有効であるかどうかを検証することを含む。例えば、ネットワークノード804のそれぞれは、公開鍵を使用してEPOCH_CHANGEメッセージを含んだECHOメッセージ内の秘密鍵署名のセットを認証することができる。

10

【0129】

816において、ネットワークノード804のそれぞれは、バックアップノード802によって送られたEPOCH_CHANGEメッセージが有効であることの検証に応答して、バックアップノード802にNEW_EPOCHメッセージを送る。NEW_EPOCHメッセージは、新しい1次ノードにバックアップノードになることについての承認を示す。例えば、ネットワークノード804によって送られたNEW_EPOCHメッセージは、バックアップノード802が新しいエポックにおける新しい1次ノードになることをネットワークノード804が承認するという指示を含む。一方で、ネットワークノード804のそれぞれは、他のネットワークノード804にもNEW_EPOCHメッセージを送る。

20

【0130】

図9を参照すると、NEW_EPOCHメッセージは、<epoch+1, i, j, seq, ec_digest>というフォーマットを有するものとして生成しており、ここで、「epoch+1」は、新しい1次ノードと関連付けられた新しいコンセンサスのラウンド(すなわち新しいエポック)を表し、「i」は、新しいエポックにおける新しい1次ノードを表し、「j」は、NEW_EPOCHメッセージを送るネットワークノード804を表し、「ec_digest」は、EPOCH_CHANGEメッセージのダイジェストを表す。いくつかの実装形態において、EPOCH_CHANGEメッセージのダイジェストは、EPOCH_CHANGEメッセージのハッシュ値を含む。いくつかの実装形態において、NEW_EPOCHメッセージは、例えば追加のフィールドまたは異なるフィールドを含めることによって、異なるフォーマットを有することができる。

30

【0131】

図8を再度参照すると、818において、バックアップノード802は、ネットワークノード804によって送られたNEW_EPOCHメッセージが有効であるかどうかを検証する。いくつかの実装形態において、バックアップノード802は、NEW_EPOCHメッセージ内のEPOCH_CHANGEメッセージのダイジェストが有効であるかどうかを検証することによってNEW_EPOCHメッセージを検証する。ダイジェストはEPOCH_CHANGEメッセージの情報を含むので、ダイジェストは、EPOCH_CHANGEメッセージ内の署名も含む。バックアップノード802は、EPOCH_CHANGEメッセージ内の署名のセットが有効であるかどうかを検証することによってEPOCH_CHANGEメッセージのダイジェストを検証することができる。

【0132】

820において、バックアップノード802は、818において判断されるような有効なNEW_EPOCHメッセージの数が所定の閾値を超過するかどうかを判断する。いくつかの実装形態において、所定の閾値は定足数(例えば $2f+1$)である。

40

【0133】

822において、バックアップノード802は、判断されたような有効なNEW_EPOCHメッセージの数が所定の閾値を超過したとの判断に応答して、バックアップノード802が新しいエポックにおける新しい1次ノードであると判断する。ネットワークノード804のそれぞれは、バックアップノード802が行うのと同じステップ818～ステップ822を行い、ネットワークノード804およびバックアップノード802は、ステップ818～ステップ822を並列に行うことができるということに留意されたい。例えば、ネットワークノード804のそれぞれは、他のネットワークノード804から送られたNEW_EPOCHメッセージのセットを検証し、有効な

50

NEW_EPOCHメッセージの数が所定の閾値を超過しているかどうかを判断し、新しい1次ノードを判断することができる。

【0134】

上述のようなエポック変更処理(例えば処理700または800)は、ブロックチェーンシステム全体の動作を改善し、ネットワークのボトルネックを緩和させるのに役立つ多くの特徴を含む。例えば、本明細書におけるエポック変更処理は、コンセンサス処理の3つのフェーズに個々の重みを割り当てること、3つのフェーズの個々の重みに基づいて加重和を判断すること、および加重和に基づいて新しい1次ノードを判断することを含む。ラウンドロビン方法の代わりに加重和に基づくエポック変更処理は、故障していない新しい1次ノードをタイムリに選ぶことを容易にすることができる。ラウンドロビン方法は、新しい1次ノードに対して一列に並んだ複数のネットワークノードが故障しているときに、1次ノードの頻繁な変更を引き起こす可能性がある。これは、故障していない1次ノードを見つける際にレイテンシまたは遅延をもたらすことによってブロックチェーンサービスに著しく影響を及ぼす。ラウンドロビン方法とは異なり、本明細書におけるエポック変更処理は、加重和に応じて新しい1次ノードを選択し、これは、故障していない新しい1次ノードを見つける際の時間を低減させることができる。これは、ブロックチェーンサービスを提供する際の全体的なブロックチェーンシステムの効率性をさらに改善することができる。

【0135】

ブロックチェーンネットワークの動作中、いくつかのネットワークノードの実行速度は、ネットワークジタリング(network jittering)、突然の電源障害、ディスク障害、および同様のものために、ほとんどのネットワークノードの実行速度より遅れることがある。このシナリオにおいて、システムにおけるネットワークノードの1/3超が、故障する可能性がある。BFTは、ネットワークノードの1/3未満がシステムの存続時間中に故障する場合の安全性および生存性をもたらす。しかし、これらの保証は、上述のようなシナリオにおいて上限を超過する可能性があるため、長寿命のシステムには不十分である。したがって、ネットワークノードの存続期間にわたってf個超の障害をシステムが許容できるように、故障しているネットワークノードを再度正しく動作させ、その後のコンセンサス処理に参加し続けるという回復処理が望ましい。さらに、説明される回復処理は、コンセンサス処理(例えばコンセンサス処理300または400)をまだ行っている1つまたは複数のネットワークノードを回復させることができ、ネットワークノード全体の間でコンセンサスが達せられるまで待つ必要はない。したがって、説明される回復処理は、システムのレイテンシをさらに低減させ、ブロックチェーンネットワークの効率性を改善することができる。

【0136】

図10は、本明細書の実装形態に従って実行されることが可能な分散システム(例えばブロックチェーンネットワーク102および212)のネットワークノード(例えばノード214または404)の回復処理を行うための処理1000の例を描写する。具体的には、図10は、本明細書によるネットワークノードの回復処理を行う方法1000の例示的实施形態を提示する図を示す。図10において示されたように、処理1000は、いくつかのフェーズおよびステージを含む。

【0137】

第1のフェーズ1010において、対象シーケンス番号 R_0 の対象取引を回復させることを望むネットワークノード(例えばネットワークノード R_0)は、ネットワークノードが回復されることになることを示す状態要求メッセージ(例えばQUERY_STATEメッセージ)を他のネットワークノードにマルチキャストする。状態要求メッセージは、ネットワークノード R_0 が回復させることを望む対象シーケンス番号を含むことができる。第2のフェーズ1020において、他のネットワークノードは、状態要求メッセージを受け取り、状態リプライメッセージ(例えばREPLY_STATEメッセージ)をネットワークノード R_0 に送る。第3のフェーズ1030において、ネットワークノード R_0 は、他のネットワークノードのそれぞれからのECHOメッセージを要求する要求メッセージ(例えばFETCH_ECHOメッセージ)を他のネットワークノードに送る。ECHOメッセージは、図3～図6を参照しながら上記で論じられたような、コンセ

10

20

30

40

50

ンサス処理300の第2のフェーズ320において個々の他のネットワークノードによって送られた同じECHOメッセージであることが可能である。第4のフェーズ1040において、他のネットワークノードのそれぞれは、FETCH_ECHOメッセージに応答して、ネットワークノードR₀にECHOメッセージを送る。第5のフェーズ1050において、ネットワークノードR₀はECHOメッセージを検証し、例えば図4を参照しながら上述されたような再構築技法の例に従って、ECコードに従って対象取引を回復させる。第6のフェーズ1060において、ネットワークノードR₀は、ネットワークノードが回復されたことを示すACCEPTメッセージを他のネットワークノードに送る。

【0138】

処理1000は、例証のためだけに4つのネットワークノードと共に実行されものとして示されているということに留意されたい。処理1000は、任意の適切な数のネットワークノードと共に実行されることが可能である。処理1000、QUERY_STATEメッセージのフォーマット、およびREPLY_STATEメッセージのフォーマットは、図11～図12を参照しながら、より詳細に下記で論じられることになる。

【0139】

図11は、本明細書の実装形態に従って実行されることが可能な分散システム(例えばブロックチェーンネットワーク102または212)におけるネットワークノードの回復処理を行うための処理1100の例を描写する。いくつかの実装形態において、処理1100は、1つまたは複数のコンピューティングデバイスを使用して実行される1つまたは複数のコンピュータ実行可能プログラムを使用して行われることが可能である。表現の明瞭さのために、続く説明は、全体的に、本説明における他の図の背景において方法1100を説明する。方法1100は、例えば任意の適切なシステム、環境、ソフトウェア、およびハードウェア、またはシステム、環境、ソフトウェア、およびハードウェアの組合せによって、必要に応じて行われることが可能であるということが理解されよう。いくつかの実装形態において、方法1100の様々なステップは、並行に、組み合わせて、ループさせて、または任意の順序で実行されることが可能である。

【0140】

処理1100は、1106において始まり、ここで、ネットワークノード1102は、他のネットワークノード1104に状態要求メッセージをマルチキャストする。状態要求メッセージは、対象シーケンス番号の対象取引をネットワークノード1102が回復させることになるという指示を含む。ネットワークノード1102は、1次ノードまたはバックアップノードであることが可能である。

【0141】

本明細書のいくつかの実装形態において、図12を参照すると、QUERY_STATEメッセージは、状態要求メッセージの例として、<j, seq>というフォーマットを有し、ここで、「j」は、QUERY_STATEメッセージを送るネットワークノード1102を表し、「seq」は、現在のコンセンサス処理におけるネットワークノード1102に対する最大のシーケンス番号または最も新しいシーケンス番号を表す。いくつかの実装形態において、QUERY_STATEメッセージは、例えば追加のフィールドまたは異なるフィールドを含めることによって、異なるフォーマットを有することができる。

【0142】

他のネットワークノード1104にQUERY_STATEメッセージをブロードキャストすることによって、ネットワークノード1102は、他のネットワークノードの最も新しいシーケンス番号をネットワークノード1102に送ることを、他のネットワークノード1104に要求しており、このことにより、ブロックチェーンシステムの最新のブロック情報を入手する。また、ブロックチェーンシステム全体の最新のブロック情報を入手することによって、ネットワークノード1102は、システム全体の最新のステータスに同期できる可能性があり、このことにより、ネットワークノード1102自体を回復させ、コンセンサス処理に参加し続ける。

【0143】

図11を再度参照すると、1108において、他のネットワークノード1104のそれぞれは、状

10

20

30

40

50

態要求メッセージの受領に応答して、状態リプライメッセージ(例えばREPLY_STATEメッセージ)をネットワークノード1102に送る。いくつかの実装形態において、状態リプライメッセージは、ネットワークノード1104と関連付けられた以前のシーケンス番号を含む。

【0144】

いくつかの実装形態において、図12を参照すると、REPLY_STATEメッセージは、状態リプライメッセージの例として、<j, last_seq>というフォーマットを有し、ここで、「j」は、REPLY_STATEメッセージを送るネットワークノード1104を表し、「last_seq」は、現在のコンセンサス処理におけるネットワークノード1104に対する以前のシーケンス番号を表す。いくつかの実装形態において、REPLY_STATEメッセージは、例えば追加のフィールドまたは異なるフィールドを含めることによって、異なるフォーマットを有することができる。

10

【0145】

図11を再度参照すると、1110において、ネットワークノード1102は、受け取られた状態リプライメッセージの数が所定の閾値を超過するかどうかを判断する。例えば、ネットワークノード1102は、受け取られたREPLY_STATEメッセージの数が定足数(例えば $2f+1$ または $n-f$)を超過するかどうかを判断することができる。いくつかの実装形態において、ネットワークノード1102は、受け取られたREPLY_STATEメッセージの定足数が同一のシーケンス番号を含むかどうかをさらに判断する。受け取られたREPLY_STATEメッセージの定足数は、ネットワークノード1104の大多数がシステム全体の共通の状態に合意することを意味する同一のシーケンス番号を含む。

20

【0146】

1112において、ネットワークノード1102は、ネットワークノード1104から受け取られた同一のシーケンス番号を含む状態リプライメッセージの数が所定の閾値を超過するとネットワークノード1102が判断すると、同一のシーケンス番号に基づいて対象シーケンス番号を判断する。例えば、ネットワークノード1102は、対象シーケンス番号が同一のシーケンス番号(例えば「last_seq」)の増分(例えば「last_seq+1」)であると判断することができる。

【0147】

1114において、ネットワークノード1102は、要求メッセージ(例えばFETCH_ECHOメッセージ)を他のネットワークノード1104に送る。FETCH_ECHOメッセージは、他のネットワークノード1104のそれぞれからECHOメッセージを要求するために、ネットワークノード1102によって送られる。図3～図6を参照しながら上記で論じられたように、ECHOメッセージは、対象取引に対するネットワークノード1104間のコンセンサスを実現するためにネットワークノード1104によって送信されたメッセージである。ECHOメッセージは、対象取引の一部(例えばECブロック)、およびECHOメッセージを送るネットワークノード1104の署名を含む。

30

【0148】

いくつかの実装形態において、図12を参照すると、FETCH_ECHOメッセージは、要求メッセージの例として、<j, last_seq+1>というフォーマットを有し、ここで、「j」は、FETCH_ECHOメッセージを送るネットワークノード1102を表し、「last_seq+1」は、ネットワークノード1102が要求している他のネットワークノード1104からのECHOメッセージと関連付けられた対象シーケンス番号を表す。いくつかの実装形態において、FETCH_ECHOメッセージは、例えば追加のフィールドまたは異なるフィールドを含めることによって、異なるフォーマットを有することができる。

40

【0149】

本明細書で論じられるようなFETCH_ECHOメッセージは、他のネットワークノード1104からの最も新しいシーケンス番号または最大のシーケンス番号を含むECHOメッセージを要求するために、ネットワークノード1102によって送られる。他のネットワークノード1104からの最も新しいシーケンス番号または最大のシーケンス番号を含むECHOメッセージを収集することによって、ネットワークノード1102は、最も新しいシーケンス番号と関連付けら

50

れた最も新しい状態に回復させることができる可能性がある。

【0150】

図11を再度参照すると、1116において、ネットワークノード1104のそれぞれは、FETCH_ECHOメッセージの受領に応答して、ネットワークノード1102にECHOメッセージを送る。いくつかの実装形態において、ネットワークノード1104のそれぞれは、ネットワークノード1102にECHOメッセージを送る前にFETCH_ECHOメッセージを検証する。いくつかの実装形態において、ネットワークノード1104のそれぞれは、FETCH_ECHOメッセージに含まれるシーケンス番号が、ネットワークノード1104と関連付けられた最も新しいシーケンス番号を超過しているかどうかを判断することによって、FETCH_ECHOメッセージを検証する。FETCH_ECHOメッセージに含まれるシーケンス番号が、ネットワークノード1104と関連付けられた最も新しいシーケンス番号に等しい場合、ネットワークノード1104は、FETCH_ECHOメッセージが有効であること、およびECHOメッセージがネットワークノード1102に送られることになると判断する。FETCH_ECHOメッセージに含まれるシーケンス番号が、ネットワークノード1104と関連付けられた最も新しいシーケンス番号を超過する場合、ネットワークノード1104は、FETCH_ECHOメッセージが無効であるということ、およびECHOメッセージがネットワークノード1102に送られることはないということを判断する。

10

【0151】

1118において、ネットワークノード1102は、ネットワークノード1104によって送られたECHOメッセージが有効であるかどうかを検証する。いくつかの実装形態において、ネットワークノード1102は、マークル木を使用してECHOメッセージを検証する。例えば、ネットワークノード1102は、ECHOメッセージに含まれるデータを使用してマークル木を再構築し、再構築されたマークル木の再構築されたルートハッシュ値を判断することができる。ネットワークノード1102は、次に、再構築されたルートハッシュ値を、ECHOメッセージに含まれるルートハッシュ値と比較することができる。再構築されたルートハッシュ値が、ECHOメッセージに含まれるルートハッシュ値にマッチする場合、ネットワークノード1102は、ECHOメッセージが有効であると判断する。再構築されたルートハッシュ値が、ECHOメッセージに含まれるルートハッシュ値にマッチしない場合、ネットワークノード1102は、ECHOメッセージが無効であると判断し、無効のECHOメッセージを廃棄することができる。

20

【0152】

いくつかの実装形態において、ネットワークノード1102は、ECHOメッセージ内の署名が有効であるかどうかをさらに検証することによって、ECHOメッセージが有効であるかどうかを検証する。例えば、ネットワークノード1102は、秘密鍵とペアになる公開鍵を使用してECHOメッセージ内の秘密鍵署名を認証して、署名を検証することができる。

30

【0153】

1120において、ネットワークノード1102は、他のネットワークノード1104から受け取られた有効なECHOメッセージの数が所定の閾値を超過するかどうかを判断する。例えば、ネットワークノード1102は、他のネットワークノード1104から受け取られた有効なECHOメッセージの数が定足数(例えば $2f+1$)を超過するかどうかを判断することができる。

【0154】

1122において、ネットワークノード1102は、有効なECHOメッセージの数が所定の閾値を超過するとの判断に回答して、対象シーケンス番号の対象取引を回復させる。いくつかの実装形態において、ネットワークノード1102は、いくつかの有効なECHOメッセージに含まれるデータを使用して対象取引を回復させる。例えば、ネットワークノード1102は、ECコードに従って対象取引を再構築するために、ECHOメッセージに含まれるECブロックのサブセットを取得することができる。

40

【0155】

1124において、ネットワークノード1102は、対象取引を回復させた後、他のネットワークノード1104にACCEPTメッセージをマルチキャストする。例えば、ネットワークノード1102は、対象取引を成功裏に再構築した後、他のネットワークノード1104にACCEPTメッセージをマルチキャストすることができる。いくつかの実装形態において、ACCEPTメッセージ

50

は、ECHOメッセージ内の署名のセット、および対象シーケンス番号を含む。署名および対象シーケンス番号を含むACCEPTメッセージを他のネットワークノード1104に送ることによって、ネットワークノード1102は、ネットワークノード1102が回復し、システムの最新の状態に同期したことを他のネットワークノード1104に示す。

【0156】

本明細書において上述されたような回復処理は、回復処理を実行するコンピュータの動作を改善し、ネットワークのボトルネックを緩和するのに役立つ多くの特徴を含む。例えば、本明細書における回復処理は、新しい1次ノードとして使用されるネットワークノードによって状態要求メッセージを送ること、他のネットワークノードから状態リプライメッセージを受け取ること、および他のネットワークノードからECHOメッセージを要求するためにネットワークノードによってFETCH_ECHOメッセージを送ること、を含む動作を含む。これらの動作は、故障しているネットワークノードの回復処理が、他の故障していないネットワークノード間のコンセンサス処理の正常動作に支障を来さないように行われる。これは、回復処理の複雑性を低減させることによって、故障しているネットワークノードを回復させるためのコンピューティングリソースおよびネットワークリソースを節約することを容易にする。

【0157】

図13を参照すると、図13は、本明細書の実装形態によるコンセンサス装置1300のモジュールを示す図である。コンセンサスを実現するための装置1300は、ブロックチェーン技術に基づくコンセンサスシステムに使用されることが可能である。例えば、装置1300は、図1～図6に示された実装形態に対応することが可能である。装置1300は、ブロックチェーンネットワーク内の1次ノードにおいて実行されることが可能である。装置1300は、取引要求を受け取るように構成された受信器または受信ユニット1302と、取引要求を使用してECコードに従っていくつかの消去コード(EC)ブロックを生成するように構成された生成ユニット1304と、いくつかの第1のメッセージを1つまたは複数のバックアップノードにそれぞれ送るように構成された送信器または送信ユニット1306であって、いくつかの第1のメッセージのそれぞれが、いくつかのECブロックと関連付けられた合成ハッシュ値を含む、送信器または送信ユニット1306と、バックアップノードのうちの少なくとも1つから少なくとも1つの第2のメッセージを受け取るようにさらに構成された受信器または受信ユニット1302であって、少なくとも1つの第2のメッセージが、いくつかの第1のメッセージのうちの1つ、およびいくつかの第1のメッセージのうちの1つと関連付けられたバックアップノードのうちの少なくとも1つの署名を含む、受信器または受信ユニット1302と、バックアップノードのうちの少なくとも1つからの少なくとも1つの第2のメッセージの受領にตอบสนองして、少なくとも1つの第2のメッセージが有効であるかどうかを検証するように構成された検証ユニット1308と、有効な第2のメッセージの数が所定の閾値を超過するかどうかを判断するように構成された判断ユニット1310と、有効な第2のメッセージの数が所定の閾値を超過するとの判断にตอบสนองして、ECコードに従っていくつかの有効な第2のメッセージのサブセットに基づいて取引要求を再構築するように構成された再構築ユニット1312と、取引要求が成功裏に再構築されたとの判断にตอบสนองして、他のネットワークノードに第3のメッセージを送るようさらに構成された送信器または送信ユニット1306であって、第3のメッセージが、有効な第2のメッセージ内にある署名のセットを含む、送信器または送信ユニット1306と、バックアップノードのうちの少なくとも1つから少なくとも1つの第3のメッセージを受け取るようにさらに構成された受信器または受信ユニット1302と、同一である所定の数の第3のメッセージの受領にตอบสนองして、取引要求を実行するように構成された実行ユニット1314と、を含む。

【0158】

任意選択の実装形態において、取引要求は、シーケンス番号と関連付けられる。

【0159】

任意選択の実装形態において、ECコードに従って複数のECブロックを生成することは、ECコードを使用して取引要求をECメッセージに変換することと、ECメッセージをいくつか

10

20

30

40

50

のECブロックに分割することと、を含む。

【0160】

任意選択の実装形態において、いくつかのECブロックの合成ハッシュ値は、ハッシュ木を使用して生成される。

【0161】

任意選択の実装形態において、ハッシュ木はマークル木を含み、合成ハッシュ値はマークル木のルートハッシュ値である。

【0162】

任意選択の実装形態において、いくつかの第1のメッセージのうちの1つと関連付けられたバックアップノードのうちの少なくとも1つの署名は、いくつかの第1のメッセージのうちの1つと関連付けられたバックアップノードのうちの少なくとも1つの秘密鍵署名を含む。

10

【0163】

任意選択の実装形態において、少なくとも1つの第2のメッセージは、いくつかのECブロックのうちの少なくとも1つをさらに含む。

【0164】

任意選択の実装形態において、少なくとも1つの第2のメッセージが有効であるかどうかを検証することは、少なくとも1つの第2のメッセージ内のいくつかのECブロックのうちの少なくとも1つを使用して、再構築されたハッシュ木を生成することと、再構築されたハッシュ木の再構築された合成ハッシュ値を判断することと、再構築された合成ハッシュ値が少なくとも1つの第2のメッセージ内の合成ハッシュ値にマッチするかどうかを判断することと、を含む。

20

【0165】

任意選択の実装形態において、判断ユニット1310は、再構築された合成ハッシュ値が第2のメッセージ内の合成ハッシュ値にマッチするとの判断に応答して、少なくとも1つの第2のメッセージが有効であると判断するようにさらに構成される。

【0166】

任意選択の実装形態において、同一である所定の数の第3のメッセージは、署名の同一セットを有する所定の数の第3のメッセージを含む。

【0167】

図13は、コンセンサス装置1300の内部機能モジュールおよび構造を示す概略図である。実行体は本質的に電子デバイスであることが可能であり、電子デバイスは、少なくとも1つのプロセッサと、少なくとも1つのプロセッサの実行可能命令を格納するように構成されたメモリと、を含む。

30

【0168】

少なくとも1つのプロセッサは、取引要求を受け取ることと、取引要求を使用してECコードに従っていくつかの消去コード(EC)ブロックを生成することと、いくつかの第1のメッセージを1つまたは複数のバックアップノードにそれぞれ送ることと、いくつかの第1のメッセージのそれぞれが、いくつかのECブロックと関連付けられた合成ハッシュ値を含む、送ることと、バックアップノードのうちの少なくとも1つから少なくとも1つの第2のメッセージを受け取ることと、少なくとも1つの第2のメッセージが、いくつかの第1のメッセージのうちの1つ、およびいくつかの第1のメッセージのうちの1つと関連付けられたバックアップノードのうちの少なくとも1つの署名を含む、受け取ることと、バックアップノードのうちの少なくとも1つからの少なくとも1つの第2のメッセージの受領に応答して、少なくとも1つの第2のメッセージが有効であるかどうかを検証することと、有効な第2のメッセージの数が所定の閾値を超過するかどうかを判断することと、有効な第2のメッセージの数が所定の閾値を超過するとの判断に応答して、ECコードに従っていくつかの有効な第2のメッセージのサブセットに基づいて取引要求を再構築することと、取引要求が成功裏に再構築されたとの判断に応答して、他のネットワークノードに第3のメッセージを送ることと、第3のメッセージが、有効な第2のメッセージ内にある

40

50

署名のセットを含む、送ることと、バックアップノードのうちの少なくとも1つから少なくとも1つの第3のメッセージを受け取ることと、同一である所定の数の第3のメッセージの受領に応答して、取引要求を実行することと、を行うように構成される。

【0169】

任意選択で、取引要求は、シーケンス番号と関連付けられる。

【0170】

任意選択で、ECコードに従って複数のECブロックを生成することは、ECコードを使用して取引要求をECメッセージに変換することと、ECメッセージをいくつかのECブロックに分割することと、を含む。

【0171】

任意選択で、いくつかのECブロックの合成ハッシュ値は、ハッシュ木を使用して生成される。

【0172】

任意選択で、ハッシュ木はマークル木を含み、合成ハッシュ値はマークル木のルートハッシュ値である。

【0173】

任意選択で、いくつかの第1のメッセージのうちの1つと関連付けられたバックアップノードのうちの少なくとも1つの署名は、いくつかの第1のメッセージのうちの1つと関連付けられたバックアップノードのうちの少なくとも1つの秘密鍵署名を含む。

【0174】

任意選択で、少なくとも1つの第2のメッセージは、いくつかのECブロックのうちの少なくとも1つをさらに含む。

【0175】

任意選択で、少なくとも1つの第2のメッセージが有効であるかどうかを検証することは、少なくとも1つの第2のメッセージ内のいくつかのECブロックのうちの少なくとも1つを使用して、再構築されたハッシュ木を生成することと、再構築されたハッシュ木の再構築された合成ハッシュ値を判断することと、再構築された合成ハッシュ値が少なくとも1つの第2のメッセージ内の合成ハッシュ値にマッチするかどうかを判断することと、を含む。

【0176】

任意選択で、少なくとも1つのプロセッサは、再構築された合成ハッシュ値が第2のメッセージ内の合成ハッシュ値にマッチするとの判断に応答して、少なくとも1つの第2のメッセージが有効であると判断するようにさらに構成される。

【0177】

任意選択で、同一である所定の数の第3のメッセージは、署名の同一セットを有する所定の数の第3のメッセージを含む。

【0178】

図14を参照すると、図14は、本明細書の実装形態によるコンセンサス装置1400のモジュールを示す図である。コンセンサスを実現するための装置1400は、ブロックチェーン技術に基づくコンセンサスシステムに使用されることが可能である。装置1400は、図1～図6に示された実装形態に対応することが可能である。例えば、装置1400は、ブロックチェーンネットワークのバックアップノード内で実行されることが可能である。装置1400は、1次ノードから第1のメッセージを受け取るように構成された受信器または受信ユニット1402であって、第1のメッセージが、いくつかのECブロックと関連付けられた合成ハッシュ値を含み、いくつかのECブロックが、取引要求を使用してECコードに従って1次ノードによって生成される、受信器または受信ユニット1402と、第1のメッセージの受領に응答して、他のネットワークノードに第2のメッセージをバックアップノードによって送るように構成された送信器または送信ユニット1404であって、第2のメッセージが、第1のメッセージおよび第1のメッセージと関連付けられたバックアップノードの署名を含む、送信器または送信ユニット1404と、このバックアップノード以外の少なくとも1つのバックアップ

10

20

30

40

50

ノードから少なくとも1つの第2のメッセージを受け取るようにさらに構成された受信器または受信ユニット1402と、少なくとも1つのバックアップノードからの少なくとも1つの第2のメッセージの受領に応答して、少なくとも1つの第2のメッセージが有効であるかどうかを検証するように構成された検証ユニット1406と、有効な第2のメッセージの数が所定の閾値を超過するかどうかを判断するように構成された判断ユニット1408と、有効な第2のメッセージの数が所定の閾値を超過するとの判断に応答して、ECコードに従っていくつかの有効な第2のメッセージのサブセットに基づいて取引要求を再構築するように構成された再構築ユニット1410と、取引要求が成功裏に再構築されたとの判断に応答して、他のネットワークノードに第3のメッセージを送るように構成された送信器または送信ユニット1404であって、第3のメッセージが、有効な第2のメッセージ内にある署名のセットを含む、送信器または送信ユニット1404と、バックアップノードのうちの少なくとも1つから少なくとも1つの第3のメッセージを受け取るようにさらに構成された受信器または受信ユニット1402と、同一である所定の数の第3のメッセージの受領に応答して、取引要求を実行するように構成された実行ユニット1412と、を含む。

10

【0179】

任意選択の実装形態において、ECコードに従って複数のECブロックを生成することは、ECコードを使用して取引要求をECメッセージに変換することと、ECメッセージをいくつかのECブロックに分割することと、を含む。

【0180】

任意選択の実装形態において、複数のECブロックの合成ハッシュ値は、ハッシュ木を使用して生成される。

20

【0181】

任意選択の実装形態において、ハッシュ木はマール木を含み、合成ハッシュ値はマール木のルートハッシュ値である。

【0182】

任意選択の実装形態において、第1のメッセージと関連付けられたバックアップノードの署名は、第1のメッセージと関連付けられたバックアップノードの秘密鍵署名を含む。

【0183】

任意選択の実装形態において、少なくとも1つの第2のメッセージは、いくつかのECブロックのうちの少なくとも1つをさらに含む。

30

【0184】

任意選択の実装形態において、少なくとも1つの第2のメッセージが有効であるかどうかを検証することは、少なくとも1つの第2のメッセージ内のいくつかのECブロックのうちの少なくとも1つを使用して、再構築されたハッシュ木を生成することと、再構築されたハッシュ木の再構築された合成ハッシュ値を判断することと、再構築された合成ハッシュ値を少なくとも1つの第2のメッセージ内の合成ハッシュ値と比較することと、再構築された合成ハッシュ値が少なくとも1つの第2のメッセージ内の合成ハッシュ値にマッチするかどうかを判断することと、を含む。

【0185】

任意選択の実装形態において、判断ユニット1408は、再構築された合成ハッシュ値が第2のメッセージ内の合成ハッシュ値にマッチするとの判断に応答して、少なくとも1つの第2のメッセージが有効であると判断するようにさらに構成される。

40

【0186】

任意選択の実装形態において、同一である所定の数の第3のメッセージは、署名の同一セットを有する所定の数の第3のメッセージを含む。

【0187】

図14は、コンセンサス装置1400の内部機能モジュールおよび構造を示す概略図である。実行体は本質的に電子デバイスであることが可能であり、電子デバイスは、少なくとも1つのプロセッサと、少なくとも1つのプロセッサの実行可能命令を格納するように構成されたメモリと、を含む。

50

【0188】

少なくとも1つのプロセッサは、1次ノードから第1のメッセージを受け取ることであって、第1のメッセージが、いくつかのECブロックと関連付けられた合成ハッシュ値を含み、いくつかのECブロックが、取引要求を使用してECコードに従って1次ノードによって生成される、受け取ることと、第1のメッセージの受領に応答して、他のネットワークノードに第2のメッセージをバックアップノードによって送ることと、第2のメッセージが、第1のメッセージおよび第1のメッセージと関連付けられたバックアップノードの署名を含む、送ることと、このバックアップノード以外の少なくとも1つのバックアップノードから少なくとも1つの第2のメッセージを受け取ることと、少なくとも1つのバックアップノードからの少なくとも1つの第2のメッセージの受領に応答して、少なくとも1つの第2のメッセージが有効であるかどうかを検証することと、有効な第2のメッセージの数が所定の閾値を超過するかどうかを判断することと、有効な第2のメッセージの数が所定の閾値を超過するとの判断に応答して、ECコードに従っていくつかの有効な第2のメッセージのサブセットに基づいて取引要求を再構築することと、取引要求が成功裏に再構築されたとの判断に応答して、他のネットワークノードに第3のメッセージを送ることと、第3のメッセージが、有効な第2のメッセージ内にある署名のセットを含む、送ることと、バックアップノードのうちの少なくとも1つから少なくとも1つの第3のメッセージを受け取ることと、同一である所定の数の第3のメッセージの受領に応答して、取引要求を実行することと、を行うように構成される。

10

【0189】

20

任意選択で、ECコードに従って複数のECブロックを生成することは、ECコードを使用して取引要求をECメッセージに変換することと、ECメッセージをいくつかのECブロックに分割することと、を含む。

【0190】

任意選択で、複数のECブロックの合成ハッシュ値は、ハッシュ木を使用して生成される。

【0191】

任意選択で、ハッシュ木はマークル木を含み、合成ハッシュ値はマークル木のルートハッシュ値である。

【0192】

30

任意選択で、第1のメッセージと関連付けられたバックアップノードの署名は、第1のメッセージと関連付けられたバックアップノードの秘密鍵署名を含む。

【0193】

任意選択で、少なくとも1つの第2のメッセージは、いくつかのECブロックのうちの少なくとも1つをさらに含む。

【0194】

任意選択で、少なくとも1つの第2のメッセージが有効であるかどうかを検証することは、少なくとも1つの第2のメッセージ内のいくつかのECブロックのうちの少なくとも1つを使用して、再構築されたハッシュ木を生成することと、再構築されたハッシュ木の再構築された合成ハッシュ値を判断することと、再構築された合成ハッシュ値を少なくとも1つの第2のメッセージ内の合成ハッシュ値と比較することと、再構築された合成ハッシュ値が少なくとも1つの第2のメッセージ内の合成ハッシュ値にマッチするかどうかを判断することと、を含む。

40

【0195】

任意選択で、少なくとも1つのプロセッサは、再構築された合成ハッシュ値が第2のメッセージ内の合成ハッシュ値にマッチするとの判断に応答して、少なくとも1つの第2のメッセージが有効であると判断するようにさらに構成される。

【0196】

任意選択で、同一である所定の数の第3のメッセージは、署名の同一セットを有する所定の数の第3のメッセージを含む。

50

【0197】

図15を参照すると、図15は、本明細書の実装形態による1次ノード変更装置1500のモジュールを示す図である。1次ノードを変更するための装置1500は、ブロックチェーン技術に基づくコンセンサスシステムに使用されることが可能である。装置1500は、図7～図9において示された実装形態に対応することが可能である。例えば、装置1500は、ブロックチェーンネットワークのバックアップノードにおいて実行されることが可能である。装置1500は、エポック変更が行われる必要があると判断するように構成された判断ユニット1502であって、エポック変更は、現在の1次ノードによる現在のエポックから、新しい1次ノードによる新しいエポックへの変更を引き起こし、現在のエポックは、1次ノードを使用していくつかのネットワークノード間のコンセンサスを実現するためのコンセンサス処理を含み、コンセンサス処理が3つのフェーズを含む、判断ユニット1502と、現在のエポックにおけるコンセンサス処理の3つのフェーズのそれぞれと関連付けられたバックアップノードの個々の重みを判断するようにさらに構成された判断ユニット1502であって、重みが、新しい1次ノードになるためのバックアップノードの資格の測定基準である、判断ユニット1502と、現在のエポックにおける3つのフェーズのそれぞれと関連付けられたバックアップノードの個々の重みに基づいてバックアップノードに対する加重和を判断するようにさらに構成された判断ユニット1502と、加重和が第1の所定の閾値に達したとの判断に
10
20
30
応答して、このネットワークノード以外のいくつかのネットワークノードにEPOCH_CHANGEメッセージを送るように構成された送信器または送信ユニット1504であって、EPOCH_CHANGEメッセージが、現在の1次ノードによる現在のエポックから、バックアップノードが新しい1次ノードである新しいエポックへの変更のための要求を示し、EPOCH_CHANGEメッセージがバックアップノードの加重和を含む、送信器または送信ユニット1504と、このバックアップノード以外のいくつかのネットワークノードのうちの少なくとも1つから少なくとも1つのNEW_EPOCHメッセージを受け取るように構成された受信器または受信ユニット1506であって、NEW_EPOCHメッセージが、新しい1次ノードにバックアップノードがなることについての承認を示す、受信器または受信ユニット1506と、少なくとも1つのNEW_EPOCHメッセージが有効であるかどうかを検証するように構成された検証ユニット1508と、少なくとも1つのNEW_EPOCHメッセージからの有効なNEW_EPOCHメッセージの数が第2の所定の閾値を超過するかどうかを判断するようにさらに構成された判断ユニット1502と、有効なNEW_EPOCHメッセージの数が第2の所定の閾値を超過するとの判断に
40
50
応答して、バックアップノードが新しいエポックにおける新しい1次ノードであると判断するようにさらに構成された判断ユニット1502と、を含む。

【0198】

任意選択の実装形態において、現在のエポックにおけるコンセンサス処理の3つのフェーズのそれぞれと関連付けられたバックアップノードの個々の重みを判断することは、コンセンサス処理の第1のフェーズに対するバックアップノードの重みが第1の値であると判断することを含む。

【0199】

任意選択の実装形態において、現在のエポックにおけるコンセンサス処理の3つのフェーズのそれぞれと関連付けられたバックアップノードの個々の重みを判断することは、現在のエポックにおけるコンセンサス処理の第2のフェーズにおいて定足数検証に失敗したとの判断に
40
50
応答して、コンセンサス処理の第2のフェーズに対するバックアップノードの重みが第1の値であると判断することと、現在のエポックにおけるコンセンサス処理の第2のフェーズにおいて定足数検証に成功したとの判断に
60
70
応答して、コンセンサス処理の第2のフェーズに対するバックアップノードの重みが第2の値であると判断することであって、第1の値が第2の値より小さい、判断することと、を含む。

【0200】

任意選択の実装形態において、ネットワークノードに対する第2のフェーズにおける定足数検証は、他のネットワークノードから所定の数のECHOメッセージを受け取ることを含む。

10

20

30

40

50

【0201】

任意選択の実装形態において、現在のエポックにおけるコンセンサス処理の3つのフェーズのそれぞれと関連付けられたバックアップノードの個々の重みを判断することは、現在のエポックにおけるコンセンサス処理の第3のフェーズにおいて定足数検証に失敗したとの判断に応答して、コンセンサス処理の第3のフェーズに対するバックアップノードの重みが第3の値であると判断することと、現在のエポックにおけるコンセンサス処理の第3のフェーズにおいて定足数検証に成功したとの判断に応答して、コンセンサス処理の第3のフェーズに対するバックアップノードの重みが第4の値であると判断することであって、第3の値が第4の値より小さい、判断することと、を含む。

【0202】

任意選択の実装形態において、ネットワークノードに対する第3のフェーズにおける定足数検証は、他のネットワークノードから所定の数の受け入れメッセージを受け取ることであって、他のネットワークノードからの受け入れメッセージのそれぞれが、他のネットワークノードのそれぞれが所定の数のECHOメッセージを受け入れたことを示す、受け取ることを含む。

【0203】

任意選択の実装形態において、EPOCH_CHANGEメッセージは、いくつかのネットワークノードからのネットワークノードのセットと関連付けられた署名のセットをさらに含み、NEW_EPOCHメッセージは、EPOCH_CHANGEメッセージのダイジェストを含む。

【0204】

任意選択の実装形態において、少なくとも1つの有効なNEW_EPOCHメッセージが有効であるかどうかを検証することは、少なくとも1つのNEW_EPOCHメッセージ内のEPOCH_CHANGEメッセージのダイジェストが有効であるかどうかを検証することを含み、少なくとも1つのNEW_EPOCHメッセージ内のEPOCH_CHANGEメッセージのダイジェストが有効であるかどうかを検証することは、EPOCH_CHANGEメッセージ内の署名のセットが有効であるかどうかを検証することを含む。

【0205】

任意選択の実装形態において、エポック変更が行われる必要があると判断することは、所定の期間内に古いエポックにおいてコンセンサスが実現されなかったとの判断に応答して、エポック変更が行われる必要があると判断することを含む。

【0206】

任意選択の実装形態において、1次ノード変更装置1500は、新しい1次ノードによる新しいエポックにおいて動作するように構成された動作ユニット1510であって、新しいエポックが、新しい1次ノードを使用して複数のネットワークノード間のコンセンサスを実現するためのコンセンサス処理を含む、動作ユニット1510をさらに含む。

【0207】

図15は、1次ノード変更装置1500の内部機能モジュールおよび構造を示す概略図である。実行体は本質的に電子デバイスであることが可能であり、電子デバイスは、少なくとも1つのプロセッサと、少なくとも1つのプロセッサの実行可能命令を格納するように構成されたメモリと、を含む。

【0208】

少なくとも1つのプロセッサは、エポック変更が行われる必要があると判断することであって、エポック変更が、現在の1次ノードによる現在のエポックから、新しい1次ノードによる新しいエポックへの変更を引き起こし、現在のエポックが、1次ノードを使用していくつかのネットワークノード間のコンセンサスを実現するためのコンセンサス処理を含み、コンセンサス処理が3つのフェーズを含む、判断することと、現在のエポックにおけるコンセンサス処理の3つのフェーズのそれぞれと関連付けられたバックアップノードの個々の重みを判断することであって、重みが、新しい1次ノードになるためのバックアップノードの資格の測定基準である、判断することと、現在のエポックにおける3つのフェーズのそれぞれと関連付けられたバックアップノードの個々の重みに基づいてバックアッ

10

20

30

40

50

プノードに対する加重和を判断することと、加重和が第1の所定の閾値に達したとの判断に
応答して、このネットワークノード以外のいくつかのネットワークノードにEPOCH_CHAN
GEメッセージを送ることと、EPOCH_CHANGEメッセージが、現在の1次ノードによる
現在のエポックから、バックアップノードが新しい1次ノードである新しいエポックへの
変更のための要求を示し、EPOCH_CHANGEメッセージがバックアップノードの加重和を含
む、送ることと、このバックアップノード以外のいくつかのネットワークノードのうちの少
なくとも1つから少なくとも1つのNEW_EPOCHメッセージを受け取ることであり、NEW_EPO
CHメッセージが、新しい1次ノードにバックアップノードになることについての承認を示
す、受け取ることと、少なくとも1つのNEW_EPOCHメッセージが有効であるかどうかを検証
することと、少なくとも1つのNEW_EPOCHメッセージからの有効なNEW_EPOCHメッセージの
数が第2の所定の閾値を超過するかどうかを判断することと、有効なNEW_EPOCHメッセ
ージの数が第2の所定の閾値を超過するとの判断に応答して、バックアップノードが新し
いエポックにおける新しい1次ノードであると判断することと、を行うように構成される。

10

【0209】

任意選択で、現在のエポックにおけるコンセンサス処理の3つのフェーズのそれぞれと
関連付けられたバックアップノードの個々の重みを判断することは、コンセンサス処理の
第1のフェーズに対するバックアップノードの重みが第1の値であると判断することを含む。

【0210】

任意選択で、現在のエポックにおけるコンセンサス処理の3つのフェーズのそれぞれと
関連付けられたバックアップノードの個々の重みを判断することは、現在のエポックにお
けるコンセンサス処理の第2のフェーズにおいて定足数検証に失敗したとの判断に応答し
て、コンセンサス処理の第2のフェーズに対するバックアップノードの重みが第1の値であ
ると判断することと、現在のエポックにおけるコンセンサス処理の第2のフェーズにおい
て定足数検証に成功したとの判断に応答して、コンセンサス処理の第2のフェーズに対す
るバックアップノードの重みが第2の値であると判断することと、第1の値が第2の
値より小さい、判断することと、を含む。

20

【0211】

任意選択で、ネットワークノードに対する第2のフェーズにおける定足数検証は、他の
ネットワークノードから所定の数のECHOメッセージを受け取ることを含む。

【0212】

30

任意選択で、現在のエポックにおけるコンセンサス処理の3つのフェーズのそれぞれと
関連付けられたバックアップノードの個々の重みを判断することは、現在のエポックにお
けるコンセンサス処理の第3のフェーズにおいて定足数検証に失敗したとの判断に応答し
て、コンセンサス処理の第3のフェーズに対するバックアップノードの重みが第3の値であ
ると判断することと、現在のエポックにおけるコンセンサス処理の第3のフェーズにおい
て定足数検証に成功したとの判断に応答して、コンセンサス処理の第3のフェーズに対す
るバックアップノードの重みが第4の値であると判断することと、第3の値が第4の
値より小さい、判断することと、を含む。

【0213】

任意選択で、ネットワークノードに対する第3のフェーズにおける定足数検証は、他の
ネットワークノードから所定の数の受け入れメッセージを受け取ることであり、他のネ
ットワークノードからの受け入れメッセージのそれぞれが、他のネットワークノードのそ
れぞれが所定の数のECHOメッセージを受け入れたことを示す、受け取ることを含む。

40

【0214】

任意選択で、EPOCH_CHANGEメッセージは、いくつかのネットワークノードからのネッ
トワークノードのセットと関連付けられた署名のセットをさらに含み、NEW_EPOCHメッセ
ージは、EPOCH_CHANGEメッセージのダイジェストを含む。

【0215】

任意選択で、少なくとも1つの有効なNEW_EPOCHメッセージが有効であるかどうかを検証
することは、少なくとも1つのNEW_EPOCHメッセージ内のEPOCH_CHANGEメッセージのダイジ

50

エストが有効であるかどうかを検証することを含み、少なくとも1つのNEW_EPOCHメッセージ内のEPOCH_CHANGEメッセージのダイジェストが有効であるかどうかを検証することが、EPOCH_CHANGEメッセージ内の署名のセットが有効であるかどうかを検証することを含む。

【0216】

任意選択で、エポック変更が行われる必要があると判断することは、所定の期間内に古いエポックにおいてコンセンサスが実現されなかったとの判断に応答して、エポック変更が行われる必要があると判断することを含む。

【0217】

任意選択で、少なくとも1つのプロセッサは、新しい1次ノードによる新しいエポックにおいて動作するようにさらに構成され、新しいエポックは、新しい1次ノードを使用して複数のネットワークノード間のコンセンサスを実現するためのコンセンサス処理を含む。

【0218】

図16を参照すると、図16は、本明細書の実装形態による1次ノード変更装置1600のモジュールを示す図である。1次ノードを変更するための装置1600は、ブロックチェーン技術に基づくコンセンサスシステムに使用されることが可能である。装置1600は、図7～図9において示された実装形態に対応する。例えば、装置1600は、ブロックチェーンネットワークのネットワークノードにおいて実行されることが可能である。装置1600は、このネットワークノード以外のバックアップノードからEPOCH_CHANGEメッセージを受け取るように構成された受信器または受信ユニット1602であって、EPOCH_CHANGEメッセージが、エポック変更が行われる必要があるという指示を含み、エポック変更が、現在の1次ノードによる現在のエポックから、新しい1次ノードによる新しいエポックへの変更を引き起こす、受信器または受信ユニット1602と、EPOCH_CHANGEメッセージが有効であるかどうかを検証するように構成された検証ユニット1604と、EPOCH_CHANGEメッセージが有効であることの検証に
20
応答して、他のネットワークノードにNEW_EPOCHメッセージを送るように構成された送信器または送信ユニット1606であって、NEW_EPOCHメッセージが、EPOCH_CHANGEメッセージのダイジェストを含む、送信器または送信ユニット1606と、このネットワークノード以外のいくつかのネットワークノードのうちの少なくとも1つから少なくとも1つのNEW_EPOCHメッセージを受け取るようにさらに構成された受信器または受信ユニット1602と、少なくとも1つのNEW_EPOCHメッセージが有効であるかどうかを検証するようにさらに構成された検証ユニット1604と、少なくとも1つのNEW_EPOCHメッセージからの有効なNEW_EPOCH
30
メッセージの数が所定の閾値を超過するかどうかを判断するように構成された判断ユニット1608と、有効なNEW_EPOCHメッセージの数が所定の閾値を超過するとの判断に
応答して、バックアップノードが新しいエポックにおける新しい1次ノードであると判断するよう
にさらに構成された判断ユニット1608と、を含む。

【0219】

任意選択の実装形態において、EPOCH_CHANGEメッセージは、バックアップノードと関連付けられた加重和、およびいくつかのネットワークノードからのネットワークノードのセットと関連付けられた署名のセットを含む。

【0220】

任意選択の実装形態において、EPOCH_CHANGEメッセージが有効であるかどうかを検証することは、EPOCH_CHANGEメッセージ内の加重和が有効であるかどうかを検証することを含み、EPOCH_CHANGEメッセージ内の加重和が有効であるかどうかを検証することは、署名の
40
セットが有効であるかどうかを検証することを含む。

【0221】

任意選択の実装形態において、少なくとも1つのNEW_EPOCHメッセージが有効であるかどうかを検証することは、少なくとも1つのNEW_EPOCHメッセージ内のEPOCH_CHANGEメッセージのダイジェストが有効であるかどうかを検証することを含み、少なくとも1つのNEW_EPOCHメッセージ内のEPOCH_CHANGEメッセージのダイジェストが有効であるかどうかを検証することは、EPOCH_CHANGEメッセージ内の署名のセットが有効であるかどうかを検証することを含む。

10

20

30

40

50

【0222】

図16は、1次ノード変更装置1600の内部機能モジュールおよび構造を示す概略図である。実行体は本質的に電子デバイスであることが可能であり、電子デバイスは、少なくとも1つのプロセッサと、少なくとも1つのプロセッサの実行可能命令を格納するように構成されたメモリと、を含む。

【0223】

少なくとも1つのプロセッサは、このネットワークノード以外のバックアップノードからEPOCH_CHANGEメッセージを受け取ることであって、EPOCH_CHANGEメッセージが、エポック変更が行われる必要があるという指示を含み、エポック変更が、現在の1次ノードによる現在のエポックから、新しい1次ノードによる新しいエポックへの変更を引き起こす、受け取ることと、EPOCH_CHANGEメッセージが有効であるかどうかを検証することと、EPOCH_CHANGEメッセージが有効であることの検証に応答して、他のネットワークノードにNEW_EPOCHメッセージを送ることと、NEW_EPOCHメッセージが、EPOCH_CHANGEメッセージのダイジェストを含む、送ることと、このネットワークノード以外のいくつかのネットワークノードのうちの少なくとも1つから少なくとも1つのNEW_EPOCHメッセージを受け取ることと、少なくとも1つのNEW_EPOCHメッセージが有効であるかどうかを検証することと、少なくとも1つのNEW_EPOCHメッセージからの有効なNEW_EPOCHメッセージの数が所定の閾値を超過するかどうかを判断することと、有効なNEW_EPOCHメッセージの数が所定の閾値を超過するとの判断に応答して、バックアップノードが新しいエポックにおける新しい1次ノードであると判断することと、を行うように構成される。

【0224】

任意選択で、EPOCH_CHANGEメッセージは、バックアップノードと関連付けられた加重和、およびいくつかのネットワークノードからのネットワークノードのセットと関連付けられた署名のセットを含む。

【0225】

任意選択で、EPOCH_CHANGEメッセージが有効であるかどうかを検証することは、EPOCH_CHANGEメッセージ内の加重和が有効であるかどうかを検証することを含み、EPOCH_CHANGEメッセージ内の加重和が有効であるかどうかを検証することは、署名のセットが有効であるかどうかを検証することを含む。

【0226】

任意選択で、少なくとも1つのNEW_EPOCHメッセージが有効であるかどうかを検証することが、少なくとも1つのNEW_EPOCHメッセージ内のEPOCH_CHANGEメッセージのダイジェストが有効であるかどうかを検証することを含み、少なくとも1つのNEW_EPOCHメッセージ内のEPOCH_CHANGEメッセージのダイジェストが有効であるかどうかを検証することが、EPOCH_CHANGEメッセージ内の署名のセットが有効であるかどうかを検証することを含む。

【0227】

図17を参照すると、図17は、本明細書の実装形態による回復装置1700のモジュールを示す図である。回復のための装置1700は、ブロックチェーン技術に基づくコンセンサスシステムに使用されることが可能である。装置1700は、図10～図12において示された実装形態に対応することが可能である。例えば、装置1700は、ブロックチェーンネットワークのネットワークノード内で実行されることが可能である。装置1700は、ブロックチェーンネットワークのいくつかの他のネットワークノードに状態要求メッセージをブロックチェーンネットワークのネットワークノードによってブロードキャストするように構成されたブロードキャストユニット1702であって、ネットワークノードが、対象シーケンス番号の対象取引を回復させるためのものである、ブロードキャストユニット1702と、いくつかの他のネットワークノードからいくつかの状態リプライメッセージを受け取るように構成された受信器1704または受信ユニット1704であって、いくつかの状態リプライメッセージのそれぞれがシーケンス番号を含む、受信器1704または受信ユニット1704と、状態リプライメッセージの数が所定の閾値を超過するとの判断に応答して、対象シーケンス番号に基づいて対象シーケンス番号を識別するように構成された識別ユニット1706であって、いくつかの

状態リプライメッセージのそれぞれが対象シーケンス番号を含む、識別ユニット1706と、いくつかの他のネットワークノードに要求メッセージを送るように構成された送信器1708または送信ユニット1708であって、要求メッセージが、いくつかの他のネットワークノードのそれぞれからのECHOメッセージを要求し、ECHOメッセージが、対象シーケンス番号を有する対象取引に対するいくつかの他のネットワークノード間のコンセンサスを実現するためにいくつかの他のネットワークノードのそれぞれによって送信されたメッセージであり、ECHOメッセージが、対象取引の一部、およびいくつかの他のネットワークノードのそれぞれの署名を含む、送信器1708または送信ユニット1708と、いくつかの他のネットワークノードからいくつかのECHOメッセージを受け取るようにさらに構成された受信器1704または受信ユニット1704と、いくつかのECHOメッセージからいく

10

つかの有効なECHOメッセージを判断するように構成された判断ユニット1710であって、いくつかの有効なECHOメッセージのそれぞれが対象シーケンス番号を含む、判断ユニット1710と、有効なECHOメッセージの数が所定の閾値を超過するとの判断に応答して、いくつかの有効なECHOメッセージに基づいてネットワークノードにおいて対象シーケンス番号を有する対象取引を回復させるように構成された回復ユニット1712と、ネットワークノードが回復されたことを示すメッセージをいくつかの他のネットワークノードに送るようにさらに構成された送信器1708と、を含む。

【0228】

任意選択の実装形態において、いくつかのネットワークノードは、1次ノードおよび1つまたは複数のバックアップノードを含む。

20

【0229】

任意選択の実装形態において、ネットワークノードは、1次ノードまたはバックアップノードである。

【0230】

任意選択の実装形態において、要求メッセージは、対象シーケンス番号を含む。

【0231】

任意選択の実装形態において、回復装置1700は、ネットワークノードにECHOメッセージを送る前に、要求メッセージをこのネットワークノード以外のいくつかの他のネットワークノードのそれぞれによって検証するように構成された検証ユニット1714をさらに含む。

【0232】

30

任意選択の実装形態において、検証ユニット1714は、ECHOメッセージのそれぞれが有効であるかどうかを検証するようにさらに構成され、ECHOメッセージのそれぞれが有効であるかどうかを検証することは、マークル木を使用してECHOメッセージのそれぞれが有効であるかどうかを検証することを含む。

【0233】

任意選択の実装形態において、ECHOメッセージのそれぞれが有効であるかどうかを検証することは、ECHOメッセージ内の署名が有効であるかどうかを検証することをさらに含む。

【0234】

任意選択の実装形態において、ECHOメッセージのそれぞれは、対象取引と関連付けられたいくつかの消去コード(EC)ブロックのうちの少なくとも1つをさらに含み、いくつかのECブロックは、対象取引を使用してECコードに従って生成される。

40

【0235】

任意選択の実装形態において、いくつかの有効なECHOメッセージに基づいてネットワークノードにおいて対象シーケンス番号を有する対象取引を回復させることは、いくつかの有効なECHOメッセージ内にある複数のECブロックのサブセットを使用して対象取引を再構築することを含む。

【0236】

任意選択の実装形態において、ネットワークノードが回復されたことを示すいくつかの他のネットワークノードへのメッセージは、いくつかの有効なECHOメッセージ内の署名の

50

セット、および対象シーケンス番号を含む。

【0237】

以前の実装形態において示されたシステム、装置、モジュール、またはユニットは、コンピュータチップまたはエンティティを使用することによって実行されることが可能であり、また一定の機能を有する製品を使用することによって実行されることが可能である。典型的な実行デバイスはコンピュータであり、コンピュータは、パーソナルコンピュータ、ラップトップコンピュータ、セルラー電話、カメラ電話、スマートフォン、パーソナルデジタルアシスタント、メディアプレーヤ、ナビゲーションデバイス、eメール送受信デバイス、ゲーム機、タブレット型コンピュータ、ウェアラブルデバイス、またはこれらのデバイスの任意の組合せであることが可能である。

10

【0238】

装置内の各ユニットの機能および役割の実行処理については、以前の方法における対応するステップの実行処理に対する参照が行われることが可能である。簡略化のために詳細はここでは省略される。

【0239】

装置の実装形態は基本的に方法の実装形態に対応するので、関連部分については、方法の実装形態における関連説明に対する参照が行われることが可能である。前述の装置の実装形態は、1つの例にすぎない。別々の部分として説明されたユニットは、物理的に別のものであっても、そうでなくてもよく、ユニットとして表示された部分は、物理的なユニットであっても、そうでなくてもよく、1つの場所に配置されてもよく、また複数のネットワークユニット上に分散されてもよい。モジュールのいくつかまたは全ては、本明細書の解決策の目的を実現するための実際の要望に基づいて選択されることが可能である。当業者は、創造的な努力をすることなく本出願の実装形態を理解し実行することができる。

20

【0240】

図17は、回復装置1700の内部機能モジュールおよび構造を示す概略図である。実行体は本質的に電子デバイスであることが可能であり、電子デバイスは、少なくとも1つのプロセッサと、少なくとも1つのプロセッサの実行可能命令を格納するように構成されたメモリと、を含む。

【0241】

少なくとも1つのプロセッサは、ブロックチェーンネットワークのいくつかの他のネットワークノードに状態要求メッセージをブロックチェーンネットワークのネットワークノードによってブロードキャストすることであって、ネットワークノードが、対象シーケンス番号の対象取引を回復させるためのものである、ブロードキャストすることと、いくつかの他のネットワークノードからいくつかの状態リプライメッセージを受け取ることであって、いくつかの状態リプライメッセージのそれぞれがシーケンス番号を含む、受け取ることと、状態リプライメッセージの数が所定の閾値を超過するとの判断に応答して、対象シーケンス番号に基づいて対象シーケンス番号を識別することであって、いくつかの状態リプライメッセージのそれぞれが対象シーケンス番号を含む、識別することと、いくつかの他のネットワークノードに要求メッセージを送ることであって、要求メッセージが、いくつかの他のネットワークノードのそれぞれからのECHOメッセージを要求し、ECHOメッセージが、対象シーケンス番号を有する対象取引に対するいくつかの他のネットワークノード間のコンセンサスを実現するためにいくつかの他のネットワークノードのそれぞれによって送信されたメッセージであり、ECHOメッセージが、対象取引の一部、およびいくつかの他のネットワークノードのそれぞれの署名を含む、送ることと、複数の他のネットワークノードからいくつかのECHOメッセージを受け取ることと、いくつかのECHOメッセージからいくつかの有効なECHOメッセージを判断することであって、いくつかの有効なECHOメッセージのそれぞれが対象シーケンス番号を含む、判断することと、有効なECHOメッセージの数が所定の閾値を超過するとの判断に応答して、いくつかの有効なECHOメッセージに基づいてネットワークノードにおいて対象シーケンス番号を有する対象取引を回復させることと、ネットワークノードが回復されたことを示すメッセージをいくつかの他のネットワ

30

40

50

ークノードに送ることと、を行うように構成される。

【0242】

任意選択で、いくつかのネットワークノードは、1次ノードおよび1つまたは複数のバックアップノードを含む。

【0243】

任意選択で、ネットワークノードは、1次ノードまたはバックアップノードである。

【0244】

任意選択で、要求メッセージは、対象シーケンス番号を含む。

【0245】

任意選択で、少なくとも1つのプロセッサは、ネットワークノードにECHOメッセージを送る前に、要求メッセージをこのネットワークノード以外のいくつかの他のネットワークノードのそれぞれによって検証するようにさらに構成される。

10

【0246】

任意選択で、少なくとも1つのプロセッサは、ECHOメッセージのそれぞれが有効であるかどうかを検証することであって、ECHOメッセージのそれぞれが有効であるかどうかを検証することが、マークル木を使用してECHOメッセージのそれぞれが有効であるかどうかを検証することを含む、検証すること、を行うようにさらに構成される。

【0247】

任意選択で、ECHOメッセージのそれぞれが有効であるかどうかを検証することは、ECHOメッセージ内の署名が有効であるかどうかを検証することをさらに含む。

20

【0248】

任意選択で、ECHOメッセージのそれぞれは、対象取引と関連付けられたいくつかの消去コード(EC)ブロックのうちの少なくとも1つをさらに含み、いくつかのECブロックは、対象取引を使用してECコードに従って生成される。

【0249】

任意選択で、いくつかの有効なECHOメッセージに基づいてネットワークノードにおいて対象シーケンス番号を有する対象取引を回復させることは、いくつかの有効なECHOメッセージ内にあるいくつかのECブロックのサブセットを使用して対象取引を再構築することを含む。

30

【0250】

任意選択で、ネットワークノードが回復されたことを示すいくつかの他のネットワークノードへのメッセージは、いくつかの有効なECHOメッセージ内の署名のセット、および対象シーケンス番号を含む。

【0251】

本明細書において説明された主題の実装形態ならびにアクションおよび動作は、デジタル電子回路機器において、現実に具現化されたコンピュータソフトウェアもしくはファームウェアにおいて、本明細書およびこれらの構造上の均等物の中で開示された構造を含むコンピュータハードウェアにおいて、またはこれらの1つもしくは複数の組合せにおいて実行されることが可能である。本明細書において説明された主題の実装形態は、データ処理装置による実行のため、またはデータ処理装置の動作を制御するために、コンピュータプログラム担体上でエンコードされた、例えばコンピュータプログラム命令の1つまたは複数のモジュールといった、1つまたは複数のコンピュータプログラムとして実行されることが可能である。担体は、有形の非一時的コンピュータストレージ媒体であることが可能である。一方、またはさらに、担体は、データ処理装置による実行のために適切な受信装置に送信するための情報をエンコードするために生成される、例えば、機械生成された電氣的、光学的、または電磁氣的な信号といった、人工的に生成された伝搬信号であることが可能である。コンピュータストレージ媒体は、機械可読ストレージデバイス、機械可読ストレージ基板、ランダムもしくはシリアルアクセスメモリデバイス、またはこれらの

40

50

1つもしくは複数の組合せであることが可能であり、またこれらの一部であることが可能である。コンピュータストレージ媒体は、伝搬信号ではない。

【0252】

用語「データ処理装置」は、プログラマブルプロセッサ、コンピュータ、または複数のプロセッサもしくはコンピュータを例として含む、データを処理するための装置、デバイス、および機械の全ての種類を包含する。データ処理装置は、例えばFPGA(フィールドプログラマブルゲートアレイ)、ASIC(特定用途向け集積回路)、またはGPU(グラフィックス処理ユニット)といった特殊用途のロジック回路機器を含むことができる。装置は、ハードウェアの他に、例えば、プロセッサファームウェア、プロトコルスタック、データベース管理システム、オペレーティングシステム、またはこれらの1つまたは複数の組合せを構成するコードといった、コンピュータプログラムのための実行環境を作り出すコードを含むこともできる。

10

【0253】

プログラム、ソフトウェア、ソフトウェアアプリケーション、アプリ、モジュール、ソフトウェアモジュール、エンジン、スクリプト、もしくはコードと呼ばれること、またはこれらとして説明されることが可能なコンピュータプログラムは、コンパイルもしくは翻訳された言語、または宣言型言語もしくは手続き型言語を含む、プログラミング言語の任意の形式で書き込まれることが可能であり、コンピュータプログラムは、スタンドアロンプログラムとして、あるいはモジュール、コンポーネント、エンジン、サブルーチン、または1つもしくは複数の場所においてデータ通信ネットワークによって相互接続された1つもしくは複数のコンピュータを含むことができるコンピューティング環境における実行に適した他のユニットとして含む、任意の形式で導入されることが可能である。

20

【0254】

コンピュータプログラムは、ファイルシステムにおけるファイルに対応することができるが、そうである必要はない。コンピュータプログラムは、例えば、マークアップ言語文書に、当該のプログラムに専用の単一のファイルに、または例えば、1つもしくは複数のモジュール、サブプログラム、もしくはコードの一部を格納するファイルといった、複数のコーディネイテッドファイルに格納された1つまたは複数のスクリプトといった、他のプログラムまたはデータを保持するファイルの一部に格納されることが可能である。

【0255】

本明細書において説明された処理およびロジックフローは、1つまたは複数のコンピュータプログラムを実行して、入力データに対して演算し、出力を生成することによって動作を行う、1つまたは複数のコンピュータによって行われることが可能である。処理およびロジックフローは、例えば、FPGA、ASIC、もしくはGPUといった特殊用途のロジック回路機器によって、または特殊用途のロジック回路機器および1つもしくは複数のプログラムされたコンピュータの組合せによって行われることも可能である。

30

【0256】

コンピュータプログラムの実行に適したコンピュータは、汎用もしくは特殊用途のマイクロプロセッサまたは両方、あるいは他の任意の種類の中央処理装置に基づくことが可能である。一般に、中央処理装置は、リードオンリメモリもしくはランダムアクセスメモリまたは両方から命令およびデータを受け取ることになる。コンピュータの要素は、命令を実行するための中央処理装置、ならびに命令およびデータを格納するための1つまたは複数のメモリデバイスを含むことができる。中央処理装置およびメモリは、特殊用途のロジック回路機器によって補われることも、特殊用途のロジック回路機器に組み込まれることも可能である。

40

【0257】

一般に、コンピュータは、(コンピュータ可読メモリとも呼ばれる)少なくとも1つの非一時的コンピュータ可読ストレージ媒体に連結されることになる。コンピュータに連結されたストレージ媒体は、コンピュータの内部構成要素(例えば統合されたハードドライブ)、または外部構成要素(例えば、ユニバーサルシリアルバス(USB)ハードドライブもしくはは

50

ネットワークを超えてアクセスされるストレージシステム)であることが可能である。ストレージ媒体の例は、例えば、磁気ディスク、光磁気ディスク、もしくは光ディスク、ソリッドステートデバイス、クラウドストレージシステムなどのネットワークストレージソース、またはストレージ媒体の他のタイプを含むことができる。しかし、コンピュータは、このようなデバイスを有する必要はない。さらに、コンピュータは、例えば、ほんの数例を挙げれば、携帯電話、パーソナルデジタルアシスタント(PDA)、モバイルオーディオプレーヤもしくはモバイルビデオプレーヤ、ゲーム機、全地球測位システム(GPS)受信器、または例えばユニバーサルシリアルバス(USB)フラッシュドライブといった携帯型ストレージデバイスといった、別のデバイスに組み込まれることが可能である。

【0258】

ユーザとの対話を行うために、本明細書において説明された主題の実装形態は、ユーザに情報を表示するための例えばLCD(液晶ディスプレイ)モニタといった表示デバイス、ならびに例えばキーボード、および例えばマウス、トラックボールもしくはタッチパッドといったポインティングデバイスといった、ユーザがコンピュータに入力することができる入力デバイスを有するコンピュータ上で実行されること、またはこのコンピュータと通信するように構成されることが可能である。同様にユーザとの対話を行うために他の種類のデバイスが使用されることが可能であり、例えば、ユーザに提供されるフィードバックは、例えば、視覚フィードバック、聴覚フィードバック、または触覚フィードバックといった、知覚によるフィードバックの任意の形式であることが可能であり、ユーザからの入力、音響、会話、または触知入力を含む任意の形式で受け取られることが可能である。さらに、コンピュータは、例えば、ウェブブラウザから受け取られた要求に応答して、ユーザのデバイス上のウェブブラウザにウェブページを送ることによって、または例えばスマートフォンもしくは電子タブレットといったユーザデバイス上で動くアプリと対話することによって、ユーザによって使用されるデバイスとの間で文書をやりとりすることによってユーザと対話することができる。同様に、コンピュータは、例えば、メッセージングアプリケーションを動かしているスマートフォンといった個人のデバイスにテキストメッセージまたは他の形式のメッセージを送ること、および返信としてユーザから応答メッセージを受け取ることによって、ユーザと対話することができる。

【0259】

本明細書は、システム、装置、およびコンピュータプログラム構成要素に関連して用語「ように構成される(configured to)」を使用する。1つまたは複数のコンピュータのシステムが特定の動作またはアクションを行うように構成されることは、動作中に動作またはアクションをシステムに行わせる、ソフトウェア、ファームウェア、ハードウェア、またはこれらの組合せをシステムがシステム上にインストールしたということを意味する。1つまたは複数のコンピュータプログラムが特定の動作またはアクションを行うように構成されることは、データ処理装置によって実行されると、動作またはアクションを装置に行わせる命令を1つまたは複数のプログラムが含むということを意味する。特殊用途のロジック回路機器が特定の動作またはアクションを行うように構成されることは、動作またはアクションを行う電子ロジックを回路機器が有するということを意味する。

【0260】

本明細書は、多くの特定の实装形態の詳細を収めるが、これらは、特許請求の範囲自体によって定義される特許請求されるものの範囲を限定するものと解釈されるべきではなく、むしろ特定の实装形態に固有であることが可能な特徴の説明として解釈されるべきである。別々の実装形態の背景において本明細書において説明される一定の特徴は、単一の実装形態における組合せの中で実現されることも可能である。逆に、単一の実装形態の背景において説明される様々な特徴は、複数の実装形態において別々に、または任意の適切なサブコンビネーションにおいて実現されることも可能である。さらに、特徴は、一定の組合せにおいて作用するものとして上述されることがあり、したがって最初であったとしても特許請求されることがあるが、特許請求される組合せからの1つまたは複数の特徴は、いくつかのケースにおいて、組合せから削除されることが可能であり、特許請求の範囲は

、サブコンビネーションまたはサブコンビネーションの変形形態を対象とすることが可能である。

【0261】

同様に、特定の順序で、図面において動作が描写され、特許請求の範囲において列挙されるが、これは、示された特定の順序で、もしくは連続した順序でこのような動作が行われること、または全ての図示された動作が望ましい結果を実現するために行われることが必要であると見なされるべきではない。一定の状況において、マルチタスク処理および並列処理が有利なこともある。さらに、上述の実装形態における様々なシステムモジュールおよび構成要素の分離は、全ての実装形態においてこのような分離が必要であると見なされるべきではなく、説明されたプログラム構成要素およびシステムは、全体的に、単一のソフトウェア製品に相互に統合されることも、複数のソフトウェア製品にパッケージ化されることも可能であるということを理解されたい。

10

【0262】

主題の特定の实装形態が説明された。他の実装形態は、以下の特許請求の範囲の範囲内にある。例えば、特許請求の範囲において列挙されるアクションは、様々な順序で行われることが可能であり、依然として望ましい結果を実現する。1つの例として、添付の図において描写された処理は、望ましい結果を実現するために、示された特定の順序、または連続した順序を必ずしも必要としない。いくつかのケースにおいて、マルチタスク処理および並列処理が有利なこともある。

20

【符号の説明】

【0263】

- 100 環境、システム
- 102 コンソーシアムブロックチェーンネットワーク、ブロックチェーンネットワーク
- 106 コンピューティングデバイスまたはシステム
- 108 コンピューティングデバイスまたはシステム
- 110 ネットワーク
- 200 概念アーキテクチャ、システム
- 202 参加者システム
- 204 参加者システム
- 206 参加者システム
- 212 ブロックチェーンネットワーク
- 214 ノード
- 216 ブロックチェーン
- 216' 完全コピー
- 216'' 完全コピー
- 216''' 完全コピー
- 300 処理、コンセンサス処理、システム
- 310 ステージ、第1のフェーズ、フェーズ1: INITIALメッセージ
- 320 ステージ、第2のフェーズ、フェーズ2: ECHOメッセージ
- 330 ステージ、第3のフェーズ、フェーズ3: ACCEPTメッセージ
- 400 処理、コンセンサス処理
- 402 クライアントノード
- 404 1次ノード、ノード
- 406 バックアップノード
- 408 取引要求を生成する
- 410 取引要求
- 412 ECブロックを生成する
- 414 INITIALメッセージを生成する
- 416 INITIALメッセージ

30

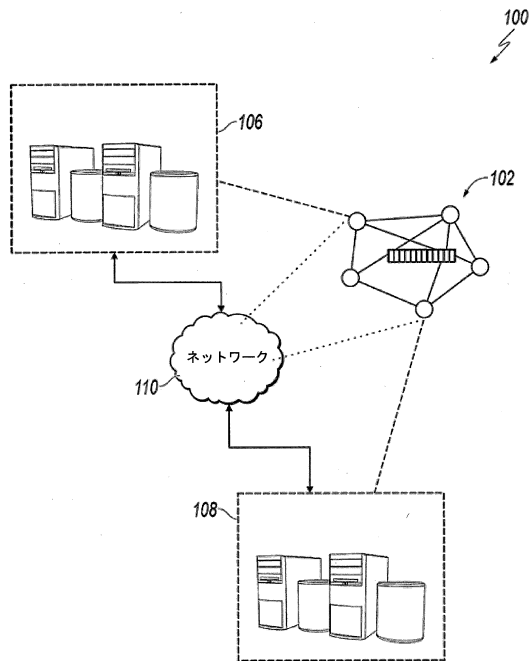
40

50

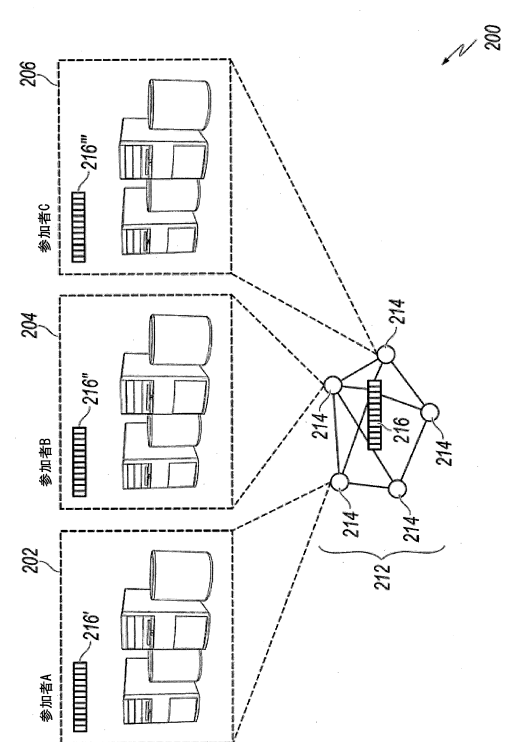
418	ECHOメッセージを生成する	
420	ECHOメッセージ	
421	他のバックアップノードにECHOメッセージを送る	
422	ECHOメッセージを検証する	
423	他のバックアップノードからECHOメッセージを受け取る	
424	有効なECHOメッセージを判断する	
426	取引要求を再構築する	
428	ACCEPTメッセージを生成する	
430	ACCEPTメッセージ	
432	ACCEPTメッセージを生成する	10
434	ACCEPTメッセージ	
436	取引要求を実行する	
438	取引結果	
500	ハッシュ木	
502	取引要求	
504	ECコード	
506	ECメッセージ	
508	ECブロック、ECブロックA	
508	ECブロック、ECブロックB	
508	ECブロック、ECブロックC	20
508	ECブロック、ECブロックD	
510	葉ノード、ハッシュ、暗号学的ハッシュ、ハッシュA	
510	葉ノード、ハッシュ、暗号学的ハッシュ、ハッシュB	
510	葉ノード、ハッシュ、暗号学的ハッシュ、ハッシュC	
510	葉ノード、ハッシュ、暗号学的ハッシュ、ハッシュD	
512	非葉ノード、ハッシュ、ハッシュAB	
512	非葉ノード、ハッシュ、ハッシュCD	
514	非葉ノード、ルートハッシュ、ハッシュABCD	
700	方法、処理	
800	処理	30
802	バックアップノード	
804	他のネットワークノード、ネットワークノード	
806	エポック変更を判断する	
808	個々の重みを判断する	
810	加重和を判断する	
812	EPOCH_CHANGEメッセージ	
814	EPOCH_CHANGEメッセージを検証する	
816	NEW_EPOCHメッセージ	
818	NEW_EPOCHメッセージを検証する	
820	有効なNEW_EPOCHメッセージを判断する	40
822	新しい1次ノードを判断する	
1000	処理	
1010	第1のフェーズ、QUERY_STATEメッセージ	
1020	第2のフェーズ、REPLY_STATEメッセージ	
1030	第3のフェーズ、FETCH_ECHOメッセージ	
1040	第4のフェーズ、ECHOメッセージ	
1050	第5のフェーズ、ローカル消去コード	
1060	第6のフェーズ、ACCEPTメッセージ	
1100	処理、方法	
1102	ネットワークノード	50

1104	ネットワークノード	
1106	状態要求メッセージ	
1108	状態リプライメッセージ	
1110	状態リプライメッセージの数を判断する	
1112	対象シーケンス番号を識別する	
1114	FETCH_ECHOメッセージ	
1116	ECHOメッセージ	
1118	ECHOメッセージを検証する	
1120	有効なECHOメッセージを判断する	
1122	対象取引を回復させる	10
1124	ACCEPTメッセージ	
1300	コンセンサス装置、装置	
1302	受信器、受信ユニット	
1304	生成ユニット	
1306	送信器、送信ユニット	
1308	検証ユニット	
1310	判断ユニット	
1312	再構築ユニット	
1314	実行ユニット	
1400	コンセンサス装置、装置	20
1402	受信器、受信ユニット	
1404	送信器、送信ユニット	
1406	検証ユニット	
1408	判断ユニット	
1410	再構築ユニット	
1412	実行ユニット	
1500	1次ノード変更装置、装置	
1502	判断ユニット	
1504	送信器、送信ユニット	
1506	受信器、受信ユニット	30
1508	検証ユニット	
1510	動作ユニット	
1600	1次ノード変更装置、装置	
1602	受信器、受信ユニット	
1604	検証ユニット	
1606	送信器、送信ユニット	
1608	判断ユニット	
1700	回復装置、装置	
1702	ブロードキャストユニット	
1704	受信器、受信ユニット	40
1706	識別ユニット	
1708	送信器、送信ユニット	
1710	判断ユニット	
1712	回復ユニット	
1714	検証ユニット	

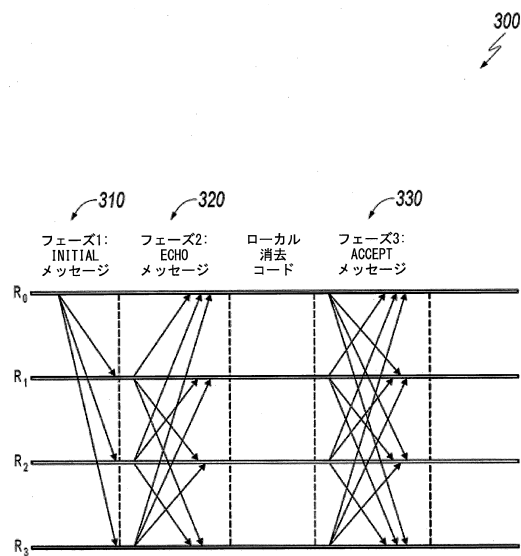
【図 1】



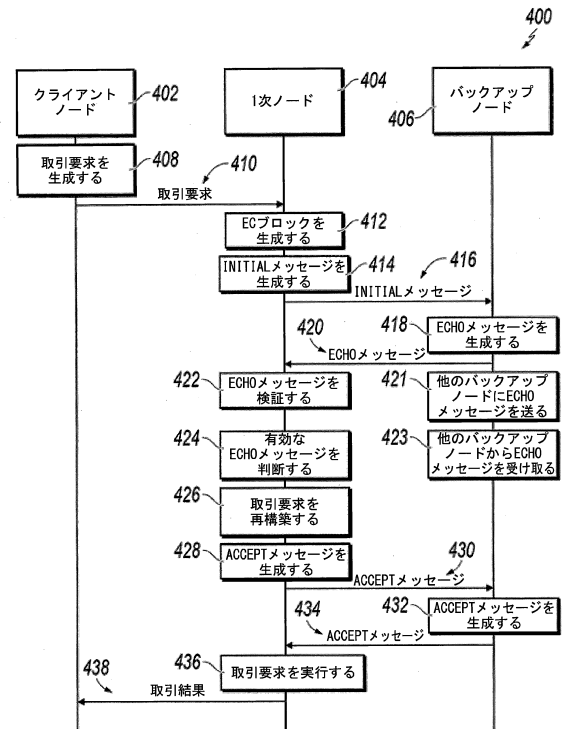
【図 2】



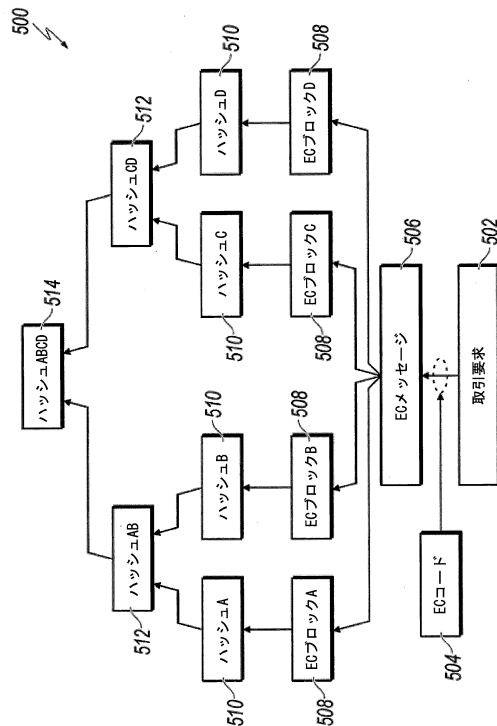
【図 3】



【図 4】



【図 5】

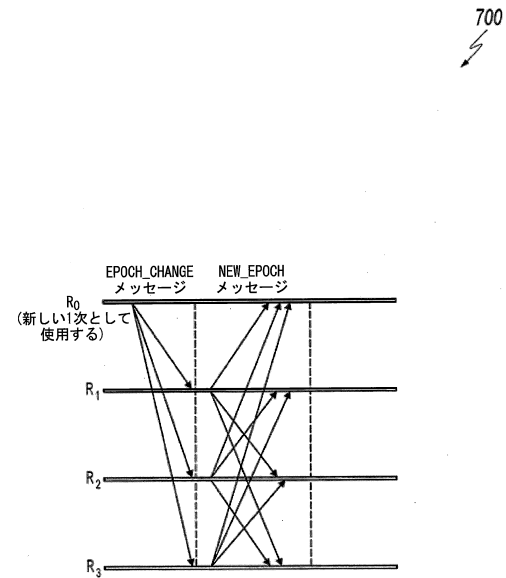


【図 6】

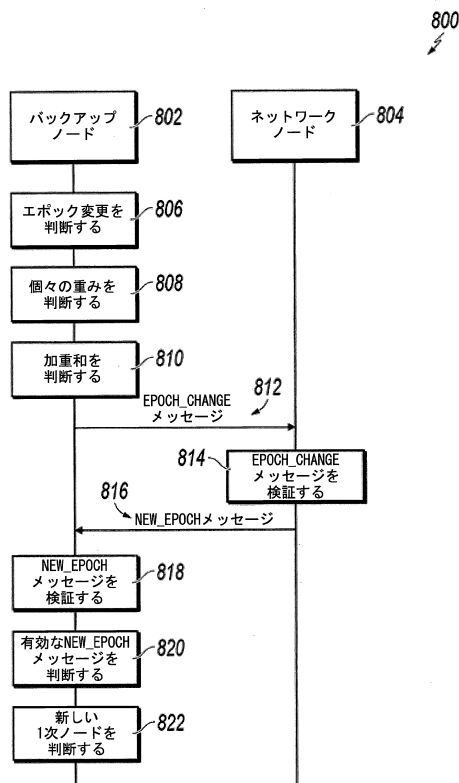
Initial<epoch, tx_root_hash, ec_block_hash, ec_block, seq, j>
 Echo<epoch, tx_root_hash, ec_block_hash, ec_block, seq, sign_proof, j>
 Accept<epoch, tx_root_hash, seq, sign_proofs, j>

FIG. 6

【図 7】



【図 8】

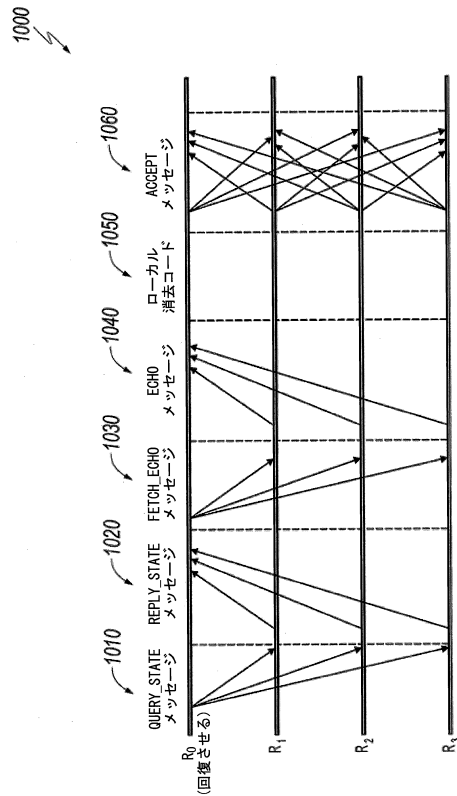


【図 9】

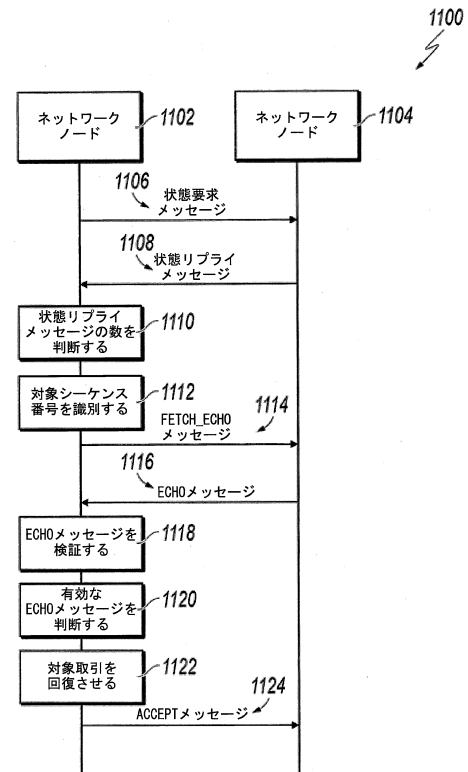
Epoch_change <weight, epoch+1, Echo {}, Accept {}, j>
 Echo {<epoch, seq, sign_proof, j>}
 Accept {<epoch, tx_root_hash, sign_proofs, seq, j>}
 New_epoch <epoch+1, i, j, seq, ec_digest>

FIG. 9

【図 10】



【図 11】

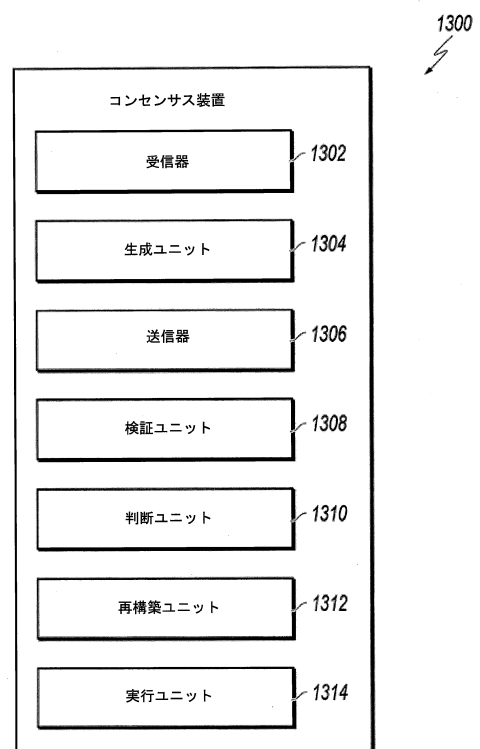


【図 12】

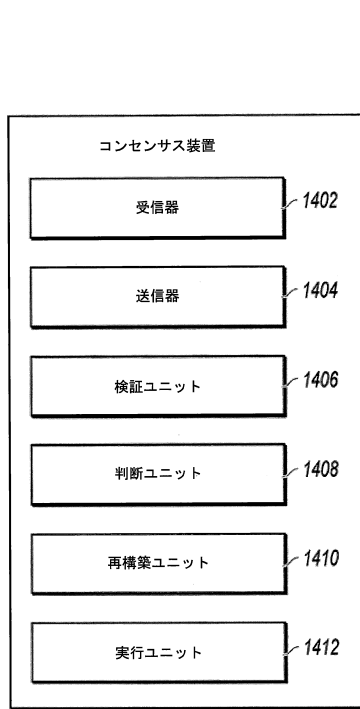
Query_state <j, seq>
 Reply_state <j, last_seq>
 Fetch_echo <j, last_seq+1>

FIG. 12

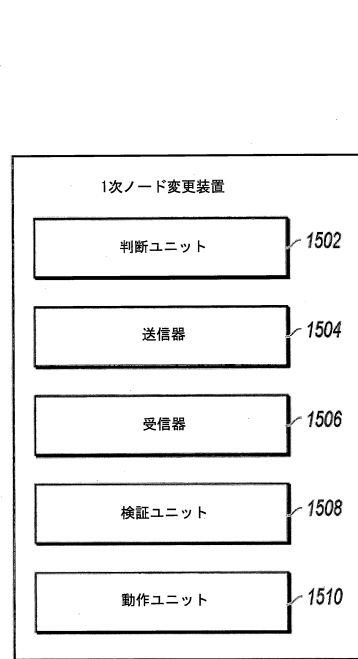
【図 13】



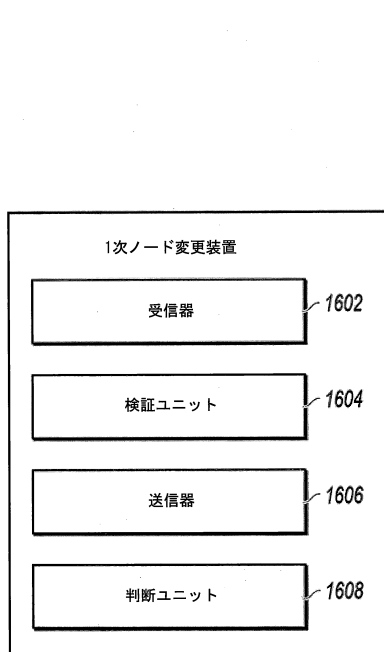
【図 1 4】



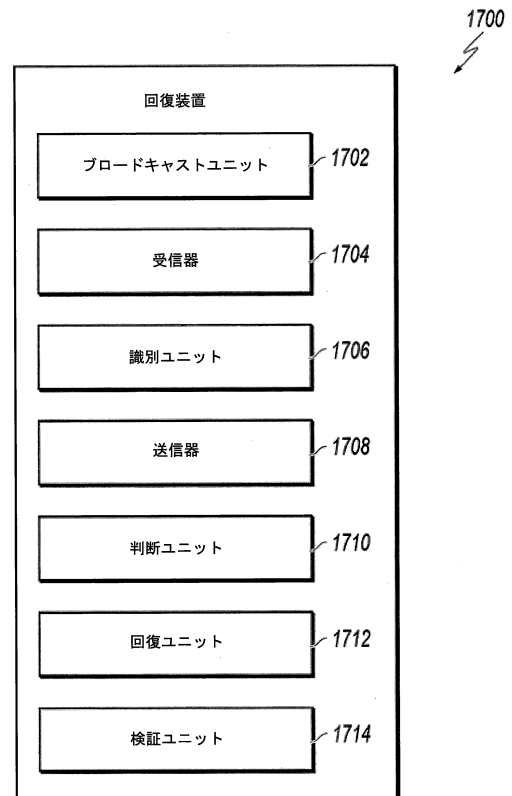
【図 1 5】



【図 1 6】



【図 1 7】



フロントページの続き

(72)発明者 ペン・リン

中華人民共和国・311121・ハンジョウ・ユ・ハン・ディストリクト・ウェスト・ウェン・イ
・ロード・ナンバー・969・ビルディング・3・5/エフ・アリババ・グループ・リーガル・デ
パートメント

審査官 桜井 茂行

(56)参考文献 米国特許出願公開第2017/0344987 (US, A1)

米国特許出願公開第2018/285412 (US, A1)

特開2015-146165 (JP, A)

特開2014-178793 (JP, A)

米国特許出願公開第2019/0124146 (US, A1)

米国特許出願公開第2019/020729 (US, A1)

(58)調査した分野(Int.Cl., DB名)

G06F 11/18

H04L 9/32

G06F 21/64