(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5557840号
(P5557840)

(45)発行日　平成26年7月23日(2014.7.23)　(24)登録日　平成26年6月13日(2014.6.13)

(51)Int.Cl.　　F I
G06F 12/00 (2006.01)　　G06F 12/00 531M
G06F 13/00 (2006.01)　　G06F 12/00 545A
　　G06F 13/00 520D
　　G06F 13/00 351N
　　G06F 13/00 351M

請求項の数 18　(全 31 頁)

(21)出願番号　特願2011-529547 (P2011-529547)
(86)(22)出願日　平成21年9月30日 (2009.9.30)
(65)公表番号　特表2012-504807 (P2012-504807A)
(43)公表日　平成24年2月23日 (2012.2.23)
(86)国際出願番号　PCT/EP2009/062714
(87)国際公開番号　WO2010/037794
(87)国際公開日　平成22年4月8日 (2010.4.8)
審査請求日　平成24年8月30日 (2012.8.30)
(31)優先権主張番号　61/102,408
(32)優先日　平成20年10月3日 (2008.10.3)
(33)優先権主張国　米国 (US)

(73)特許権者　598036300
テレフオンアクチーボラゲット　エル　エム　エリクソン (パブル)
スウェーデン国　ストックホルム　エス－１６４　８３
(74)代理人　100076428
弁理士　大塚　康徳
(74)代理人　100112508
弁理士　高柳　司郎
(74)代理人　100115071
弁理士　大塚　康弘
(74)代理人　100116894
弁理士　木村　秀二
(74)代理人　100130409
弁理士　下山　治

最終頁に続く

(54)【発明の名称】分散データベースの監視メカニズム

(57)【特許請求の範囲】
【請求項１】
データの少なくとも１つのパーティションのレプリカを格納するように各ノードが構成された複数のノードを有する分散データベースシステムを扱う方法であって、
格納されるデータをｐ個のパーティションに分割するステップ（Ｓ－００５）と、
各パーティションをｒ個のレプリカに複製するステップ（Ｓ－０１０）と、
各パーティションについて、前記ｒ個のレプリカを前記複数のノードの中から選択された対応するｒ個のノードに分配するステップ（Ｓ－０１５、Ｓ－０２０）と、
相互にアドレスを指定するために使用可能である他のノードの識別子のリストを各ノードに設定するステップ（Ｓ－０１７、Ｓ－０２０）と、
前記複数のノードの中から２つ以上のノードを起動するステップ（Ｓ－０３０、Ｓ－０３５、Ｓ－０６０、Ｓ－０７０）と、
各アクティブノードにおいて、各レプリカの最新の更新と、レプリカ状態と、各レプリカを担当するローカルリソースの状態と、各レプリカの接続状態とのうちから選択された少なくとも１つのイベントを監視するステップ（Ｓ－０５０、Ｓ－０５５、Ｓ－０９０、Ｓ－０９５）と、
前記複数のノードの中のノードの起動又は停止の際に、前記アクティブノードの中から選択された少なくとも１つのノードにおいて、前記少なくとも１つのイベントに関する情報を各アクティブノードから収集し、各レプリカについての前記収集されたイベントに依存して前記アクティブノード内の各レプリカの優先順位を決めるためのルールを適用し、

各パーティションについて最高のレプリカ優先度を有する特定のノード内のレプリカを選択するステップであって、当該選択されたレプリカが現在のマスタレプリカであり、当該特定のノードが当該パーティションについての現在のマスタノードである、ステップと、

前記分散データベースシステム内のデータの読み出し／書き込みを行うことの、ノードにおいて受信された（Ｓ－１５０；Ｓ－１６０）任意のリクエストについて、当該データが属するパーティション（１１、１２）と当該パーティションについての現在のマスタレプリカを担当する現在のマスタノード（２１０５）とを決定し、当該リクエストを当該現在のマスタノードへルーティングする（Ｓ－１５１、Ｓ－１５２、Ｓ－１５３；Ｓ－１６１）ステップと

を有することを特徴とする方法。

【請求項２】

各パーティションについて、前記ｒ個のレプリカを対応するｒ個のノードに分配する前記ステップは、他の基準が同一のレプリカ優先度を生み出す場合に適用されるデフォルトレプリカ優先度を各レプリカに設定するステップを含むことを特徴とする請求項１に記載の方法。

【請求項３】

前記アクティブノードの中から選択された少なくとも１つのノードによって、パーティションについての所与のデフォルトレプリカ優先度が設定されていることに関する情報を少なくとも１つのアクティブノードから収集するステップと、

前記アクティブノードの中から選択された少なくとも１つのノードによって、各パーティションについて最高のデフォルトレプリカ優先度を有する特定のノード内のレプリカを選択するステップであって、当該選択されたレプリカが現在のマスタレプリカであり、当該特定のノードが当該パーティションについての現在のマスタノードである、ステップとを含むことを特徴とする請求項２に記載の方法。

【請求項４】

前記複数のノードの中のノードの起動又は停止の際に、前記アクティブノードが起動された順序を決定するステップをさらに有することを特徴とする請求項１に記載の方法。

【請求項５】

最初に起動されたアクティブノードは、

前記少なくとも１つのイベントに関する前記情報を各アクティブノードから収集するステップと、

各レプリカについての前記収集されたイベントに依存して、前記アクティブノード内の各レプリカの優先順位を決めるための前記ルールを適用するステップと、

各パーティションについて前記最高のレプリカ優先度を有する前記特定のノード内の前記レプリカを選択するステップであって、当該選択されたレプリカが現在のマスタレプリカであり、当該特定のノードが当該パーティションについての現在のマスタノードである、ステップと、

各パーティションについて選択された現在のマスタレプリカと当該マスタレプリカを保持する現在のマスタノードとに関して他のアクティブノードへ通知するステップとの実行を担当するシステムマスタモニタであるとみなされることを特徴とする請求項４に記載の方法。

【請求項６】

最初に起動されたアクティブノードは、

パーティションについての所与のデフォルトレプリカ優先度が設定されていることに関する情報を少なくとも１つのアクティブノードから収集するステップと、

各パーティションについて最高のデフォルトレプリカ優先度を有する特定のノード内のレプリカを選択するステップであって、当該選択されたレプリカが現在のマスタレプリカであり、当該特定のノードが当該パーティションについての現在のマスタノードである、ステップと、

各パーティションについて選択された現在のマスタレプリカと当該現在のマスタレプ

リカを保持する現在のマスタノードとに関して他のアクティブノードへ通知するステップと
の実行を担当するシステムマスタモニタであるとみなされることを特徴とする請求項４に記載の方法。
【請求項７】
各パーティションについて各アクティブノードにおいて前記現在のマスタレプリカの内容を、当該パーティションについての前記現在のマスタレプリカ（１１２、１２１、１３２、１４２）を担当する前記現在のマスタノード（１０５）からコピーするステップをさらに有することを特徴とする請求項１に記載の方法。
【請求項８】
各アクティブノードにおいてコピーされた各レプリカについて、行われた前記最新の更新と、レプリカ状態と、前記レプリカを担当するローカルリソースの状態と、前記レプリカの接続状態とのうちの少なくとも１つを作成するステップをさらに有することを特徴とする請求項７に記載の方法。
【請求項９】
複数のノード（１、２、３、４）を有する分散データベースシステムであって、
格納されるデータは複数のパーティションに分割され、
各パーティションは、前記複数のノードから選択された対応する数のノードに分散される複数のレプリカに複製され、
前記複数のノードのうちの少なくとも１つのノードはアクティブであり、
前記少なくとも１つのノードは、
少なくとも１つのパーティション（１１２、１２３、１４３）のレプリカ（２１０１）を格納するとともに、相互にアドレスを指定するために使用可能である他のノードの識別子（１５２）を格納するためのデータ記憶装置（１５）と、
前記分散データベースシステムの他のノード（１、３、４）と通信するとともに、前記分散データベースシステムにおける読み出し／書き込み動作を要求するクライアント（５）と通信するための入出力部（３０）と、
各レプリカの最新の更新（２１０３）と、レプリカ状態（２１２、２２３、２４３）と、各レプリカを担当するローカルリソースの状態と、各レプリカ（１１２、１２３、１４３）の接続状態（３１２、３２３、３４３）とのうちから選択された少なくとも１つのイベントを監視するための監視部（６０）と、
前記データ記憶装置、前記監視部及び前記入出力部と連携して、前記少なくとも１つのイベントに関する情報を前記分散データベースシステムの各アクティブノードから収集し、各レプリカについての前記収集されたイベントに依存して前記アクティブノード内の各レプリカの優先順位を決めるためのルールを適用し、各パーティションについて最高のレプリカ優先度を有する特定のノード内のレプリカを選択する動作であって、当該選択されたレプリカが現在のマスタレプリカであり、当該特定のノードが当該パーティションについての現在のマスタノードある、動作を実行し、前記分散データベースシステム内のデータの読み出し／書き込みを行うことの、受信された任意のリクエストについて、当該データが属するパーティションと当該パーティションについての現在のマスタレプリカを担当する現在のマスタノードとを決定し、当該リクエストを当該現在のマスタノードへルーティングするための制御部（２０）と
を含むことを特徴とする分散データベースシステム。
【請求項１０】
各ノードの前記データ記憶装置（１５）は、他の基準が同一のレプリカ優先度を生み出す場合に適用されるデフォルトレプリカ優先度を示すようにレプリカごとに設定されたインジケータ（２１０２）を格納するように構成されることを特徴とする請求項９に記載の分散データベースシステム。
【請求項１１】
各ノードの前記処理部（２０）、前記監視部（６０）、前記データ記憶装置（１５）及

び前記入力部（３０）は、
パーティションについての所与のデフォルトレプリカ優先度が設定されていることに関する情報を少なくとも１つのアクティブノードから収集し、
各パーティションについて最高のデフォルトレプリカ優先度を有する特定のノード内のレプリカを選択し、当該選択されたレプリカが現在のマスタレプリカであり、当該特定のノードが当該パーティションについての現在のマスタノードである
ようにさらに構成されることを特徴とする請求項１０に記載の分散データベースシステム。
【請求項１２】
各ノードの前記処理部（２０）、前記監視部（６０）、前記データ記憶装置（１５）及び前記入力部（３０）は、前記分散データベースシステムの各アクティブノードから、前記アクティブノードが起動された順序を決定するための情報（１１０４、２１０４、３１０４、４１０４）を収集するように構成されることを特徴とする請求項９に記載の分散データベースシステム。
【請求項１３】
最初に起動されたアクティブノードはシステムマスタモニタであるとみなされ、前記システムマスタモニタの前記処理部（２０）、前記監視部（６０）、前記データ記憶装置（１５）及び前記入力部（３０）は、
前記少なくとも１つのイベントに関する前記情報を各アクティブノードから収集し、
各レプリカについての前記収集されたイベントに依存して、前記アクティブノード内の各レプリカの優先順位を決めるための前記ルールを適用し、
各パーティションについて前記最高のレプリカ優先度を有する前記特定のノード内の前記レプリカを選択し、当該選択されたレプリカが現在のマスタレプリカであり、当該特定のノードが当該パーティションについての現在のマスタノード（２１０５、４１０５）であり、
各パーティションについて選択された現在のマスタレプリカと当該マスタレプリカを保持する現在のマスタノードとに関して他のアクティブノードへ通知する
ように構成されることを特徴とする請求項１２に記載の分散データベースシステム。
【請求項１４】
最初に起動されたアクティブノードはシステムマスタモニタであるとみなされ、前記システムマスタモニタの前記処理部（２０）、前記監視部（６０）、前記データ記憶装置（１５）及び前記入力部（３０）は、
パーティションについての所与のデフォルトレプリカ優先度が設定されていることに関する情報を少なくとも１つのアクティブノードから収集し、
各パーティションについて最高のデフォルトレプリカ優先度を有する特定のノード内のレプリカを選択し、当該選択されたレプリカが現在のマスタレプリカであり、当該特定のノードが当該パーティションについての現在のマスタノード（２１０５、４１０５）であり、
各パーティションについて選択された現在のマスタレプリカと当該マスタレプリカを保持する現在のマスタノードとに関して他のアクティブノードへ通知する
ようにさらに構成されることを特徴とする請求項１２に記載の分散データベースシステム。
【請求項１５】
各ノードの前記処理部（２０）、前記監視部（６０）、前記データ記憶装置（１５）及び前記入力部（３０）は、各パーティションについて各アクティブノードにおいて前記現在のマスタレプリカの内容を、当該パーティションについての前記現在のマスタレプリカ（１１２、１２１、１３２、１４２）を担当する前記現在のマスタノード（２１０５、４１０５）からコピーするようにさらに構成されることを特徴とする請求項９に記載の分散データベースシステム。
【請求項１６】

各ノードの前記処理部（２０）、前記監視部（６０）、前記データ記憶装置（１５）及び前記入力部（３０）は、各アクティブノードにおいてコピーされた各レプリカについて、行われた前記最新の更新と、レプリカ状態と、各レプリカを担当するローカルリソースの状態と、前記レプリカの接続状態とを作成するようにさらに構成されることを特徴とする請求項１５に記載の分散データベースシステム。

【請求項１７】

入出力部と処理部とを有するコンピュータの内部メモリにロード可能なコンピュータプログラムであって、前記コンピュータで動作する場合に請求項１乃至８の何れか１項に記載の方法を実行するように構成された実行可能なコードを含むことを特徴とするコンピュータプログラム。

【請求項１８】

コンピュータにおいて読み出し可能であり、請求項１７に記載のコンピュータプログラムを含む記録媒体。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は分散データベースに関し、特に通信ネットワークのための共通データベースとして利用可能な地理的に分散したデータベースに関する。さらに具体的に、本発明は改良された分散データベースシステムだけでなく、このような分散データベースシステムを扱う方法を含む。

【背景技術】

【０００２】

本発明は通信ネットワークの分野におけるアプリケーションの多様性について共通集中データベースが有する問題を解決する。通信システムの様々に異なる世代をサポートし、有線システムまたは無線システムである通信ネットワークのほとんどは、従来、加入情報および加入者データだけでなく、通信ネットワーク内または第三者のサービスネットワーク内に存在するが当該通信ネットワークの加入者がアクセス可能である様々なアプリケーションについてのサービスデータを格納するために、１つ以上の共通集中データベースを利用する。

【０００３】

通信ネットワークが成長するにつれて、通信システムの新たな世代が現われ、既存の共通集中データベースは通信ネットワーク内のすべての通信システムのニーズに常に適合できるとは限らず、またはこのニーズに常に適切であるとは限らない。それにも係わらず、通信ネットワークは、ここで利用される任意の特定のデータベースシステムで満たされるべき極めて類似する要件を共有する。

【０００４】

従来の中央管理通信データベースは一般に、少なくとも以下の特徴をサポートする必要がある。回復性および高可用性、一貫性、高性能および低レンテイシ、大容量、拡張性、地理的冗長性、柔軟な配置およびデータモデル、単一アクセス（各地理的位置に１つ）、ならびに単一障害点の不存在。

【０００５】

この点において、従来の中央管理通信データベースについての地理的冗長性は一般に、メインノードに加えて複製されたノードを有するものとして理解されてきた。ここで、メインノードは動作中であり、複製されたノードはメインノードが何らかの理由でダウンした場合に動作を再開するためにスタンバイしている。

【０００６】

近年、純粋な集中データベースは、いくつかある欠点の中で特に、信号転送の観点で極めて高コストである。実際に、一部のデータが他のデータよりも頻繁にアクセスされるように様々なアプリケーションデータがするため、そのリソースの使用は十分にバランスが取られていない。集中データベースが位置する地理的位置に関して、経済的な観点だけで

なく負荷および障害のリスクの観点の両方において、信号転送のコストについて選択が極めて重要になりうる。明らかに、データベースクライアントとデータベース自体との間の信号経路が長くなると、ノードがその最終宛先へ信号をさらに送出することができなくなるリスクが大きくなる。同様に、この信号パスが長くなると、通信ネットワークがサポートする負荷が大きくなり、実行時間が長くなる。これとは別に、通信ネットワークは、他のネットワーク事業者に属している様々なアクセスネットワークを通じてしばしば通信される様々な遠くの領域を通じて広がる。このシナリオでは、この信号経路が長くなると、より多くの通信事業者が影響を受け、より多くのコストが引き出されうる。

【０００７】

他方で、他の技術を調査すると、分散データベースは、場合によっては中央データベース管理システムの制御の下で、物理的または論理的に分散された複数のデータベースとしてみなされてもよく、必ずしもすべての記憶装置が共通の処理部に接続される必要はない。よって、分散データベースは同一の物理的位置に位置する複数のコンピュータで構築されてもよいし、相互接続されたコンピュータのネットワーク上に点在してもよい。

【０００８】

一般的に言うと、データベースインスタンスの分散はデータ分散自体の結果である必要はなく、高可用性システムおよび地理的冗長性を得るためのデータレプリケーションのために有益である。特に、従来の集中データベースシステムを、中央管理または分散された共通バックエンドへアクセスする複数の通信データベースフロントエンドで置き換えるいわゆるデータレイヤアーキテクチャ技術は、通信データベースについての上記の要件を満たし、データ分散およびレプリケーションに利用可能な分散データベースの例示の適用性である。

【０００９】

分散データベースの様々なデータベースインスタンスにおけるデータのレプリケーションは、様々なデータベースインスタンスにおいて現存するレプリカを最新に維持するために複雑な管理を必要とする。さらに、通信ネットワークの分野の従来の分散データベースのクライアントは、近くのデータベースインスタンスにおける任意のデータベース関連の動作が常に実行可能であるとは限らない。よって、当該データベース関連の動作に必要な信号経路を常に最短化できるとは限らない。さらに、通信ネットワークの分野における従来の分散データベースのクライアントは、任意のデータベースインスタンスが利用不可能であるデータへのアクセスで問題を経験しうる。

【発明の概要】

【発明が解決しようとする課題】

【００１０】

本発明は上述の欠点を少なくとも軽減し、データの少なくとも１つのパーティションのレプリカを格納するようにそれぞれが構成された複数のノードを有する改良された分散データベースシステムと、当該分散データベースシステムを扱う方法とを提供する。

【課題を解決するための手段】

【００１１】

本発明の第１側面では、データの少なくとも１つのパーティションのレプリカを格納するようにそれぞれが構成された複数のノードを有する分散データベースシステムを扱う新たな方法が提供される。

【００１２】

この方法は、格納されるデータをｐ個のパーティションに分割するステップと、各パーティションをｒ個のレプリカに複製するステップと、各パーティションについて、前記ｒ個のレプリカを前記複数のノードの中から選択された対応するｒ個のノードに分配するステップと、相互にアドレスを指定するために使用可能である他のノードの識別子のリストを各ノードに設定するステップと、前記複数のノードの中から２つ以上のノードを起動するステップと、各アクティブノードにおいて、各レプリカの最新の更新と、レプリカ状態と、各レプリカを担当するローカルリソースの状態と、各レプリカの接続状態とのうちか

ら選択された少なくとも１つのイベントを監視するステップと、前記複数のノードの中のノードの起動又は停止の際に、前記アクティブノードの中のどのノードが各パーティションについての現在のマスタノードであり、当該パーティションについての現在のマスタレプリカを担当するとみなされるかを決定するステップと、前記分散データベースシステム内のデータの読み出し／書き込みを行うことの、ノードにおいて受信された任意のリクエストについて、当該データが属するパーティションと当該パーティションについての現在のマスタレプリカを担当する現在のマスタノードとを決定し、当該リクエストを当該現在のマスタノードへルーティングするステップとを有する。

【００１３】

前記アクティブノードの中のどのノードが各パーティションについての前記マスタノードであり、それ故当該パーティションについての現在のマスタレプリカを担当しているとみなされるかを決定する際に、前記方法は、前記アクティブノードの中の少なくとも１つのノードにおいて、各レプリカの最新の更新と、レプリカ状態と、各レプリカを担当するローカルリソースの状態と、各レプリカの接続状態とのうちから選択された前記少なくとも１つのイベントに関する情報を各アクティブノードから収集するステップと、各レプリカについての前記収集されたイベントに依存して、前記アクティブノード内の各レプリカの優先順位を決めるための事前に設定されたルールを適用するステップと、各パーティションについて最高のレプリカ優先度を有する特定のノード内のレプリカを選択するステップであって、このレプリカが前記マスタレプリカであるとみなされ、この特定のノードが前記パーティションについての前記マスタノードであるとみなされるステップとを含んでもよい。

【００１４】

本発明は、前記アクティブノードの中のどのノードが各パーティションについてのマスタノードであり前記パーティションについての現在のマスタレプリカを担当するとみなされるかを決定する際に、２つの主な実施形態、すなわち動作モードを提供する。

【００１５】

第１動作モードでは、処理の観点からすべてのノードは似たもの同士のノードであるため、すべてのノードが同じ情報を処理でき、各パーティションについて、現在のマスタレプリカを担当する同じマスタノードを決定することに到りうる。この実施形態のもとでは、前記少なくとも１つのアクティブノードではなく、分散データベースシステム内の各ノードが以下のステップを実行するように構成されてもよい。すなわち、前記アクティブノードの中の少なくとも１つのノードにおいて、各レプリカの最新の更新と、レプリカ状態と、各レプリカを担当するローカルリソースの状態と、各レプリカの接続状態とのうちから選択された前記少なくとも１つのイベントに関する情報を各アクティブノードから収集するステップと、各レプリカについての前記収集されたイベントに依存して、前記アクティブノード内の各レプリカの優先順位を決めるための事前に設定されたルールを適用するステップと、各パーティションについて最高のレプリカ優先度を有する特定のノード内のレプリカを選択するステップであって、このレプリカが前記マスタレプリカであるとみなされ、この特定のノードが前記パーティションについての前記マスタノードであるとみなされるステップとである。

【００１６】

第２動作モードでは、前記複数のノードの中の任意のノードの起動または停止の際に、前記方法は前記アクティブノードが起動された順序を決定するステップをさらに有してもよい。この場合に、前記アクティブノードが起動された順序を決定するようにすべてのノードが構成され、その結果、最初に起動されたアクティブノードは以下のステップの実行を担当するいわゆるシステムマスタモニタノードであるとみなされる。すなわち、　各レプリカの最新の更新と、レプリカ状態と、各レプリカを担当するローカルリソースの状態と、各レプリカの接続状態とのうちから選択された前記少なくとも１つのイベントに関する情報を各アクティブノードから収集するステップと、各レプリカについての前記収集されたイベントに依存して、前記アクティブノード内の各レプリカの優先順位を決めるた

めの事前に設定されたルールを適用するステップと、各パーティションについて最高のレプリカ優先度を有する特定のノード内のレプリカを選択するステップであって、このレプリカが前記マスタレプリカであるとみなされ、この特定のノードが前記パーティションについての前記マスタノードであるとみなされるステップと、各パーティションについて選択された前記マスタレプリカと当該マスタレプリカを保持する前記マスタノードとに関して他のアクティブノードへ通知するステップとである。

【００１７】

前記収集されたイベントに依存して、事前に設定されたルールの適用が２つ以上のレプリカについて同一の優先度を生み出す場合に特に有用であるが、各パーティションについて、前記ｒ個のレプリカを対応するｒ個のノードに分配するこの方法の前記ステップは、他の基準が同一のレプリカ優先度を生み出す場合に適用されるデフォルトレプリカ優先度を各レプリカに設定するステップを含んでもよい。この場合に、前記アクティブノードの中のどのノードが各パーティションについてのマスタノードであり、当該パーティションについての現在のマスタレプリカを担当するとみなされるかを決定する前記ステップは、前記少なくとも１つのアクティブノードにおいて、各パーティションについての所与のデフォルトレプリカ優先度が設定されていることに関する情報を少なくとも１つのアクティブノードから収集するステップと、各パーティションについて最高のデフォルトレプリカ優先度を有する特定のノード内のレプリカを選択するステップであって、このレプリカが前記マスタレプリカであるとみなされ、この特定のノードが前記パーティションについての前記マスタノードであるとみなされるステップとを含んでもよい。特に、上記の第１動作モードに従って前記方法が動作する場合に、前記少なくとも１つのノードではなく、前記分散データベースシステム内の各ノードがこれらのステップを実行するように構成されてもよい。

【００１８】

しかしながら、上記第２動作モードに従って前記方法が動作する場合に、最初に起動されたアクティブノードはシステムマスタモニタであるとみなされ、前記アクティブノードの中のどのノードが各パーティションについての現在のマスタノードであり、当該パーティションについての現在のマスタレプリカを担当するとみなされるかを決定するステップは、当該システムマスタモニタノードにおいて、各パーティションについての所与のデフォルトレプリカ優先度が設定されていることに関する情報を少なくとも１つのアクティブノードから収集するステップと、各パーティションについて最高のデフォルトレプリカ優先度を有する特定のノード内のレプリカを選択するステップであって、このレプリカが前記マスタレプリカであるとみなされ、この特定のノードが前記パーティションについての前記マスタノードであるとみなされるステップと、各パーティションについて選択された前記マスタレプリカと当該マスタレプリカを保持する前記マスタノードとに関して他のアクティブノードへ通知するステップとを含んでもよい。

【００１９】

一般的に言うと、すべての読み出し／書き込み動作は関連するパーティションについてのマスタレプリカ上で実行される。従って、マスタレプリカの内容と、同一のパーティションについての他のレプリカの内容とは所定の時点において異なりうる。各パーティションについての様々なレプリカの間での一貫性を維持するために、前記方法は、各パーティションについて各アクティブノードにおいて前記現在のマスタレプリカの内容を、当該パーティションについての前記現在のマスタレプリカを担当する前記現在のマスタノードからコピーするステップをさらに有してもよい。コピーするステップが行われる場合に、前記方法は、各アクティブノードにおいてコピーされた各レプリカについて、行われた前記最新の更新と、レプリカ状態と、前記レプリカを担当するローカルリソースの状態と、前記レプリカの接続状態とのうちの少なくとも１つを作成するステップをさらに有してもよい。これは特に、前記マスタレプリカを担当する現在のマスタノードに障害が生じ、ダウンまたは利用不可能になり、もしくは非アクティブノードになる場合に将来の別のマスタレプリカを選択するために有利である。

【００２０】
他方で、本発明の第２側面に従うと、複数のノードを有する分散データベースシステムであって、各ノードはデータの少なくとも１つのパーティションのレプリカを格納するように構成される改良された分散データベースシステムが提供される。この分散データベースシステムにおいて、各ノードは、格納されるデータの少なくとも１つのデータパーティションのレプリカを格納するとともに、相互にアドレスを指定するために使用可能である他のノードの識別子を格納するためのデータ記憶装置と、前記分散データベースシステムの他のノードと通信するとともに、前記分散データベースシステムにおいて読み出し／書き込み動作を要求するクライアントと通信するための入出力部と、各レプリカの最新の更新と、レプリカ状態と、各レプリカを担当するローカルリソースの状態と、各レプリカの接続状態とのうちから選択された少なくとも１つのイベントを監視するための監視部と、前記データ記憶装置、前記監視部及び前記入出力部と連携して、前記分散データベースシステム内のアクティブノードの中のどのノードが各パーティションについての現在のマスタノードであり、当該パーティションについての現在のマスタレプリカを担当するとみなされるかを決定し、前記分散データベースシステム内のデータの読み出し／書き込みを行うことの、受信された任意のリクエストについて、当該データが属するパーティションと当該パーティションについての現在のマスタレプリカを担当する現在のマスタノードとを決定し、当該リクエストを当該現在のマスタノードへルーティングするための制御部とを含む。

【００２１】
上記方法と整合して、前記分散データベースシステムが第１動作モードに従って動作するか第２動作モードに従って動作するかにかかわらず、各ノードの前記処理部、前記監視部、前記データ記憶装置及び前記入力部は、各レプリカの最新の更新と、レプリカ状態と、各レプリカを担当するローカルリソースの状態と、各レプリカの接続状態とのうちから選択された少なくとも１つのイベントに関する情報を前記分散データベースシステムの各アクティブノードから収集し、各レプリカについての前記収集されたイベントに依存して、前記アクティブノード内の各レプリカの優先順位を決めるための事前に設定されたルールを適用し、各パーティションについて最高のレプリカ優先度を有する特定のノード内のレプリカを選択し、このレプリカが前記マスタレプリカであるとみなされ、この特定のノードが前記パーティションについての前記マスタノードであるとみなされるように構成されてもよい。

【００２２】
特に、前記収集されたイベントに依存して、事前に設定されたルールの適用が２つ以上のレプリカについて同一の優先度を生み出す場合の上記方法に有利に整合して、各ノードの前記データ記憶装置は、他の基準が同一のレプリカ優先度を生み出す場合に適用されるデフォルトレプリカ優先度を示すようにレプリカごとに設定されたインジケータを格納するように構成されてもよい。この場合に、各ノードの前記処理部、前記監視部、前記データ記憶装置及び前記入力部は、パーティションについての所与のデフォルトレプリカ優先度が設定されていることに関する情報を少なくとも１つのアクティブノードから収集し、各パーティションについて最高のデフォルトレプリカ優先度を有する特定のノード内のレプリカを選択し、このレプリカが前記マスタレプリカであるとみなされ、この特定のノードが前記パーティションについての前記マスタノードであるとみなされるようにさらに構成されてもよい。

【００２３】
また、上記方法に整合して、分散データベースシステムが第２動作モードに従って動作する場合に特に適用可能であるが、各ノードの前記処理部、前記監視部、前記データ記憶装置及び前記入力部は、前記分散データベースシステムの各アクティブノードから、前記アクティブノードが起動された順序を決定するための情報を収集するように構成されてもよい。

【００２４】

この場合に、最初に起動されたアクティブノードはシステムマスタモニタであるとみなされてもよく、前記システムマスタモニタの前記処理部、前記監視部、前記データ記憶装置及び前記入力部は、各レプリカの最新の更新と、レプリカ状態と、各レプリカを担当するローカルリソースの状態と、各レプリカの接続状態とのうちから選択された少なくとも１つのイベントに関する情報を各アクティブノードから収集し、各レプリカについての前記収集されたイベントに依存して、前記アクティブノード内の各レプリカの優先順位を決めるための事前に設定されたルールを適用し、各パーティションについて最高のレプリカ優先度を有する特定のノード内のレプリカを選択し、このレプリカが前記マスタレプリカであるとみなされ、この特定のノードが前記パーティションについての前記マスタノードであるとみなされ、各パーティションについて選択されたマスタレプリカと当該マスタレプリカを保持するマスタノードとに関して他のアクティブノードへ通知するように構成されてもよい。

【００２５】

さらに、上述のデフォルト優先度が第２動作モードの下でのパーティションについてのレプリカの優先順位を決めるために関連する場合に、前記処理部、前記監視部、前記データ記憶装置及び前記入力部は、パーティションについての所与のデフォルトレプリカ優先度が設定されていることに関する情報を少なくとも１つのアクティブノードから収集し、各パーティションについて最高のデフォルトレプリカ優先度を有する特定のノード内のレプリカを選択し、このレプリカが前記マスタレプリカであるとみなされ、この特定のノードが前記パーティションについての前記マスタノードであるとみなされ、各パーティションについて選択された前記マスタレプリカと当該マスタレプリカを保持する前記マスタノードとに関して他のアクティブノードへ通知するようにさらに構成されてもよい。

【００２６】

各パーティションについての様々なレプリカの間での一貫性を維持し、それゆえ上記の方法と整合するために、各ノードの前記処理部、前記監視部、前記データ記憶装置及び前記入力部は、各パーティションについて各アクティブノードにおいて前記現在のマスタレプリカの内容を、当該パーティションについての前記現在のマスタレプリカを担当する前記現在のマスタノードからコピーするようにさらに構成される。さらに、現在のマスタレプリカが何らかの理由で非アクティブになる場合にマスタレプリカを担当する別のマスタノードをさらに選択するために、各ノードの前記処理部、前記監視部、前記データ記憶装置及び前記入力部は、各アクティブノードにおいてコピーされた各レプリカについて、行われた前記最新の更新と、レプリカ状態と、各レプリカを担当するローカルリソースの状態と、前記レプリカの接続状態とのうちの少なくとも１つを作成するようにさらに構成されてもよい。

【００２７】

他方で、本発明の第３側面に従って本発明はコンピュータプログラムで実施されてもよい。コンピュータプログラムは入出力部と処理部とを有するコンピュータの内部メモリにロード可能なコンピュータプログラムであって、上記の方法を実行するように構成された実行可能なコードを含む。特に、この実行可能なコードはコンピュータにおいて読み出し可能な記録媒体に記録されてもよい。

【００２８】

本発明の機能、目的および利点は添付の図面と併せて本明細書を読むことによって明らかになるだろう。

【図面の簡単な説明】

【００２９】

【図１Ａ】格納されるデータをｐ個のパーティションに分割し、各パーティションをｒ個のレプリカに複製するために実行される動作シーケンスの簡略図。

【図１Ｂ】各パーティションについて、ｒ個のレプリカを複数のノードの中から選択された対応するｒ個のノードに分配し、相互にアドレスを指定するために使用可能である他のノードの識別子のリストを各ノードに設定する動作シーケンスの簡略図。

【図１Ｃ】データの少なくとも１つのパーティションのレプリカを格納するように各ノードが構成された複数のノードを有する分散データベースシステムを扱う方法を実行するために、図１Ａおよび図１Ｂに説明された動作と連動して実行される動作シーケンスの簡略図。
【図２】本発明の一部の実施形態を説明するために有用なデータを有する分散データベース内の複数のノードの例示の構成の簡略図。
【図３】分散データベースに含まれる複数のノードの中のノードの例示の実装を説明する図。
【図４】本発明の一部の実施形態を説明するためにクラスタとして構成された有用なデータを有する分散データベース内の複数のノードの中のノードの例示の構成の簡略図。
【図５】分散データベース内のデータの読み出し／書き込みを行うことの、ノードにおいて受信された任意のリクエストを、このようなデータが属するパーティションについての現在のマスタレプリカを担当する現在のマスタノードへルーティングするために実行される動作シーケンスの簡略図。
【図６】アクティブノードの中のどのノードが、他のノードの調整と各レプリカのついてのマスタノードがどれであるかの決定とを担当するコントローラシステムモニタであるとみなされるかを決定するために本発明の実施形態に従って提供される例示の状態マシンを説明する図。
【図７】分散データベースシステム内の複数のノードの起動の際にアクティブノードの中のどのノードがコントローラシステムモニタであるとみなされるかを決定するために、図６に示される状態マシンのサポートのもので実行される例示の動作シーケンスを説明する図。
【図８】以前にみなされていたコントローラシステムモニタの非アクティブ化の際にアクティブノードの中のどのノードがコントローラシステムモニタであるとみなされるかを決定するために、図６に示される状態マシンのサポートのもので実行される別の例示の動作シーケンスを説明する図。
【発明を実施するための形態】
【００３０】
データの少なくとも１つのパーティションのレプリカをそれぞれが格納する複数のノードを有する改良された分散データベースシステムと当該分散データベースシステムを扱う方法との現時点で好適な実施形態を以下に説明する。
【００３１】
通信データベースシステムは複数の地理的に分散されたノードを含んでもよく、各ノードは複数のデータ記憶部を含んでもよく、各ノード内の各データ記憶部はデータの部分集合の特定のレプリカ、すなわちパーティションのレプリカを割り振ってもよい。図１Ａおよび図１Ｂに説明されるように、ノード１～４のデータ記憶部の間でのデータ集合１０の例示の分配は本発明に従って提供される複数のステップに従って実行されうる。
【００３２】
図１Ａに示されるように、データ集合１０はステップＳ－００５の間に複数のパーティション１１～１４に分割され、各パーティションはデータ集合１０の特定の部分集合を有する。次いで、各パーティションについて、ステップＳ－０１０の間に複数のレプリカが生成される。各パーティションについてのレプリカの個数はすべてのパーティションについて同じである必要はない。よって、図１Ａに示される例のように、４つのレプリカ１１１～１１４がパーティション１１について生成され、３つのレプリカ１２１～１２３とレプリカ１４１～１４３とがパーティション１２と１４とについてそれぞれ生成され、ただ２つのレプリカ１３１、１３２がパーティション１３について生成される。
【００３３】
図１Ｂに示されるように、これらのレプリカは、必要となる地理的な分散を決定する予備的なステップＳ－０１５の間にパーティションごとにグループ分けされてもよい。これとは別に、データベースシステムを構成するノードの地理的な分配を決定する際に、各ノ

ードはステップＳ－０１７の間にアドレス目的に利用可能な識別子が割り当てられる。図１Ｂの例示の説明は別個の識別子Ｎ－１ＩＤ、Ｎ－２ＩＤ、Ｎ－３ＩＤおよびＮ－４ＩＤを有する４つのノード１～４で構成される。

【００３４】

次いで、ステップＳ－０２０の間に、各パーティションについて生成されたレプリカは、データベースシステムを構成するノードに分配されてもよい。図１Ｂに説明される例では、第１パーティションの第１レプリカ１１１はノード１に格納され、第１パーティションの第２レプリカ１１２はノード２に格納され、第１パーティションの第３レプリカ１１３はノード３に格納され、第１パーティションの第４レプリカ１１４はノード４に格納され、第２パーティションの第１レプリカ１２１はノード３に格納され、第２パーティションの第２レプリカ１２２はノード１に格納され、第２パーティションの第３レプリカ１２３はノード２に格納され、第３パーティションの第１レプリカ１３１はノード４に格納され、第３パーティションの第２レプリカ１３２はノード１に格納され、第４パーティションの第１レプリカ１４１はノード３に格納され、第４パーティションの第２レプリカ１４２はノード４に格納され、第４パーティションの第３レプリカ１４３はノード２に格納される。よって、各パーティションについてのレプリカを格納するためにすべてのノードが必要であるとは限らず、本発明の側面に従って提供される分散データベースシステムの各ノードにすべてのパーティションがレプリカを有さなければならないというわけではない。

【００３５】

これとは別に、各ノードはまたこのステップの間に、他のノードの識別子を設定されてもよい。よって、ノード１はノード２、３、４を識別する識別子１５１を格納する。ノード２はノード１、３、４を識別する識別子１５２を格納する。ノード３はノード１、２、４を識別する識別子１５３を格納する。ノード４はノード１、２、３を識別する識別子１５４を格納する。

【００３６】

動作中に、以下に説明される所定のイベントに依存して、所定のノード内の１つの特定のレプリカが最高の優先度を取得してもよく、それ故パーティションについてのマスタレプリカであると決定される一方で、所定のノードは当該パーティションについてのマスタノードとみなされる。しかしながら、イベントが相異なるノード内のパーティションの相異なるレプリカについて同一の優先度を生み出し、その結果、マスタレプリカが決定され得ない状況が存在しうる。２つ以上のレプリカに同一の優先度が与えられる曖昧さを省くために、本発明は他の基準、すなわち上記のイベントの処理結果は同一のレプリカ優先度を生み出す場合に、各レプリカを適用されるデフォルトのレプリカ優先度を設定することを提供する。図２に示されるように、各ノード１～４はそれぞれ、複数のパーティションについてレプリカ１１０１、２１０１、３１０１、４１０１と、レプリカごとのデフォルト優先度１１０２、２１０２、３１０２、４１０２を含む。

【００３７】

分散データベースシステムの各ノードが上述のように構成されると、分散データベースシステムはオペレータが望むようにノードごとにまたは一斉に動作に入ることの準備が整う。

【００３８】

図１Ｃは本発明の別の側面に従って上記の分散データベースシステムを扱う方法を実行するための後続の動作シーケンスを説明する。すべてのノードが同様に振舞うが、本発明の実施形態に従って以下に説明されるように、各ノードにおける完全な動作シーケンスは相異なるノードが起動される順序に依存してもよい。よって、図１Ｃの例はノード２がステップＳ－０３０の間に最初に起動されるものであり、その後にステップＳ－０３５の間にノード３が起動され、次いでステップＳ－０６０の間にノード４であり、最後にステップＳ－０７０の間にノード１であるシナリオが説明される。

【００３９】

各ノード２、３、４、１の起動の後に、各アクティブノードにおいて当該ノードが起動された開始時刻を決定するそれぞれのステップＳ－０４０、Ｓ－０４５、Ｓ－０８０、Ｓ－０７５が続いてもよい。このオプションのステップは、以下に説明される他のノードとの連携と各パーティションについてのマスタレプリカがどれであるかの決定とを担当するシステムマスタモニタとして動作するノードを有するデータベースシステムが動作している場合に、アクティブノードが起動された順序をさらに決定するために有用である。この点において、図２に説明されるように、各ノード１～４は、開始時刻からノードが稼動している動作時間を示すそれぞれの表示１１０４、２１０４、３１０４、４１０４を含む。

【００４０】

各ノードについて開始時刻を決定するステップが実行されるかどうかに係わらず、総括的に、各ノード２、３、４、１の起動の後に、各レプリカの最新の更新と、レプリカ状態と、各レプリカを担当するローカルリソースの状態と、各レプリカの接続状態とから選択される少なくとも１つのイベントを各アクティブノード２、３、４、１において監視するステップＳ－０５０、Ｓ－０５５、Ｓ－０９５、Ｓ－０９０のそれぞれが続く。

【００４１】

この目的のために、図２に説明される例のように、各ノード１～４はそれぞれ、複数のパーティションについて、レプリカ１１０１、２１０１、３１０１、４１０１とともに、レプリカごとに最新の更新のインジケータ１１０３、２１０３、３１０３、４１０３を含み、且つ上述のようにレプリカごとのデフォルトの優先度１１０２、２１０２、３１０２、４１０２を含む。図２に示されるデータの例とは別に、図４に示されるように各ノード１～４においてレプリカごとに他のデータが格納されてもよい。よって、図４に説明される例のように、ノード２はそれぞれ複数のパーティションについてレプリカ１１２、１２３、１４３とともに、レプリカごとの接続状態のインジケータ３１２、３２３、３４３を含み、且つレプリカ２１２、２２３、２４３ごとに、レプリカがマスタレプリカであるとみなされているか、起動中であるがマスタではないアクティブレプリカであるか、起動していないとみなされうるように設定が行われているスタンバイレプリカであるかを示すレプリカ状態を含む。特に、各ノードはさらに、図示されてはいないが、各レプリカについてのローカルリソースを監督するプロセスがこのようなイベントを監視するステップから分離されうるように、レプリカごとにローカルリソースの状態のインジケータを含めるための、レプリカごとの記憶フィールドをさらに有してもよい。これに替えて、各アクティブノードにおいて選択の少なくとも１つのイベントを監視する上述のステップは、各レプリカについてローカルリソースを監督するステップと各ローカルリソースの状態を決定するステップとを含んでもよい。

【００４２】

なおも図４を参照して、この分散データベースシステムの各ノード１～４は複数のクラスタで構成されてもよく、各クラスタは、パーティションのレプリカと、レプリカについての接続状態のインジケータと、レプリカについてのレプリカ状態と、レプリカごとの最新の更新のインジケータと、レプリカごとのデフォルト優先度とを含む。よって、図４に説明された例として、ノード２はクラスタ１６０、１７０、１８０を備え、各クラスタはそれぞれレプリカについてのデータおよびインジケータとともに、パーティションのレプリカ１１２、１２３、１４３を含む。

【００４３】

図１Ｃに説明される動作シーケンスに戻り、イベントを監視するステップが各ノードにおいて実行されると、各ノードはステップＳ－０６５、Ｓ－０８５、Ｓ－１００、Ｓ－１１０のそれぞれの間に、各ノードに知られているすべての他のノードへのいわゆる「アライブ（Ａｌｉｖｅ）」メッセージの送信を開始する。

【００４４】

当業者は理解するかもしれないが、ノード４がノード１よりも前に開始されたにもかかわらず、ノード１における一部の動作はノード４における対応する動作の前に生じる。これは一般的に起こりうる。なぜなら、１つのノードにおけるプロセッサ負荷は別のノード

における負荷よりも高いかもしれず、結果として前者における性能が低下するからであり、またこれは様々なネットワーク経路を通じる様々な信号遅延に起因しうる。

【００４５】

図１Ｃに特に例示として説明されるように、ノード２は、ノード１が起動される前に、ステップＳ－０６５でいわゆる「アライブ」メッセージを送信した。ノード２はステップＳ－１００の間にノード１からの「アライブ」メッセージを受信するので、ノード２はノード１が当初の「アライブ」メッセージを受信しておらず、ノード１がすべてのノードからの完全な情報を有していないことを認識してもよい。これらの状況では、現在の例示の状況におけるノード２のようなノードはイベントを再び監視するステップＳ－１０５と、ノード２に知られているすべてのノードへ「アライブ」メッセージを再び送出するステップＳ－１１５とを実行する。さらに、これも図１Ｃに示されるように、ノード２においてイベントを監視するステップＳ－１０５と「アライブ」メッセージを送出するステップＳ－１１５との中間のステップＳ－１１０の間にノード２においてノード３からの「アライブ」メッセージが受信され、その結果としてノード２から送出された最新の「アライブ」メッセージは、各ノードがすべての他のアクティブノードを認識している最新であるとみなされうる。

【００４６】

これらの「アライブ」メッセージは分散データベースシステムのノード間で交換され、分散データベースシステムのノードの起動または停止の際に、以下に説明されるように、アクティブノードのうちのどのノードが各パーティションについての現在のマスタノードであり、当該パーティションについての現在のマスタレプリカを担当するとみなされるかを決定するために有用である。

【００４７】

上述のように、動作中に決定された所定のイベントが、パーティションについてのマスタレプリカと当該マスタレプリカを担当するマスタノードとを決定する際に考慮に入れられる。各パーティションについてのマスタレプリカがどれであるかを決定するために、以下の情報が考慮に入れられてもよい。すなわち、各パーティションについてどのレプリカが完全な情報、つまり更新レベルを有する直近に更新された内容を有するかと、明らかに稼動中のレプリカだけが適格であるので各パーティションについての各レプリカのレプリカ状態と、各パーティションについての各レプリカの接続状態と、パーティションの各レプリカについて設定されたデフォルト優先度である。さらに、デフォルト優先度は以前の基準の結果を上書きするように構成されてもよく、以前の基準がパーティションのレプリカの２つ以上について同じ結果を生み出す場合にのみ適用可能となるように構成されてもよい。

【００４８】

さらに、動作中に決定されたこれらのイベントに依存して、アクティブノード内の各レプリカの優先順位を決めるために事前に設定されたルールが適用されてもよい。例えば、事前に設定されたルールは、接続状態が更新レベルに優先度を引き継ぐようなものであってもよいし、レプリカ状態のすぐ後でデフォルト優先度が考慮に入れられるものであってもよいし、レプリカの優先順位を決めるためのイベントに関する他の如何なる基準であってもよい。

【００４９】

パーティションについての各レプリカの内容だけでなく、このような内容の更新に関して、所定の時刻に各レプリカがマスタレプリカから更新するような従来のルーチンが提供されてもよい。よって、パーティションについてのすべてではないレプリカが同時に内容を更新し、且つ当該パーティションについてのすべてではないレプリカが同じ速度で更新を進行する。更新中に、分散データベースシステム内の各ノードは、レプリカの内容が完全な情報であるとみなされうるかどうかと、交換が実行された時点とを、他のノードと交換される情報がこの点において考慮するように、どれくらい更新が進行しているかを監視する必要がある。

【００５０】
パーティションについてのマスタレプリカと当該マスタレプリカを担当するマスタノードとの決定に関して上述のイベントを考慮に入れるために、各ノードにおいて監視されたイベントは分散データベースシステム内の他のノードへ通信される。この目的のために、各ノード１～４は他のノードの仮想ＩＰアドレスを設定されてもよく、または互いに識別してアドレスを指定するために別個のノード識別子１５１～１５４を利用してもよい。各ノード１～４は定期的に、例えば連続した遅延時間が満了した後に、他のノードへ「アライブ」メッセージを送信してもよい。有利には本発明の実施形態において、「アライブ」メッセージは、ＴＣＰで簡単に検出される一方向リンク障害の可能性を回避するために、ある種のハートビートについてＵＤＰを利用する代わりに既知のＴＣＰプロトコルで送信されてもよい。

【００５１】
さらに正確に、各「アライブ」メッセージは受信側ノードに対して送信側ノードを識別するノード識別子Ｎ－１ＩＤ、Ｎ－２ＩＤ、Ｎ－３ＩＤ、Ｎ－４ＩＤを含み、パーティションの各レプリカについて、レプリカが属するパーティションの識別子、レプリカ状態、更新レベル、更新時刻、接続状態およびデフォルト優先度のうちの少なくとも１つを含んでもよい。

【００５２】
これに加えて、アクティブノードが起動された順序をさらに決定するために特に有用であるように、任意のノード１～４から分散データベースシステム内の他のノードへ送信された各「アライブ」メッセージはそれぞれ、送信側ノードが自身の開始時刻からアクティブである動作時間の表示１１０４、２１０４、３１０４、４１０１を含んでもよい。

【００５３】
本発明は２つの主な実施形態、すなわちアクティブノードのうちのどのノードが各パーティションについてのマスタノードであり、当該パーティションについての現在のマスタレプリカを担当するとみなされるかの決定における動作モードを提供する。

【００５４】
第１動作モードでは、処理の観点からすべてのノードは似たもの同士のノードであるため、各ノードはすべての他のノードから、そこで決定された監視情報を有する「アライブ」メッセージを受信し、その結果、すべてのノードが同じ情報を処理でき、各パーティションについて、現在のマスタレプリカを担当する同じマスタノードを決定することに到りうる。この実施形態のもとでは、分散データベースシステム内の各ノードは以下のステップを実行するように構成されてもよい。すなわち、各レプリカの最新の更新、レプリカ状態、各レプリカを担当するローカルリソースの状態、および各レプリカの接続状態から選択された少なくとも１つのイベントに関する情報を各アクティブノードから収集するステップと、各レプリカについて収集されたイベントに依存して、アクティブノード内の各レプリカの優先順位を決めるための事前に設定されたルールを適用するステップと、各パーティションについて最高のレプリカ優先度を有する特定のノード内のレプリカを選択するステップであって、このレプリカがマスタレプリカであるとみなされ、この特定のノードが当該パーティションについてのマスタノードであるとみなされるステップとである。

【００５５】
特に有利にはこの第１動作モードの下で如何なる特定のレプリカについてこれらのイベントが最高のレプリカ優先度を結果として生じない場合に、分散データベースシステム内の各ノードはさらに以下のステップを実行するように構成されてもよい。すなわち、パーティションについての所与のデフォルトレプリカ優先度を設定されている情報を少なくとも１つのアクティブノードから収集するステップと、各パーティションについて最高のデフォルトレプリカ優先度を有する特定のノード内のレプリカを選択するステップであって、このレプリカがマスタレプリカであるとみなされ、この特定のノードが当該パーティションについてのマスタノードであるとみなされるステップとである。

【００５６】

第２動作モードでは、アクティブノードが起動された順番を決定するためにすべてのノードが「アライブ」メッセージを処理し、その結果、最初に起動されたアクティブノードは以下のステップの実行を担当するいわゆるシステムマスタモニタノードであるとみなされる。すなわち、各レプリカの最新の更新、レプリカ状態、各レプリカを担当するローカルリソースの状態、および各レプリカの接続状態から選択された少なくとも１つのイベントに関する情報を各アクティブノードから収集するステップと、各レプリカについて収集されたイベントに依存して、アクティブノード内の各レプリカの優先順位を決めるための事前に設定されたルールを適用するステップと、各パーティションについて最高のレプリカ優先度を有する特定のノード内のレプリカを選択するステップであって、このレプリカがマスタレプリカであるとみなされ、この特定のノードが当該パーティションについてのマスタノードであるとみなされるステップと、各パーティションについて選択されたマスタレプリカと当該マスタレプリカを保持するマスタノードとに関して他のアクティブノードへ通知するステップとである。

【００５７】

特に有利にはこの第２動作モードの下で如何なる特定のレプリカについてこれらのイベントが最高のレプリカ優先度を結果として生じない場合に、分散データベースシステム内のシステムマスタモニタノードはさらに以下のステップを実行するように構成されてもよい。すなわち、パーティションについての所与のデフォルトレプリカ優先度を設定されている情報を少なくとも１つのアクティブノードから収集するステップと、各パーティションについて最高のデフォルトレプリカ優先度を有する特定のノード内のレプリカを選択するステップであって、このレプリカがマスタレプリカであるとみなされ、この特定のノードが当該パーティションについてのマスタノードであるとみなされるステップと、各パーティションについて選択されたマスタレプリカと当該マスタレプリカを保持するマスタノードとに関して他のアクティブノードへ通知するステップとである。

【００５８】

上述の方法を実行するために、本発明は総括的にノード１～４のような複数のノードを有する分散データベースシステムを提供する。各ノードはデータの少なくとも１つのパーティションのレプリカを格納するように構成される。ノード２について図３に説明される例のように、各ノードは格納されるデータの少なくとも１つのデータパーティション１１２、１２３、１４３のレプリカ２１０１を格納するとともに、相互にアドレスを指定するために用いられる他のノードの識別子１５２を格納するためのデータ記憶装置１５と、分散データベースシステムの他のノード１、３、４と通信するための入出力部３０と、各レプリカの最新の更新２１０３、レプリカ状態２１２、２２３、２４３、各レプリカを担当するローカルリソースの状態、各レプリカの接続状態３１２、３２３、３４３から選択された少なくとも１つのイベントを監視するための監視部６０と、データ記憶装置、監視部および入出力部と連動して、分散データベースシステムのアクティブノードのうちのどのノードが各パーティションについての現在のマスタノード２１０５であり、当該パーティションについての現在のマスタレプリカを担当するとみなされるかを決定するための処理部２０とを備える。

【００５９】

さらに、この分散データベースシステムでは、各ノードの処理部２０、監視部６０、データ記憶装置１５および入出力部３０は、各レプリカの最新の更新１１０３、２１０３、３１０３、４１０３、レプリカ状態２１２、２２３、２４３、各レプリカを担当するローカルリソースの状態、および各レプリカの接続状態３１２、３２３、３４３から選択された少なくとも１つのイベントに関する情報を各アクティブノードから収集し、各レプリカについて収集されたイベントに依存して、アクティブノード内の各レプリカの優先順位を決めるための上述の事前に設定されたルールを適用し、各パーティションについて最高のレプリカ優先度を有する特定のノード内のレプリカを選択するステップし、このレプリカがマスタレプリカであるとみなされ、この特定のノードが当該パーティションについてのマスタノードであるとみなされるように構成される。

【００６０】
さらに、特に有利には分散データベースシステムにおいて如何なるレプリカについて以前のイベントが最高のレプリカ優先度を結果として生じない場合に、各ノードの処理部２０、監視部６０、データ記憶装置１５および入出力部３０は、パーティションについての所与のデフォルトレプリカ優先度を設定されている情報を少なくとも１つのアクティブノードから収集し、各パーティションについて最高のデフォルトレプリカ優先度を有する特定のノード内のレプリカを選択し、このレプリカがマスタレプリカであるとみなされ、この特定のノードが当該パーティションについてのマスタノードであるとみなされるように構成される。
【００６１】
第２動作モードに従って分散データベースシステムを動作するために、各ノードの処理部２０、監視部６０、データ記憶装置１５および入出力部３０は、アクティブノードが起動された順序を決定するための情報１１０４、２１０４、３１０４、４１０４を分散データベースシステムの各アクティブノードから収集するように構成され、その結果、最初に起動されたアクティブノードがシステムマスタモニタであるとみなされる。
【００６２】
特に、ノードが分散データベースシステムのシステムマスタモニタノードであるとみなされる場合に、当該システムマスタモニタの処理部２０、監視部６０、データ記憶装置１５および入出力部３０はさらに、各パーティションについて選択されたマスタレプリカと当該マスタレプリカを保持するマスタノードとに関して他のアクティブノードへ通知するように構成されてもよい。
【００６３】
分散データベースシステムを扱う上記の方法は、分散データベーシステムのクライアントがどのように当該システムに格納された情報にアクセスしうるか、特に読み出し動作または書き込み動作のためにアクセスしうるかに関する特定の議論が必要となりうる。
【００６４】
原則として、通信ネットワークのホームロケーションレジスタ、認証センタ、またはホーム加入者サーバのような分散データベースシステムのクライアントは分散データベースシステムの任意のノードから任意のデータへアクセスできる。しかしながら、様々なレプリカ内でデータの一貫性を維持するために、レプリカのうちの１つだけが読み出しおよび／または書き込みのリクエストを受信し、これはマスタレプリカであるだろう。上述のように、マスタレプリカ内のデータは他のレプリカへ時々刻々と更新される。
【００６５】
よって、分散データベースシステムの全内容は、分散データベースシステムを構成する任意のノードからアクセス可能である。この目的ために、各ノードは１つ以上のアクセスゲートウェイ（以下、ＡＧ）を含んでもよく、ＡＧはマスタレプリカが位置するノードへデータの読み出し／書き込みを行うリクエストを転送することを担当するエンティティである。データベースプロトコルはアクセスプロトコルと異なりうるため、ＡＧはデータの読み出し／書き込みを行うリクエストをクライアントから受信し、分散データベースシステム内の他のノードへアクセスすることを担当する。性能を最大化するために各ノード内に２つ以上のＡＧが提供されうるため、２つ以上のＡＧの間でトラフィックを分配するためにロードバランサ（以下、ＬＢ）が提供されてもよい。しかしながら、このようなＬＢはただ１つのＡＧを有するノード構成では必要ではない。
【００６６】
図５が説明するように、ノード２はＬＢ１０９ａおよび３つのＡＧ１９１ａ～１９３ａを備えてもよく、第１パーティション１１についてのレプリカ１１２、第２パーティション１２についてのレプリカ１２３、および第４パーティション１４についてのレプリカ１４３を担当してもよく、レプリカ１１２は第１パーティションについてのマスタレプリカであり、ノード２は第１パーティションについての当該マスタレプリカを担当するマスタノードである。一方で、ノード３はＬＢ１９０ｂおよび３つのＡＧ１９１ｂ～１９３ｂを

備えてもよく、第１パーティション１１についてのレプリカ１１３、第２パーティション１２についてのレプリカ１２１、および第４パーティション１４についてのレプリカ１４１を担当してもよく、レプリカ１２１は第２パーティションについてのマスタレプリカであり、ノード３は第２パーティションについての当該マスタレプリカを担当するマスタノードである。

【００６７】

よって、分散データベースシステムを扱うこの方法は、分散データベースシステム内のデータの読み出し／書き込みを行うためにノードにおいて受信された任意のリクエストについて、当該データが属するパーティションと当該パーティションについての現在のマスタレプリカを担当する現在のマスタノードとを決定するステップと、当該リクエストを当該現在のマスタノードへルーティングするステップとを含む。

【００６８】

図５に説明される例のように、分散データベースシステム内のデータの読み出し／書き込みを行うためのクライアント５からのリクエストはノード２のような任意のノードで受信されうる。図５の例はステップＳ－１５０の間にＬＢ１９０ａにおいて受信されたデータの読み出し／書き込みを行うリクエストを説明する。このリクエストはステップＳ－１５１の間にＡＧ１９３ａに割り当てられ、ＡＧ１９３ａは読み出し／書き込みを行うデータが第２パーティション１２に属することを判定し、このＡＧはまた、当該パーティションを担当する現在のマスタノードがノード３であることを判定する。次いで、ＡＧ１９３ａはステップＳ－１５２の間にリクエストをノード３へルーティングする。このリクエストは当該ノード３に２つ以上のＡＧが存在するならばノード３のＬＢ１９０ｂにおいて受信されてもよいし、ノード３が１つのＡＧのみを含むならば唯一のＡＧにおいて受信されてもよいし、この図５に示される例のように、設定手段によってノード３の特定のＡＧ１９１ｂがＡＧ１９３ａに知られているならばこのようなＡＧにおいて受信されてもよい。リクエストを受信するノード３のＡＧ１９１ｂはステップＳ－１５２の間にリクエストに従ってデータの読み出し／書き込みを行うために第２パーティション１２についてのマスタレプリカ１２１にアクセスする。

【００６９】

他方で、図５はまた、データが属するパーティションについてのマスタレプリカを保持するマスタノードにおいてリクエストが受信される場合を説明する。この例の場合では、データの読み出し／書き出しを行うリクエストはステップＳ－１６０の間にノード２のＬＢ１９０ａにおいて受信される。

【００７０】

このリクエストはステップＳ－１６１の間にＡＧ１９１ａに割り当てられてもよく、ＡＧ１９１ａは読み出し／書き込みを行うデータが第１パーティション１１に属することを判定し、このＡＧはまた、当該パーティションを担当する現在のマスタノードがこのノード２であることを判定する。次いで、ＡＧ１９１ａはステップＳ－１６２の間に、リクエストに従ってデータの読み出し／書き込みを行うために、第１パーティション１１についてのマスタレプリカ１１２へアクセスするようにリクエストを内部的にルーティングする。

【００７１】

ノード２のような、分散データベースシステムの任意のノードのＬＢは、図３に説明されるようなクライアント５との通信専用の入出力部５０と、ノードの負荷および性能のバランスをとるために適切なＡＧを選択するように構成された処理部２０のリソースとで構築されてもよい。この入出力部５０は、分散データベースシステムの各ノードが備える入出力部３０の一体部分であってもよいし、当該入出力部３０に含まれる別個のモジュールであってもよい。

【００７２】

ノード２のような、分散データベースシステムの任意のノードのＡＧは図３に説明されるような他のノード１、３、４との通信専用の入出力部４０と、読み出し／書き込みを行

うデータが属するパーティションを決定し、当該パーティションについてのマスタレプリカを担当するマスタノードが現在のノードであるかまたは分散データベースシステムの別のノードであるかを判定し、当該パーティションについてのマスタレプリカを担当するマスタノードが現在のノードである場合に当該データへアクセスするためにデータ記憶装置１５へアクセスし、それ以外の場合に現在のマスタノードであると判定された別のノードへリクエストをルーティングするように構成された処理部２０のリソースとで構築されてもよい。

【００７３】

本発明の実施形態に従うと、監視部６０は上述のイベントを監視し蓄積する一意のユニットである、本明細書におけるいわゆるローカルシステムモニタ（以下、ＬＳＭ）を含んでもよく、またはアクティブＬＳＭとスタンバイＬＳＭとを含み、前者に障害が生じた場合に後者が動作を引き継げるようにしてもよい。以下において、監視部またはＬＳＭへの如何なる言及も、言及されているノード内のアクティブＬＳＭを意味すると解釈される。

【００７４】

特に、本発明が上述の第２動作モードに従って動作する場合に、いわゆるシステムマスタモニタノードの監視部６０がコントローラシステムモニタ（以下、ＣＳＭ）であるとみなされるが、分散データベースシステム内の他のノードの各監視部はなおもＬＳＭとして言及される。よって、ＣＳＭは分散データベースシステム内の各ノードのＬＳＭから受信されたイベント情報を考慮に入れ、且つアクティブノード内の各レプリカの優先順位を決めるための事前に設定されたルールを適用することによって、マスタレプリカでどれであるかを決定してもよい。ＣＳＭは各パーティションについてのマスタノードが何であるかを他のノード内の各ＬＳＭと通信する。

【００７５】

第２動作モードにおいてアクティブノードのうちのどのノードがシステムマスタモニタとみなされるかを決定する際に、図６に説明される例示の状態マシンが適用される。状態間の遷移は他のノードからの「アライブ」メッセージの受信か、タイマの満了かのいずれかに起因する。簡単のために、分散データベースシステム内の他のノードに対してシステムマスタモニタノードを言及するのではなく、以下の議論ではこれらの別個の監視部６０、すなわち当該他のノードの各ＬＳＭに対するシステムマスタモニタノードのＣＳＭを言及する。

【００７６】

図６に説明される本発明の実施形態では、取りうる状態および遷移は以下でありうる。

‐非アクティブ。これは各ＬＳＭが開始した際の状態である。ＬＳＭがこの状態にあるならば、ノードはリクエストに応答せず、マスタレプリカのホストになりえない。この状態へ遷移すると、ＬＳＭは残りのノードへのアライブメッセージの送信を開始する。

‐アクティブ。アクティブ状態では、各ＬＳＭは他のノードに関する情報を有する「アライブ」メッセージをリッスンし、特に存在するならばマスタレプリカを有するＣＳＭからの「アライブ」メッセージをリッスンする。各ＬＳＭはまた、自身のノードに関する情報を送信し、自身のノード内で任意のローカルＡＧへマスタレプリカに関する情報を内部的に分配し、場合によっては「アライブ」メッセージの情報を残りのノードへ転送する。

‐ポテンシャルＣＳＭ。所定の設定可能な期間、すなわち本明細書におけるいわゆるＤＥＬＡＹ＿ＴＩＭＥの間、ＬＳＭがこの状態に留まるならば、このようなＬＳＭはＣＳＭになる。これはまた、ＤＥＬＡＹ＿ＴＩＭＥが満了する前にシステム内の残りのすべてのノードから「アライブ」メッセージを受信する場合にも生じる。

‐ＣＳＭ。この状態に到達するノードは分散データベースシステム内のすべてのパーティションについてどのレプリカがマスタレプリカであるかを決定する。

【００７７】

これにもかかわらず、状態、遷移、またはその両方の他の実施形態が同様に予見可能である。

【００７８】

簡単のために任意の状態から非アクティブ状態への遷移が示されていない図６に示される状態間の遷移に関して、任意のＬＳＭは非アクティブ状態から開始し、ＬＳＭを有するノードが分散データベースシステム内のすべての他のノードへ「アライブ」メッセージを送信するとすぐに、このＬＳＭはアクティブ状態へ移行する。すなわち、非アクティブ状態からアクティブ状態への遷移ＳＴ－１は、非アクティブ状態のＬＳＭを有するノードから分散データベースシステム内のすべての他のノードへ「アライブ」メッセージを送信することである。

【００７９】

２つ以上のサブネットワークがアクティブになることを回避することによってスプリットブレイン状況における一貫性を解決するために提供される本発明のオプションの実施形態に従うと、非アクティブ状態への遷移は、１つ以上のノードに障害が発生し、ＣＳＭを含む（ｎ＋１）／２個未満の稼動中のノードが存在するか、ＣＳＭを含まないｎ／２＋１個未満が存在する場合に発生してもよい。ここで「ｎ」は分散データベースシステム内の全ノード数である。例えば、３つのノードで構成される分散データベースシステムがあり、ＣＳＭのホストであるノードが隔離されるならば、そのサブネットワークは１つだけ、すなわち（３＋１）／２＝２未満のノードを有する。よって、ＣＳＭのホストであるこのノードは非アクティブ状態へ移行する。他方のサブネットワークは２つのノードを有し、これは２／２＋１＝２個以上のノードを有することを意味し、よってこれらはアクティブに留まり、新たなＣＳＭが選択される。このオプションの構成では、各ＬＳＭは、設定可能な期間である、本明細書におけるいわゆるＩＮＡＣＴＩＶＥ＿ＴＩＭＥの後に各ＬＳＭが１つ以上のノードから「アライブ」メッセージを受信しないならば、当該１つ以上のノードがダウンしていることを検出する。非アクティブへの遷移により、任意のＬＳＭは自身の実行時間をリセットし、すなわち再び「若く」になる。これは、隔離された以前のＣＳＭが再びＣＳＭになり、使用されるべきでない情報を送信することを防ぐ。

【００８０】

このオプションの実施形態は分散データベースシステムを適切に構成することによって実行されうる。このような実施形態が望まれないならば、適切な設定パラメータがリセットされ、その結果、システムから分離された複数のノードのｘｘｘが、ノードをアクティブ状態へ移行させ、ノード間で分離されている場合であっても２つ以上のサブネットワークで動作させることに無関係になる。

【００８１】

図６に示される状態間の遷移に戻り、上述のオプションの実施形態が動作するように構成されていると仮定すると、アクティブ状態にあるＬＳＭを有する任意のノードは、他のノードから十分な「アライブ」メッセージが得られないならば、すなわちＣＳＭを含む稼働中の（ｎ＋１）／２個未満またはＣＳＭを含まないｎ／２＋１個未満であるならば、非アクティブ状態へ移行できる。

【００８２】

そうではなく、他のノードから十分な「アライブ」メッセージが受信され、送信側ノードが後から起動したことを受信された「アライブ」メッセージ内の情報が示し、且つメッセージを送信するノードが残っていない場合に、受信側ノード内のＬＳＭはＣＳＭになる。すなわち、アクティブ状態からＣＳＭ状態への遷移ＳＴ－２．１は、他のノードからの十分な「アライブ」メッセージの受信であり、分散データベース内にメッセージを送信するノードが残っていないことである。特に２ノードシステムにおいて、メッセージが受信されることなくＤＥＬＡＹ＿ＴＩＭＥが経過するかも知れず、同様に当該ノードはＣＳＭになる。しかしながら、他のノードから、当該ノードが後から起動したことを示す「アライブ」メッセージを受信したが、「アライブ」メッセージを受信できると予想されるノードがさらに存在する場合に、遷移ＳＴ－２．２が行われ、受信側ノード内のＬＳＭはポテンシャルＣＳＭになる。

【００８３】

以前の遷移について説明したのと同じ理由ために、ポテンシャルＣＳＭ状態から、ノー

ドは非アクティブ状態へ戻りうる。そうでなければ、自身のＬＳＭが確定したＣＳＭであることを示す、さらに予想された「アライブ」メッセージがノードから受信された場合、または他のＬＳＭが先に起動したことを示す、さらに予想された「アライブ」メッセージがノードから受信された場合に、図６に説明される遷移ＳＴ－３．１が行われ、ポテンシャルＣＳＭ状態のＬＳＭは、これらの予想される「アライブ」メッセージのいずれかを受信するノードにおいてアクティブ状態へ移行する。他方で、古いノードを示すさらに予想された「アライブ」メッセージをさらに受信することなくいわゆるＤＥＬＡＹ＿ＴＩＭＥが経過した場合、または先に起動したノードが存在しないことを示す「アライブ」メッセージが残りのノードから受信された場合に、図６に説明される遷移ＳＴ－３．２が行われ、ポテンシャルＣＳＭ状態にあるＬＳＭは、ＤＥＬＡＹ＿ＴＩＭＥが経過したか、これらの予想された「アライブ」メッセージのいずれかが受信されたノードにおいてＣＳＭ状態へ移行する。

【００８４】

上述のように、２種類のタイマが存在する。１つは本明細書におけるいわゆるＤＥＬＡＹ＿ＴＩＭＥタイマであり、自身がＣＳＭであると宣言する前に古いＬＳＭに関して通知する、残りのノードの「アライブ」メッセージをＬＳＭが待つ時間である。もう１つは本明細書におけるいわゆるＩＮＡＣＴＩＶＥ＿ＴＩＭＥタイマであり、ノードがダウンしており利用不可能であることを結論付ける前にこのようなノードからの「アライブ」メッセージをＬＳＭが待つ時間である。

【００８５】

上記の処理が終了すると、ＣＳＭとして確定された監視部６０を有するノードはマスタレプリカのリストを含む「アライブ」メッセージの送信を開始する。

【００８６】

図６に説明される状態マシンの処理に加えて、図７は、３つの例示のノードの中のどのノードがＣＳＭとして確定された監視部６０を有するノードであるかを決定するための分散データベースシステムのノード間での例示の動作シーケンスを示す。上述のように、ＣＳＭは最初に起動されたＬＳＭである。これを決定するために、各ＬＳＭは起動時に図２に示された例のように動作時間２１０４、４１０４を有する「アライブ」メッセージを残りのノードへ送信する。システムの一部であるノードは設定によって知られている。「アライブ」メッセージを送信した後に、ＬＳＭは所定の期間であるＤＥＬＡＹ＿ＴＩＭＥの間、他のノードから「アライブ」メッセージが受信されるのを待つ。ＣＳＭを確立するフェーズは、この時間が経過した場合、または他のノードから十分な「アライブ」メッセージが受信されたならばその前に終了する。このフェーズの間に、この時間までに受信された情報に従ってポテンシャルＣＳＭが割り当てられるかもしれないが、このフェーズが終了するまでそれは確定されないだろう。

【００８７】

よって、図７に説明されるように、ステップＳ－２００の間に起動される最初のノードはノード１である。ＬＳＭ、すなわちノード１の監視部６０が起動するとすぐに、ステップＳ－２０５の間に分散データベースシステムの他の２つのノード１、２へ「アライブ」メッセージを送信し、ＤＥＬＡＹ＿ＴＩＭＥ秒のタイマを開始する。他のノードはまだ稼動していないので、対応する各ＬＳＭはこのような「アライブ」メッセージを受信しない。

【００８８】

次いで、ノード２のＬＳＭはステップＳ－２１０の間に稼動を開始し、ステップＳ－２２０の間に「アライブ」メッセージをノード１、３へ送信し、ＤＥＬＡＹ＿ＴＩＭＥＲ秒の自身のタイマを開始する。ノード２のＬＳＭが起動した後ではあるが、ノード２のＬＳＭが「アライブ」メッセージを送信する前に、ノード３のＬＳＭはステップＳ－２１５の間に起動し、ＤＥＬＡＹ＿ＴＩＭＥＲ秒の自身のタイマを開始する。

【００８９】

ノード１がノード２から「アライブ」メッセージを受信する場合に、ノード１はステッ

プＳ－２２５の間に自身をポテンシャルＣＳＭとして認定する。なぜなら、ノード１が有する情報を用いて、ノード１は、ノード３からの情報を待つものの、早くに起動したＬＳＭだからである。この段階で、ノード３はステップＳ－２３０の間にノード１、２へ「アライブ」メッセージを送信できる。

【００９０】

ノード２がノード３から「アライブ」メッセージを受信する場合に、ノード２はステップＳ－２４０の間に自身をポテンシャルＣＳＭとして認定する。なぜなら、ノード２が有する情報を用いて、ノード２のＬＳＭはノード３のＬＳＭよりも早く起動したからである。同様に、ノード３がノード２から「アライブ」メッセージを受信する場合に、ノード３はステップＳ－２４５の間にノード２をポテンシャルＣＳＭとして認定する。なぜなら、ノード３が有する情報を用いて、ノード１、２はノード１からの情報を待っており、ノード３にはまだ知られていないが、ノード２のＬＳＭは早くに起動したからである。

【００９１】

ノード１がノード３から「アライブ」メッセージを受信する場合に、ノード１はステップＳ－２３５の間に自身を確定したＣＳＭとして認定する。なぜなら、ノード１はシステム内のすべてのノードから「アライブ」メッセージを受信しており、この情報を用いて、ノード１のＬＳＭは早くに起動したものであるからである。よって、ノード１はステップＳ－２５０の間に、ノード１が確定したＣＳＭであることを通知する「アライブ」メッセージを他のノードへ送信する。ノード１から最終的に「アライブ」メッセージを受信するノード２、３はステップＳ－２５５の間にノード１の監視部がＣＳＭであり、現在の状況において他のノードがこの役割を負わないことを認識し、ステップＳ－２６０の間に記録する。

【００９２】

障害の場合にＣＳＭの役割を再割当てできるように、「アライブ」メッセージは各ノードから残りのノードへ定期的に送信される。このように、ノードの追加または削除のような、分散データベースシステムの構成の任意の変化が即座にすべてのノードにおいて知られうる。

【００９３】

よって、本発明の両方の実施形態、すなわち両方の動作モードに従うと、各ノードは、他のノードから受信された情報に基づいて、どのノードがＣＳＭであるとみなされるか、すなわちアクティブノードの中で、より長い動作時間を有するものがどれかを決定してもよい。ノードがＣＳＭになるために、このようなノードは、任意の他のポテンシャルＣＳＭから遠隔の「アライブ」メッセージを受信するための何らかの時間が存在することを保証するために、少なくともＤＥＬＡＹ＿ＴＩＭＥを待つことができるだろう。ＣＳＭ状態に到達したノードは、曖昧さが存在しないことを保証するために残りのノードへ自身の決定を通信してもよい。しかしながら、すべてのノードが第１動作モードに従って振舞う場合にこの通信は必要ない。この通信は信頼性を保証するためにＴＣＰベースであってもよい。再設定時間を低減するために、この振る舞いは、ＣＳＭが他のノードへレプリカ構成を通信する際に間接的に実現されてもよい。この点において、レプリカ状態は「アライブ」メッセージに含まれるので、ひとたび選択されると、ＣＳＭはどのレプリカ状態の設定が動作に適するかがわかる。実際に、この情報はすべてのノードに知られるかもしれないが、これらは第２動作モードのもとでＣＳＭが確認するのを待ってもよい。

【００９４】

各ノードは、当該ノードが「アライブ」メッセージを受信した、現在のノードを有するリストを、当該メッセージ内で受信されたレプリカ状態とともに管理できるだろう。「アライブ」メッセージが受信されるごとに、送信側ノードがアクティブとして設定されてもよい。パーティションについてのマスタノードはアクティブノードの中だけから選択されてもよい。ノードは、それからメッセージを受信することなく、いわゆるＩＮＡＣＴＩＶＥ＿ＰＥＲＩＯＤという期間が経過している場合に利用可能でないとして設定されてもよい。この時間はメッセージ受信時間の平均時間の２倍または３倍でありえ、これは最初に

前述のＤＥＬＡＹ＿ＴＩＭＥに設定されうる。ノードはまた、それに送信された「アライブ」メッセージが届かない場合に利用不可能であるとして設定されてもよい。このようにノードの利用可能性は非常に高速に検出される。

【００９５】

一般的に言うと、ＣＳＭであるとみなされる監視部６０を有するノードは、分散データベースシステム内の他のノードへ、自身のノード識別子、動作時間、各レプリカについてのレプリカ状態、各レプリカについてのマスタノード、状態マシンからのノード状態、（オフラインでさえも）「アライブ」メッセージが受信されているアクティブノードのリスト、現在のマスタレプリカ情報（以下、ＭＲＩ）、およびＭＲＩを設定するＣＳＭについての（実行時間を含む）更新時間およびノードＩＤを送出してもよい。

【００９６】

特に、いわゆるＭＲＩメッセージはホストノードを有するレプリカのリストを含みうる。本発明の実施形態では、ＭＲＩメッセージはまた、ＭＲＩについての確認を受信するために、オフラインのノードへも送信されてもよい。この観点で、オフラインのノードは実ＣＳＭノードといわゆるサブＣＳＭノードとの間のリンクであってもよい。後者は、マスタであると信じているがそうではないノード内で稼動しているＬＳＭである。いずれにせよ、上述のように、オフラインのノードのレプリカはマスタとして設定されえない。従って、この問題を回避するために、ＣＳＭ処理はＭＲＩを送信する前にＤＥＬＡＹ＿ＴＩＭＥを待ってもよい。

【００９７】

一般的に言うと、ＣＳＭ選択メカニズムは、すべての監視プロセスが同期されるようなものである。上述のように、最も古いプロセスがＣＳＭになることがアイデアである。最も古いＬＳＭがどれかを決定する際に、複数の実施形態が予見可能である。第１実施形態では、本発明は動作時間を考慮に入れる。すなわち、各プロセスは、自身のローカル時間を用いて、起動から何秒動作しているかを決定し、この情報を「アライブ」メッセージ内で送信する。この場合に、恐らくはレイテンシが重要な役割を果たすため、以下の欠点が存在するかもしれない。受信側ノードは送信された作業時間を見るだろうが、レイテンシを見ないので、受信側ノードが送信側ノードよりも若いか古いかを正確に判断することは難しいだろう。レイテンシ時間はピングメッセージを介して測定されえ、平均が確立されうるだろう。

【００９８】

第２実施形態では、本発明は起動時刻を考慮に入れる。この実施形態の下で、すべてのプロセスは自身の起動時刻を「アライブ」メッセージ内で送信する。この時刻は各ノードにおけるローカルマシン時刻に関連するだろう。従って、システム内のすべてのマシンを同期する必要があるかもしれない。リナックスシステムおよび一般のオペレーティングシステムはＮＴＰを用いて何年も前にこの問題を解決している。

【００９９】

これらの第１実施形態および第２実施形態の下で、ＣＳＭ選択処理の間に、図６に説明される例示の状態マシンについての非アクティブ状態にＬＳＭが到達するといつでも、個別の動作時間または起動時刻は、劣化したＣＳＭが回復した際に再びＣＳＭ状態に到達することを回避するためにリセットされる必要があるかもしれない。

【０１００】

上述のように、分散データベースシステムの任意のノードの起動または停止の際に、アクティブノードの中で、どのノードが各パーティションについてのマスタレプリカを担当する現在のマスタノードであるかを判定するステップが存在する。このマスタノードがどれであるかを判定する際に、本発明は２つの実施形態を提供する。第１実施形態では、どれがマスタノードであるかを判定するためにすべてのノードは独立して動作する。第２の実施形態では、各パーティションについてどれがマスタノードであるかを決定するためにＣＳＭが決定される。

【０１０１】

図８は、例ではノード３である現在みなされているＣＳＭがダウンして、他のノード、すなわちノード１、２に対して利用不可能になる例示の状況を説明する。

【０１０２】

図８に示されるように、ＣＳＭとして動作するノード３はステップＳ－３００の間に他のノード１、２へ「アライブ」メッセージの最新の集合を送信した。このようなメッセージがそれぞれ受信されると、ノード１、２の両方は、新たな「アライブ」メッセージがノード３から受信されるまでリセットされないいわゆる非活動期間を開始する。ノード３からさらに「アライブ」メッセージを受信することなくノード１、２の両方において非活動期間が満了するため、ノード１、２の両方はそれぞれステップＳ－３２０、Ｓ－３３０の間にノード３を利用不可能として記録する。この例示の場合では、ノード２はノード１よりも古く、この情報はノード２に知られている。なぜなら、ノード２はステップＳ－３１０の間にノード１から最新の定期的な「アライブ」メッセージを受信しているからである。従って、ノード２はステップＳ－３４０の間に、図６に説明される状態マシンに従って、自身をポテンシャルＣＳＭ状態へ移行する。次いで、ノード２はステップＳ－３５０の間に、自身の定期的な「アライブ」メッセージをノード１とノード３とへ送信するが、後者はこのようなメッセージを受信できない。ノード２から「アライブ」メッセージを受信したノード１は、ノード２が古いことを認識し、ステップＳ－３６０の間にノード２が現在のＣＳＭであるという結論にいたる。ノード２においていわゆるＤＥＬＡＹ＿ＴＩＭＥが満了した後に、ノード２の監視部６０はステップＳ－３７０の間にＣＳＭとして確定され、ノード２は現在のシステムマスタモニタである。

【０１０３】

さらに、図には示されていないが、ノード３が再び復旧した場合に、ノード３は自身の定期的な「アライブ」メッセージをノード１、２へ送信する。ノード１、２においてこのようなメッセージを受信すると、これらはともに、ノード３を再び利用可能として記録する。上述のように、ノード３はオプションとして自身の動作時間または起動時刻をリセットし、その結果、状況は変化せず、ＤＥＬＡＹ＿ＴＩＭＥの期間にポテンシャルＣＳＭ状態にあった後に、現在のＣＳＭはなおもノード２の監視部であり、これは図６に示される状態マシンに従ってＣＳＭ状態へ到達する。

【０１０４】

以前に分散データベースシステムに含まれていなかった新たなノードの追加に関して、ＣＳＭ選択の観点から、状況は、ノードがダウンし再び復旧した状況と非常に類似している。唯一の違いは、障害が発生したノードは設定テーブルに提示されているためＣＳＭに知られているが、全く新たなノードは既存のノードに知られていない点である。この点において、ＣＳＭだけでなく他の監視プロセスも、設定されていないノードから受信された「アライブ」メッセージに注意を払わない。従って、新たなノードを導入する重要な側面は、新たなノードの識別子を図１Ｂに説明されるような既存のノードの対応する設定テーブル１５１～１５４に含めることと、このような新たなノード内にレプリカの構成を含めることとである。これが完了すると、新たなノードの監視部は起動しうる。このように、新たなノードは他の既存のアクティブノードへ「アライブ」メッセージを送信し、他の既存のアクティブノードは新たなノードを認識するだろう。ＣＳＭは新たなノード内の新たなレプリカを認識することになるので、これに応じてＣＳＭは必要ならばシステムを再構成する。上記の第１動作モードに従って、すなわちＣＳＭを有さずにシステムが動作する観点で、すべてのノードはそれぞれ、「アライブ」メッセージ内で受信された情報を処理し、各パーティションについてのマスタレプリカを担当するマスタノードがどれであるかに関して同じ結論に到達する。原則として、「アライブ」メッセージはいわゆるＭＲＩメッセージ内に情報を有する。それにもかかわらず、ＭＲＩメッセージは「アライブ」メッセージとは別に送信されてもよい。

【０１０５】

他方で、稼動中の分散データベースシステムへの新たなパーティションの追加は上述の振る舞いを用いて簡単な作業でありうる。１つのノードにただ１つのレプリカを有する場

合よりも高可用性を得るために、それぞれ２つの既存のノードへ追加される少なくとも２つのレプリカ上にパーティションを複製することを考えてもよい。上述のように、任意のノードの構成について、レプリカとは別に、レプリカごとの上述のすべての対応するインジケータが同様に構成されなければならない。よって、第２動作モードでのＣＳＭを有する選択されたシステムマスタモニタノード、または第１動作モードでのすべてのノードは、遅かれ早かれすべてのノードから「アライブ」メッセージを受信する。ここで、新たなレプリカについての情報は当該新たなレプリカのホストとなるノードから受信される。

【０１０６】

高可用性を有する分散データベースシステムを効率的に提供するために、監視プロセスも高可用性を有することが期待される。従って、上述のように、２つのＬＳＭであるアクティブＬＳＭとスタンバイＬＳＭとが各ノードで動作してもよい。動作中に、アクティブＬＳＭはすべての受信メッセージ、特に「アライブ」メッセージと、任意のＭＲＩメッセージが受信されるならこのようなＭＲＩメッセージとをスタンバイＬＳＭへ転送する。このようにして、アクティブＬＳＭがクラッシュした場合に、スタンバイＬＳＭが即座に引継ぎ、その結果、単に監視プロセスに障害が発生したからといってノードはダウンすることがない。

【０１０７】

本発明の目的のために、監視プロセスは場合によっては処理部２０と連動する監視部６０によって実行されてもよい。

【０１０８】

上記とは別に他の個別の実施形態が予見可能である。例えば、設定データとともに格納される代わりに、すべての動的データが動的に管理され、時々刻々と更新されてもよい。この動的データはＭＲＩ情報だけでなく、レプリカおよび通常は「アライブ」メッセージで送信されるノードに関する情報を含む。これは、リクエストに応答するプロセスで用いられるポートとは異なるポートにおけるマルチキャスト技術を用いて容易に実現されうる。

【０１０９】

この点において、他のノードから受信されたすべての「アライブ」メッセージだけでなく、設定されたノードへ送信された「アライブ」メッセージは、マルチキャストにより送信されてもよく、特別な処理は必要ない。同様に、ノードの任意のＬＳＭにより受信された、またはシステムマスタモニタのＣＳＭにより送信されたＭＲＩメッセージはマルチキャストにより受信または送信されてもよい。

【０１１０】

ＭＲＩは２相コミットであるため、以下の特別な検討が必要になる。

・ＬＳＭプロセスであり、ＭＲＩ＿ＡＣＫが送信されていないならば、ＣＳＭが接続を再開しうるため問題はない（いずれにせよ、ちょうど中間に発生する可能性は非常に低い）。マルチキャストによって何も送信されない。

・ＬＳＭプロセスであり、ＭＲＩ＿ＡＣＫが送信されているならば、「確認待機中」を示すフラグを有するＭＲＩをマルチキャストする（ＭＲＩ＿ＮＡＣＫについて、確認されないだろうために何も行われてはならない）。このように、スタンバイプロセスは確認を理解することができてもよい。

・ＬＳＭプロセスであり、ＭＲＩ＿ＣＯＮＦＩＲＭが受信されているならば、「確認済」としてのフラグを有するＭＲＩをマルチキャストする。スタンバイプロセスは直近のものとしてＭＲＩを解釈してもよい。

・ＣＳＭプロセスであり、ＭＲＩがどのノードへも送信されていないならば、何もマルチキャストされてはならない（アクティブになるので、スタンバイプロセスは自身の現在のＭＲＩが精密なものに一致しないことを検出してもよく、プロセスを開始してもよい。）

・ＣＳＭプロセスであり、ＭＲＩが何れかのノードへ送信されているならば、「確認のペンダント」としてそれをマルチキャストする。スタンバイが制御を得るならば、すべて

のＬＭとの接続を再開してもよく（すでに開いている接続はタイムアウトしてもよく、リモートＳＭプロセスは最初のＭＲＩ表示を無視してもよい）、ＭＲＩを再送する。

‐ＣＳＭプロセスであり、ＭＲＩ＿ＣＯＮＦＩＲＭが何れかのノードへ送信されているならば、「確認中」というフラグを有するＭＲＩをマルチキャストする。スタンバイが制御を得るならば、再び確認するためにすべてのＬＳＭプロセスとの接続を再開してもよい（すでに確認されたものはこのメッセージを無視してもよい）。

‐ＣＳＭプロセスであり、ＭＲＩ＿ＣＯＮＦＩＲＭがすべてのノードへ送信されているならば、「確認済」としてＭＲＩをマルチキャストし、その結果、スタンバイはこれを新たなものとして解釈してもよいが、確認手順を開始しなくてもよい。

【０１１１】

上述のように、ＭＲＩメッセージは、構文解析のための共通コードを用いるために、末端において「確認」状態についての追加のフラグをマルチキャストするために必要であってもよい。いずれにせよ、すべてのマルチキャストメッセージは、アクティブとスタンバイとの間の競合状態を解決するために、プロセスの重みを表す１バイトを最後の部分に含んでもよいだろう。

【０１１２】

このセクションは、アクティブプロセスとスタンバイプロセスとがどのようにそれらの状態を検出しえるか、そしてこれらが何を行うことが想定されるかを説明する。プロセスが起動するときには常に、分散データベースシステム内の任意のマルチメディアメッセージを受信するためにリッスンを開始してもよい。何らかのマルチキャストが受信されると、スタンバイとして設定されてもよいし、受信した状態に従って何れかの内部データを更新してもよい（自身の選択された状態がアライブメッセージから受信されるので、状態マシンを処理しなくてもよい）。プロセスが起動後または以前のマルチキャスト受信パケットの後、ＤＥＬＡＹ＿ＴＩＭＥ期間内に何もマルチキャストメッセージを受信しないならば、「アライブ」メッセージまたはＭＲＩメッセージについてリッスン（ポートのオープン）を開始してもよく、処理重み（これはノード内の任意のプロセスについて異なるだろう）とともに（最初の始動時に空であるだろう）自身の現在のＭＲＩを送信してもよい。アクティブプロセスが自身のものよりも大きな重みを有するマルチキャストメッセージと空でないＭＲＩとを受信したならば、以前のポートでのリッスンを停止してもよく、スタンバイへ低下してもよい（これは最初の競合状態を解消してもよく、これは通常動作の間に発生しないと想定される）。空のＭＲＩを比較することの目的は、非常に速く再開される場合に制御を得るために、より大きな重みを有するプロセスを回避することである（いずれにせよ、再起動の後にＤＥＬＡＹ＿ＴＩＭＥを待つことができ、それによってこれは、現在のアクティブプロセスが自身の「アライブ」メッセージを送信するのと同時、且つ隣接ノードも自身のマルチキャストされる「アライブ」メッセージを送信するのと同時に、それが起動しない限りこれは発生し得ないだろう）。

【０１１３】

以下の表は本発明の実施形態に従う上記の分散データベースシステムのノード間で用いられるプロトコルの取りうる実装を説明する。

【０１１４】

位置(バイト)	長さ(バイト)	名称	説明
0	1	Msgタイプ	0x01はアライブメッセージである。
1	1	バージョン	
2	2	ノードId	ノードId
4	4	実行時間	現在のノードの実行時間
8	2	m	ローカルで監視されるレプリカ数。利用可能なレプリカだけが送信される。
10	2	レプリカマスタ状態	送信されるレプリカがノード内のマスタとしてみなされるか否かを通知する2バイト論理和。最初のノードが最初のビット(2⁰)でありバイトはビッグインディアンであるとみなされる。 例:0x5(0..0101)は送信される1番目のノードと3番目のノードとがマスタであることを意味する。
12	1	ノード状態	ノード状態:イベントマシンに従って非アクティブ、アクティブ、ポテンシャルCSM、CSM(1,2,3,4)。
P=13	2	レプリカid	送信される最初のレプリカのid。
P+1	1	メモリ使用率&D状態	レプリカに使用されるメモリの割合
P+3	2	レプリカidグループ	第2レプリカのid
P+5	1	メモリ使用率	第2レプリカのメモリ使用率
…			残りの利用可能なレプリカ
P+3*(m-1)	2	レプリカid	最後のレプリカのid
P+3*(m-1)+2	1	メモリ&St	最後のDについての割合
P+3*m	2	ノードn	利用可能なノード数
Q=P+3*m+2	2	ノードid	最初の利用可能なノードのid
Q+2	2	ノードid	2番目の利用可能なノードのid
…			
Q+2*(n-1)	2	ノードid	最後の利用可能なノードのid
Q+2*n	8+5*D	MRI	マスタレプリカ情報。 Dはこのフィールド内のレプリカ数を表す。

【０１１５】

本発明は、入出力部だけでなく処理部を有するコンピュータの内部メモリにロード可能なコンピュータプログラムで実施されてもよい。このコンピュータプログラムはこの目的のためにコンピュータで実行される場合に上述の方法のステップを実行するように適応された実行可能なコードを有する。特に、実行可能なノードはコンピュータにおいて読み取り可能な媒体手段に記録されてもよい。

【０１１６】

本発明は例示であり非限定的であることが意図される様々な実施形態に関連して上述された。当業者がこれらの実施形態を変形してもよいことが期待される。本発明の範囲は明細書と図面に連動して特許請求の範囲により規定され、特許請求の範囲に含まれるすべての変形が本明細書に含まれることが意図される。

【図１Ａ】

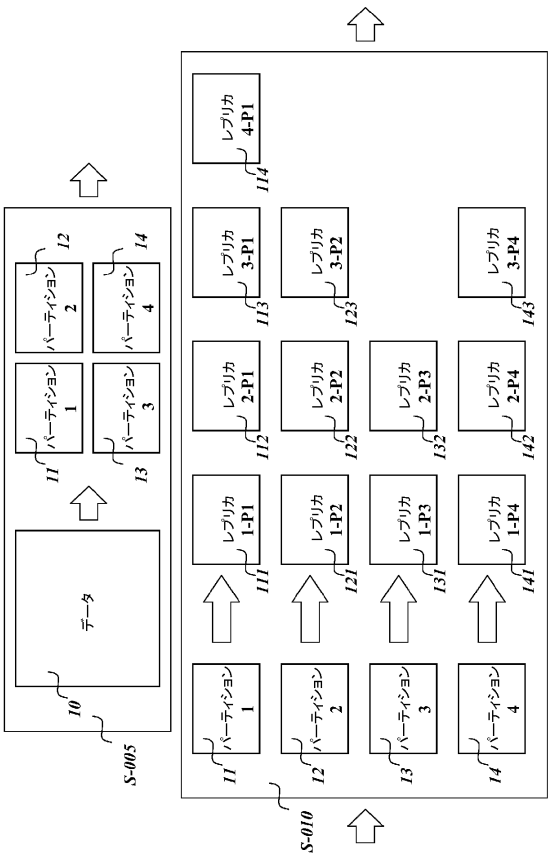

FIG.-1A-

【図１Ｂ】

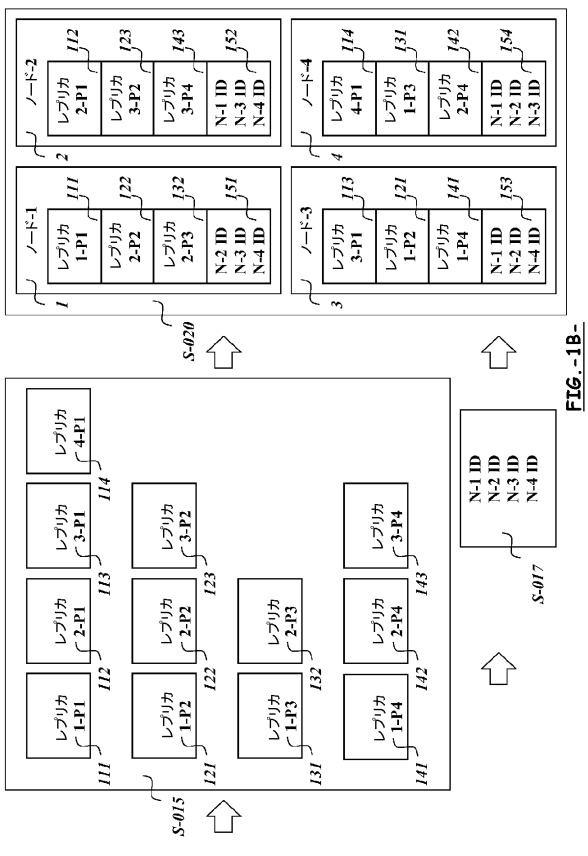

FIG.-1B-

【図１Ｃ】

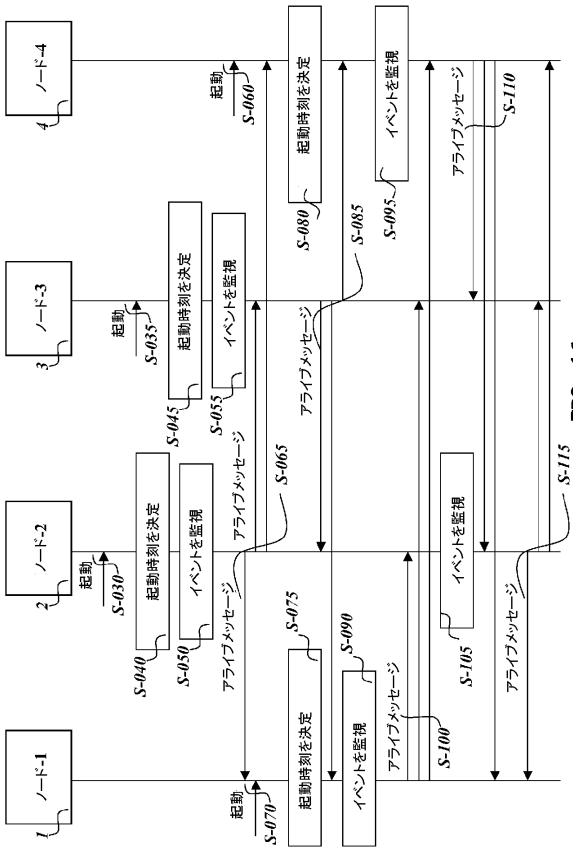

FIG.-1C-

【図２】

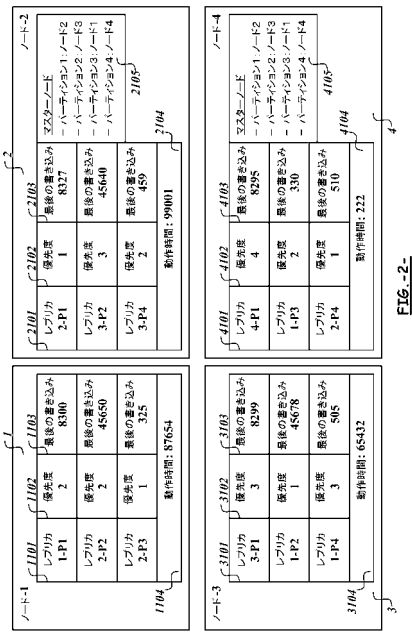

FIG.-2-

【図３】

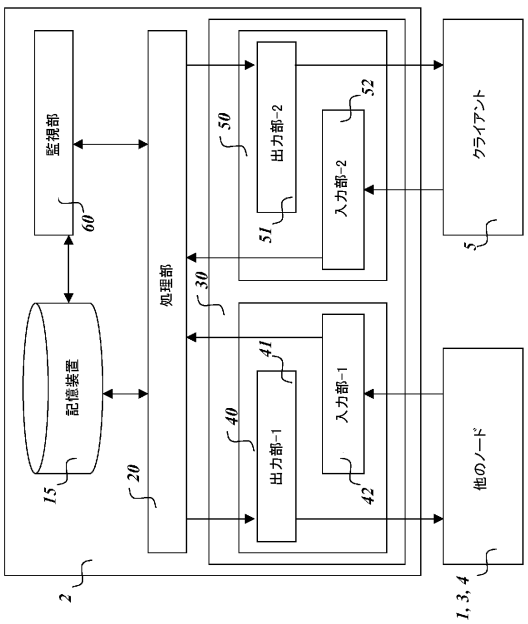

FIG.-3-

【図４】

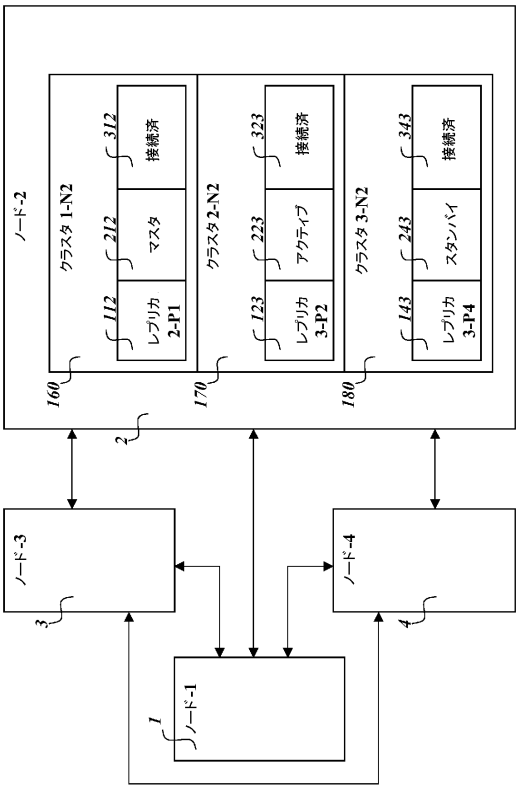

FIG.-4-

【図５】

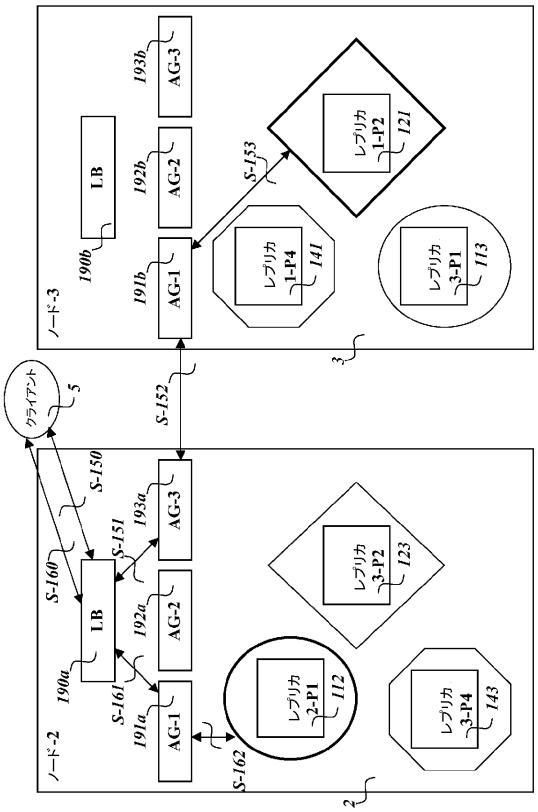

FIG.-5-

【図６】

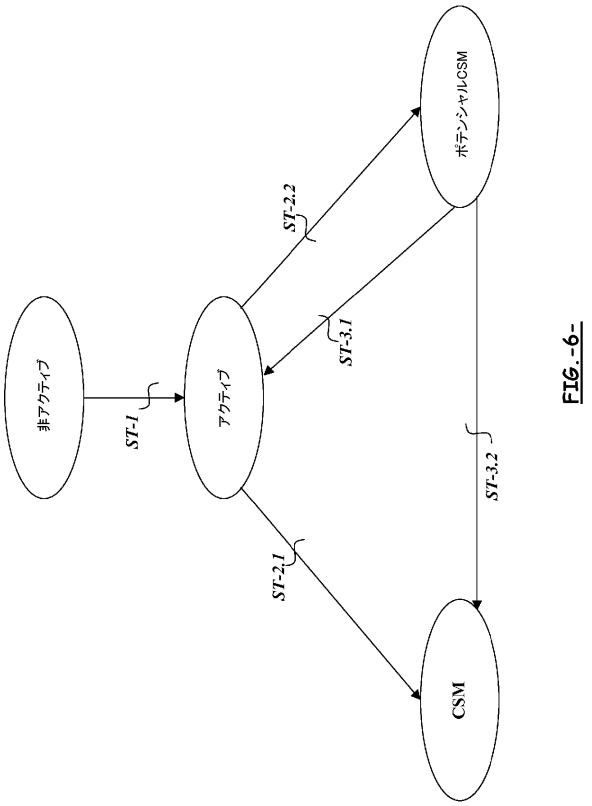

FIG.-6-

【図７】

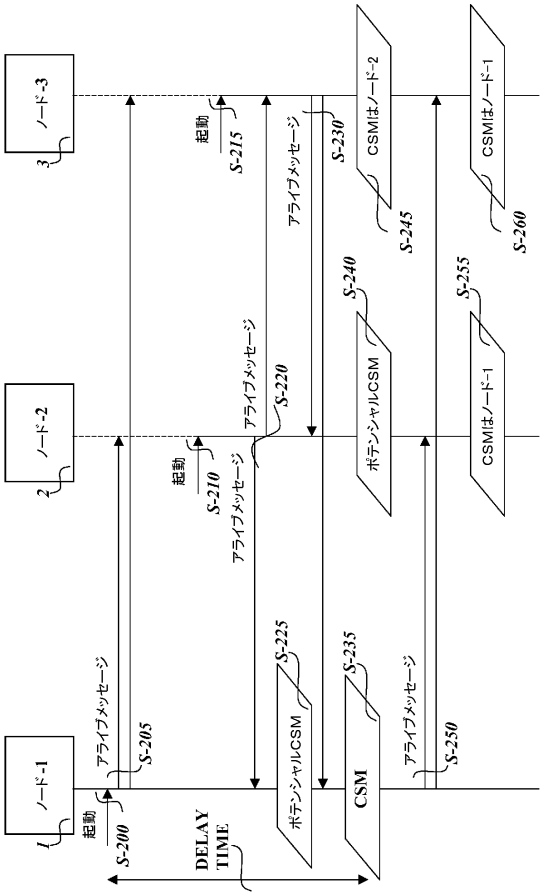

FIG.-7-

【図８】

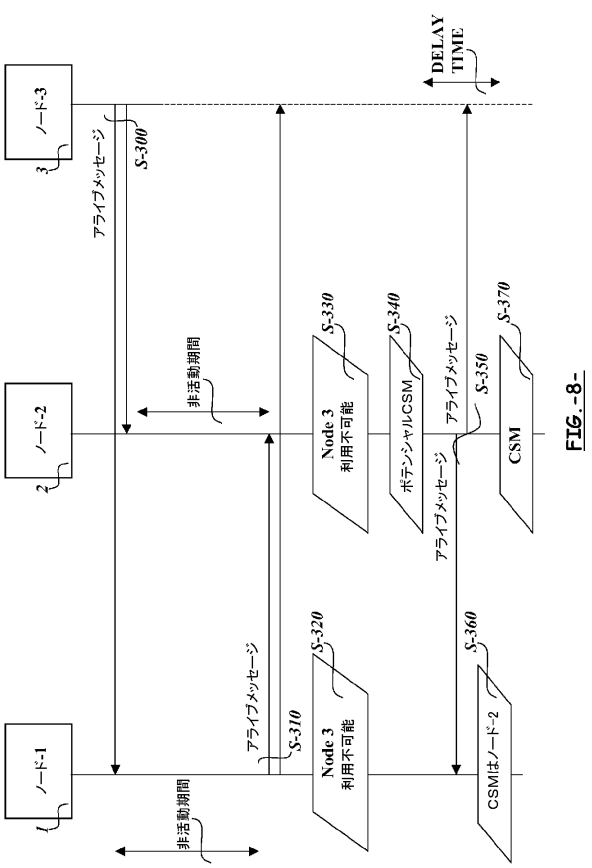

FIG.-8-

フロントページの続き

(74)代理人 100161399
弁理士 大戸 隆広
(72)発明者 ヘンリクセン, デニス
デンマーク国 コペンハーゲン ディーケー－２２００, ブロフスゲーテ ２０, ２．ティーエイチ
(72)発明者 ネヴァド ヒメネス, ホルヘ
スペイン国 マドリード イー－２８４３６, アルペデレテ, プラド デ ラ イグレシア ２, ビージェイ４
(72)発明者 サン マルティン アリバス, マルタ
スペイン国 マドリード イー－２８００７, カヴァニリェス ６０

審査官 加内 慎也

(56)参考文献 米国特許第０６５３９３８１（ＵＳ，Ｂ１）
米国特許第０７７８８２３３（ＵＳ，Ｂ１）
Ruben de Juan-Marin ET AL, Revisiting Hot Passive Replication, Second International Conference on Availability, Reliability and Security (ARES'07), ２００７年 ４月 １日

(58)調査した分野(Int.Cl., ＤＢ名)
Ｇ０６Ｆ １２／００
Ｇ０６Ｆ １３／００