

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6545709号  
(P6545709)

(45) 発行日 令和1年7月17日 (2019.7.17)

(24) 登録日 令和1年6月28日 (2019.6.28)

(51) Int. Cl.

F I

G O 6 F 16/2458 (2019.01)

G O 6 F 16/2458

G O 6 F 16/182 (2019.01)

G O 6 F 16/182

請求項の数 11 (全 32 頁)

(21) 出願番号	特願2016-565068 (P2016-565068)	(73) 特許権者	502303739
(86) (22) 出願日	平成27年3月27日 (2015.3.27)		オラクル・インターナショナル・コーポレイション
(65) 公表番号	特表2017-514239 (P2017-514239A)		アメリカ合衆国カリフォルニア州94065レッドウッド・シティー、オラクル・パークウェイ500
(43) 公表日	平成29年6月1日 (2017.6.1)		
(86) 国際出願番号	PCT/US2015/023120	(74) 代理人	110001195
(87) 国際公開番号	W02015/167724		特許業務法人深見特許事務所
(87) 国際公開日	平成27年11月5日 (2015.11.5)	(72) 発明者	シェン, シュガン
審査請求日	平成30年1月24日 (2018.1.24)		中華人民共和国、100193 ペイジン、ハイディアン・ディストリクト、チョングアンツン・ソフトウェア・パーク、ビルディング・ナンバー・24、オラクル・ビルディング
(31) 優先権主張番号	61/985, 135		
(32) 優先日	平成26年4月28日 (2014.4.28)		
(33) 優先権主張国	米国 (US)		
(31) 優先権主張番号	14/587, 468		
(32) 優先日	平成26年12月31日 (2014.12.31)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 トランザクション環境におけるリソースマネージャ (RM) インスタンス検知に基づいた共通のトランザクション識別子 (X I D) 最適化およびトランザクションアフィニティをサポートする

(57) 【特許請求の範囲】

【請求項 1】

トランザクションシステムにおいてトランザクション処理をサポートするための方法であって、前記トランザクションシステムは、

複数のトランザクショングループを含み、前記複数のトランザクショングループは、

複数のトランザクションアプリケーションサーバと、

前記複数のトランザクションアプリケーションサーバ上のグローバルトランザクションの処理を管理することに関与する複数のトランザクションマネージャとを含み、同じトランザクショングループの複数のトランザクションアプリケーションサーバは1つのトランザクションブランチを共有し、前記トランザクションシステムはさらに、

データベースに接続する複数のリソースマネージャインスタンスを含み、前記方法は、

コーディネータトランザクションマネージャを構成する第1のトランザクショングループのパーティシパントトランザクションマネージャが、共通のトランザクション識別子およびリソースマネージャインスタンスについての情報を、1つ以上の他のトランザクショングループに含まれる1つ以上の他のパーティシパントトランザクションマネージャに対して伝播させるステップと、

前記リソースマネージャインスタンスを前記コーディネータトランザクションマネージャと共有している前記他のパーティシパントトランザクションマネージャのうち1つ以上が、前記共通のトランザクション識別子を用いることを可能にするステップと、

前記コーディネータトランザクションマネージャの1つのトランザクションブランチを

10

20

用いて、かつ前記リソースマネージャインスタンスを前記コーディネータトランザクションマネージャと共有している前記他のパーティシパントトランザクションマネージャの前記トランザクションブランチを無視して、前記コーディネータトランザクションマネージャおよび前記共有されているリソースマネージャインスタンスによって、前記グローバルトランザクションを処理するステップとを含む、方法。

【請求項 2】

前記コーディネータトランザクションマネージャに関連付けられたトランザクション識別子を前記共通のトランザクション識別子として用いるステップをさらに含む、請求項 1 に記載の方法。

【請求項 3】

前記リソースマネージャインスタンスをクラスタ化されたデータベースに関連付けることを可能にするステップをさらに含む、請求項 1 または 2 に記載の方法。

【請求項 4】

前記他のパーティシパントトランザクションマネージャのうち前記 1 つ以上のために前記コーディネータトランザクションマネージャが前記グローバルトランザクションをコミットまたはロールバックするステップをさらに含む、請求項 1 ~ 3 のいずれか 1 項に記載の方法。

【請求項 5】

前記リソースマネージャインスタンスを共有しない別のパーティシパントトランザクションマネージャのために、別のトランザクションブランチを用いて前記グローバルトランザクションを処理するステップをさらに含む、請求項 1 ~ 4 のいずれか 1 項に記載の方法。

【請求項 6】

前記グローバルトランザクションに含まれるすべてのパーティシパントが前記リソースマネージャインスタンスを共有する場合、前記リソースマネージャインスタンス上に、直接、1 相コミット動作を呼出すステップをさらに含む、請求項 1 ~ 5 のいずれか 1 項に記載の方法。

【請求項 7】

リモートトランザクションリソースを解放するためにリモートノード上のパーティシパントに対して削除要求を送信するステップをさらに含む、請求項 1 ~ 6 のいずれか 1 項に記載の方法。

【請求項 8】

さまざまなドメインにおける複数のパーティシパントが前記グローバルトランザクションに含まれることを可能にするステップをさらに含む、請求項 1 ~ 7 のいずれか 1 項に記載の方法。

【請求項 9】

前記共通のトランザクション識別子および前記リソースマネージャインスタンスについての前記情報を、リモートドメインへのゲートウェイを介して受信するステップをさらに含む、請求項 1 ~ 8 のいずれか 1 項に記載の方法。

【請求項 10】

トランザクション環境におけるトランザクション処理をサポートするためのシステムであって、

複数のトランザクショングループを含み、前記複数のトランザクショングループは、各々が 1 つ以上のマイクロプロセッサを含む複数のトランザクションアプリケーションサーバと、

前記 1 つ以上のマイクロプロセッサ上で実行される複数のトランザクションマネージャとを含み、前記複数のトランザクションマネージャは前記複数のトランザクションアプリケーションサーバ上のグローバルトランザクションの処理を管理することに関与し、同じトランザクショングループの複数のトランザクションアプリケーションサーバは 1 つのトランザクションブランチを共有し、前記システムはさらに、

10

20

30

40

50

データベースに接続する複数のリソースマネージャインスタンスを含み、前記複数のリソースマネージャインスタンスは前記1つ以上のマイクロプロセッサ上で実行され、

コーディネータトランザクションマネージャを構成する第1のトランザクショングループのパーティシパントトランザクションマネージャは、

共通のトランザクション識別子およびリソースマネージャインスタンスについての情報を、1つ以上の他のトランザクショングループに含まれる1つ以上の他のパーティシパントトランザクションマネージャに伝播させるステップと、

前記リソースマネージャインスタンスを前記コーディネータトランザクションマネージャと共有している前記他のパーティシパントトランザクションマネージャのうちの1つ以上が、前記共通のトランザクション識別子を用いることを可能にするステップと、

前記コーディネータトランザクションマネージャの1つのトランザクションブランチを用いて、かつ前記リソースマネージャインスタンスを前記コーディネータトランザクションマネージャと共有している前記他のパーティシパントトランザクションマネージャの前記トランザクションブランチを無視して、前記コーディネータトランザクションマネージャおよび前記共有されているリソースマネージャインスタンスによって、前記グローバルトランザクションを処理するステップとを、実行するように動作する、システム。

【請求項11】

プログラム命令を含むコンピュータプログラムであって、前記プログラム命令は、システムによって実行されると、請求項1～9のいずれか1項に記載の方法を前記システムに実行させる、コンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

著作権表示：

この特許文献の開示の一部は、著作権保護の対象となる題材を含んでいる。著作権の所有者は、特許商標庁の包袋または記録に掲載されるように特許文献または特許情報開示を誰でも複製できることに對して異議はないが、その他の点ではすべての如何なる著作権をも保有する。

【0002】

発明の分野：

本発明は、一般的に、コンピュータシステムおよびソフトウェアに関し、特に、トランザクションシステムに関する。

【背景技術】

【0003】

背景：

トランザクションミドルウェアシステムまたはトランザクション指向型ミドルウェアは、組織内のさまざまなトランザクションを処理することができるエンタープライズアプリケーションサーバを含む。高性能ネットワークおよびマルチプロセッサコンピュータなどの新技術の開発によって、トランザクションミドルウェアの性能をさらに改善する必要がある。これらは、本発明の実施形態が対処しようとする一般的領域である。

【発明の概要】

【課題を解決するための手段】

【0004】

概要：

トランザクション環境においてトランザクション処理をサポートすることができるシステムおよび方法がこの明細書中に記載される。グローバルトランザクションのためのコーディネータは、トランザクション環境におけるグローバルトランザクションの1つ以上のパーティシパントに、リソースマネージャインスタンスについての共通のトランザクション識別子および情報を伝播(propagate)させるように動作する。コーディネータは、リソースマネージャインスタンスをコーディネータと共有している上記1つ以上のパーティ

10

20

30

40

50

シパントが共通のトランザクション識別子を用いることを可能にする。さらに、コーディネータは、1つのトランザクションブランチを用いて、リソースマネージャインスタンスを共有する上記1つ以上のパーティシパントのためにグローバルトランザクションを処理することができる。

#### 【0005】

トランザクション環境においてトランザクション処理をサポートすることができるシステムおよび方法がこの明細書中に記載される。トランザクションシステムは、リソースマネージャ(resource manager: RM)インスタンスに接続されているトランザクションサーバに要求をルーティングするように動作する。さらに、トランザクションシステムは、トランザクションサーバが関連付けられているRMインスタンスを示しているアフィニティコンテキスト(affinity context)をトランザクションサーバに割り当てることができる。さらに、トランザクションシステムは、アフィニティコンテキストに基づいて、要求に関連する1つ以上の後続の要求をトランザクションサーバにルーティングすることができる。

10

#### 【0006】

トランザクション環境においてトランザクション処理をサポートすることができるシステムおよび方法がこの明細書中に記載される。トランザクションサーバは、1つ以上のRMインスタンスに関連付けられているデータソースからリソースマネージャ(RM)インスタンス情報を受信するように動作する。この場合、受信されたインスタンス情報は、トランザクションサーバが現在接続されているのがどのRMインスタンスであるかを当該トランザクションサーバが検知することを可能にする。さらに、トランザクションサーバは、受信されたインスタンス情報を、トランザクションサーバに関連付けられた1つ以上のテーブルに保存するように動作する。次いで、トランザクションサーバは、1つ以上のテーブルに保存されたインスタンス情報に基づいてグローバルトランザクションを処理することができる。

20

#### 【図面の簡単な説明】

#### 【0007】

【図1】本発明の一実施形態に従った、トランザクション環境においてリソースマネージャ(RM)インスタンス検知をサポートする例を示す図である。

【図2】本発明の一実施形態に従った、トランザクション環境においてさまざまな状態テーブルを維持する例を示す図である。

30

【図3】本発明の一実施形態に従った、トランザクション環境においてサーバテーブル(server table: ST)をサポートする例を示す図である。

【図4】本発明の一実施形態に従った、トランザクション環境においてインスタンス情報を更新する例を示す図である。

【図5】本発明の一実施形態に従った、トランザクション環境においてさまざまなチェックポイントを用いてトランザクションプロセスをサポートする例を示す図である。

【図6】本発明の一実施形態に従った、トランザクション環境においてリソースマネージャ(RM)インスタンス検知をサポートするための例示的なフローチャートである。

【図7】本発明の一実施形態に従った、トランザクション環境におけるインスタンス検知に基づいてトランザクションアフィニティをサポートする例を示す図である。

40

【図8】本発明の一実施形態に従った、トランザクション環境におけるトランザクションアフィニティルーティングをサポートする例を示す図である。

【図9】本発明の一実施形態に従った、トランザクション環境においてアフィニティコンテキストを含むメッセージを送信する例を示す図である。

【図10】本発明の一実施形態に従った、トランザクション環境においてアフィニティコンテキストを含むメッセージを受信する例を示す図である。

【図11】本発明の一実施形態に従った、トランザクション環境においてクライアントコンテキスト内におけるアフィニティルーティングをサポートする例を示す図である。

【図12】本発明の一実施形態に従った、トランザクション環境においてさまざまなドメ

50

インにわたってアフィニティコンテキストを伝播させる例を示す図である。

【図 1 3】本発明の一実施形態に従った、トランザクション環境においてアプリケーションサーバにアフィニティコンテキストを伝播させる例を示す図である。

【図 1 4】本発明の一実施形態に従った、トランザクション環境においてインスタンス検知に基づいてトランザクションアフィニティをサポートするための例示的なフローチャートである。

【図 1 5】本発明の一実施形態に従った、さまざまなトランザクション識別子 (transaction identifier: X I D) を用いて、トランザクション環境においてグローバルトランザクション処理する例を示す図である。

【図 1 6】本発明の一実施形態に従った、共通のトランザクション識別子 (X I D) を用いてトランザクション環境においてグローバルトランザクション処理する例を示す図である。

【図 1 7】本発明の一実施形態に従った、トランザクション環境においてデータベースインスタンス検知に基づいて 1 相コミット (one-phase commit: 1 P C) 処理モデルをサポートする例を示す図である。

【図 1 8】本発明の一実施形態に従った、トランザクション環境においてデータベースインスタンス検知に基づいてグローバルトランザクション処理する例を示す図である。

【図 1 9】本発明の一実施形態に従った、共通の X I D を用いてトランザクション環境における複数のドメインにわたってグローバルトランザクション処理する例を示す図である。

【図 2 0】本発明の一実施形態に従った、共通の X I D を用いてトランザクション環境においてグローバルトランザクション処理するための例示的なフローチャートである。

【発明を実施するための形態】

【0008】

詳細な説明：

本発明は、同様の参照符号で同様の要素を示している添付の図面において、限定によってではなく例示によって説明されている。この開示における「ある」または「1つの」または「いくつかの」実施形態への言及は、必ずしも同じ実施形態に対するものではなく、そのような言及は「少なくとも1つ」という意味である。

【0009】

以下の本発明の説明は、トランザクションミドルウェアマシン環境についての一例として Tuxedo 環境を用いている。他のタイプのトランザクションミドルウェアマシン環境が限定なしに使用可能であることが当業者にとって明らかになるだろう。

【0010】

トランザクションミドルウェアマシン環境などのトランザクション環境においてトランザクション処理をサポートすることができるシステムおよび方法がこの明細書中に記載される。

【0011】

トランザクションミドルウェアマシン環境

本発明の一実施形態によれば、迅速に用意することができ且つオンデマンドで拡張することができる大規模並列処理インメモリグリッドを含む J a v a (登録商標) E E アプリケーションサーバ複合体を提供するために、当該システムは、64ビットプロセッサ技術などの高性能ハードウェア、高性能大容量のメモリ、冗長インフィニバンド (InfiniBand) およびイーサネット (登録商標) ネットワーク、ならびに W e b L o g i c (登録商標) スイートなどのアプリケーションサーバまたはミドルウェア環境の組合せを含む。一実施形態によれば、当該システムは、アプリケーションサーバグリッド、ストレージエリアネットワーク、およびインフィニバンド (I B) ネットワークを提供するフルラック、ハーフラックまたはクォーターラックもしくは他の構成として展開可能である。ミドルウェアマシンソフトウェアは、アプリケーションサーバ、ミドルウェアおよび他の機能、たとえば、W e b L o g i c S e r v e r、J R o c k i t もしくは H o t s p o t J V

10

20

30

40

50

M、Oracle Linux（登録商標）もしくはSolaris、およびOracle（登録商標）VMなどを提供することができる。一実施形態によれば、当該システムは、IBネットワークを介して互いに通信する複数の計算ノード、IBスイッチゲートウェイ、およびストレージノードまたはストレージユニットを含むことができる。ラック構成として実装される場合、ラックの未使用部分は、空のままにしてもよく、充填物で充填されてもよい。

#### 【0012】

本発明の一実施形態によれば、当該システムは、OracleミドルウェアSWスイートまたはWebLogicなどのミドルウェアまたはアプリケーションサーバソフトウェアをホストするために、展開容易な解決案を提供する。本明細書に説明されるように、一実施形態に従ったシステムは、ミドルウェアアプリケーションをホストするために必要とされる1つ以上のサーバ、ストレージユニット、ストレージネットワーク用のIBファブリック、およびすべての他の要素を含む「グリッド・イン・ア・ボックス」(grid in a box)である。たとえば、リアルアプリケーションクラスタおよびExalogicオープンストレージなどを使用する大規模並列グリッドアーキテクチャを活用することによって、すべての種類のミドルウェアアプリケーションに高い性能を提供することができる。このシステムは、線形I/O拡張性によって改善した性能を提供し、使用および管理が簡単であり、ミッションクリティカルな可用性および信頼性を提供する。

#### 【0013】

本発明の一実施形態に従うと、Oracle Tuxedoシステムなどのトランザクションミドルウェアシステムは、Oracle Exalogicミドルウェアマシンなど複数のプロセッサや、IBネットワークなどの高性能ネットワーク接続を備えた高速マシンを利用することができる。付加的には、Oracle Tuxedoシステム（単に「Tuxedo」とも称する）は、キャッシュアーキテクチャが共有されているクラスタ化されたデータベースであるOracleのリアルアプリケーションクラスタ（Real Application Clusters: RAC）エンタープライズデータベースなどのクラスタ化されたデータベースを利用することができ、かつ、クラウドアーキテクチャのコンポーネントにもなり得る。Oracle RACは、従来のシェアード・ナッシング・アプローチおよびシェアード・ディスク・アプローチの制限事項を克服して、ビジネス用途でできるように高度に拡張可能かつ利用可能なデータベースソリューションを提供することができる。

#### 【0014】

本発明の一実施形態によれば、Oracle Tuxedoシステムは、高性能分散型ビジネスアプリケーションの構築、実行および管理を可能にするソフトウェアモジュールのセットであり、トランザクションミドルウェアとして多くの複層アプリケーション開発ツールに使用されてきた。Tuxedoは、分散コンピューティング環境において分散トランザクション処理を管理するために使用することができるミドルウェアプラットフォームである。Tuxedoは、無制限の拡張性および標準ベースの相互運用性を提供しながら、エンタープライズレガシーアプリケーションのロックを解除し、エンタープライズレガシーアプリケーションをサービス指向アーキテクチャに拡張するための有効なプラットフォームである。

#### 【0015】

付加的には、Oracle Tuxedoシステムは、言語国際化のための、2相コミット（two-phase commit: 2PC）処理のためのXA基準と、X/Open ATMI APIと、X/Open Portability Guide（XPG）基準とについてのサポートを含む、オーブングループのX/Open基準に準拠し得る。トランザクションアプリケーションサーバは、XA基準を用いる場合にはXAサーバと称されてもよい。たとえば、Tuxedoグループに属する各々のTuxedoアプリケーションサーバは、OPENINFO特性を用いて構成することができる。TuxedoグループにおけるXAサーバはすべて、OPENINFO特性を用いてリソースマネージャ（resource manager: RM）への接続を確立することができる。

## 【0016】

## インスタンス検知 (Instance Awareness)

図1は、本発明の一実施形態に従った、トランザクション環境においてリソースマネージャ (RM) インスタンス検知をサポートする例を示す。図1に示されるように、トランザクション環境100におけるトランザクションシステム101は、データベースなどのデータソース102に関連付けられた1つ以上のリソースマネージャ (RM) インスタンスを用いて、トランザクション処理をサポートすることができる。

## 【0017】

本発明の一実施形態に従うと、トランザクションシステム101は、データソース102におけるRMインスタンス情報104を検知することができる。たとえば、トランザクションシステム101におけるトランザクションサーバ103は、ユーザコールバック110を利用することによってデータソース102からRMインスタンス情報104を取得することができる。トランザクションシステム101は、ユーザコールバック110を登録するためのさまざまなメカニズム、たとえば静的登録メカニズムおよび動的登録メカニズム、を用いることができる。

## 【0018】

静的登録メカニズムは、XAサーバをサポートする場合に用いることができる。XAサーバは、一様なXAインターフェイスを用いてトランザクションを制御するサーバである。たとえば、Tuxedoにおいては、静的な登録は、xa\_\_open()関数が呼出された後に、tpopen()関数で呼出すことができる。登録が成功すると、トランザクションサーバ103がデータベース (たとえばOracleデータベース) への接続を確立したとき、ユーザコールバック110が呼出し可能となる。付加的には、ユーザコールバック110は、xa\_\_close()関数が呼出される前に、tpclose()関数で登録解除することができる。

## 【0019】

代替的には、トランザクションサーバ103は、たとえば、データソース102に関連付けられた共有ライブラリ105に基づいてユーザコールバック110を動的に登録することができる。たとえば、Tuxedoは、ユーザが非XAサーバを用いて (たとえばOCIまたはPro\*c/c++によって) Oracleデータベースに接続したときに、動的にコールバックを登録することができる。Tuxedoは、最初に、Oracle OCIライブラリOCI APIを動的にロードして、関連するOCI環境ハンドルを取得することができる。次いで、Tuxedoは、OCISessionBegin関数で、OCIUserCallbackRegisterによってユーザコールバックを登録することができる。

## 【0020】

図1に示されるように、システムは、トランザクションサーバ103に関連付けられた関連するコンテキスト106に、取得したインスタンス情報104を保存することができる。加えて、トランザクションサーバ103は、インスタンス情報104を共有メモリ107におけるさまざまな状態テーブル108 (たとえば、Tuxedoにおけるグローバルな掲示板 (bulletin board: BB) に記憶することができる。これらのテーブル108は、さまざまなノードに同期させることができ、複数のトランザクションサーバ (たとえば、サーバ111-112) および/またはネイティブ (native) クライアントによってアクセス可能である。

## 【0021】

図2は、本発明の一実施形態に従った、トランザクション環境においてさまざまな状態テーブルを維持する例を示す。図2に示されるように、トランザクションシステム200はインスタンス情報220を共有メモリ201におけるさまざまな状態テーブルに記憶することができる。

## 【0022】

これらの状態テーブルは、固有のRM/データベース名を記憶するリソースマネージャ

10

20

30

40

50

テーブル 2 1 1、R M / データベースインスタンス名を記憶するインスタンステーブル 2 1 2、および R M / データベースサービス名を記憶するサービステーブル 2 1 3 を含み得る。このような情報により、他のサーバが特定のサーバに関連付けられたインスタンス情報を検知するのが容易になり得る。

【 0 0 2 3 】

加えて、トランザクションシステム 2 0 0 は共有メモリ 2 0 1 におけるサーバテーブル ( S T ) 2 1 4 を維持することができる。S T 2 1 4 は、1 つ以上のサーバテーブルエントリ ( server table entry : S T E ) を含んでいてもよく、その各々はインスタンステーブル 2 1 2 にインデックスを記憶することができる。たとえば、各々の S T E は、サーバがシングルスレッドサーバである場合にインスタンス識別子 ( instance identifier : I D ) を記憶することができる。

10

【 0 0 2 4 】

図 2 に示されるように、サーバテーブル 2 1 4 は、共有メモリ 2 0 1 内の他のテーブル 2 1 1 - 2 1 3 を指し示すことができる。このため、トランザクションシステム 2 0 0 は、インスタンス情報 ( 特定のサーバが現在接続されている R M インスタンスについての情報など ) を取得するためにサーバテーブル 2 1 4 を用いることができ、トランザクションシステム 2 0 0 は、さまざまな状態テーブル 2 1 1 - 2 1 3 に直接記憶されているインスタンス情報を用いない可能性もある。

【 0 0 2 5 】

図 3 は、本発明の一実施形態に従った、トランザクション環境においてサーバテーブル ( S T ) をサポートする例を示す。図 3 に示されるように、トランザクションシステム 3 0 0 におけるサーバテーブル ( S T ) 3 1 0 は、1 つ以上のサーバテーブルエントリ ( S T E ) 3 1 1 - 3 1 2 を含み得る。サーバテーブルエントリ ( S T E ) 3 1 1 - 3 1 2 の各々は 1 つ以上のインスタンス識別子 ( I D ) を含み得る。たとえば、S T E 3 1 1 はインスタンス I D 3 2 1 - 3 2 2 を含み得る。S T E 3 1 2 はインスタンス I D 3 2 3 - 3 2 4 を含み得る。

20

【 0 0 2 6 】

本発明の一実施形態に従うと、各々のインスタンス I D 3 2 1 - 3 2 4 はさまざまなインスタンス情報を含み得る。図 3 に示されるように、インスタンス I D 3 2 2 はインスタンス名 3 0 1、データベース名 3 0 2 およびサービス名 3 0 3 を識別することができる。

30

【 0 0 2 7 】

たとえば、インスタンス I D は、整数 (たとえば、ビット 0 - 1 1、ビット 1 2 - 1 9 およびビット 2 0 - 3 1 という 3 つのセクションを含む 3 2 ビット整数) を用いて実現することができる。第 1 のセクションであるビット 0 - 1 1 は、R M / データベースインスタンス名 3 0 1 についてのエントリインデックスを記憶することができる。第 2 のセクションであるビット 1 2 - 1 9 は、R M / データベース名 3 0 2 についてのエントリインデックスを記憶することができる。第 3 のセクションであるビット 2 0 - 3 1 は、R M / データベースサービス名 3 0 3 についてのエントリインデックスを記憶することができる。加えて、無効なインスタンス I D を示すために特別の値 0 x F F F F F F F F を用いることができる。

40

【 0 0 2 8 】

本発明の一実施形態に従うと、トランザクションシステム 3 0 0 は、関連するビット同士を単純に比較することによってインスタンス I D 3 2 2 におけるインスタンス情報をチェックすることができる。これにより、システムは、ストリング比較動作の方がビット比較動作よりも費用がかかるせいで生じる、ストリング比較による性能の問題を回避することができる。

【 0 0 2 9 】

図 4 は、本発明の一実施形態に従った、トランザクション環境においてインスタンス情報を更新する例を示す。図 4 に示されるように、トランザクション環境 4 0 0 におけるトランザクションシステム 4 0 1 は、たとえばデータベースにおけるリソースマネージャ (

50



RM) 402 から、関連するインスタンス情報 404 を取得することができる。

【0030】

ユーザコールバック 410 が呼出されると、最新のインスタンス情報 404 を RM 402 から検索することができ、コンテキスト 405 に記憶することができる。加えて、トランザクションシステム 401 は、最新のインスタンス情報 404 を受信したことを示すフラグ 409 を設定することができる。

【0031】

本発明の一実施形態に従うと、トランザクションプロセス 403 は 1 つ以上のチェックポイントで構成することができる。たとえば、チェックポイントは、サービス呼出しの後、および初期化ルーチンの後にトリガすることができる。また、接続が確立されるか中断されたときにチェックポイントをトリガすることができる。

10

【0032】

図 4 に示されるように、チェックポイント 408 において、トランザクションプロセス 403 がフラグ 409 をチェックすることができる。フラグ 409 が上がっていれば、トランザクションサーバ 403 は、検索されたインスタンス情報 404 に基づいてトランザクションコンテキスト 407 を更新し、検索されたインスタンス情報 404 を（共有メモリにおける）状態テーブル 406 に記憶することができる。

【0033】

図 5 は、本発明の一実施形態に従った、トランザクション環境においてさまざまなチェックポイントを用いてトランザクションプロセスをサポートする例を示す。図 5 に示されるように、ステップ 501 において、トランザクションプロセスが開始される。次いで、トランザクションプロセスは、ステップ 502 において初期化プロセスに進み得る。

20

【0034】

初期化プロセス 502 は、1 つ以上のインスタンスチェックポイント、たとえばチェックポイント 510、を含み得る。たとえば、Tuxedo において、動的な登録が用いられる場合、チェックポイント 510 は、初期化ルーチン（たとえば、tpsvrinit）の後にサーバのスタートアップルーチンに配置されていてもよい。また、サーバが RM への接続を確立しようと試みる場合（たとえば、xa\_open（）関数呼出しが、tpopen（）関数呼出しでの呼出しに成功した後）、チェックポイント 510 をトリガすることができる。

30

【0035】

加えて、ステップ 512 において、トランザクションプロセスは、初期化プロセス 502 中に、インスタンス検知能力を使用可能にするためにインスタンス情報を検索することができる。

【0036】

さらに、ステップ 503 において、トランザクションプロセスは、サービス要求があるかどうかをチェックすることができる。ステップ 505 において、トランザクションプロセスはサービスディスパッチャ、たとえば、tmstvcdsp（）、を呼出すことができる。図 5 に示されるように、サービスルーチン 507 が要求メッセージを処理するために呼出される前にチェックポイント 506 をトリガすることができる。加えて、サービスルーチン 507 が完了した後、別のチェックポイント 508 をトリガすることができる。ステップ 509 において、トランザクションプロセスはサービスディスパッチャを終了させることができる。

40

【0037】

それ以外の場合には、サービス要求がなくてトランザクションプロセスが遮断されるべきであれば、ステップ 504 において、トランザクションプロセスはシャットダウン処理 511 を開始することができる。シャットダウンプロセス 511 では、チェックポイント 513 をトリガしてインスタンス ID 情報をクリーンにすることができる。たとえば、Tuxedo において、サーバが RM への接続を閉鎖しようとする場合（たとえば、xa\_close（）関数呼出しが tpclose（）関数呼出しで呼出される前）に、チェッ

50

クポイント513をトリガすることができる。最後に、トランザクションプロセスがステップ514において終了する。

#### 【0038】

本発明の一実施形態に従うと、システムは、動的な登録ではなく静的な登録を用いる場合、チェックポイント512およびチェックポイント513におけるのとは異なるやり方で動作する可能性もある。イニシエータをチェックすることなく、直接、インスタンス情報を検索して更新することができる。たとえば、静的な登録が用いられる場合、XAサーバは、`tpopen()`/`tpclose()`関数呼出しを用いて、RMへの接続を確立/削除することができる。また、`tpopen()`関数は、カスタマイズされた`tpsvrinit()`関数呼出しで呼出すことができ、`tpclose()`関数は、カスタマイズされた`tpsvrdone()`関数で呼出すことができる。

10

#### 【0039】

図6は、本発明の一実施形態に従った、トランザクション環境においてリソースマネージャ(RM)インスタンス検知をサポートするための例示的なフローチャートを示す。図6に示されるように、ステップ601において、トランザクションサーバは、1つ以上のRMインスタンスに関連付けられたリソースソースからリソースマネージャ(RM)インスタンス情報を受信することができる。この場合、受信されたインスタンス情報により、トランザクションサーバが現在接続されているのがどのRMインスタンスであるかを当該トランザクションサーバが検知することが可能となる。次いで、ステップ602において、システムは、受信されたインスタンス情報を、トランザクションサーバに関連付けられた1つ以上のテーブルに保存することができる。さらに、ステップ603において、システムは、トランザクションサーバが、1つ以上のテーブルに保存されたインスタンス情報に基づいてグローバルトランザクションを処理することを可能にする。

20

#### 【0040】

##### トランザクションアフィニティ

図7は、本発明の一実施形態に従った、トランザクション環境におけるインスタンス検知に基づいてトランザクションアフィニティをサポートする例を示す。図7に示されるように、トランザクション環境700は、1つ以上のリソースマネージャ(RM)(たとえばデータベース704に関連付けられたRMインスタンス702)を用いて、グローバルトランザクション710の処理をサポートすることができる。

30

#### 【0041】

トランザクションシステム701は、(たとえば、デフォルトのルーティングポリシーを用いて)トランザクションサーバ703に対してデータベース接続についての要求711をルーティングすることができる。加えて、システムは、トランザクションサーバ703にアフィニティコンテキスト705を割当てることができる。RMインスタンス702を識別する情報を含むアフィニティコンテキスト705は、共有メモリ(たとえば、`Tuxedo`)におけるグローバルトランザクションテーブル(`global transaction table: GTT`))に記憶可能であり、メッセージを用いて伝播させることができる。たとえば、トランザクションサーバ703は、上述のインスタンス検知特徴に基づいて、アフィニティコンテキスト705を介して、RMインスタンス名、RM/データベース名およびRM/データベースサービス名を取得することができる。

40

#### 【0042】

さらに、後続の要求712は、アフィニティコンテキスト705に基づいてトランザクションサーバ703にルーティングすることができる。加えて、他の後続の要求もまた、グローバルトランザクション710が完了する(かまたはクライアントコンテキストが終了する)まで、RMインスタンス702に接続されたトランザクションサーバ703にルーティングすることができる。

#### 【0043】

本発明の一実施形態に従うと、トランザクションアフィニティにより、RMインスタンス702に接続されている関連するデータベース要求711-712を、確実に、同じト

50

ランザクションサーバ703にルーティングすることができるようになる。こうして、ランザクションアフィニティはデータベースクラスタを最大限に活用することによってアプリケーション性能を向上させることができる。なぜなら、ランザクションアフィニティは、キャッシュヒットの可能性を増大させることによってデータベース性能を向上させることができるからである。

【0044】

本発明の一実施形態に従うと、システムは、グローバルランザクション710を実行するための他のルーティングポリシーと共にランザクションアフィニティルーティングポリシーを適用することができる。たとえば、以下のルーティング優先順位は、Tuxedoでサポートすることができる。

【0045】

1. ドメインのためのランザクション優先順位ルーティング
2. クライアント/サーバアフィニティルーティング
3. ランザクションアフィニティルーティング
4. サービスロードに従ったロードバランシング

図8は、本発明の一実施形態に従った、ランザクション環境におけるランザクションアフィニティルーティングをサポートする例を示す。図8に示されるように、ランザクションシステムは、ランザクションアフィニティルーティングポリシーを用いて、グローバルランザクションの処理をサポートすることができる。

【0046】

ステップ801において、システムは、データベース接続を含むランザクション要求を受信することができる。次いで、ステップ802において、システムは、既存のアフィニティコンテキストがあるかどうかをチェックすることができる。

【0047】

グローバルランザクションにおいてアフィニティコンテキストが含まれない場合、システムは、ステップ809において、ロードバランシングルーティングを実行することができる。

【0048】

グローバルランザクションに既存のアフィニティコンテキストが含まれている場合、システムは、サーバを発見するためにランザクションアフィニティルーティングポリシーを適用することができる。

【0049】

ステップ803において、システムは、同じインスタンス名、同じデータベース(database:DB)名および同じサービス名に関連付けられたサーバを発見しようと試みてもよい。

【0050】

システムがサーバを発見することができない場合、ステップ804において、システムは、同じDB名および同じサービス名に関連付けられておりかつ現在のグローバルランザクションには含まれないグループにあるサーバを発見しようと試みてもよい。

【0051】

システムがサーバを発見することができない場合、ステップ805において、システムは、同じDB名および同じインスタンス名に関連付けられたサーバを発見しようと試みてもよい。

【0052】

ランザクションシステムがサーバを発見することができない場合、ステップ806において、システムは、同じDB名に関連付けられたサーバを発見しようと試みてもよい。

【0053】

ステップ807において、システムは、既存のアフィニティコンテキストに基づいてサーバを発見することができるかもしれない。他方では、ステップ808において、システムはサーバを発見することができないかもしれない。次いで、ステップ809において、

10

20

30

40

50

システムは、ロードバランシングルーティングに従ってサーバを発見しようと試みることができる。

【0054】

図9は、本発明の一実施形態に従った、トランザクション環境においてアフィニティコンテキストを含むメッセージを送信する例を示す。図9に示されるように、トランザクションシステム901はトランザクション環境900におけるトランザクション処理をサポートすることができる。さらに、トランザクションシステム901におけるトランザクションサーバ903は、トランザクションコンテキスト907（たとえば、TuxedoにおけるTUXC）に基づいてトランザクション処理をサポートすることができる。

【0055】

図9に示されるように、トランザクションサーバ903は、共有メモリ902から関連するアフィニティコンテキスト904（たとえば、TuxedoにおけるGTT）を取得することができ、関連するアフィニティコンテキスト914を用いてトランザクションコンテキスト907を更新することができる。チェックポイント908がトリガされると、システムは、関連するアフィニティコンテキスト914を、トランザクションコンテキスト907からメッセージキュー905内のメッセージ906にコピーすることができる。

【0056】

これにより、トランザクションシステム901は、サービスにメッセージに906を送信する前に、サービスルーティングのために、トランザクションコンテキスト907における関連するアフィニティコンテキスト914を参照することができる。

【0057】

図10は、本発明の一実施形態に従った、トランザクション環境においてアフィニティコンテキストを含むメッセージを受信する例を示す。図10に示されるように、トランザクション環境1000におけるトランザクションシステム1001は、メッセージキュー1005を用いて、1つ以上のメッセージ（たとえば、メッセージ1006）を受信することができる。

【0058】

トランザクションシステム1001におけるトランザクションサーバ1003は、最初のブート後に連続的に（要求を含む）メッセージをデキューすることができる。図10に示されるように、トランザクションサーバ1003はメッセージキュー1005からメッセージ1006を読み出すことができ、メッセージ1006内のサービス要求を処理する。

【0059】

サービス要求の処理中に、システムは、アフィニティコンテキストをメッセージ1006からトランザクションコンテキスト1007にコピーするためのチェックポイント1008をトリガすることができる。次いで、システムは、トランザクションコンテキスト1007内のアフィニティコンテキスト1014を用いて、共有メモリ1002内のアフィニティコンテキスト1004を更新することができる。

【0060】

メッセージ1006内の要求が処理されると、トランザクションサーバ1003は、メッセージキュー1005からより多くのメッセージを読み出すことができる。そうでない場合、トランザクションサーバ1003は、次の要求が到達するまで、メッセージキュー1005上で待機することができる。

【0061】

図11は、本発明の一実施形態に従った、トランザクション環境においてクライアントコンテキスト内におけるアフィニティルーティングをサポートする例を示す。図11に示されるように、トランザクション環境1100におけるトランザクションシステム1101は、1つ以上のリソースマネージャ（RM）（たとえばデータベース1104に関連付けられるRMインスタンス1102）を用いて、クライアントコンテキスト1110におけるトランザクション処理をサポートすることができる。たとえば、クライアントコンテキスト1110内では、ウェブ会話は回数ごとに接続したり遮断したりすることができる

10

20

30

40

50

。これらの各々の接続中、会話は、ショッピングカートなどの同じ（または同様の）データに対する参照および／またはアクセスを実行してもよい。

【 0 0 6 2 】

本発明の一実施形態に従うと、システムは、（たとえば、デフォルトのルーティングポリシーに基づいて）トランザクションサーバ 1 1 0 3 に対してデータベース接続についての要求 1 1 1 1 をルーティングすることができる。加えて、システムは、R M 1 1 0 2 を示すアフィニティコンテキスト 1 1 0 5 をトランザクションサーバ 1 1 0 3 に割当てることができる。

【 0 0 6 3 】

さらに、クライアントコンテキスト 1 1 1 0 内の 1 つ以上の後続の要求（たとえば、要求 1 1 1 2 ）は、クライアントコンテキスト 1 1 1 0 が終了するかまたは関連するトランザクションが完了するまで、アフィニティコンテキスト 1 1 0 5 に基づいてトランザクションサーバ 1 1 0 3 にルーティングすることができる。これにより、トランザクションシステム 1 1 0 1 は、クライアントコンテキスト 1 1 1 0 内のさまざまなデータベース動作を同じ R M インスタンス 1 1 0 2 に方向付けることを確実にすることができる。

【 0 0 6 4 】

本発明の一実施形態に従うと、トランザクションシステム 1 1 0 1 は、クライアントコンテキスト 1 1 1 0 内のアフィニティを暗示しているさまざまなロードバランスアドバイザリイベントをデータベースから受信することができる。たとえば、T u x e d o においては、データベースから受信されたロードバランシングアドバイザリイベントは、パラメータである A F F I N I T Y H I N T を含み得る。これは、アフィニティが特定のインスタンスとサービスとの組合せに対してアクティブとなるかまたはイナクティブとなるかどうかを示すフラグである。A F F I N I T Y H I N T パラメータは、ウェブセッションの期間中継続する一時的なアフィニティであって、サービスに対して目標を設定することによってロードバランシングアドバイザリが使用可能になると、自動的に使用可能にすることができる。加えて、同じサービスを提供するさまざまなインスタンスは、A F F I N I T Y H I N T についてのさまざまな設定を有し得る。

【 0 0 6 5 】

本発明の一実施形態に従うと、トランザクションシステム 1 1 0 1 は、関連するデータベース動作がトランザクション内にある場合には、クライアントコンテキストベースのアフィニティポリシーではなく、トランザクションアフィニティルーティングポリシーを適用してもよい。他方では、システムは、デフォルトの t u x e d o ロードバランスルートポリシーに基づいて、クライアントコンテキストベースのアフィニティルーティングポリシーを実現することができる。

【 0 0 6 6 】

図 1 2 は、本発明の一実施形態に従った、トランザクション環境におけるさまざまなドメインにわたってアフィニティコンテキストを伝播させる例を示す。図 1 2 に示されるように、トランザクション環境 1 2 0 0 は、1 つ以上のリソースマネージャ（R M ）、たとえばデータベース 1 2 1 5 に関連付けられた R M インスタンス 1 2 0 5、を用いてトランザクション処理をサポートすることができる。

【 0 0 6 7 】

本発明の一実施形態に従うと、システムは、（たとえば、デフォルトのルーティングポリシーを用いて）トランザクションサーバ 1 2 0 3 に対してデータベース接続についての要求 1 2 1 1 をルーティングすることができる。加えて、システムは、R M インスタンス 1 2 0 2 を示すアフィニティコンテキスト 1 2 0 7 をトランザクションサーバ 1 2 0 3 に割当てることができる。

【 0 0 6 8 】

さらに、トランザクション環境 1 2 0 0 におけるトランザクションドメインは、要求 1 2 1 1 がさまざまなドメイン間で転送されるべきである場合、ドメインにわたってアフィニティコンテキスト情報を伝播させることができる。

10

20

30

40

50

## 【 0 0 6 9 】

図 1 2 に示されるように、トランザクションドメイン 1 2 0 1 は、アフィニティコンテキスト 1 2 0 7 をアフィニティキースtring 1 2 0 8 に変換してから、このアフィニティキースtring 1 2 0 8 をリモートドメイン 1 2 0 2 に送信することができる。アフィニティキースtring 1 2 0 8 を受信した後、トランザクションドメイン 1 2 0 2 は、アフィニティキースtring 1 2 0 8 を、トランザクションドメイン 1 2 0 2 においてトランザクションサーバ 1 0 0 4 が使用可能なアフィニティコンテキスト 1 2 0 6 に変換することができる。

## 【 0 0 7 0 】

これにより、1 つ以上の後続の要求（たとえば、要求 1 2 1 2 ）をアフィニティコンテキスト 1 2 0 6 に基づいて R M インスタンス 1 2 0 2 に方向付けることができる。

10

## 【 0 0 7 1 】

図 1 3 は、本発明の一実施形態に従った、トランザクション環境においてアプリケーションサーバにアフィニティコンテキストを伝播させる例を示す。図 1 3 に示されるように、トランザクション環境 1 3 0 0 におけるトランザクションシステム 1 3 0 1 は、1 つ以上のリソースマネージャ（R M ）（たとえばデータベース 1 3 1 5 に関連付けられた R M インスタンス 1 3 0 5 ）を用いて、トランザクション処理をサポートすることができる。

## 【 0 0 7 2 】

本発明の一実施形態に従うと、システムは、（たとえば、デフォルトのルーティングポリシーを用いて）トランザクションサーバ 1 3 0 3 に対してデータベース接続についての要求 1 3 1 1 をルーティングすることができる。加えて、システムは、R M インスタンス 1 3 0 5 を示すアフィニティコンテキスト 1 3 0 7 をトランザクションサーバ 1 3 0 3 に割当てることができる。

20

## 【 0 0 7 3 】

さらに、トランザクションシステム 1 3 0 1 （たとえば、T u x e d o T D o m a i n ）は、アフィニティキースtring 1 3 0 8 を介して、アプリケーションサーバ 1 3 0 2 （たとえば W e b L o g i c アプリケーションサーバ）にアフィニティコンテキスト 1 3 0 7 情報を伝播させることができる。たとえば、アフィニティコンテキスト 1 3 0 7 は、コネクタ 1 3 0 4 （たとえば T u x e d o W T C ）を介してトランザクションシステム 1 3 0 1 とアプリケーションサーバ 1 3 0 2 との間でやり取りすることができる。

30

## 【 0 0 7 4 】

トランザクションシステム 1 3 0 1 がコネクタ 1 3 0 4 に要求を送信すると、トランザクションシステム 1 3 0 1 はアフィニティコンテキスト 1 3 0 7 をアフィニティキースtring 1 3 0 8 に変換することができる。コネクタ 1 3 0 4 がトランザクションシステム 1 3 0 1 から要求を受信すると、コネクタ 1 3 0 4 は、アフィニティキースtring 1 3 0 8 を、アプリケーションサーバ 1 3 0 2 が使用可能なトランザクションコンテキスト 1 3 0 6 に変換することができる。

## 【 0 0 7 5 】

コネクタ 1 3 0 4 がトランザクションシステム 1 3 0 1 に要求を送信すると、コネクタ 1 3 0 4 は、アプリケーションサーバ 1 3 0 2 に関連付けられたトランザクションコンテキスト 1 3 0 6 からアフィニティコンテキストを取得することができる。トランザクションシステム 1 3 0 1 がコネクタ 1 3 0 4 から要求を受信すると、トランザクションシステム 1 3 0 1 は、アフィニティキースtring 1 3 0 8 をアフィニティコンテキスト 1 3 0 7 に変換することができる。

40

## 【 0 0 7 6 】

これにより、アプリケーションサーバ 1 3 1 2 における 1 つ以上の後続の要求（たとえば要求 1 3 1 2 ）を、アフィニティコンテキスト 1 3 0 5 に基づいて R M インスタンス 1 3 0 2 に方向付けることができる。

## 【 0 0 7 7 】

図 1 4 は、本発明の一実施形態に従った、トランザクション環境においてインスタンス

50

検知に基づいてトランザクションアフィニティをサポートするための例示的なフローチャートを示す。図 14 に示されるように、ステップ 1401 において、システムは、リソースマネージャ (RM) インスタンスに接続されているトランザクションサーバに要求をルーティングすることができる。次いで、ステップ 1402 において、システムは、トランザクションサーバが関連付けられている RM インスタンスを示しているアフィニティコンテキストをトランザクションサーバに割当てることができる。さらに、ステップ 1403 において、システムは、要求に関連する 1 つ以上の後続の要求を、アフィニティコンテキストに基づいて、トランザクションサーバにルーティングすることができる。

【0078】

共通のトランザクション識別子 (XID)

10

図 15 は、本発明の一実施形態に従った、さまざまなトランザクション識別子 (XID) を用いて、トランザクション環境においてグローバルトランザクションを処理する例を示す。図 15 に示されるように、トランザクションシステム 1500 は、さまざまなリソースマネージャ (RM) インスタンス (たとえば、データベース 1506 に接続する RM インスタンス 1504 - 1505) を用いて、グローバルトランザクション 1510 の処理をサポートすることができる。

【0079】

本発明の一実施形態に従うと、グローバルトランザクション 1510 はグローバルトランザクション識別子 (global transaction identifier: GTRID 1520) に関連付けることができる。グローバルトランザクション 1510 内では、同じグループ内にある関連するトランザクションサーバは 1 つのトランザクションブランチを共有することができ、別々のグループにおけるトランザクションサーバはそれぞれ別々のトランザクションブランチを用いてもよい。

20

【0080】

図 15 に示されるように、トランザクションシステム 1500 は、グローバルトランザクション 1510 を処理するために複数のブランチ (たとえばブランチ A 1512 - C 1523) を用いてもよい。ブランチ A 1521 - C 1523 の各々をブランチ修飾子 (たとえば BQUAL A 1531 - C 1533) に関連付けることもできる。

【0081】

本発明の一実施形態に従うと、トランザクションシステム 1500 は、さまざまなブランチ A 1521 - C 1523 上でのグローバルトランザクション 1510 の処理を管理するために、さまざまなトランザクションマネージャ (transactional manager: TM) (たとえば、TM A 1501 - TM C 1503) を用いることができる。

30

【0082】

たとえば、TM A 1501 はトランザクション識別子 (XID) A 1511 に関連付けられており、ブランチ A 1521 上でのグローバルトランザクション 1510 の処理を管理するための役割を果たし得る。TM B 1502 はトランザクション識別子 (XID) B 1512 に関連付けられており、ブランチ B 1522 上でのグローバルトランザクション 1510 の処理を管理するための役割を果たし得る。TM C 1503 は、トランザクション識別子 (XID) C 1513 に関連付けられており、ブランチ C 1523 上でのグローバルトランザクション 1510 の処理を管理するための役割を果たし得る。

40

【0083】

図 15 に示されるように、グローバルトランザクション 1510 におけるさまざまなブランチ A 1521 - C 1523 のための XID A 1511 - XID C 1513 は、同じ GTRID 1520 (およびフォーマット ID) を共有することができ、かつ、さまざまなブランチ修飾子 (すなわち BQUAL A 1531 - BQUAL C 1533) を有していてもよい。

【0084】

本発明の一実施形態に従うと、システムは、トランザクションサーバの 2 つ以上のグループがグローバルトランザクション 1510 に含まれている場合、グローバルトランザク

50

ション 1 5 1 0 上において 2 相コミット ( 2 P C ) プロセスを呼出すことができる。

【 0 0 8 5 】

図 1 5 に示されるように、システムは、グローバルトランザクション 1 5 1 0 におけるさまざまな参加トランザクショングループ (たとえば、T M A 1 5 0 1 - T M B 1 5 0 2 ) が、実際に同じ R M インスタンス 1 5 0 4 に関連付けられている場合であっても、グローバルトランザクション 1 5 1 0 を処理するために 2 P C モデルを用いる可能性がある。

【 0 0 8 6 】

本発明の一実施形態に従うと、システムは、2 つ以上のグループが同じリソースマネージャインスタンス 1 5 0 4 上で実行されている場合に、共通の X I D を用いることによってグローバルトランザクション 1 5 1 0 を処理する性能を向上させることができる。

10

【 0 0 8 7 】

図 1 6 は、本発明の一実施形態に従った、共通の X I D を用いてトランザクション環境におけるグローバルトランザクションを処理する例を示す。図 1 6 に示されるように、トランザクションシステム 1 6 0 0 は、データベース 1 6 0 6 に接続するリソースマネージャ ( R M ) インスタンス 1 6 0 4 - 1 6 0 5 を用いて、G R T I D 1 6 2 0 に関連付けられたグローバルトランザクション 1 6 1 0 の処理をサポートすることができる。

【 0 0 8 8 】

さらに、トランザクションシステム 1 6 0 0 は、さまざまなトランザクショングループ (たとえば、ブランチ A 1 6 2 1 - C 1 6 2 3 ) におけるさまざまなトランザクションアプリケーションサーバ上でグローバルトランザクション 1 6 1 0 の処理を管理するためにトランザクションマネージャ ( T M ) A 1 6 0 1 - C 1 6 0 3 を用いることができる。

20

【 0 0 8 9 】

本発明の一実施形態に従うと、リソースマネージャ ( R M ) 1 6 0 4 - 1 6 0 5 の各々は、たとえばデータベース名、サーバ名およびインスタンス名に基づいて固有に識別することができる。トランザクションシステム 1 6 0 0 におけるトランザクションサーバは、データベースインスタンス検知能力に基づいて、それが現在接続しているのがどの R M インスタンスであるかを検知することができる。

【 0 0 9 0 】

図 1 6 に示されるように、トランザクションシステム 1 6 0 0 は T M A 1 6 0 1 などのコーディネータを含み得る。コーディネータ T M A 1 6 0 1 は、R M インスタンス 1 6 0 4 を識別するインスタンス I D 1 6 4 1 に関連付けられている。

30

【 0 0 9 1 】

加えて、グローバルトランザクション 1 6 1 0 は、コーディネータ 1 6 0 1 がリモートサーバノードにおいて配置されているかまたは配置され得るローカルサーバノード上に位置し得る 1 つ以上の参加サーバ (たとえば、パーティシパント T M B 1 6 0 2 - T M C 1 6 0 3 ) を含み得る。パーティシパント T M B 1 6 0 2 - T M C 1 6 0 3 の各々はまた、それら各々が接続している R M インスタンスを識別するインスタンス I D (たとえば、インスタンス I D 1 6 4 2 - I D 1 6 4 3 ) に関連付けることもできる。

【 0 0 9 2 】

40

本発明の一実施形態に従うと、グローバルトランザクション 1 6 1 0 を処理するために共通の X I D 特徴が使用可能になると、コーディネータ T M A 1 6 0 1 についての X I D 1 6 1 1 はグローバルトランザクション 1 6 1 0 内で共有可能となる (すなわち、X I D 1 6 1 1 が共通の X I D として用いられる)。これにより、複数のグループを含み、O r a c l e データベースなどのクラスタ化されたデータベース上で実行されるトランザクションアプリケーションは、データベースインスタンス検知を利用することによってトランザクション性能を向上させることができる。

【 0 0 9 3 】

図 1 6 に示されるように、グローバルトランザクション 1 6 1 0 のためのコーディネータ T M A 1 6 0 1 は、グローバルトランザクション 1 6 1 0 のライフサイクル内で、X

50



ＩＤ１６１１およびインスタンスＩＤ１６４１などのさまざまなタイプの情報をさまざまなパーティシパントＴＭ Ｂ１６０２ - ＴＭ Ｃ１６０３に伝播させることができる。

【００９４】

さらに、パーティシパントＴＭ Ｂ１６０２ - ＴＭ Ｃ１６０３の各々は、受信されたインスタンスＩＤ１６４１をそれ自体のインスタンスＩＤと比較することによって、それがコーディネータＴＭ Ａ１６０１と同じＲＭを共有しているかどうかを判断することができる。インスタンスＩＤが同じであれば、パーティシパントＴＭ Ｂ１６０２ - ＴＭ Ｃ１６０３はそれら自体を共通のＸＩＤサーバ（またはグループ）としてマーク付けすることができる。

【００９５】

たとえば、システムは、ＴＭ Ｂ１６０２がＴＭ Ａ１６０１と同じＲＭインスタンス１６０４を共有しているので、ブランチＢ１６２２上に一致を発見する可能性がある。これにより、ＴＭ Ｂ１６０２は、トランザクション処理をサポートするためにそれ自体のＸＩＤではなく共通のＸＩＤ１６１１を用いることができる。次いで、ＴＭ Ｂ１６０２は、それが共通のＸＩＤ １６１１を用いていることをコーディネータＴＭ Ａ１６０１に通知することができる。このような場合、コーディネータＴＭ Ａ１６０１がグローバルトランザクション１６１０をコミットまたはロールバックするように動作すると、システムは、（ＢＱＵＡＬ Ａ１６３１に基づいた）共通のＸＩＤ１６１１を用いているので、ブランチＢ１６２２を無視することができる。

【００９６】

他方では、ＴＭ Ｃ１６０３が（インスタンスＩＤ １６４３を用いて）別のＲＭ １６０５に関連付けられているので、システムは、（ＸＩＤ Ｃ１６１３およびＢＱＵＡＬ Ｃ１６３３を用いる）ブランチＣ １６２３上で一致を発見しない可能性がある。任意には、ＴＭ Ｃ１６０３は、それが共通のＸＩＤ１６１１用いていないことをコーディネータＴＭ Ａ１６０１に通知することができる。次いで、システムは、２相コミット（２ＰＣ）処理モデルに従ってトランザクションブランチＣ１６２３を処理することができる。

【００９７】

本発明の一実施形態に従うと、さまざまな共通のＸＩＤグループ、すなわち、コーディネータ１６０１と同じＲＭインスタンス１６０４に関連付けられているトランザクションサーバのグループは、共通のＸＩＤ１６１１を介してＲＭインスタンス１６０４にアクセスすることができる。

【００９８】

さらに、コミット要求が呼出されると、コーディネータ１６０１はローカルの共通ＸＩＤグループに対して如何なるメッセージも送信しなくなる可能性がある。システムは、各々のローカルの共通ＸＩＤグループの状態を同時に読取り専用に変更することができる。また、リモートの共通ＸＩＤグループは、コーディネータ１６０１から準備要求を受信することができ、実際に如何なるデータベース動作も行なうことなく、その状態を読取り専用に変更してもよい。これにより、システムは、これらのグループのうちの１つ（たとえば、コーディネータのグループ）を準備／コミットするだけでよいかもしれない。

【００９９】

加えて、システムは、インスタンスＩＤが変化した場合に、共通ＸＩＤグループを非共通ＸＩＤグループに変更することができる。たとえば、ブランチＢ１６２２が別のＲＭインスタンスを用いるように変化した場合、システムは、代わりに（たとえば、ＢＱＵＡＬ Ｂ１６３２に基づいて）２相コミット（２ＰＣ）プロセスを呼出すことができる。

【０１００】

図１７は、本発明の一実施形態に従った、トランザクション環境においてデータベースインスタンス検知に基づいて１相コミット（１ＰＣ）処理モデルをサポートする例を示す。図１７に示されるように、トランザクションシステム１７００は、データベース１７０６に接続するリソースマネージャ（ＲＭ）インスタンス１７０４を用いて、ＧＲＴＩＤ

10

20

30

40

50

１７２０に関連付けられたグローバルトランザクション１７１０の処理をサポートすることができる。

【０１０１】

トランザクションシステム１７０１は、複数のトランザクションマネージャ（ＴＭ）Ａ１７０１ - Ｃ１７０３を含み得る。これら複数のトランザクションマネージャ（ＴＭ）Ａ１７０１ - Ｃ１７０３は、さまざまなトランザクショングループ（すなわち、ブランチＡ１７２１ - Ｃ１７２３）におけるグローバルトランザクション１７１０の処理を管理するために用いられる。

【０１０２】

さらに、ＴＭ Ａ１７０１ - ＴＭ Ｃ１７０３は、単一のリソースマネージャ（ＲＭ）インスタンス１７０４に基づいてグローバルトランザクション１７１０の処理を管理することができる。ＴＭ Ａ１７０１ - ＴＭ Ｃ１７０３の各々は、インスタンス識別子（ＩＤ）、たとえばインスタンスＩＤ１７４１ - ＩＤ１７４３、を維持することができる。

【０１０３】

図１７に示されるように、トランザクションシステム１７００はコーディネータ（たとえば、ＴＭ Ａ１７０１）を含み得る。グローバルトランザクション１７１０のためのコーディネータＴＭ Ａ１７０１は、グローバルトランザクション１７１０のライフサイクル内において、共通のＸＩＤ１７１１およびインスタンスＩＤ１７４１などのさまざまなタイプの情報をさまざまな参加トランザクションサーバ（たとえば、ＴＭ Ｂ１７０２ - ＴＭ Ｃ１７０３）に伝播させることができる。

【０１０４】

本発明の一実施形態に従うと、トランザクションアフィニティ性能に基づいて、システムは、グローバルトランザクション１７１０における関連するすべての要求を同じＲＭインスタンス１７０４にルーティングすることができる。さらに、インスタンス検知能力に基づいて、ＴＭ Ａ１７０１ - ＴＭ Ｃ１７０３は、単一のＲＭインスタンス１７０４だけがグローバルトランザクション１７１０において用いられることを検知することができる。なぜなら、別々のインスタンスＩＤ１７４１ - ＩＤ１７４３がすべて、同じＲＭインスタンス１７０４を識別するからである。これにより、コーディネータＴＭ Ａ１７０１は、トランザクション環境１７００においてグローバルトランザクション１７１０を調整するために共通のＸＩＤ１７１１（たとえば、ＢＱＵＡＬ Ａ１７３１およびＧＲＴＩＤ １７２０に基づいたそれ自体のＸＩＤ）を用いることができる。

【０１０５】

図１７に示されるように、コーディネータＴＭ Ａ１７０１は、コミット段階において、如何なる「準備／コミット」要求も他のグループに送信しない可能性がある。なぜなら、それらはすべて共通のＸＩＤグループであるからである。さらに、システムは１相コミット（１ＰＣ）処理モデルを利用することができる。

【０１０６】

本発明の一実施形態に従うと、読取り専用の１相コミット最適化により、システム性能を著しく向上させることができる。これにより、他のすべてのグループが読取り専用に戻った場合に、保存されたグループに対して１ＰＣ処理を実行することが可能となる。グローバルトランザクションのブランチがすべて、たとえば同じインスタンスまたは同じＲＡＣで密に連結されている場合、当該性能を向上させることができる。

【０１０７】

たとえば、トランザクション環境１７００は、グローバルトランザクション１７１０を処理するためのＮ個（この場合、 $N > 1$ ）の参加グループを有し得る。それらのうち、Ｍ個（この場合 $M < N$ ）の参加グループはコーディネータと同じインスタンスＩＤを有している可能性がある。

【０１０８】

データベースインスタンスが検知されていない２相コミット（２ＰＣ）処理モデルを用いる場合、システムは、（たとえば、図１５に示されるように）データベース上で、Ｎ個

10

20

30

40

50

の準備動作および1個のコミット動作を実行してもよい。また、システムは、トランザクションログを書込む必要があるかもしれない。

【0109】

代替的には、(たとえば、図16に示されるように)データベースインスタンス検知に基づいて、システムは、データベース上で、 $N - M$ 個の準備動作および1個のコミット動作を実行する必要があるかもしれない( $M$ 個の準備動作が減じられている)。

【0110】

さらに、 $M = N - 1$ である場合、これは、グローバルトランザクションにおける他のすべての参加グループがコーディネータと同じトランザクションブランチを共有し得ることを示している。さらに、グローバルトランザクションを処理する際にブランチは1つしか存在しない。システムは、1つのコミット操作を実行するだけでよいかもしれず、この場合、 $N$ 個(または $M + 1$ 個)の準備動作が減じられている。また、システムはトランザクションログを書込む必要はないかもしれない。

【0111】

図18は、本発明の一実施形態に従った、トランザクションミドルウェア環境においてデータベースインスタンス検知に基づいてグローバルトランザクションを処理する例を示す。図18に示されるように、トランザクションシステム、たとえばOracle Tuxedoシステムは、複数のトランザクショングループ(たとえばTuxedoグループA1802 - B1803)を用いて、グローバルトランザクションの処理をサポートすることができる。

【0112】

さらに、TuxedoグループA1802 - B1803の各々は、1セットのトランザクション管理サーバ(transaction manager server: TMS)を有し得る。たとえば、グループA1802は、サーバA1804と、コーディネータとして機能することができるTMS A1806とを含む。加えて、グループA1802は、共有メモリ(たとえばTuxedo掲示板(BB)A1808)を含み得る。さらに、グループB1803は、サーバB1805およびTMS B1807、ならびに共有メモリ(たとえばTuxedo BB B1809)を含む。

【0113】

図18に示される例においては、ステップ1811において、クライアント1801は、関数呼出しtpcall(サービスA)を呼出すことによってサーバA1804上のサービス(たとえば、サービスA)にアクセスすることができる。次いで、ステップ1812において、サーバA1804は、Tuxedo BB A1808において、関連するグローバルトランザクションテーブルエントリ(global transaction table entry: GTE)を作成することができる。

【0114】

加えて、ステップ1813において、クライアント1801は、関数呼出しtpcall(サービスB)を呼出すことによってサーバB1805上で別のサービス(たとえばサービスB)にアクセスすることができる。ステップ1814において、サーバA1804は、グループA1802についての関連する情報をTuxedo BB B1809に追加することができる。また、ステップ1815において、クライアント1801は、グループB1803についての関連情報をTuxedo BB A1808に追加することができる。

【0115】

さらに、ステップ1816において、クライアント1801は、関数呼出しtpcommit()を呼出すことによってトランザクションをコミットするように要求することができる。Tuxedoは、グローバルトランザクションに含まれるすべてのグループが同じRMインスタンス上で実行される場合に、グローバルトランザクション上に直接IPCを呼出すことができる。ステップ1817において、コーディネータTMS A1806は次にグローバルトランザクションをコミットすることができる。

## 【0116】

1 P C呼出しが成功した場合、ステップ1818において、コーディネータT M S A 1806は、ローカルノードにおけるG T T Eを削除することができる。次いで、ステップ1819において、コーディネータ1806は、リモートの共通X I DグループであるグループB 1803に対して、そのブランチを削除 (forget) するように通知することができる。最後に、ステップ1820において、T M S B 1807はT u x e d o B B B 1809を更新することができる。

## 【0117】

図19は、本発明の一実施形態に従った、共通のX I Dを用いてトランザクション環境において複数のドメインにわたってグローバルトランザクションを処理する例を示す。図19に示されるように、トランザクションシステム1900は、データベース1906に接続するリソースマネージャ (R M) インスタンス1904に基づいて、複数のドメイン (たとえば、ドメインA 1951 - B 1952) にわたってグローバルトランザクション1910の処理をサポートすることができる。

10

## 【0118】

加えて、さまざまなブランチA 1921 - B 1923は、トランザクションシステム1900においてG T R I D 1920を共有することができる。ローカルドメインA 1951におけるコーディネータT M A 1901は、インスタンスI D 1941に基づいたトランザクション識別子 (X I D) 1911およびインスタンス情報をリモートドメインB 1952に伝播させることができる。

20

## 【0119】

本発明の一実施形態に従うと、ドメインA 1951内において固有であるインスタンスI D 1941は、サーバ立上げシーケンスが異なっているせいで、別のドメイン (たとえば、ドメインB 1952) において、異なるものになる可能性もある。

## 【0120】

図19に示されるように、ドメインに直接クロスさせてインスタンスI D 1941を伝播させるのではなく、コーディネータT M A 1901は、インスタンスI D 1941をフォーマットされたストリング1908に変換してから、ドメインにわたって伝播させることができる。たとえば、フォーマットされたストリング1908は、データベース名、サーバ名およびインスタンス名を含み得る。

30

## 【0121】

加えて、ドメインゲートウェイサーバ1905は、ローカルドメインA 1951とリモートドメインB 1952との間の通信をサポートするために用いることができる。ドメインゲートウェイサーバ1905のアウトバウンドインターフェイスは、インスタンスI D 1941からのインスタンス情報をフォーマットされたストリング1908にマッピングすることができる。ドメインゲートウェイサーバ1905のインバウンドにより、フォーマットされたストリング1908からのインスタンス情報をインスタンスI D 1941にマッピングすることができる。

## 【0122】

たとえば、T u x e d oにおいて、ユーザは、ビジネス上の理由から、T u x e d oグループをさまざまなドメインに分割してもよい。G W T D O M A I Nサーバなどのゲートウェイサーバは異なるドメイン間での通信をサポートするために用いることができる。さらに、コーディネータドメインにおけるG W T D O M A I Nサーバはプロキシとして機能することもできる。加えて、他のドメインにおけるG W T D O M A I Nサーバを介するすべての関与サーバが共通のX I Dを用いるように設定されている場合、G W T D O M A I Nサーバは、共通のX I Dを用いるように構成することができる。

40

## 【0123】

本発明の一実施形態に従うと、リモートドメインB 1952は、共通のX I D 1911を、インポートされたX I D 1913として記憶することができる。図19に示されるように、インポートされたX I D 1913が存在し、インポートされたX I D 1913に関

50

連付けられた B Q U A L A 1 9 3 1 が有効である場合、ブランチ B 1 9 2 3（すなわち共通の X I D グループ）はデータベースにアクセスするためにこのインポートされた X I D 1 9 4 3 を用いることができる。

【 0 1 2 4 】

本発明の一実施形態に従うと、ドメイン間トランザクションが単一の R M インスタンスを含む場合、システムはまた、データベースインスタンス検知に基づいて、ドメイン間トランザクションを処理する際に 1 相コミット（ 1 P C ）モデルを利用することもできる。

【 0 1 2 5 】

図 2 0 は、本発明の一実施形態に従った、トランザクション環境において共通の X I D を用いてグローバルトランザクションを処理するための例示的なフローチャートを示す。図 2 0 に示されるように、ステップ 2 0 0 1 において、グローバルトランザクションのためのコーディネータは、トランザクション環境におけるグローバルトランザクションの 1 つ以上のパーティシパントに対してリソースマネージャインスタンスについての共通のトランザクション識別子および情報を伝播させることができる。次いで、ステップ 2 0 0 2 において、システムは、リソースマネージャインスタンスをコーディネータと共有している上記 1 つ以上のパーティシパントが、共通のトランザクション識別子を用いることを可能にする。さらに、ステップ 2 0 0 3 において、コーディネータは、1 つのトランザクションブランチを用いて、リソースマネージャインスタンスを共有する上記 1 つ以上のパーティシパントのためのグローバルトランザクションを処理することができる。

【 0 1 2 6 】

本発明の一実施形態に従うと、トランザクション環境においてトランザクション処理をサポートするための方法が提供される。当該方法は、リソースマネージャ（ R M ）インスタンスに接続されているトランザクションサーバに要求をルーティングするステップと、トランザクションサーバが関連付けられている R M インスタンスを示しているアフィニティコンテキストをトランザクションサーバに割当てするステップと、アフィニティコンテキストに基づいて、要求に関連する 1 つ以上の後続の要求をトランザクションサーバにルーティングするステップとを含む。

【 0 1 2 7 】

本発明の一実施形態に従うと、当該方法はさらに、上記 R M インスタンスが、クラスタ化されたデータベースにおけるインスタンスとなることを可能にするステップを含む。

【 0 1 2 8 】

本発明の一実施形態に従うと、当該方法はさらに、上記要求および上記 1 つ以上の後続の要求をグローバルトランザクションに関連付けることを可能にするステップを含む。

【 0 1 2 9 】

本発明の一実施形態に従うと、当該方法はさらに、 R M 名、インスタンス名およびサービス名に基づいたアフィニティルーティングポリシーを用いて、1 つ以上の後続の要求をルーティングするステップを含む。

【 0 1 3 0 】

本発明の一実施形態に従うと、当該方法はさらに、上記要求および上記 1 つ以上の後続の要求をクライアントコンテキストに関連付けることを可能にするステップを含む。

【 0 1 3 1 】

本発明の一実施形態に従うと、当該方法はさらに、メッセージを送信する前に、トランザクションサーバを介してメッセージにアフィニティコンテキストを含めるステップを含む。

【 0 1 3 2 】

本発明の一実施形態に従うと、当該方法はさらに、別のトランザクションサーバから受信したメッセージに基づいて、トランザクションサーバにおけるアフィニティコンテキストを更新するステップを含む。

【 0 1 3 3 】

本発明の一実施形態に従うと、当該方法はさらに、アフィニティコンテキストストリン

10

20

30

40

50

グを用いて、アフィニティコンテキストをトランザクションドメインから別のトランザクションドメインに転送するステップを含む。

【0134】

本発明の一実施形態に従うと、当該方法はさらに、コネクタを用いて、アフィニティコンテキストをトランザクションシステムからアプリケーションサーバに転送するステップを含む。

【0135】

本発明の一実施形態に従うと、当該方法はさらに、コネクタを用いて、アプリケーションサーバからアフィニティコンテキストを受信するステップを含む。

【0136】

本発明の一実施形態に従うと、コンピュータプログラムは、システムによって実行されると当該システムに上述の方法のうちの方法を実行させる命令を含む。

【0137】

本発明の一実施形態に従うと、コンピュータプログラムは、システムによって実行されると当該システムに上述の方法のうちの方法を実行させる命令を含む。発明の一実施形態に従うと、トランザクションミドルウェア環境におけるグローバルトランザクションの処理をサポートするためのシステムが提供される。当該システムは、1つ以上のマイクロプロセッサと、1つ以上のマイクロプロセッサ上で実行されるトランザクションシステムとを含む。トランザクションシステムは、リソースマネージャ(RM)インスタンスに接続されたトランザクションサーバに要求をルーティングするステップと、トランザクションサーバが関連付けられているRMインスタンスを示すアフィニティコンテキストをトランザクションサーバに割当てするステップと、アフィニティコンテキストに基づいて、要求に関連する1つ以上の後続の要求をトランザクションサーバにルーティングするステップとを実行するように動作する。

【0138】

本発明の一実施形態に従うと、上述のシステムにおいては、上記RMインスタンスはクラスタ化されたデータベースにおけるインスタンスである。

【0139】

本発明の一実施形態に従うと、上述のシステムにおいては、上記要求および上記1つ以上の後続の要求はグローバルトランザクションに関連付けられている。

【0140】

本発明の一実施形態に従うと、上述のシステムにおいては、トランザクションシステムは、RM名、インスタンス名およびサービス名に基づいているアフィニティルーティングポリシーを用いて1つ以上の後続の要求をルーティングするように動作する。

【0141】

本発明の一実施形態に従うと、上述のシステムにおいては、上記要求および上記1つ以上の後続の要求はクライアントコンテキストに関連付けられている。

【0142】

本発明の一実施形態に従うと、上述のシステムにおいては、トランザクションシステムは、メッセージにアフィニティコンテキストを含めてからメッセージを送信するように動作する。

【0143】

本発明の一実施形態に従うと、上述のシステムにおいては、トランザクションシステムは、別のトランザクションサーバから受信されたメッセージに基づいて、トランザクションサーバにおけるアフィニティコンテキストを更新するように動作する。

【0144】

本発明の一実施形態に従うと、上述のシステムにおいては、トランザクションシステムは、アフィニティコンテキストストリングを用いて、アフィニティコンテキストをトランザクションドメインから別のトランザクションドメインに転送するように動作する。

【0145】

10

20

30

40

50

本発明の一実施形態に従うと、上述のシステムにおいては、トランザクションシステムは、コネクタを用いて、トランザクションシステムからアプリケーションサーバへのアフィニティコンテキストの転送と、アプリケーションサーバからのアフィニティコンテキストの受信とのうち少なくとも1つを実行するように動作する。

【0146】

本発明の一実施形態に従うと、命令が格納された非一時的な機械読取可能な記憶媒体が提供される。当該命令は、実行されると、システムに、リソースマネージャ(RM)インスタンスに接続されたトランザクションサーバに要求をルーティングするステップと、トランザクションサーバが関連付けられているRMインスタンスを示すアフィニティコンテキストをトランザクションサーバに割当てするステップと、アフィニティコンテキストに基づいて、上記要求に関連する1つ以上の後続の要求をトランザクションサーバにルーティングするステップとを実行させる。

10

【0147】

本発明の一実施形態に従うと、トランザクション環境においてトランザクション処理をサポートするための方法が提供される。当該方法は、1つ以上のRMインスタンスに関連付けられたデータソースから、トランザクションサーバを介して、リソースマネージャ(RM)インスタンス情報を受信するステップを含み、受信されたインスタンス情報は、トランザクションサーバが現在接続されているのがどのRMインスタンスであるかを当該トランザクションサーバが検知することを可能にし、当該方法はさらに、受信されたインスタンス情報を、トランザクションサーバに関連付けられた1つ以上のテーブルに保存するステップと、トランザクションサーバが、1つ以上のテーブルに保存されたインスタンス情報に基づいてグローバルトランザクションを処理することを可能にするステップとを含む。

20

【0148】

本発明の一実施形態に従うと、当該方法はさらに、複数のトランザクションサーバによってアクセス可能な共有メモリに上記1つ以上のテーブルを記憶するステップを含む。

【0149】

本発明の一実施形態に従うと、当該方法はさらに、RMにコールバック動作を登録するステップを含む。トランザクションサーバは、コールバック動作がトリガされとき、インスタンス情報をRMから検索するように動作する。

30

【0150】

本発明の一実施形態に従うと、当該方法はさらに、コールバック動作を静的にまたは動的に登録することを可能にするステップを含む。

【0151】

本発明の一実施形態に従うと、当該方法はさらに、受信されたインスタンス情報をコンテキストに記憶するステップと、インスタンス情報の受信を示すフラグを使用可能にするステップとを含む。

【0152】

本発明の一実施形態に従うと、当該方法はさらに、トランザクションを実行している間に1つ以上のチェックポイントにおいてフラグをチェックするステップと、インスタンス情報が更新されたときに、1つ以上のテーブルに保存されたインスタンス情報を更新するステップとを含む。

40

【0153】

本発明の一実施形態に従うと、当該方法はさらに、上記1つ以上のテーブルが、1つ以上の固有のデータベース名を記憶するRMテーブル、1つ以上のインスタンス名を記憶するインスタンステーブル、および、1つ以上のデータベースサービス名を記憶するサービステーブルのうち少なくとも1つを含むことを可能にするステップを含む。

【0154】

本発明の一実施形態に従うと、当該方法はさらに、上記1つ以上のテーブルが、1つ以上のサーバテーブルエントリを含むサーバテーブルを含むことを可能にするステップを含

50

む。

【0155】

本発明の一実施形態に従うと、当該方法はさらに、各々の上記サーバテーブルエントリが、トランザクションサーバが現在接続されているRMインスタンスに関連付けられているインスタンス識別子(ID)を含むことを可能にするステップを含む。

【0156】

本発明の一実施形態に従うと、当該方法はさらに、各々の上記インスタンスIDが複数のセクションを含む整数となることを可能にするステップを含む。

【0157】

本発明の一実施形態に従うと、コンピュータプログラムは、システムによって実行されると当該システムに上述の方法のうちの方法を実行させる命令を含む。

10

【0158】

本発明の一実施形態に従うと、トランザクション環境におけるトランザクション処理をサポートするためのシステムが提供される。当該システムは、1つ以上のマイクロプロセッサと、1つ以上のマイクロプロセッサ上で実行されるトランザクションサーバとを含む。トランザクションサーバは、トランザクションサーバが現在接続されているのがどのRMのインスタンスであるかを当該トランザクションサーバが検知することを可能にするリソースマネージャ(RM)インスタンス情報を1つ以上のRMインスタンスに関連付けられたデータソースから受信するステップと、受信されたインスタンス情報を、トランザクションサーバに関連付けられた1つ以上のテーブルに保存するステップと、1つ以上のテーブルに保存されたインスタンス情報に基づいてグローバルトランザクションを処理するステップとを実行するように動作する。

20

【0159】

本発明の一実施形態に従うと、上述のシステムにおいては、トランザクションサーバは、複数のトランザクションサーバによってアクセス可能な共有メモリに上記1つ以上のテーブルを記憶するように動作する。

【0160】

本発明の一実施形態に従うと、上述のシステムにおいては、トランザクションサーバはRMにコールバック動作を登録するように動作する。トランザクションサーバは、コールバック動作がトリガされたとき、RMからインスタンス情報を検索するように動作する。

30

【0161】

本発明の一実施形態に従うと、上述のシステムにおいては、コールバック動作は、静的にまたは動的に登録されるように適合される。

【0162】

本発明の一実施形態に従うと、上述のシステムにおいては、トランザクションサーバは、受信されたインスタンス情報をコンテキストに記憶し、インスタンス情報の受信を示すフラグを使用可能にするように動作する。

【0163】

本発明の一実施形態に従うと、上述のシステムにおいては、トランザクションサーバは、トランザクションを実行している間に1つ以上のチェックポイントにおいてフラグをチェックし、インスタンス情報が更新されたときに1つ以上のテーブルに保存されたインスタンス情報を更新するように動作する。

40

【0164】

本発明の一実施形態に従うと、上述のシステムにおいては、上記1つ以上のテーブルは、1つ以上の固有のデータベース名を記憶するRMテーブル、1つ以上のインスタンス名を記憶するインスタンステーブル、および、1つ以上のデータベースサービス名を記憶するサービステーブル、のうち少なくとも1つを含む。

【0165】

本発明の一実施形態に従うと、上述のシステムにおいては、上記1つ以上のテーブルは、1つ以上のサーバテーブルエントリを含むサーバテーブルを含む。

50



## 【 0 1 6 6 】

本発明の一実施形態に従うと、上述のシステムにおいては、各々の上記サーバテーブルエントリは、トランザクションサーバが現在接続されている R M インスタンスに関連付けられているインスタンス I D を含み、上記インスタンス I D は、複数のセクションを含む整数である。

## 【 0 1 6 7 】

本発明の一実施形態に従うと、命令が格納された非一時的な機械読取可能な記憶媒体であって、当該命令が実行されると、システムに：トランザクションサーバが現在接続されているのがどの R M インスタンスであるのかを当該トランザクションサーバが検知することを可能にするリソースマネージャ（ R M ）インスタンス情報を、トランザクションサーバを介して、1つ以上の R M インスタンスに関連付けられたデータソースから受信するステップと；トランザクションサーバに関連付けられた1つ以上のテーブルに、受信されたインスタンス情報を保存するステップと；トランザクションサーバが1つ以上のテーブルに保存されたインスタンス情報に基づいてグローバルトランザクションを処理することを可能にするステップと；を含むステップを実行させる。

10

## 【 0 1 6 8 】

本発明の一実施形態に従うと、コンピュータプログラムは、システムによって実行されたときに当該システムに上記方法のうちの方法を実行するための方法を実行させる命令を含む。

## 【 0 1 6 9 】

本発明の多くの特徴は、ハードウェア、ソフトウェア、ファームウェアまたはそれらの組合せにおいて、それらを用いて、またはそれらの支援により、実行可能である。従って、本発明の特徴は、（たとえば、1つ以上のプロセッサを含む）処理システムを用いて実現され得る。

20

## 【 0 1 7 0 】

当業者であれば、上述の特徴が互いに適宜組合わせられ得ること、および／または、所与のいずれかの実現例の特定の環境および要件に従って添付の特許請求の範囲に規定された特徴と適宜組み合わせられ得ることを検知するだろう。

## 【 0 1 7 1 】

この発明の特徴は、ここに提示された特徴のうちのいずれかを行なうように処理システムをプログラミングするために使用可能な命令を格納した記憶媒体またはコンピュータ読取可能媒体であるコンピュータプログラム製品において、それを使用して、またはその助けを借りて実現され得る。記憶媒体は、フロッピー（登録商標）ディスク、光ディスク、DVD、CD-ROM、マイクロドライブ、および光磁気ディスクを含む任意のタイプのディスク、ROM、RAM、EPROM、EEPROM、DRAM、VRAM、フラッシュメモリ装置、磁気カードもしくは光カード、ナノシステム（分子メモリICを含む）、または、命令および／もしくはデータを格納するのに好適な任意のタイプの媒体もしくは装置を含み得るものの、それらに限定されない。

30

## 【 0 1 7 2 】

この発明の特徴は、機械読取可能媒体のうちのいずれかに格納された状態で、処理システムのハードウェアを制御するために、および処理システムがこの発明の結果を利用する他の機構とやり取りすることを可能にするために、ソフトウェアおよび／またはファームウェアに取込まれ得る。そのようなソフトウェアまたはファームウェアは、アプリケーションコード、装置ドライバ、オペレーティングシステム、および実行環境／コンテナを含み得るものの、それらに限定されない。

40

## 【 0 1 7 3 】

この発明の特徴はまた、たとえば、特定用途向け集積回路（application specific integrated circuit：ASIC）などのハードウェアコンポーネントを使用して、ハードウェアにおいて実現されてもよい。ここに説明された機能を行なうようにハードウェアステートマシンを実現することは、関連技術の当業者には明らかであろう。

50

## 【 0 1 7 4 】

加えて、この発明は、この開示の教示に従ってプログラミングされた1つ以上のプロセッサ、メモリおよび/またはコンピュータ読取可能記憶媒体を含む、1つ以上の従来の汎用または特殊デジタルコンピュータ、コンピューティング装置、マシン、またはマイクロプロセッサを使用して都合よく実現されてもよい。ソフトウェア技術の当業者には明らかであるように、この開示の教示に基づいて、適切なソフトウェアコーディングが、熟練したプログラマによって容易に準備され得る。

## 【 0 1 7 5 】

この発明のさまざまな実施形態が上述されてきたが、それらは限定のためではなく例示のために提示されたことが理解されるべきである。この発明の精神および範囲から逸脱することなく、形状および詳細のさまざまな変更を行なうことができることは、関連技術の当業者には明らかであろう。

10

## 【 0 1 7 6 】

この発明は、特定された機能およびそれらの関係の実行を示す機能的構築ブロックの助けを借りて上述されてきた。説明の便宜上、これらの機能的構築ブロックの境界は、この明細書中ではしばしば任意に規定されてきた。特定された機能およびそれらの関係が適切に実行される限り、代替的な境界を規定することができる。このため、そのようないかなる代替的な境界も、この発明の範囲および精神に含まれる。

## 【 0 1 7 7 】

この発明の前述の説明は、例示および説明のために提供されてきた。それは、網羅的であるよう、またはこの発明を開示された形態そのものに限定するよう意図されてはいない。この発明の幅および範囲は、上述の例示的な実施形態のいずれによっても限定されるべきでない。多くの変更および変形が、当業者には明らかになるだろう。これらの変更および変形は、開示された特徴の関連するあらゆる組合せを含む。実施形態は、この発明の原理およびその実用的応用を最良に説明するために選択され説明されたものであり、それにより、考えられる特定の使用に適したさまざまな実施形態についての、およびさまざまな変更例を有するこの発明を、当業者が理解できるようにする。この発明の範囲は、請求項およびそれらの同等例によって定義されるよう意図されている。

20

【図 1】

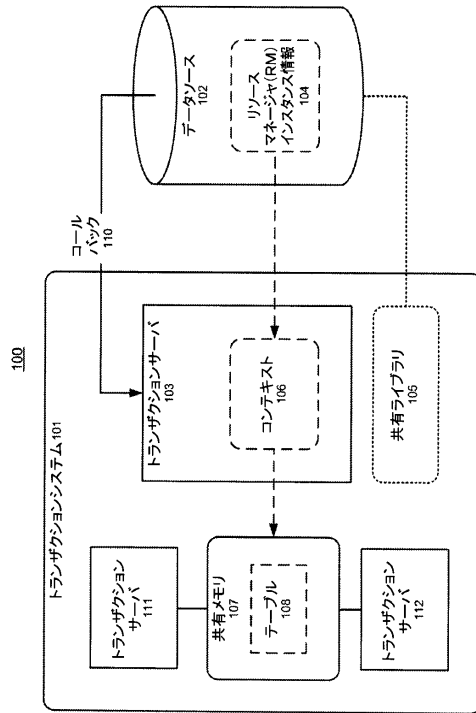


FIGURE 1

【図 2】

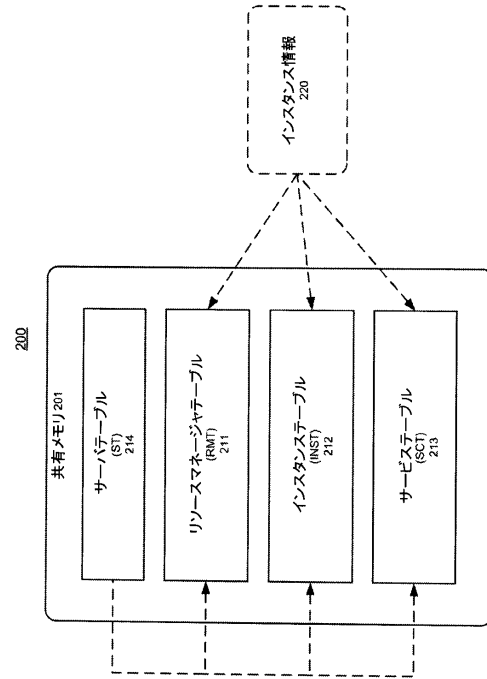


FIGURE 2

【図 3】

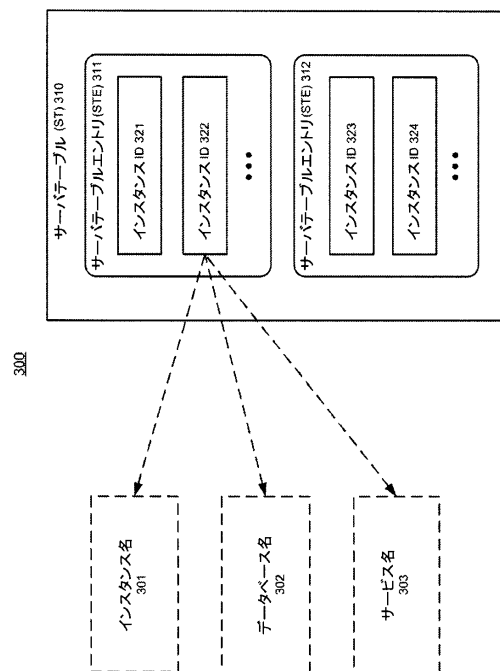


FIGURE 3

【図 4】

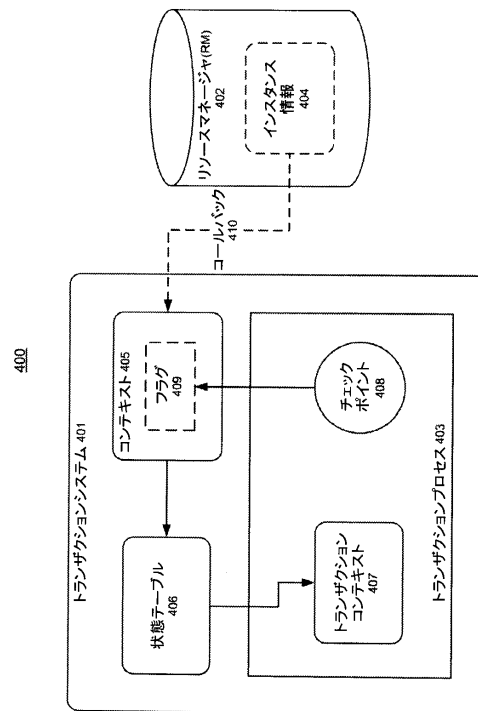
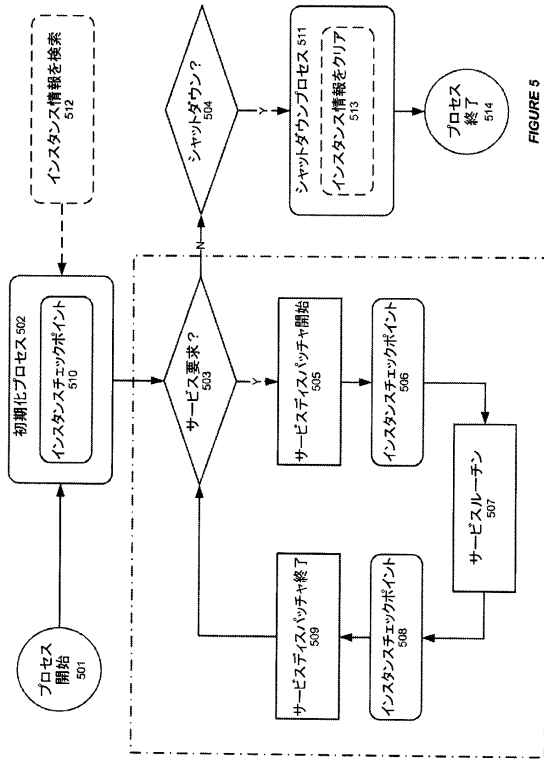
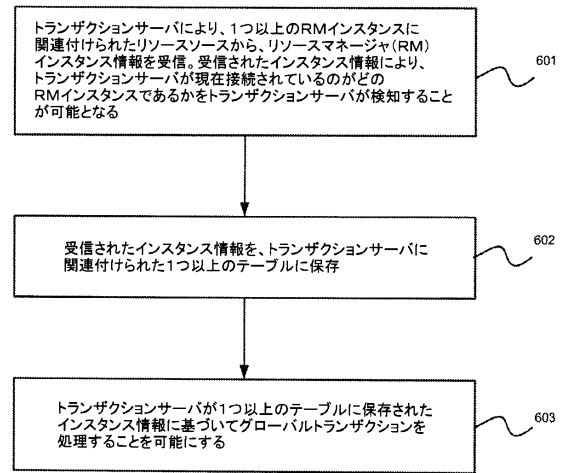


FIGURE 4

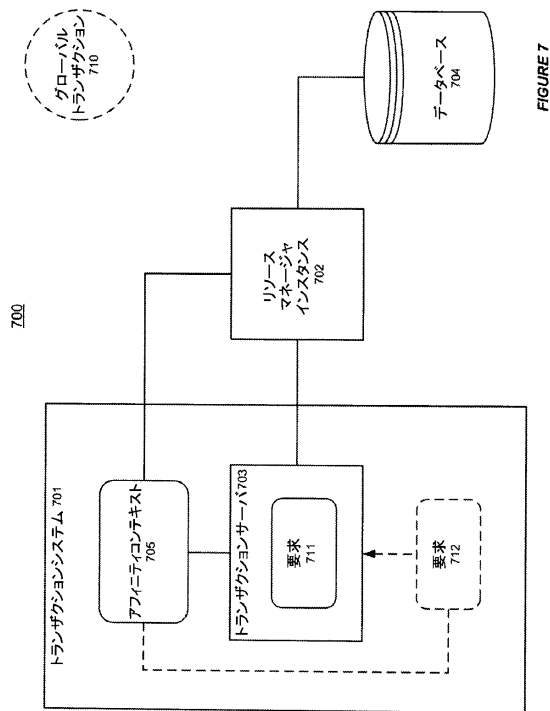
【図 5】



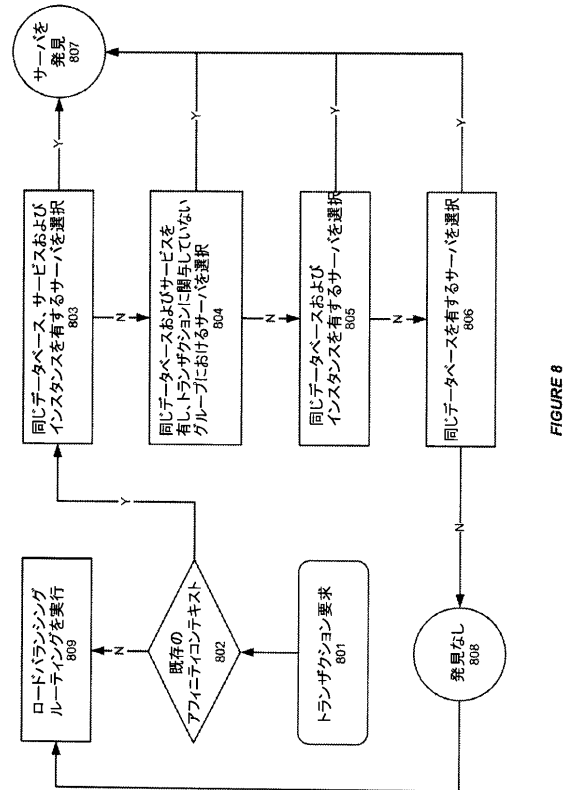
【図 6】



【図 7】



【図 8】



【図 9】

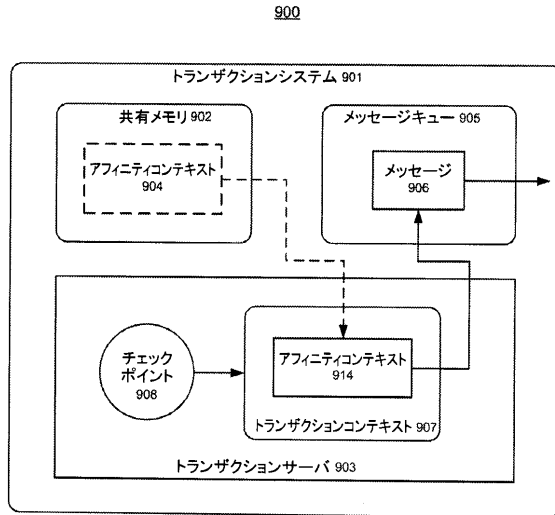


FIGURE 9

【図 10】

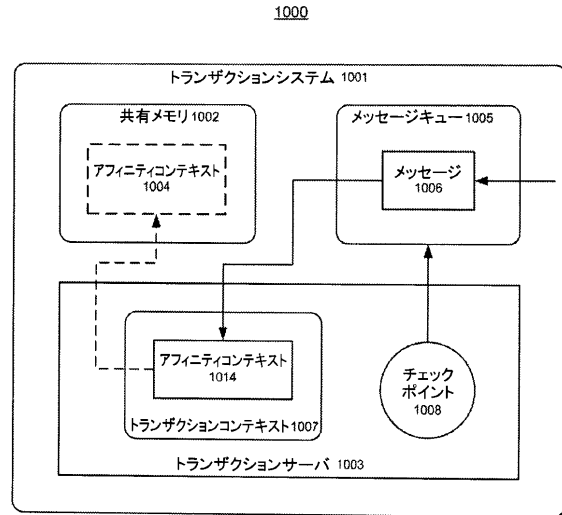


FIGURE 10

【図 11】

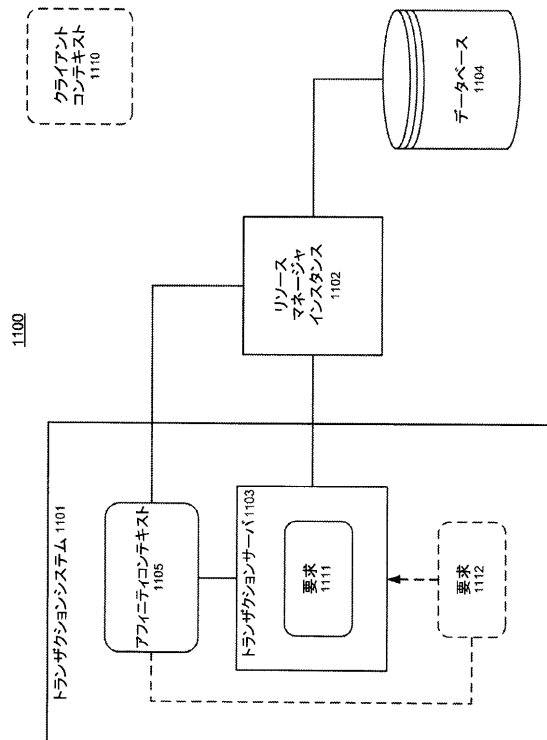


FIGURE 11

【図 12】

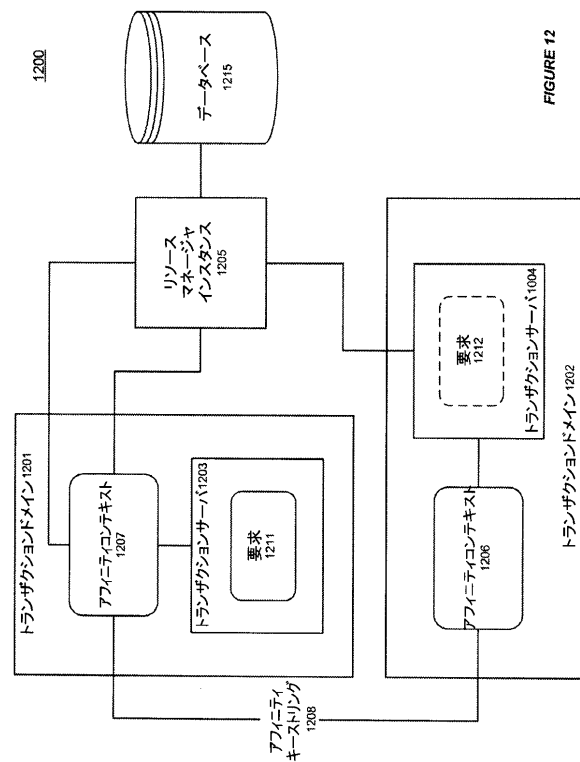


FIGURE 12

【図 13】

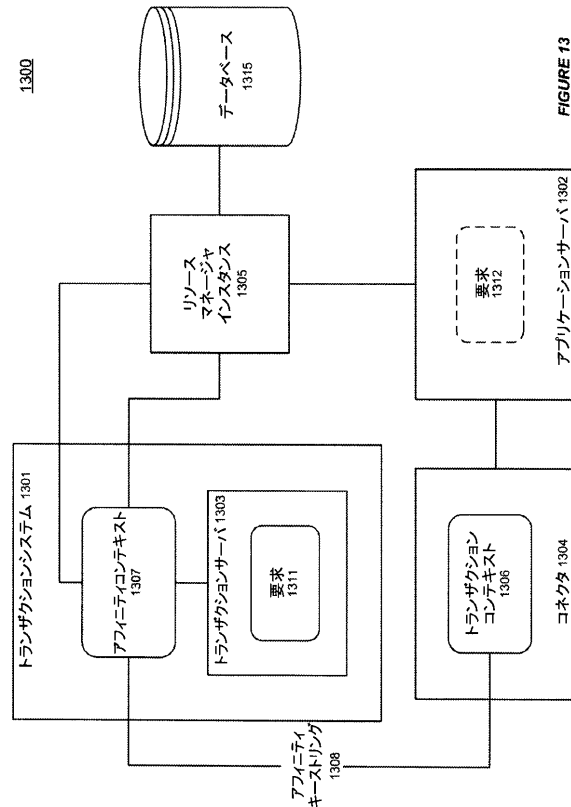


FIGURE 13

【図 14】

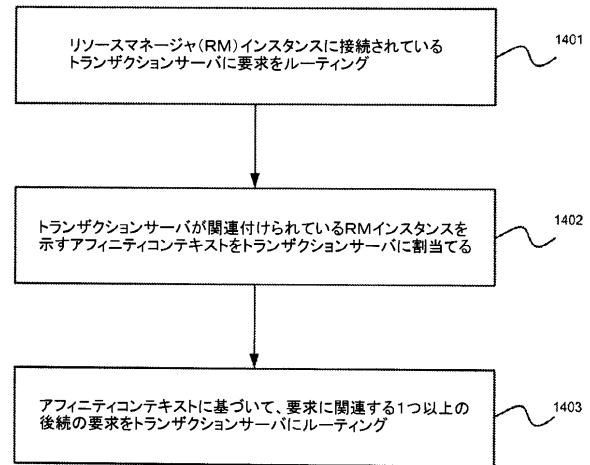


FIGURE 14

【図 15】

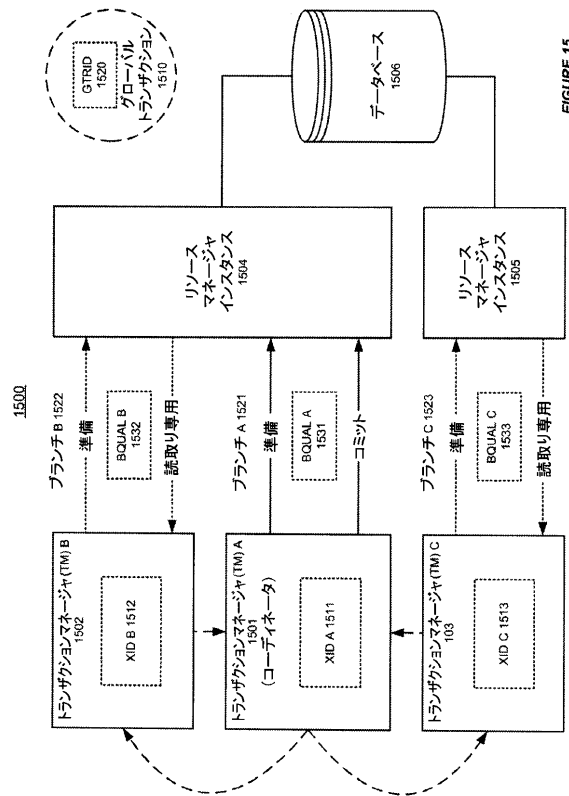


FIGURE 15

【図 16】

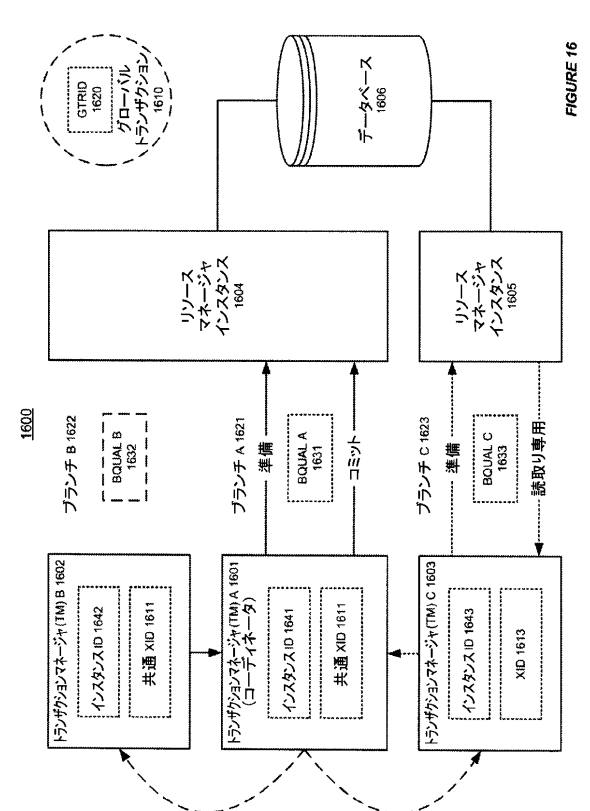


FIGURE 16

【図 17】

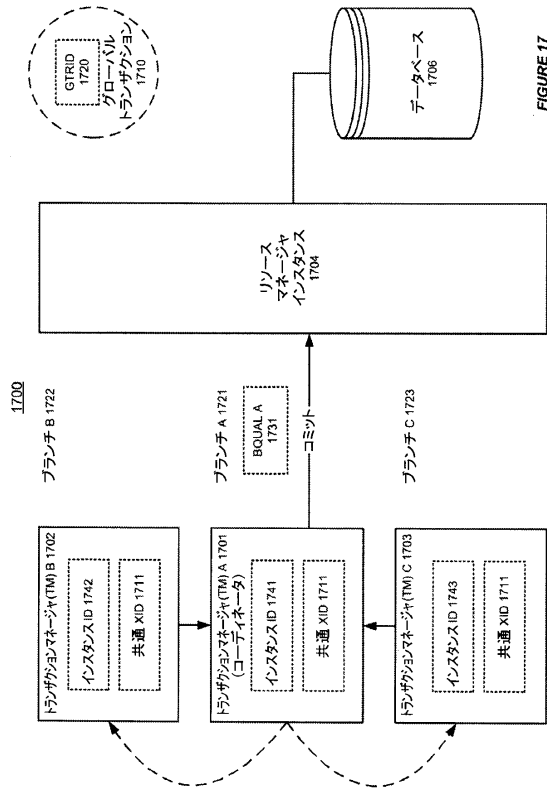


FIGURE 17

【図 18】

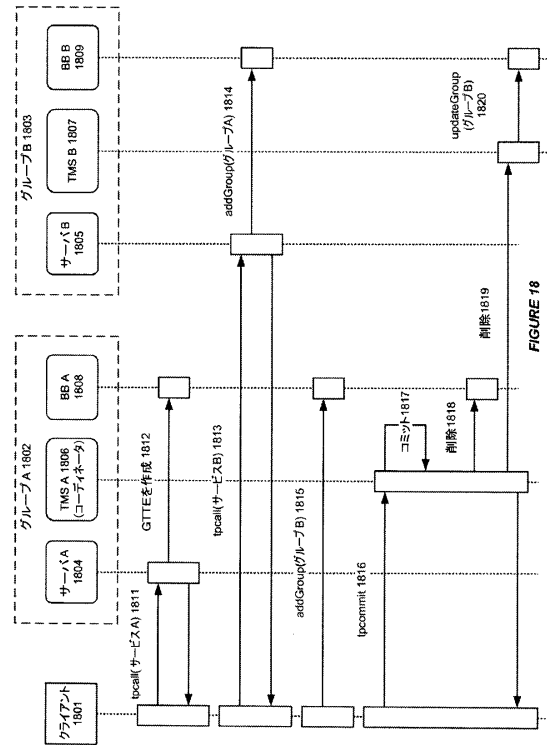


FIGURE 18

【図 19】

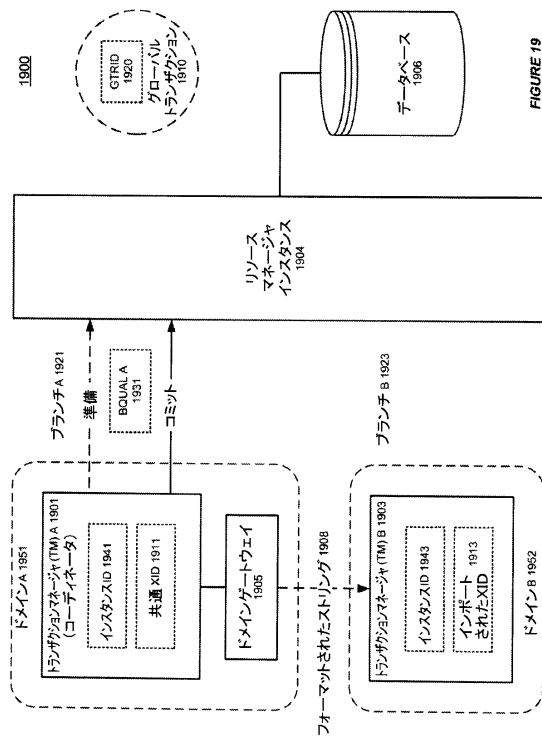


FIGURE 19

【図 20】

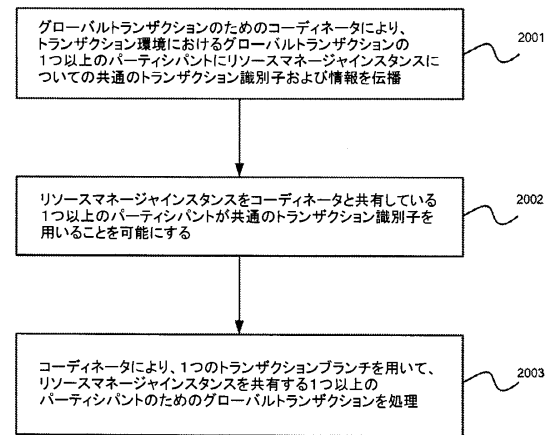


FIGURE 20

## フロントページの続き

(31)優先権主張番号 14/587,474

(32)優先日 平成26年12月31日(2014.12.31)

(33)優先権主張国 米国(US)

(31)優先権主張番号 14/587,476

(32)優先日 平成26年12月31日(2014.12.31)

(33)優先権主張国 米国(US)

(72)発明者 チャン, チンシェン

中華人民共和国、100193 ベイジン、ハイディアン・ディストリクト、チョングアンツン・ソフトウェア・パーク、ビルディング・ナンバー・24、オラクル・ビルディング

(72)発明者 リトル, トッド・ジェイ

アメリカ合衆国、60067-6675 イリノイ州、パラタイン、ウエスト・イリノイ・アベニュー、1155

(72)発明者 ジン, ヨンシュン

中華人民共和国、100193 ベイジン、ハイディアン・ディストリクト、チョングアンツン・ソフトウェア・パーク、ビルディング・ナンバー・24、オラクル・ビルディング

審査官 大桃 由紀雄

(56)参考文献 特開2004-295272(JP, A)

特開2007-011550(JP, A)

国際公開第2013/138770(WO, A1)

(58)調査した分野(Int.Cl., DB名)

G06F 16/2458

G06F 16/182

(54)【発明の名称】トランザクション環境におけるリソースマネージャ(RM)インスタンス検知に基づいた共通のトランザクション識別子(XID)最適化およびトランザクションアフィニティをサポートするためのシステムおよび方法