

(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 1

G06F 16/2458 (2019.01)
G06F 16/182 (2019.01)G06F 16/2458
G06F 16/182

請求項の数 11 (全 32 頁)

(21) 出願番号 特願2016-565068 (P2016-565068)
 (86) (22) 出願日 平成27年3月27日 (2015.3.27)
 (65) 公表番号 特表2017-514239 (P2017-514239A)
 (43) 公表日 平成29年6月1日 (2017.6.1)
 (86) 國際出願番号 PCT/US2015/023120
 (87) 國際公開番号 WO2015/167724
 (87) 國際公開日 平成27年11月5日 (2015.11.5)
 審査請求日 平成30年1月24日 (2018.1.24)
 (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)

(73) 特許権者 502303739
 オラクル・インターナショナル・コーポレーション
 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・パーカウェイ500
 (74) 代理人 110001195
 特許業務法人深見特許事務所
 (72) 発明者 シエン、シュガン
 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、ジョングアンツン・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

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(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

(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ー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ー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ー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に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プランチを無視して、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プ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プランチを共有し、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プランチを用いて、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プランチを無視して、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対して異議はないが、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プランチを用いて、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。

【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。

【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。

【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。

【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

10

20

30

40

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13】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14】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15】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 : 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である。

【図16】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（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である。 10

【図17】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 : 1PC）処理モデルをサポートする例を示す図である。

【図18】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を処理する例を示す図である。

【図19】本発明の一実施形態に従った、共通のXID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を処理する例を示す図である。 20

【図20】本発明の一実施形態に従った、共通の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。 30

【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Java（登録商標）EE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WebLogic（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（IB）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オ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、WebLogic Server、RockitもしくはHotspot JV 40

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。

10

【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、従来のシェアド・ナッシング・アプローチおよびシェアド・ディスク・アプローチの制限事項を克服して、ビジネス用途で使用できるよう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。

20

【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してきた。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。

30

【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 ATMIAPIと、X/Open Portability Guide(XPG)基準についてのサポートを含む、オープングループの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。

40

50

【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が呼び出しが可能となる。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 211、RM / 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 212、および RM / データベースサービス名を記憶するサービステーブル 213 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。

【0023】

加えて、トランザクションシステム 200 は共有メモリ 201 におけるサーバテーブル (S T) 214 を維持することができる。S T 214 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 212 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

【0024】

図 2 に示されるように、サーバテーブル 214 は、共有メモリ 201 内の他のテーブル 211 - 213 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 200 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 RM 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 214 を用いることができ、トランザクションシステム 200 は、さまざまな状態テーブル 211 - 213 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。

【0025】

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 300 におけるサーバテーブル (S T) 310 は、1つ以上のサーバテーブルエントリ (S T E) 311 - 312 を含み得る。サーバテーブルエントリ (S T E) 311 - 312 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 311 はインスタンス I D 321 - 322 を含み得る。S T E 312 はインスタンス I D 323 - 324 を含み得る。20

【0026】

本発明の一実施形態に従うと、各々のインスタンス I D 321 - 324 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 322 はインスタンス名 301、データベース名 302 およびサービス名 303 を識別することができる。30

【0027】

たとえば、インスタンス I D は、整数（たとえば、ビット 0 - 11、ビット 12 - 19 およびビット 20 - 31 という3つのセクションを含む 32 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 - 11 は、RM / データベースインスタンス名 301 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 12 - 19 は、RM / データベース名 302 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 20 - 31 は、RM / データベースサービス名 303 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 × F F F F F F F F を用いることができる。40

【0028】

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 300 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 322 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。

【0029】

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 400 におけるトランザクションシステム 401 は、たとえばデータベースにおけるリソースマネージャ (リソースマネージャ) によるリソースの割り当てや解放操作を実行する。50

R M) 4 0 2 から、関連するインスタンス情報 4 0 4 を取得することができる。

【 0 0 3 0 】

ユーザコールバック 4 1 0 が呼出されると、最新のインスタンス情報 4 0 4 を R M 4 0 2 から検索することができ、コンテキスト 4 0 5 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 4 0 1 は、最新のインスタンス情報 4 0 4 を受信したことを示すフラグ 4 0 9 を設定することができる。

【 0 0 3 1 】

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 4 0 3 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、サービス呼出しの前後、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

【 0 0 3 2 】

図 4 に示されるように、チェックポイント 4 0 8 において、トランザクションプロセス 4 0 3 がフラグ 4 0 9 をチェックすることができる。フラグ 4 0 9 が上がっていれば、トランザクションサーバ 4 0 3 は、検索されたインスタンス情報 4 0 4 に基づいてトランザクションコンテキスト 4 0 7 を更新し、検索されたインスタンス情報 4 0 4 を（共有メモリにおける）状態テーブル 4 0 6 に記憶することができる。

【 0 0 3 3 】

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 5 0 1 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 5 0 2 において初期化プロセスに進み得る。
20

【 0 0 3 4 】

初期化プロセス 5 0 2 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 5 1 0 、を含み得る。たとえば、T u x e d o 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 5 1 0 は、初期化ルーチン（たとえば、t p s v r i n i t ）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 R M への接続を確立しようと試みる場合（たとえば、x a _ o p e n () 関数呼出しが、t p o p e n () 関数呼出しでの呼出しに成功した後）、チェックポイント 5 1 0 をトリガすることができる。
30

【 0 0 3 5 】

加えて、ステップ 5 1 2 において、トランザクションプロセスは、初期化プロセス 5 0 2 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。

【 0 0 3 6 】

さらに、ステップ 5 0 3 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 5 0 5 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、t m s v c d s p () 、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 5 0 7 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 5 0 6 をトリガすることができる。加えて、サービスルーチン 5 0 7 が完了した後、別のチェックポイント 5 0 8 をトリガすることができる。ステップ 5 0 9 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

【 0 0 3 7 】

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 5 0 4 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 5 1 1 を開始することができる。シャットダウンプロセス 5 1 1 では、チェックポイント 5 1 3 をトリガしてインスタンス I D 情報をクリーンにすることができる。たとえば、T u x e d o において、サーバが R M への接続を閉鎖しようとする場合（たとえば、x a _ c l o s e () 関数呼出しが t p c l o s e () 関数呼出して呼出される前）に、チェック

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`tpsverinit()` 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`tpsverdone()` 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、ト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、ト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、グローバルトランザクション 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。

10

【0045】

1. ドメインのためのト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. ト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、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 8 に示されるように、トランザクションシステムは、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。

【0046】

20

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ト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】

グローバルト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】

グローバルト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することができる。

30

【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 (data base : 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ト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】

40

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】

ト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、

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。

【0062】

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1103に対してデータベース接続についての要求1111をルーティングすることができる。加えて、システムは、RM1102を示すアフィニティコンテキスト1105をトランザクションサーバ1103に割当てることができる。

【0063】

さらに、クライアントコンテキスト1110内の1つ以上の後続の要求（たとえば、要求1112）は、クライアントコンテキスト1110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1105に基づいてトランザクションサーバ1103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1101は、クライアントコンテキスト1110内のさまざまなデータベース動作を同じRMインスタンス1102に方向付けることを確実にすることができます。10

【0064】

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1101は、クライアントコンテキスト1110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マップ（アフィニティ）をデータベースから受信することができる。たとえば、Tuxedo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AFFINITY_HINT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。AFFINITY_HINT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、AFFINITY_HINTについてのさまざまな設定を有し得る。20

【0065】

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1101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Tuxedo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。30

【0066】

図12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12に示されるように、トランザクション環境1200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1215に関連付けられたRMインスタンス1205、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。40

【0067】

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1203に対してデータベース接続についての要求1211をルーティングすることができる。加えて、システムは、RMインスタンス1202を示すアフィニティコンテキスト1207をトランザクションサーバ1203に割当てることができる。

【0068】

さらに、トランザクション環境1200におけるトランザクションドメインは、要求1211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。50

【 0 0 6 9 】

図12に示されるように、トランザクションドメイン1201は、アフィニティコンテキスト1207をアフィニティキースtring1208に変換してから、このアフィニティキースtring1208をリモートドメイン1202に送信することができる。アフィニティキースtring1208を受信した後、トランザクションドメイン1202は、アフィニティキースtring1208を、トランザクションドメイン1202においてトランザクションサーバ1004が使用可能なアフィニティコンテキスト1206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1212）をアフィニティコンテキスト1206に基づいてRMインスタンス1202に方向付けることができる。 10

【 0 0 7 1 】

図13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13に示されるように、トランザクション環境1300におけるトランザクションシステム1301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1315に関連付けられたRMインスタンス1305）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1303に対してデータベース接続についての要求1311をルーティングすることができる。加えて、システムは、RMインスタンス1305を示すアフィニティコンテキスト1307をトランザクションサーバ1303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1301（たとえば、Tuxedo TDomain）は、アフィニティキースtring1308を介して、アプリケーションサーバ1302（たとえばWebLogicアプリケーションサーバ）にアフィニティコンテキスト1307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1307は、コネクタ1304（たとえばTuxedo WTC）を介してトランザクションシステム1301とアプリケーションサーバ1302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1301がコネクタ1304に要求を送信すると、トランザクションシステム1301はアフィニティコンテキスト1307をアフィニティキースtring1308に変換することができる。コネクタ1304がトランザクションシステム1301から要求を受信すると、コネクタ1306は、アフィニティキースtring1308を、アプリケーションサーバ1302が使用可能なトランザクションコンテキスト1306に変換することができる。

【 0 0 7 5 】

コネクタ1304がトランザクションシステム1301に要求を送信すると、コネクタ1304は、アプリケーションサーバ1302に関連付けられたトランザクションコンテキスト1306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1301がコネクタ1304から要求を受信すると、トランザクションシステム1301は、アフィニティキースtring1308をアフィニティコンテキスト1307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1312における1つ以上の後続の要求（たとえば要求1312）を、アフィニティコンテキスト1305に基づいてRMインスタンス1302に方向付けることができる。

【 0 0 7 7 】

図14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、さまざまなりソースマネージャ(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: GTRID1520)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プ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プ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プランチ(たとえばプランチA1512-C1523)を用いてもよい。プランチA1521-C1523の各々をプランチ修飾子(たとえばBQUAL A1531-C1533)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、さまざまなプランチA1521-C1523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 A1501-TM C1503)を用いることができる。

30

【0082】

たとえば、TM A1501はトランザクション識別子(XID)A1511に関連付けられており、プランチA1521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 B1502はトランザクション識別子(XID)B1512に関連付けられており、プランチB1522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 C1503は、トランザクション識別子(XID)C1513に関連付けられており、プランチC1523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プランチA1521-C1523のためのXID A1511-XID C1513は、同じGTRID1520(およびフォーマットID)を共有することができ、かつ、さまざまなプランチ修飾子(すなわちBQUAL A1531-BQUAL C1533)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 1510 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 15 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 1510 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、TM A1501 - TM B1502）が、実際に同じ RM インスタンス 1504 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 1510 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 1504 上で実行されている場合に、共通の XID を用いることによってグローバルトランザクション 1510 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 16 は、本発明の一実施形態に従った、共通の XID 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 16 に示されるように、トランザクションシステム 1600 は、データベース 1606 に接続するリソースマネージャ (RM) インスタンス 1604 - 1605 を用いて、GRRID1620 に関連付けられたグローバルトランザクション 1610 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 1600 は、さまざまなトランザクショングループ（たとえば、プランチ A1621 - C1623）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 1610 の処理を管理するためにトランザクションマネージャ (TM) A1601 - C1603 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 (RM) 1604 - 1605 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 1600 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 RM 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 16 に示されるように、トランザクションシステム 1600 は TM A1601 などのコーディネータを含み得る。コーディネータ TM A1601 は、RM インスタンス 1604 を識別するインスタンス ID1641 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 1610 は、コーディネータ 1601 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 TM B1602 - TM C1603）を含み得る。パーティシパント TM B1602 - TM C1603 の各々はまた、それら各自が接続している RM インスタンスを識別するインスタンス ID（たとえば、インスタンス ID1642 - ID1643）に関連付けることもできる。

【 0 0 9 2 】

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 1610 を処理するために共通の XID 特徴が使用可能になると、コーディネータ TM A1601 についての XID1611 はグローバルトランザクション 1610 内で共有可能となる（すなわち、XID1611 が共通の XID として用いられる）。これにより、複数のグループを含み、Oracle 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。 40

【 0 0 9 3 】

図 16 に示されるように、グローバルトランザクション 1610 のためのコーディネータ TM A1601 は、グローバルトランザクション 1610 のライフサイクル内で、X 50

I D 1 6 1 1 およびインスタンス I D 1 6 4 1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 T M B 1 6 0 2 - T M C 1 6 0 3に伝播させることができる。

【 0 0 9 4 】

さらに、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、受信されたインスタンス I D 1 6 4 1をそれ自体のインスタンス I Dと比較することによって、それがコーディネータ T M A 1 6 0 1と同じ R M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 I Dが同じであれば、パーティシパント T M B 1 6 0 2 - T M C 1 6 0 3はそれら自体を共通の X I D 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。

【 0 0 9 5 】

たとえば、システムは、T M B 1 6 0 2がT M A 1 6 0 1と同じ R Mインスタンス 1 6 0 4を共有しているので、プランチ B 1 6 2 2上に一致を発見する可能性がある。これにより、T M B 1 6 0 2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 X I Dではなく共通の X I D 1 6 1 1を用いることができる。次いで、T M B 1 6 0 2は、それが共通の X I D 1 6 1 1を用いていることをコーディネータ T M A 1 6 0 1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 T M A 1 6 0 1がグローバルトランザクション 1 6 1 0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、(B Q U A L A 1 6 3 1に基づいた)共通の X I D 1 6 1 1を用いているので、プランチ B 1 6 2 2を無視することができる。

【 0 0 9 6 】

他方では、T M C 1 6 0 3が(インスタンス I D 1 6 4 3を用いて)別の R M 1 6 0 5に関連付けられているので、システムは、(X I D C 1 6 1 3およびB Q U A L C 1 6 3 3を用いる)プランチ C 1 6 2 3上で一致を発見しない可能性がある。任意には、T M C 1 6 0 3は、それが共通の X I D 1 6 1 1用いていないことをコーディネータ T M A 1 6 0 1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(2 P C)処理モデルに従ってトランザクションプランチ C 1 6 2 3を処理することができる。

【 0 0 9 7 】

本発明の一実施形態に従うと、さまざまな共通の X I D グループ、すなわち、コーディネータ 1 6 0 1と同じ R Mインスタンス 1 6 0 4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 X I D 1 6 1 1を介して R Mインスタンス 1 6 0 4にアクセスすることができる。

【 0 0 9 8 】

さらに、コミット要求が呼出されると、コーディネータ 1 6 0 1はローカルの共通 X I D 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 X I D グループの状態を同時に読み取り専用に変更することができる。また、リモートの共通 X I D グループは、コーディネータ 1 6 0 1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、その状態を読み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。

【 0 0 9 9 】

加えて、システムは、インスタンス I Dが変化した場合に、共通 X I D グループを非共通 X I D グループに変更することができる。たとえば、プランチ B 1 6 2 2が別の R M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、B Q U A L B 1 6 3 2に基づいて)2相コミット(2 P C)プロセスを呼出すことができる。

【 0 1 0 0 】

図 1 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(1 P C)処理モデルをサポートする例を示す。図 1 7 に示されるように、トランザクションシステム 1 7 0 0 は、データベース 1 7 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 7 0 4を用いて、G R T I D

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)A 1701-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)A 1701-C1703は、さまざまなトランザクショングループ(すなわち、プランチA 1721-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) 10 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。 20

【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。 30

【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、読み取り専用の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み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プランチがすべて、たとえば同じインスタンスまたは同じRACT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。 40

【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 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。

【 0 1 0 9 】

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)。

【 0 1 1 0 】

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プ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プランチは 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知れず、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知れない。10

【 0 1 1 1 】

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。20

【 0 1 1 2 】

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。

【 0 1 1 3 】

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 : GTTE)を作成することができる。30

【 0 1 1 4 】

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。40

【 0 1 1 5 】

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 tpcommitt()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。50

【 0 1 1 6 】

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、そのプランチを削除 (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 Tuxedo BB B 1809を更新することができる。

【 0 1 1 7 】

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

【 0 1 1 8 】

加えて、さまざまなプランチ 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

【 0 1 1 9 】

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。

【 0 1 2 0 】

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

【 0 1 2 1 】

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。

【 0 1 2 2 】

たとえば、Tuxedoにおいて、ユーザは、ビジネス上の理由から、Tuxedo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

【 0 1 2 3 】

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に關

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、プランチ 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ー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ートが、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プランチを用いて、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ー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。

【 0 1 5 5 】

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。

【 0 1 5 6 】

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。

【 0 1 5 7 】

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

【 0 1 5 8 】

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

【 0 1 5 9 】

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。

【 0 1 6 0 】

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

【 0 1 6 1 】

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。

【 0 1 6 2 】

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。

【 0 1 6 3 】

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

【 0 1 6 4 】

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。

【 0 1 6 5 】

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RMインスタンスに関連付けられているインスタンスIDを含み、上記インスタンスID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RM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(RM)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；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。

【 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。

【 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さ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、それらに限定されない。

【 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、それらに限定されない。

【 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トーム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。

10

20

30

40

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 】

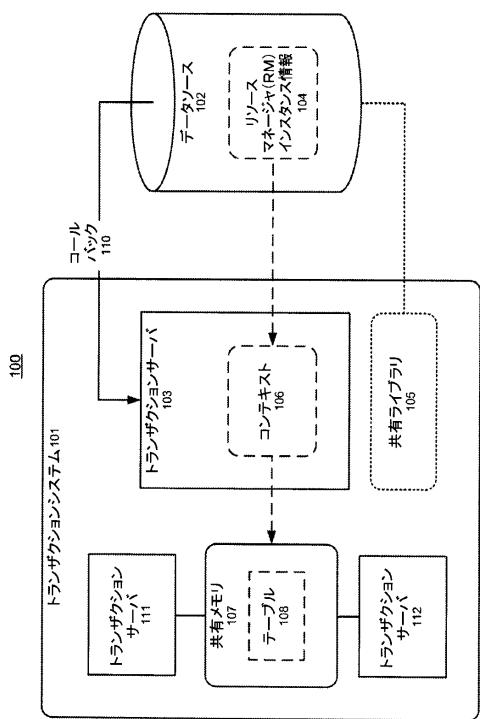
この発明は、特定された機能およびそれらの関係の実行を示す機能的構築ブロックの助けを借りて上述されてきた。説明の便宜上、これらの機能的構築ブロックの境界は、この明細書中ではしばしば任意に規定されてきた。特定された機能およびそれらの関係が適切に実行される限り、代替的な境界を規定することができる。このため、そのようないかなる代替的な境界も、この発明の範囲および精神に含まれる。

【 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】



【図2】

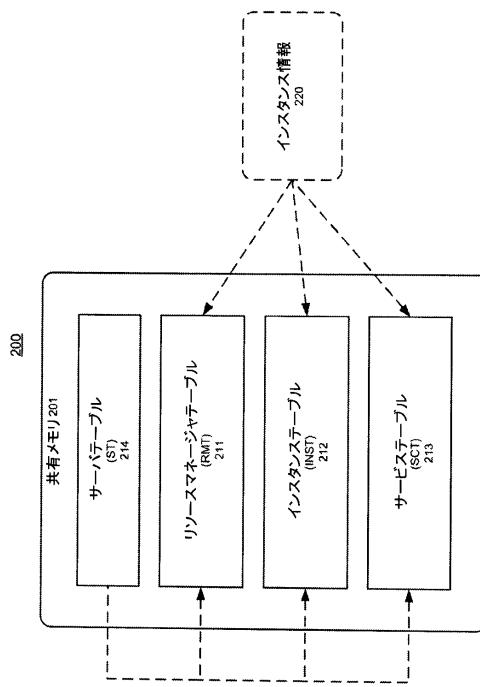
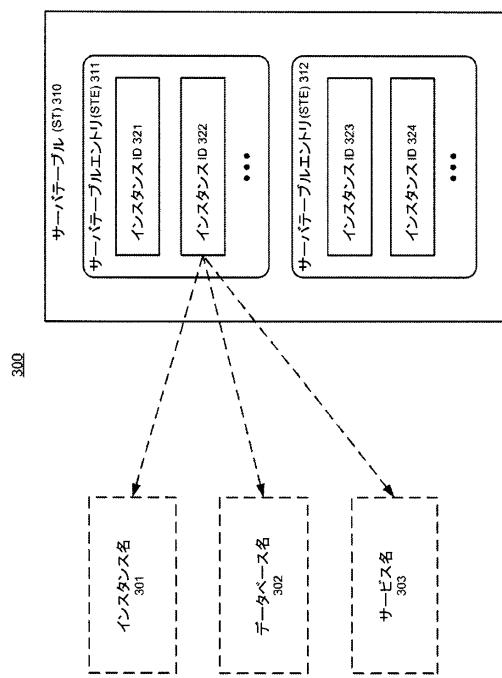


FIGURE 1

FIGURE 2

【図3】



【図4】

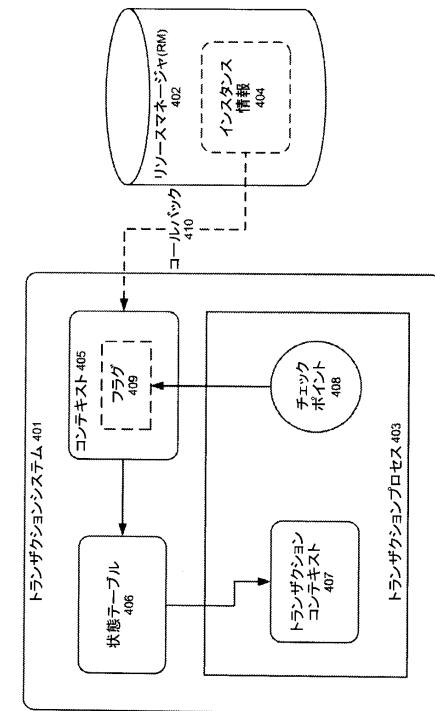


FIGURE 3

FIGURE 4

【図5】

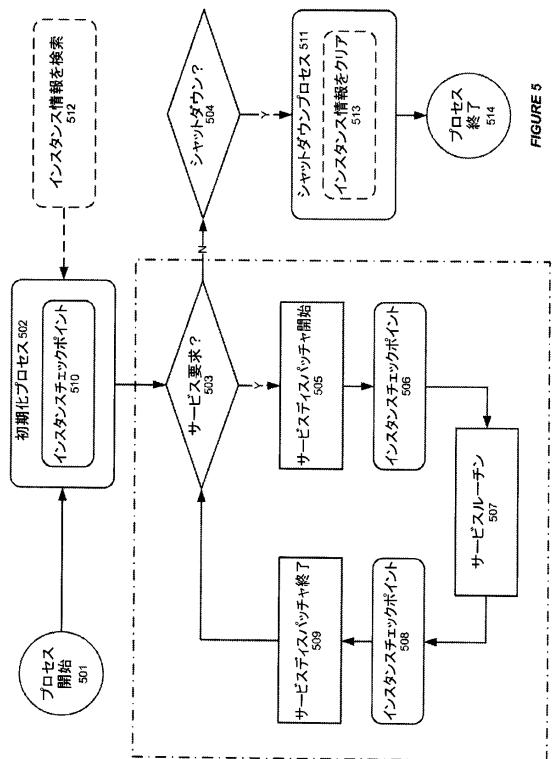


FIGURE 5

【図6】

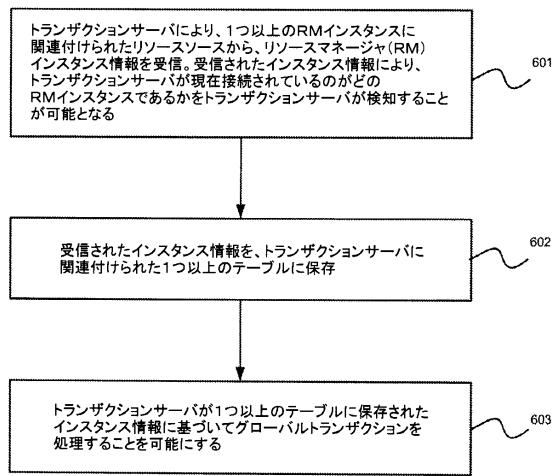


FIGURE 6

【図7】

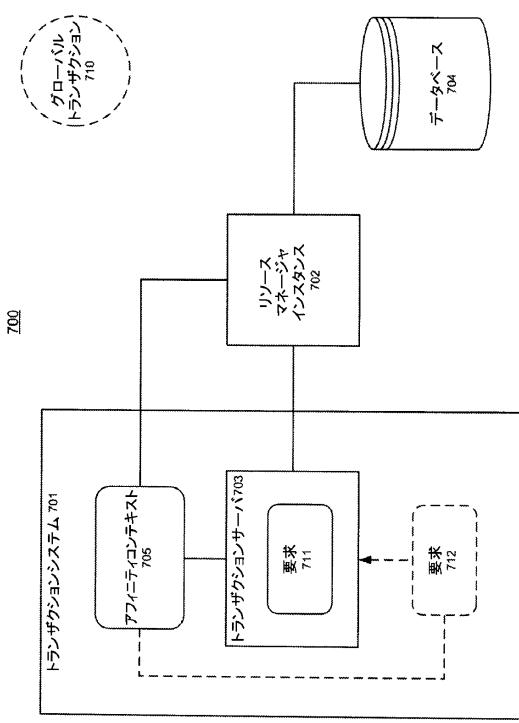


FIGURE 7

【図8】

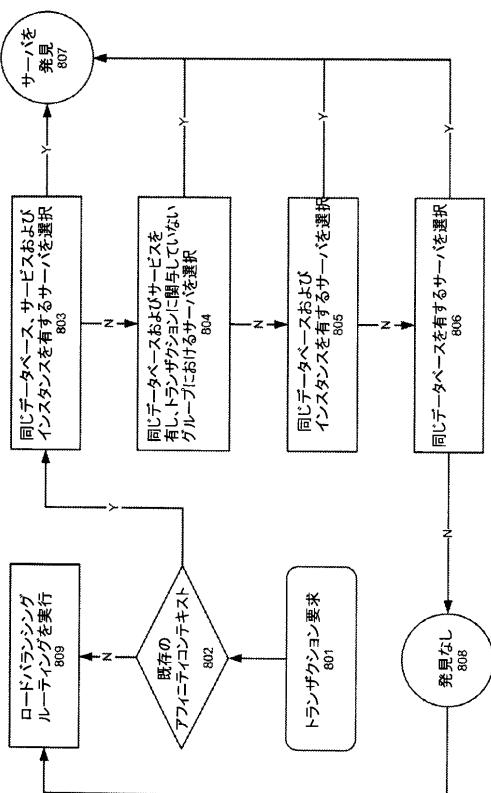
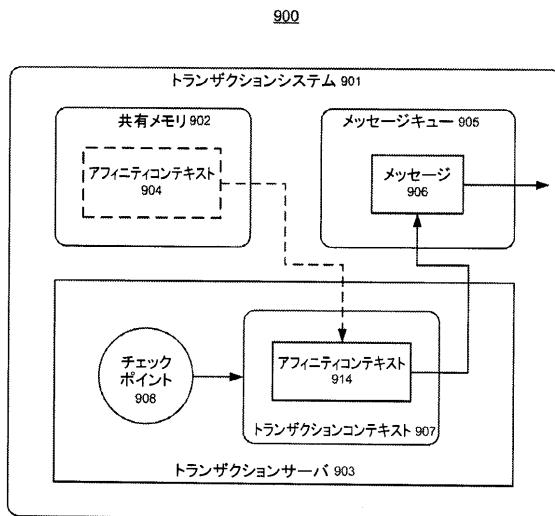
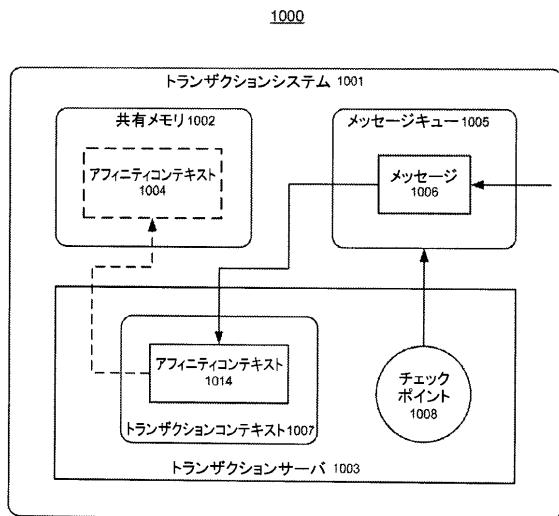


FIGURE 8

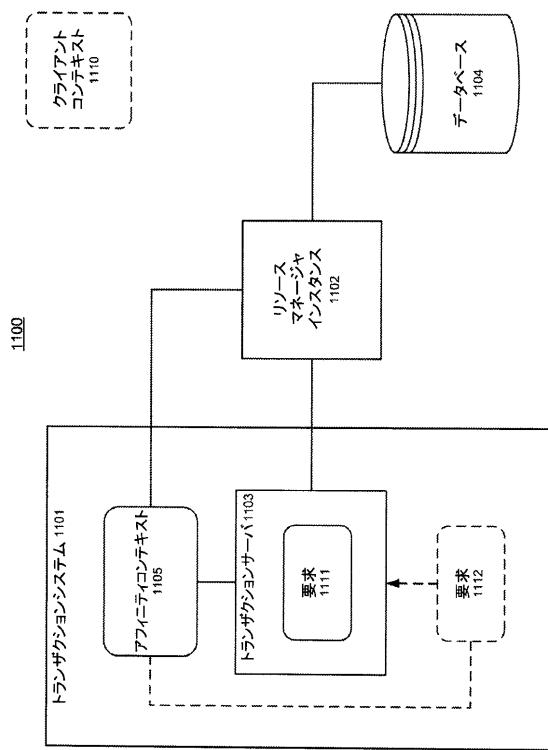
【図9】



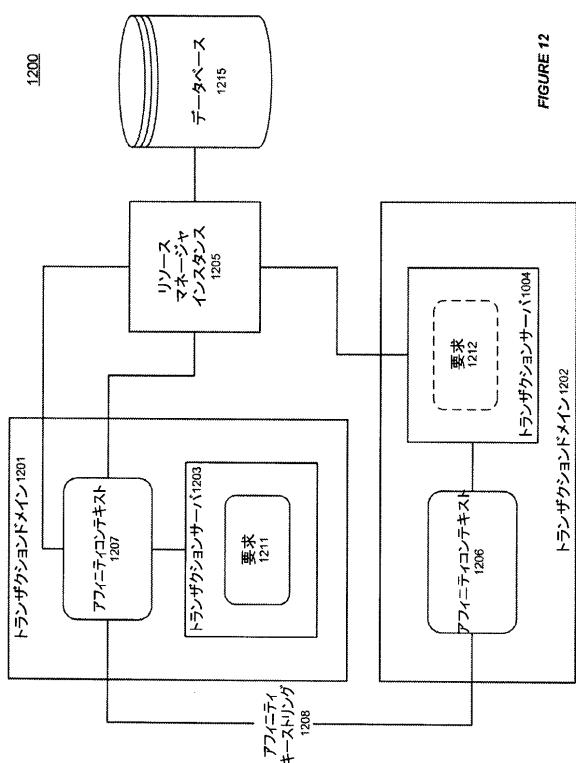
【図10】



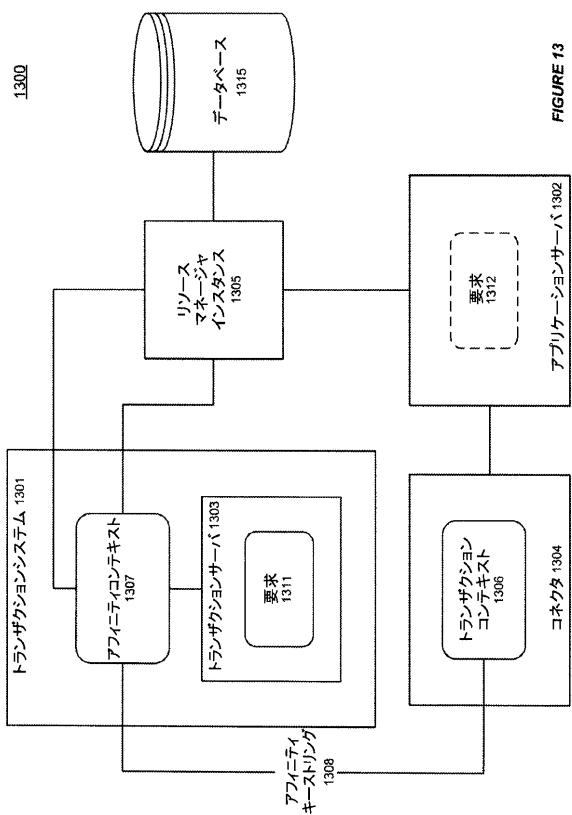
【図11】



【図12】



【図13】



【図14】

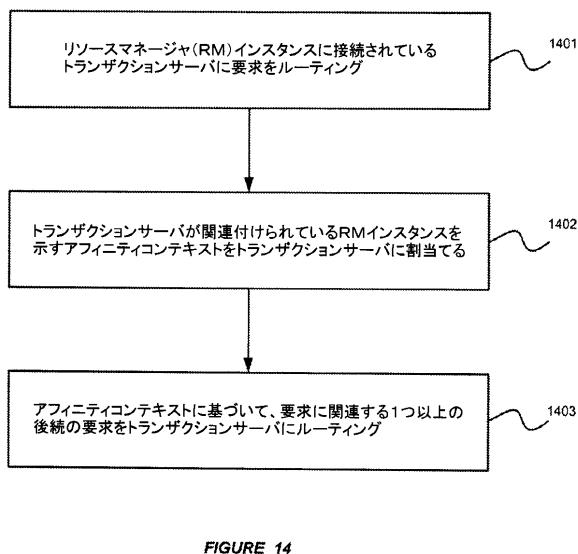
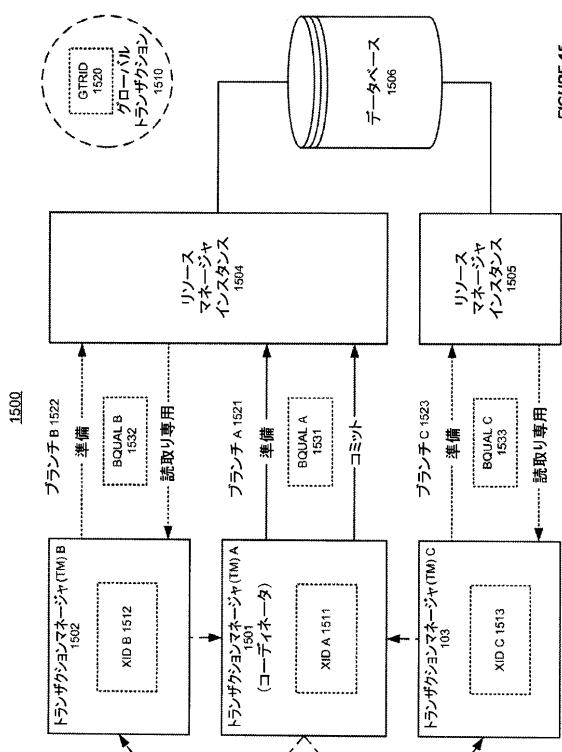


FIGURE 14

【図15】



【図16】

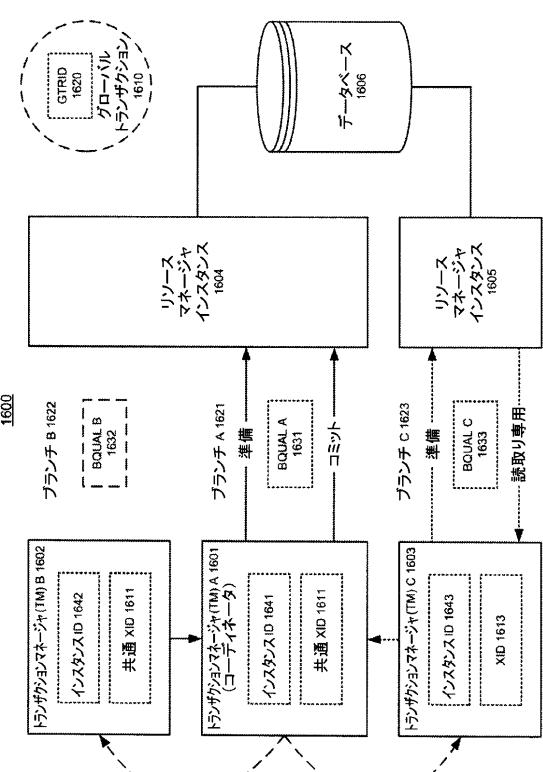
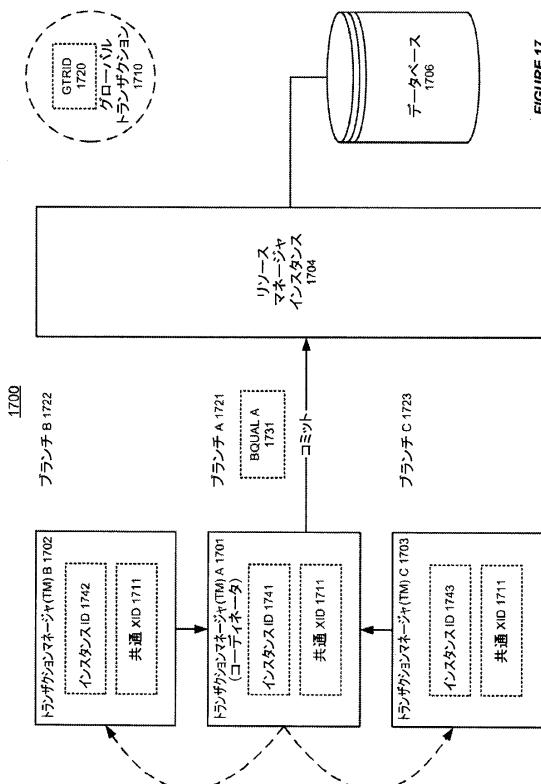
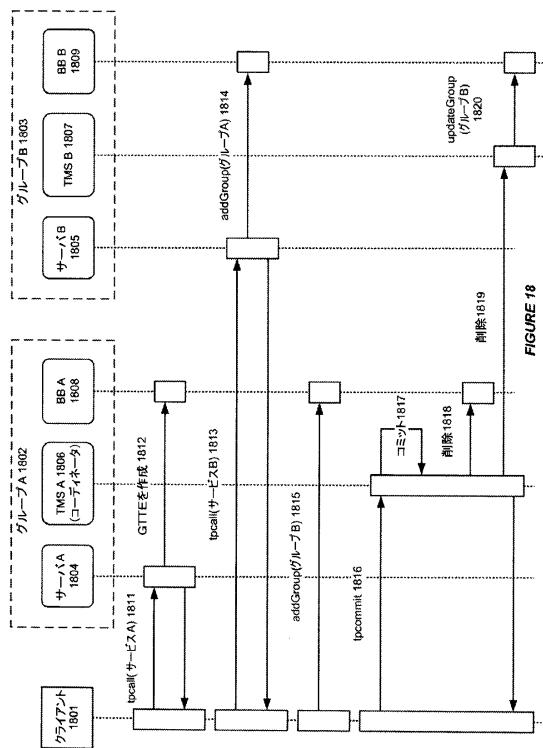


FIGURE 16

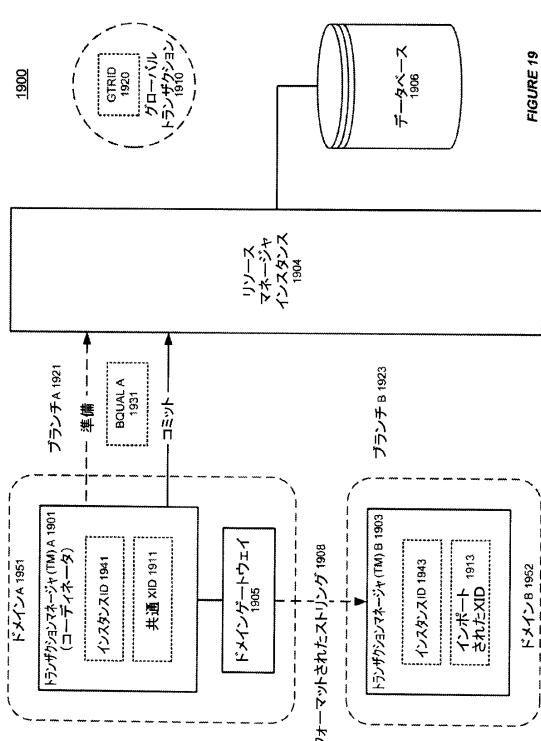
【図17】



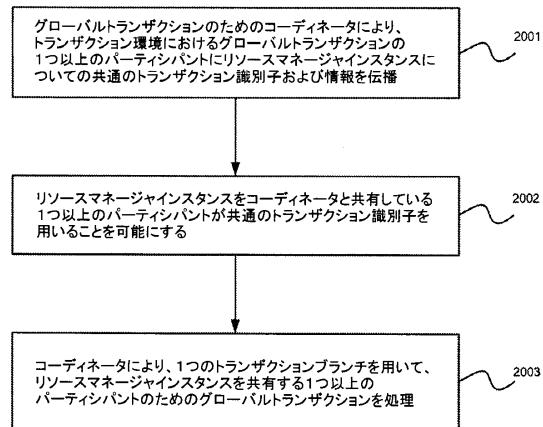
【図18】



【図19】



【図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、チャングアンツン・
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、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、チャングアンツン・
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)
G 06 F 16 / 2458
G 06 F 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