

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7023845号
(P7023845)

(45)発行日 令和4年2月22日(2022.2.22)

(24)登録日 令和4年2月14日(2022.2.14)

(51)国際特許分類

G 0 6 F 16/182 (2019.01)

F I

G 0 6 F 16/182

請求項の数 11 (全17頁)

(21)出願番号	特願2018-533870(P2018-533870)	(73)特許権者	502303739 オラクル・インターナショナル・コーポ レイション アメリカ合衆国カリフォルニア州940 65レッドウッド・シティー, オラクル ・パークウェイ500
(86)(22)出願日	平成29年8月3日(2017.8.3)		
(65)公表番号	特表2019-528489(P2019-528489 A)		
(43)公表日	令和1年10月10日(2019.10.10)		
(86)国際出願番号	PCT/US2017/045283	(74)代理人	110001195
(87)国際公開番号	WO2018/027026	(72)発明者	特許業務法人深見特許事務所 ドウ・ラバルネ, ジャン フランス、78000 ベルサイユ、リ ュ・デュ・マル・ドウ・ラトル・ドウ タシニー、16 バーマ, サウラブ インド、560029 バンガロール、 チッカ・アダゴディ、セカンド・クロス
(87)国際公開日	平成30年2月8日(2018.2.8)		
審査請求日	令和2年3月26日(2020.3.26)		
(31)優先権主張番号	15/227,897	(72)発明者	最終頁に続く
(32)優先日	平成28年8月3日(2016.8.3)		
(33)優先権主張国・地域又は機関	米国(US)		

(54)【発明の名称】 マルチテナント・データベース環境においてテナントの動的再配置を提供するためのシステムおよび方法

(57)【特許請求の範囲】

【請求項1】

接続プールの利用と、テナントの動的再配置のサポートとを含む、マルチテナント環境においてデータベースへのアクセスを提供するためのシステムであって、
 プロセッサ、および前記プロセッサを実行するアプリケーションサーバまたはデータベース環境のうちの少なくとも一方を含むコンピュータと、
 ソフトウェアアプリケーションが前記接続プールに接続を要求でき、且つ、提供される接続をデータベースにアクセスするために利用できるようにする、接続プールとを備え、
 特定のテナントに対応付けられたデータソースを第1のデータベース位置から第2のデータベース位置へ移行する命令を受信したことに応答して、前記接続プールは、クライアントアプリケーションに対応付けられたテナントを複数のデータベース位置の範囲内で再配置されるようにし、前記再配置できるようにすることは、
 前記テナントに本来対応付けられていた前記第1のデータベース位置への存在している接続をドレインすることを制御することと、
 前記テナントに対応付けられる前記第2のデータベース位置への新たな接続要求を転送することを含み、
 前記第1のデータベース位置および前記第2のデータベース位置に対応付けられるリスナーが、前記新たな接続要求が前記第2のデータベース位置へ送られるようにするために、前記クライアントアプリケーションにデータソースインスタンスに対応する接続文字列の変更を要求することなく、データベースドライバにリダイレクトを送るように構成される

、システム。

【請求項 2】

前記クライアントアプリケーションが、前記接続プールに1つ以上のシャード・キーを接続プールに提供すると、前記接続プールが前記1つ以上のシャード・キーについてのマッピングをすでに有している場合、前記接続要求が適切なシャードおよびチャunkに直接転送され、前記接続プールが前記1つ以上のシャード・キーについてのマッピングを有していない場合、前記接続要求が適切なシャード・ディレクタまたはリスナーに転送される、請求項1記載のシステム。

【請求項 3】

接続プールの利用と、テナントの動的再配置のサポートとを含む、マルチテナント環境においてデータベースへのアクセスを提供するための方法であって、

前記方法は、

プロセッサ、および前記プロセッサを実行するアプリケーションサーバまたはデータベース環境のうちの少なくとも一方を含むコンピュータにおいて、接続プールを提供するステップを備え、前記接続プールは、接続オブジェクトを含み、ソフトウェアアプリケーションが前記接続プールに接続を要求でき、且つ、提供される接続をデータベースにアクセスするために利用できるようにし、前記方法は、さらに、

特定のテナントに対応付けられたデータソースを第1のデータベース位置から第2のデータベース位置へ移行する命令を受信したことに応答して、前記接続プールによって、クライアントアプリケーションに対応付けられたテナントを複数のデータベース位置の範囲内で再配置するステップを備え、前記再配置するステップは、

前記テナントに本来対応付けられていた前記第1のデータベース位置への存在している接続をドレインすることを制御するステップと、

前記テナントに対応付けられる前記第2のデータベース位置への新たな接続要求を転送するステップとを含み、

前記第1のデータベース位置および前記第2のデータベース位置に対応付けられるリスナーが、前記新たな接続要求が前記第2のデータベース位置へ送られるようにするために、前記クライアントアプリケーションにデータソースインスタンスに対応する接続文字列の変更を要求することなく、データベースドライバにリダイレクトを送る、方法。

【請求項 4】

前記クライアントアプリケーションが、前記接続プールに1つ以上のシャード・キーを接続プールに提供すると、前記接続プールが前記1つ以上のシャード・キーについてのマッピングをすでに有している場合、前記接続要求が適切なシャードおよびチャunkに直接転送され、前記接続プールが前記1つ以上のシャード・キーについてのマッピングを有していない場合、前記接続要求が適切なシャード・ディレクタまたはリスナーに転送される、請求項3記載の方法。

【請求項 5】

存在している接続をドレインし、第1のコンテナ・データベースの第1のプラガブル・データベースから第2のコンテナ・データベースの新しい位置へ新しい接続を移行している間、

前記第2のコンテナ・データベースの第2のプラガブル・データベースは、オープンされており、

前記第1のプラガブル・データベース上のクライアントセッションは、終了され、前記新しい位置に対応付けられた、移行されたサービスに再接続できるようにされる、請求項3または4に記載の方法。

【請求項 6】

システム・イベントは、前記テナントに本来対応付けられていた前記データベース位置がシャットダウンすることを前記接続プールに通知し、および、対応付けられた接続をクローズして移行の準備を行うために、利用される、請求項3～5のいずれか1項に記載の方法。

10

20

30

40

50

【請求項 7】

ソフトウェアアプリケーションが特定の接続状態に特定のラベルを対応付けることができるようとする、請求項 3 ~ 6 のいずれか 1 項に記載の方法。

【請求項 8】

前記接続プールは、各テナントに対応付けられた異なるデータベース位置を含む、複数のテナントをサポートする、請求項 3 ~ 7 のいずれか 1 項に記載の方法。

【請求項 9】

コンピュータシステムに、請求項 3 ~ 8 のいずれか 1 項に記載の方法を実行させるためのコンピュータプログラム。

【請求項 10】

請求項 3 ~ 8 のいずれか 1 項に記載の方法を実行するための手段を備える、装置。

10

【請求項 11】

接続プールの利用と、テナントの動的再配置のサポートとを含む、マルチテナント環境においてデータベースへのアクセスを提供するためのシステムであって、

プロセッサ、および前記プロセッサを実行するアプリケーションサーバまたはデータベース環境のうちの少なくとも一方を含むコンピュータを備え、

前記コンピュータは、接続プールからの接続を要求するソフトウェアアプリケーションからの要求を受信する接続プールにおける接続オブジェクトの作成および利用を制御し、データベースへアクセスするために、提供された接続を利用し、前記データベース環境は、コンテナデータベースと、プラガラブル・データベースとして複数のデータベース位置を有し、

前記複数のデータベース位置の各データベース位置は、1 以上のサービスを介して、マルチテナントソフトウェアアプリケーションにアクセスする 1 以上のテナントと対応付けられ、前記システムは、各特定のテナントの前記データベース位置での特定のデータソースインスタンスへのマッピングを維持し、前記マルチテナントソフトウェアアプリケーションは、テナントデータソースと対応する接続要求を作り、前記データベース位置への接続を利用するためサービスをスイッチするように適合され、

前記システムは、前記ソフトウェアアプリケーションが特定の接続状態に特定のラベルを対応付けることができるようにし、

特定のテナントに対応付けられたデータソースを第 1 のデータベース位置から第 2 のデータベース位置へ移行する命令を受信したことに応答して、前記接続プールは、クライアントアプリケーションに対応付けられたテナントを複数のデータベース位置の範囲内で再配置するようにし、前記再配置できるようにすることは、

30

前記テナントに本来対応付けられていた前記第 1 のデータベース位置への存在している接続をドレンインすることを制御することと、

前記テナントに対応付けられる前記第 2 のデータベース位置への新たな接続要求を転送することを含み、

前記第 1 のデータベース位置および前記第 2 のデータベース位置に対応するリスナーが、前記新たな接続要求が前記第 2 のデータベース位置へ送られるようにするために、前記クライアントアプリケーションに前記データソースインスタンスに対応する接続文字列の変更を要求することなく、データベースドライバにリダイレクトを送るように構成される、システム。

40

【発明の詳細な説明】**【技術分野】****【0 0 0 1】****著作権表示**

本特許文献の開示の一部には、著作権保護の対象となる資料が含まれる。特許文献または特許開示は、米国特許庁の特許ファイルおよび記録に掲載されているため、著作権者は、何人によるその複製に対しても異議はないが、そうでない場合には、例外なくすべての著作権を保有する。

50

【0002】

優先権の主張

本願は、2016年8月3日に出願され、「SYSTEM AND METHOD FOR PROVIDING DYNAMIC RELOCATION OF TENANTS IN A MULTITENANT DATABASE ENVIRONMENT」と題された米国特許出願第15/227,897号に基づく優先権の利益を主張し、当該出願を、引用により本明細書に援用する。

【0003】

発明の分野

本発明の実施形態は、全体的に、ソフトウェアアプリケーションサーバおよびデータベースに関し、具体的には、接続プールの利用と、テナントの動的再配置のサポートとを含む、マルチテナント環境においてデータベースへのアクセスを提供するためのシステムおよび方法に関する。

10

【発明の概要】

【発明が解決しようとする課題】

【0004】

背景

一般的に説明すると、データベース環境において、接続プールは、接続オブジェクトのキッシュとして動作する。接続オブジェクトの各々は、ソフトウェアアプリケーションがデータベースに接続するために使用できる接続を表す。実行時、アプリケーションは、接続プールに接続を要求することができる。接続プールが特定の要求に応えることができる接続を含む場合、接続プールは、その接続を使用のためにアプリケーションに返すことができる。場合によっては、適した接続が見つからない場合、新しい接続を作成してアプリケーションに返すことができる。アプリケーションは、データベースにアクセスして作業を行うために接続を借り、その後、接続をプールに返却することができ、接続は、返却されると、同じまたはその他のアプリケーションからの次の接続要求のために利用可能にすることができる。

20

【課題を解決するための手段】

【0005】

概要

30

本明細書において、接続プールの利用と、テナントの動的再配置のサポートとを含む、マルチテナント環境においてデータベースへのアクセスを提供するためのシステムおよび方法を説明する。実施形態によると、ソフトウェアアプリケーションは、テナントに代わって接続プールから接続を獲得することができ、これによって、ソフトウェアアプリケーションまたはテナントをデータベースにアクセスできるようにする。再配置プロセスによって、マルチテナントまたはその他のクライアントアプリケーションに対応付けられたテナントを、クライアントアプリケーションにとってニアゼロ(near-zero)ダウンタイムで、たとえば、複数のコンテナ・データベース間で再配置できるようにし、当該再配置できるようにすることは、基礎をなすアプリケーションを変更する必要なしに、存在している接続のドレイン(dRAINING)を管理することと、新しい接続を移行することとを含む。

40

【図面の簡単な説明】

【0006】

【図1】実施形態に係る、接続プールを備えるシステムを説明する図である。

【図2】実施形態に係る、シャード・データベースの利用のサポートを含む、接続プールを備えるシステムをさらに説明する図である。

【図3】実施形態に係る、マルチテナント環境での利用のサポートを含む、接続プールを備えるシステムをさらに説明する図である。

【図4】実施形態に係る、接続プール環境におけるテナントの動的再配置のサポートを説明する図である。

50

【図5】実施形態に係る、接続プール環境におけるテナントの動的再配置のサポートをさらに説明する図である。

【図6】実施形態に係る、接続プール環境におけるテナントの動的再配置のサポートをさらに説明する図である。

【図7】実施形態に係る、接続プール環境におけるテナントの動的再配置のサポートをさらに説明する図である。

【図8】実施形態に係る、接続プール環境におけるテナントの動的再配置のサポートをさらに説明する図である。

【図9】実施形態に係る、接続プール環境におけるテナントの動的再配置のサポートをさらに説明する図である。

【図10】実施形態に係る、接続プール環境におけるテナントの動的再配置のサポートを提供するための方法を説明する図である。

【発明を実施するための形態】

【0007】

詳細な説明

上述したように、接続プールは、接続オブジェクトのキャッシュとして動作する。接続オブジェクトの各々は、ソフトウェアアプリケーションがデータベースに接続するために使用できる接続を表す。実行時、アプリケーションは、接続プールに接続を要求できる。接続プールが特定の要求に応えることができる接続を含む場合、接続プールは、その接続を使用のためにアプリケーションに返すことができる。場合によっては、適した接続が見つからない場合、新しい接続を作成してアプリケーションに返すことができる。アプリケーションは、データベースにアクセスして作業を行うために接続を借り、その後、接続をプールに返却することができ、接続は、返却されると、同じまたはその他のアプリケーションからの次の接続要求のために利用可能にすることができる。

【0008】

接続オブジェクトを作成することは、時間とリソースの観点からいえば、高くつく傾向がある。たとえば、ネットワーク通信、認証、トランザクション参加、およびメモリ割り当てなどのタスクは、すべて、特定の接続オブジェクトを作成するのにかかる時間とリソースの量の一因となる。接続プールがこのような接続オブジェクトの再利用を可能にするため、さまざまなオブジェクトを作成しなければならない回数を減らすことに役立つ。

【0009】

接続プールの一例に、JDBC (Java (登録商標) Database Connectivity) 接続をキャッシュするための接続プールを提供するUCP (Oracle Universal Connection Pool) がある。たとえば、接続プールは、データベースへの接続を作成するために、JDBC ドライバと共に動作することができ、接続は、その後、プールによって維持され、要求側ソフトウェアアプリケーションの性能要件および可用性要件に基づいて、プールの挙動をさらに最適化するために使用されるプロパティを有して構成できる。

【0010】

接続のラベル付け

図1は、実施形態に係る、接続プールを備えるシステムを説明する図である。

【0011】

図1に示すように、実施形態によると、アプリケーションサーバまたはデータベース環境100は、物理的なコンピュータ資源101 (たとえば、プロセッサ / CPU、メモリ、およびネットワークコンポーネント)、たとえば、Oracle WebLogic Server、Oracle Fusion Middleware、またはその他のアプリケーションサーバもしくはデータベース環境を含み、Oracle データベースなどのデータベース102、またはその他の種類のデータベースへのアクセスを含むまたは提供することができる。

【0012】

10

20

30

40

50

図1にさらに示すように、実施形態によると、システムは、接続プールロジック104またはプログラムコードも備え、接続プールロジック104またはプログラムコードは、コンピュータによって実行されると、たとえば、ソフトウェアアプリケーションが現在使用中の接続(108)、および、アイドル状態の接続(110)または現在使用されていない接続を含む、接続プール106における接続オブジェクトの作成および使用を制御(105)する。

【0013】

ソフトウェアアプリケーションは、接続プールから取り出された接続を、データベースにアクセスしたり、データベースで作業を行ったりするために当該接続を使用する前に、初期化することができる。たとえば、初期化には、アプリケーションコード内でメソッド呼び出しが必要な単純状態再初期化、または、ネットワークのラウンドトリップ(`round trip`)が必要なデータベース動作を含んだ、より複雑な初期化などがある。後者のタイプの初期化の計算コストは、かなり大きいだろう。

10

【0014】

接続プール(たとえば、UCP)によっては、接続プールプロパティを使用して構成できるものがあり、接続プールは、`get`メソッドおよび`set`メソッドを有し、プールが有効なデータソースインスタンスを通して利用可能である。これらの`get`メソッドおよび`set`メソッドは、プールをプログラムで構成するための便利な方法を提供する。プールプロパティが設定されていない場合、接続プールは、デフォルトのプロパティ値を使用する。

20

【0015】

実施形態によると、接続をラベル付けすることによって、クライアントソフトウェアアプリケーションは、任意の名前/値ペアを接続に結びつけることができる。次に、アプリケーションは、所望のラベルを有する接続を接続プールに要求できる。特定のラベルを特定の接続状態と対応付けることによって、アプリケーションは、すでに初期化された接続をプールから取り出せる可能性があり、再初期化にかかる時間とコストを回避できる。接続のラベル付けは、ユーザ定義キーまたは値になんら意味を加えない。すべてのユーザ定義キーおよび値の意味は、アプリケーションによってのみ定義される。

【0016】

たとえば、図1に示すように、実施形態によると、接続プールは、ここでは、接続A 1 1 2 およびB 1 1 4と示される、ソフトウェアアプリケーションによって現在使用中の複数の接続を含めることができる。接続の各々はラベル付けができる、たとえば、接続Aには(青)がラベル付けされ、接続Bには(緑)がラベル付けされる。これらのラベル/色は、説明のために提供されており、上述したように、クライアントアプリケーションによって接続に結びつけられた任意の名前/値ペアであり得る。さまざまな実施形態によると、互いに異なる接続の種類を区別するために、異なる種類のラベルを利用でき、互いに異なるアプリケーションは、互いに異なるラベル/色を特定の接続の種類に結びつけることができる。

30

【0017】

また、図1にさらに示すように、実施形態によると、接続プールは、ここでは接続C 1 1 6、D 1 1 8、E 1 2 0、F 1 2 2、G 1 2 4、およびN 1 2 6と示される、アイドル状態またはソフトウェアアプリケーションによって現在使用されていない複数の接続を含めることができる。アイドル状態の接続の各々は、同様に、この例示において、(青)または(緑)とラベル付けができる、ここでも、これらのラベル/色は、説明のために提供される。

40

【0018】

図1にさらに示すように、実施形態によると、ソフトウェアアプリケーション130が、特定の種類の接続、たとえば、(赤)接続を利用してデータベースへの要求を行いたい場合、アプリケーションは、「`get Connection`(赤)」要求132を行うことができる。これに応答して、接続プールロジックは、ここではX 1 3 4(赤)と示され

50

る、新しい（赤）接続を作成する、または、ここでは E 135（赤）と示される、存在しているアイドル状態の接続を（青または緑）から（赤）へ転用する。

【 0 0 1 9 】

シャード・データベース

実施形態によると、シャーディングは、複数の独立した物理データベース間にデータを水平分割するデータベースのスケーリング技術である。各物理データベースに格納されたデータ部分は、シャードと呼ばれる。ソフトウェア・クライアント・アプリケーションの観点からみると、すべての物理データベースの集まりは、1つの論理データベースに見える。

【 0 0 2 0 】

実施形態によると、システムは、シャード・データベースと共に接続プールを利用するとのサポートを含めることができる。シャード・ディレクタまたはリスナーによって、ソフトウェア・クライアント・アプリケーションはデータベース・シャードへアクセスできるようになる。接続プール（たとえば、U C P）およびデータベース・ドライバ（たとえば、J D B C ドライバ）は、接続チェックアウト中またはその後にクライアントアプリケーションにシャード・キーを提供させ、クライアントアプリケーションが指定したシャード・キーを認識し、クライアントアプリケーションによる特定のシャードまたはチャネルへの接続を可能にするように構成できる。この手法は、接続リソースの効率的な再利用を可能にし、且つ、適切なシャードへのより速いアクセスを可能にする。

10

【 0 0 2 1 】

図2は、実施形態に係る、シャード・データベースの利用のサポートを含む、接続プールを備えるシステムをさらに説明する図である。

20

【 0 0 2 2 】

実施形態によると、たとえば、特定のシャード内で各ロー（r o w）が格納される場所を決定する1つ以上のカラム（c o l u m n）としてシャード・キー（S H A R D _ K E Y）を利用して、データベース・テーブルを分割できる。シャード・キーは、接続データ（C O N N E C T _ D A T A）の属性として、接続文字列または接続記述子に設けることができる。たとえば、シャード・キーには、データベースのV A R C H A R 2、C H A R、D A T E、N U M B E R、またはT I M E S T A M Pなどがある。また、実施形態によると、シャード・データベースは、シャード・キーまたはシャードグループキーなしで接続を受け付けることができる。

30

【 0 0 2 3 】

実施形態によると、システム性能およびデータ可用性へのシャーディングの影響を低減するため、各シャードは、小さな部分またはチャネルにさらに分割できる。各チャネルは、あるシャードから別のシャードへ移動できるシャーディングの単位となる。また、チャネルは、シャード・キー・マッピングへの間接参照を追加することによって、ルーティングを簡素化する。

【 0 0 2 4 】

たとえば、各チャネルをシャード・キー値の範囲に自動的に対応付けることができる。ユーザ提供のシャード・キーを特定のチャネルにマッピングし、そのチャネルを特定のシャードにマッピングすることができる。データベース動作が特定のシャードに存在しないチャネルに対して動作を行おうとした場合、エラーが発生する。シャードグループを使用する場合、各シャードグループは、特有の値のシャードグループ識別子を有するチャネルの集まりである。

40

【 0 0 2 5 】

シャードアウェア・クライアントアプリケーションは、シャード・データベース構成とともに作動でき、1つ以上のシャーディング方法に基づいてデータが分割された1つまたは複数のデータベース・シャードに接続する機能を含む。データベース動作が必要となるたびに、クライアントアプリケーションは、接続する必要があるシャードを決定できる。

【 0 0 2 6 】

実施形態によると、シャーディング方法は、個々のシャードにシャード・キー値をマッピ

50

ングするために使用できる。さまざまなシャーディング方法をサポートできる。たとえば、ハッシュ値の範囲が各チャunkに割り当てられるハッシュ・ベース・シャーディング、これにより、データベース接続を確立すると、システムは、所定の値のシャーディング・キーにハッシュ関数を適用して対応するハッシュ値を算出し、次に、その値が属する範囲に基づいて、対応するハッシュ値がチャunkにマッピングされる。シャード・キー値の範囲が個々のシャードに直接割り当てられるレンジ・ベース・シャーディング、および、各シャードがシャード・キー値の一覧に対応付けられるリスト・ベース・シャーディングがある。

【0027】

図2に示すように、実施形態によると、シャード・データベース140は、ここでは「D B East」、DBEと示される第1のデータベース領域A 141を備えることができ、第1のデータベース領域A 141は、チャunk A1、A2、...Anとして格納されるシャードAを有するシャード・データベース・インスタンス「DBE-1」142と、チャunk B1、B2、...Bnとして格納されるシャードBを有するシャード・データベース・インスタンス「DBE-2」143とを含む。

10

【0028】

図2にさらに示すように、実施形態によると、第2のデータベース領域B（ここでは、「DB West」、DBWと示される）144は、チャunk C1、C2、...Cnとして格納されるシャードCを有するシャード・データベース・インスタンス「DBW-1」145と、チャunk D1、D2、...Dnとして格納されるシャードDを有する「DBW-2」146とを含む。

20

【0029】

実施形態によると、各データベース領域、またはシャード・データベース・インスタンスのグループは、シャード・ディレクタまたはリスナー（たとえば、Oracle Global Service Managers (GSM) リスナー、または別の種類のリスナー）に対応付けることができる。たとえば、図2に示すように、シャード・ディレクタまたはリスナー147は、第1のデータベース領域Aに対応付けることができ、別のシャード・ディレクタまたはリスナー148は、第2のデータベース領域Bに対応付けることができる。システムは、シャード・トポロジー・レイヤ154を維持するデータベース・ドライバ（たとえば、JDBCドライバ）152を備えることができる。シャード・トポロジー・レイヤ154は、ある期間にわたってシャード・キー範囲を学習し、シャード・データベースにおける各シャードの位置にシャード・キー範囲のキャッシュを保持する。

30

【0030】

実施形態によると、クライアントアプリケーションは、接続要求（162）の間、接続プールに1つ以上のシャード・キーを提供でき、接続プールは、1つ以上のシャード・キー、およびシャード・トポロジー・レイヤによって提供された情報に基づいて、接続要求を正しいまたは適切なシャードに送ることができる。

【0031】

また、実施形態によると、接続プールは、特定のシャードまたはチャunkへの接続を、そのシャード・キーによって識別でき、同じシャード・キーの要求を特定のクライアントアプリケーションから受け付けた場合、接続を再利用させることができる。

40

【0032】

たとえば、図2に示すように、実施形態によると、特定のチャunk（たとえば、チャunk A1）への接続は、そのチャunkへ接続する（174）ために使用できる。特定のシャードまたはチャunkへの利用可能な接続がプールにない場合、システムは、存在している利用可能な接続を別のシャードまたはチャunkに転用し、その接続を再利用しようとすることができる。データベースにおけるシャードおよびチャunk間のデータ分散は、クライアントアプリケーションに対して透過的にすることができ、チャunkをリシャーディングすることのクライアントへの影響も最小限に抑える。

【0033】

50

シャードアウェア・クライアントアプリケーションが接続要求に対応付けて1つ以上のシャード・キーを接続プールに提供すると、接続プールまたはデータベース・ドライバがシャード・キーについてのマッピングをすでに有している場合、適切なシャードおよびチャンク、この例においてチャンク C 2、に接続要求を直接転送できる。

【 0 0 3 4 】

シャードアウェア・クライアントアプリケーションが接続要求に対応付けてシャード・キーを提供しない場合、または、提供されたシャード・キーのマッピングを接続プールもしくはデータベース・ドライバが有しない場合、接続要求を適切なシャード・ディレクタまたはリストナーに転送できる。

【 0 0 3 5 】

マルチテナント環境

実施形態によると、システムは、接続のラベル付けを利用したクラウドベースまたはマルチテナント環境のサポートを含めることができる。たとえば、マルチテナント・クラウド環境は、クラウドベース環境において複数のテナントまたはテナントアプリケーションが使用するためのデータベースへのアクセスを含むまたは提供するアプリケーションサーバまたはデータベース環境を含めることができる。

【 0 0 3 6 】

図3は、実施形態に係る、マルチテナント環境での利用のサポートを含む、接続プールを備えたシステムをさらに説明する図である。

【 0 0 3 7 】

クラウドまたはその他のネットワークを介してテナントがアクセスできるソフトウェアアプリケーションは、上述の環境と同様に、接続プールから取り出された接続を、使用する前に初期化してもよい。

【 0 0 3 8 】

上述したように、たとえば、初期化には、アプリケーションコード内でメソッド呼び出しが必要な単純状態再初期化や、ネットワークのラウンドトリップが必要なデータベース動作を含んだ、より複雑な初期化などがある。

【 0 0 3 9 】

また、上述したように、接続をラベル付けすることによって、アプリケーションは、任意の名前 / 値ペアを接続に結びつけることができ、これにより、アプリケーションは、所望のラベルを有する接続を接続プールに要求でき、ラベル付けすることは、すでに初期化された接続をプールから取り出し、再初期化にかかる時間とコストを回避する機能を含む。

【 0 0 4 0 】

図3に示すように、実施形態によると、マルチテナント・データベース環境 1 8 0 は、たとえば、コンテナ・データベース (C D B) 1 8 1 、ならびに、ここでは、「 P D B - 1 」 1 8 2 、「 P D B - 2 」 1 8 3 、および「 P D B - 3 」 1 8 4 と示される1つ以上のプログラガブル・データベース (P D B) を備えることができる。

【 0 0 4 1 】

実施形態によると、各 P D B は、アプリケーションサーバもしくはデータベース環境 1 0 0 がホストであるまたは外部クライアントアプリケーション 1 8 6 として提供されるマルチテナント・アプリケーションの、ここでは、「テナント - 1 」、「テナント - 2 」、および「テナント - 3 」と示されるテナントに対応付けることができる。各 P D B は、この例において「 R A C - インスタンス - 1 」および「 R A C - インスタンス - 2 」を含む1つ以上の R A C (O r a c l e R e a l A p p l i c a t i o n C l u s t e r) インスタンス 1 8 6 、 1 8 8 と、この例において「サービス - 1 」、「サービス - 2 」、および「サービス - 3 」を含む1つ以上のサービスと、テナントのサービスへのマッピング 1 9 0 とを利用して、データベース環境へのアクセスを提供する。

【 0 0 4 2 】

図3に示す例において、データベース環境にアクセスするためにテナントによって使用中のアプリケーションは、そのテナントのデータソース 1 9 2 、 1 9 4 、 1 9 6 に対応付け

10

20

30

40

50

られた接続要求を行うことができ、システムは、存在している R A C インスタンスまたは P D B への接続を利用するため、必要であれば、サービスを切り替える (1 9 8) ことができる。

【 0 0 4 3 】

サーバ側接続プール

実施形態によると、システムは、たとえば、 D R C P (O r a c l e D a t a b a s e R e s i d e n t C o n n e c t i o n P o o l) が提供するような、サーバ側接続プールのタグ付け機能を利用することができる。サーバ側接続プールのタグ付け機能によって、ユーザアプリケーションまたはクライアントは、データベース環境への接続を、そのデータベース環境が理解する 1 つのタグの利用に基づいて選択的に取得できるようになる。

10

【 0 0 4 4 】

実施形態によると、1 つのタグのみが接続ごとに応付ける。データベースサーバは、ユーザアプリケーションまたはクライアントに、タグ値を伝達しないで、タグの一致を伝達する (たとえば、 B o o l e a n 値として) 。

【 0 0 4 5 】

プールにおけるテナントの動的再配置

実施形態によると、ソフトウェアアプリケーションは、テナントの動的再配置のサポートを含めることができる。ソフトウェアアプリケーションは、テナントに代わって接続プールから接続を獲得することができ、これによって、ソフトウェアアプリケーションまたはテナントがデータベースにアクセスできるようにする。再配置プロセスによって、マルチテナントまたはその他のクライアントアプリケーションに対応付けられたテナントを、クライアントアプリケーションにとってニアゼロダウンタイムで、たとえば、複数のコンテナ・データベース間で再配置できるようにし、当該再配置できるようにすることは、基礎をなすアプリケーションを変更する必要なしに、存在している接続のドレインを管理することと、新しい接続を移行することとを含む。

20

【 0 0 4 6 】

図 4 ~ 図 9 は、実施形態に係る、接続プール環境におけるテナントの動的再配置のサポートを説明する図である。

【 0 0 4 7 】

図 4 に示すように、実施形態によると、コンテナ・データベース (たとえば、「 C D B - 1 」 2 0 2) などのデータベース、または別の種類のデータベースが、複数の接続 2 0 4 をサポートする。

30

【 0 0 4 8 】

アプリケーションサーバまたはデータベース環境のいずれかがホストであるマルチテナントまたはその他のクライアントアプリケーションに対応付けられた、または外部クライアントアプリケーションとして提供されたテナントは、データベースサービスを介して、コンテナ・データベースのプラガブル・データベースにアクセスすることを適宜含む、データベースにアクセスするために、接続プールを利用できる。

【 0 0 4 9 】

たとえば、実施形態によると、各特定のテナントは、コンテナ・データベースにおいて、テナント自体の特定のプラガブル・データベースに対応付けることができる。各特定のテナントは、そのテナントに対応付けられた特定のプラガブル・データベースに、特定のプラガブル・データベースに対応付けられたデータベースサービスを介してアクセス (たとえば、 2 0 5) するために、接続プールが提供する接続を利用できる。

40

【 0 0 5 0 】

実施形態によると、データベース環境が変化した場合、たとえば、システムに第 2 のコンテナ・データベース (たとえば、「 C D B - 2 」 2 0 3) が追加された場合、または、接続プールをホストしているアプリケーションサーバがデータベース環境からサービスダウン・イベント 2 0 6 を受信することに応答して、システムは、特定のテナントが使用するための、新しいデータベース位置への新しい接続 2 0 7 を提供することができる。

50

【 0 0 5 1 】

たとえば、実施形態によると、システムは、テナントが利用するために、プラガブル・データベースの移行を開始でき、当該移行は、元のプラガブル・データベース位置に対応付けられた接続およびその関連するデータベースサービス（たとえば、「CDB-1」202において「PDB-1」182に対応付けられた接続）をドレインすることと、これらの接続の可用性を新しいプラガブル・データベース位置および関連する（たとえば、「CDB-2」203において、ここで「PDB-1」210と示される）データベースサービスへ移行または再配置する（208）ことを含む。

【 0 0 5 2 】

これによって、クライアントまたはテナントアプリケーションに対して透過的な方法で、テナントの元の位置に対応付けられた存在している接続をドレインし、テナントの新しい位置を指す新しい接続を作成することによって、接続プールがニアゼロダウンタイムのテナント再配置をサポートできるようにする。

10

【 0 0 5 3 】

たとえば、マルチテナント環境において、システムは、特定のテナントに対応付けられたプラガブル・データベースを、第1のRAC(Oracle Real Application Cluster)データベースから第2のRACデータベースへ移動すること、または、第1のコンテナ・データベースから第2のコンテナ・データベースへ移動することをサポートする。

【 0 0 5 4 】

しかしながら、これらのプラガブル・データベースは、概ね、互いに異なる／別々のデータベースとして動作する。これによって、接続が失われてしまう可能性がある。

20

【 0 0 5 5 】

これに対処するために、実施形態によると、元のコンテナ・データベース（たとえば、「CDB-1」）のリスナー212を指す接続文字列を現在使用中であるアプリケーションの場合、リスナーは、新しい位置またはコンテナ・データベース（たとえば、「CDB-2」）への接続要求をリダイレクトするように構成できる。これによって、リスナーは、アプリケーションサーバのデータベース・ドライバにリダイレクトを送信することができ、アプリケーションサーバは、データベース・ドライバに、新しいコンテナ・データベースへの新しい接続要求を送らせる。

30

【 0 0 5 6 】

これに加えて、存在している接続要求を元のコンテナ・データベースから引き離さなければならない。しかしながら、新しいコンテナ・データベースは異なるデータベースと考えられるため、その存在についてプールはまだ知らない可能性がある。

【 0 0 5 7 】

これに対処するために、実施形態によると、システム・イベント通知（たとえば、Oracle Notification Serviceイベント）は、プラガブル・データベースがシャットダウンすることを接続プールに通知し、および、対応付けられた接続をクローズし、新しい位置に対応付けられた新しいデータベースサービスへの移行の準備を行うために、使用できる。

40

【 0 0 5 8 】

一般に、新しい接続をサポートするために新しいデータベース位置が直ちに利用可能にならない期間は短い。この間、存在している接続はクローズされ、接続プールは、新しい要求を受け付けるまで新しい接続を作成しない。これによって、たとえば、信頼できる情報源を新しい位置に切り替えるためにredoログを停止することを含んだredoログの更新が必要であることから、わずかなシステム・ダウンタイムが発生する。

【 0 0 5 9 】

たとえば、図4に示す例において、プラガブル・データベース（たとえば、「PDB-1」）を第1のコンテナ・データベース（たとえば、「CDB-1」）から第2のコンテナ・データベース（たとえば、「CDB-2」）へ移行することが望ましく、実施形態によ

50

ると、プラガブル・データベースの再配置に関するプロセスは、以下を含む。

【0060】

1. プラガブル・データベースの再配置を開始する。たとえば、図5に示すように、サーバは、「変更プラガブル・データベース再配置」コマンドを実行することによって、プラガブル・データベースの再配置を開始することができる。当該コマンドは、元のプラガブル・データベース上で実行中のセッション220に影響を与える。

【0061】

2. 新しい位置にあるプラガブル・データベースをオープンして、元のインスタンスコンテナ・データベース上のすべてのクライアントセッションを終了する。たとえば、図6に示すように、システムは、コンテナ・データベース・インスタンス「CDB-2」のプラガブル・データベース「PDB-1」をオープンして、元のコンテナ・データベース・インスタンス「CDB-1」上のすべてのクライアントセッションを終了することによって、「変更プラガブル・データベース再配置」コマンドに応答できる。その後、システムは、「CDB-1」上のプラガブル・データベース「PDB-1」をクローズし、そのバックファ・キャッシュをフラッシュする。

10

【0062】

3. クライアントが新しいデータベース位置に再接続できるようにする。たとえば、図7に示すように、次に、クライアントは、(たった今移行された)サービス226に自身を再接続する必要がある。接続プールは、これを、アプリケーションに透過的な方法、たとえば、図8に示すように、サーバからサービスダウン・イベントを受け付けた際に存在している接続をドレンし、移行されたプラガブル・データベースへの新しい接続を作成し直すことによって、可能にする。

20

【0063】

4. 新しい位置への接続要求を転送する。たとえば、図9に示すように、移行が完了すると、サーバ側において、リスナーは、接続プールからの新しい対象コンテナ・データベース(たとえば、「CDB-2」)への新しい接続要求を転送する(228)。アプリケーションは、接続文字列を変更する必要がない。これによって、再配置プロセスがアプリケーションに対して透過的に行われる。

【0064】

動的再配置プロセス

30

図10は、実施形態に係る、接続プール環境におけるテナントの動的再配置のサポートを提供する方法を説明する図である。

【0065】

図10に示すように、実施形態によると、ステップ231において、接続プールにおける接続オブジェクトの作成および利用を制御する接続プールロジックまたはプログラムコードが、アプリケーションサーバまたはデータベース環境において提供され、ソフトウェアアプリケーションは、接続プールに接続を要求し、提供される接続を、データベースにアクセスするために利用できる。

【0066】

図10に示すように、実施形態によると、ステップ233において、テナントに対応付けられたプラガブル・データベースを第1のコンテナ・データベース・インスタンスから第2のコンテナ・データベース・インスタンスの新しい位置に移行するための命令を受け付ける。

40

【0067】

ステップ235において、サーバは、プラガブル・データベース上で実行中のセッションに影響を与える、プラガブル・データベースの再配置を開始する。

【0068】

ステップ237において、システムは、新しい位置でプラガブル・データベースをオープンして、第1のコンテナ・データベース・インスタンス上のすべてのクライアントセッションを終了することによって、応答する。

50

【 0 0 6 9 】

ステップ 239において、クライアントが新しい位置に対応付けられた（移行された）サービスに再接続できるようにする。

【 0 0 7 0 】

ステップ 241において、移行が完了すると、サーバ側において、リスナーは、接続プールからの（新しい）コンテナ・データベース位置への新しい接続要求を転送する。

【 0 0 7 1 】

本発明の実施形態は、本開示の教示に従ってプログラムされた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み取り可能な記憶媒体を含む、1つ以上の従来の汎用または専用のデジタル・コンピュータ、コンピューティング・デバイス、機械、またはマイクロプロセッサによって好都合に実現されてもよい。当業者であるプログラマーは、ソフトウェア技術を身につけた者に明らかのように、本開示の教示に基づいて、適切なソフトウェア・コーディングを容易に用意できる。10

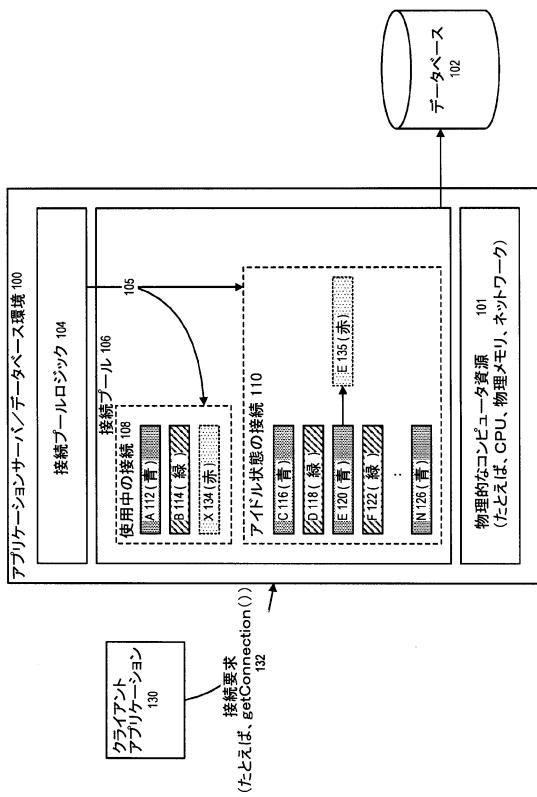
【 0 0 7 2 】

いくつかの実施形態において、本発明は、指示を格納した非一時的な記憶媒体またはコンピュータ読み取り可能な記憶媒体（複数のコンピュータ読み取り可能な記憶媒体）であるコンピュータ・プログラム・プロダクトを含む。コンピュータ・プログラム・プロダクトを使用して、コンピュータを本発明のいずれの処理も実行するようにプログラムできる。たとえば、記憶媒体は、これらに限定されないが、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種類のディスク、ROM、RAM、EPROM、EEPROM、DRAM、VRAM、フラッシュメモリデバイス、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）、または指示および／またはデータを格納するのに適した任意の種類の媒体またはデバイスを含むことができる。20

【 0 0 7 3 】

上記の本発明の実施形態の説明は、例示および説明のために提供されている。これは、本発明を包括的であったり、開示された厳密な形態に限定したりすることを意図していない。多くの変更例および変形例は、当業者に明らかであろう。変更例および変形例は、開示の特徴の任意の適切な組み合わせを含む。実施形態は、本発明の原理およびその実用的な適用を最も適切に説明するために選ばれて記載されているため、他の当業者が、考えられる特定の使用に適したさまざまな変更例を用いて、さまざまな実施形態について本発明を理解できるようになっている。30

【図1】
【図2】



【図2】

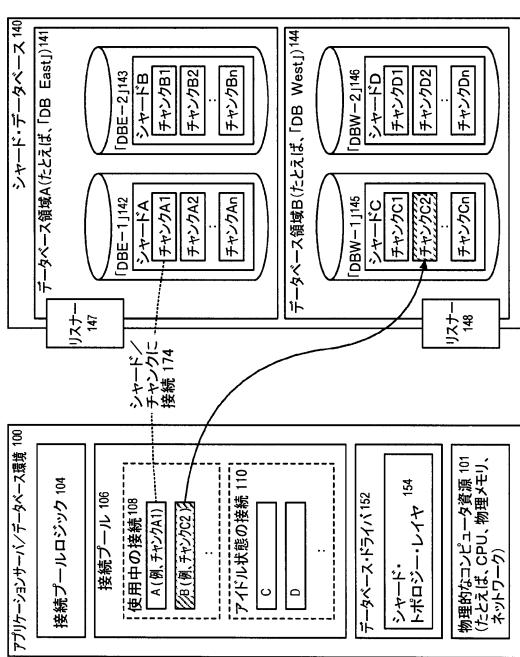


FIGURE 1

【図3】

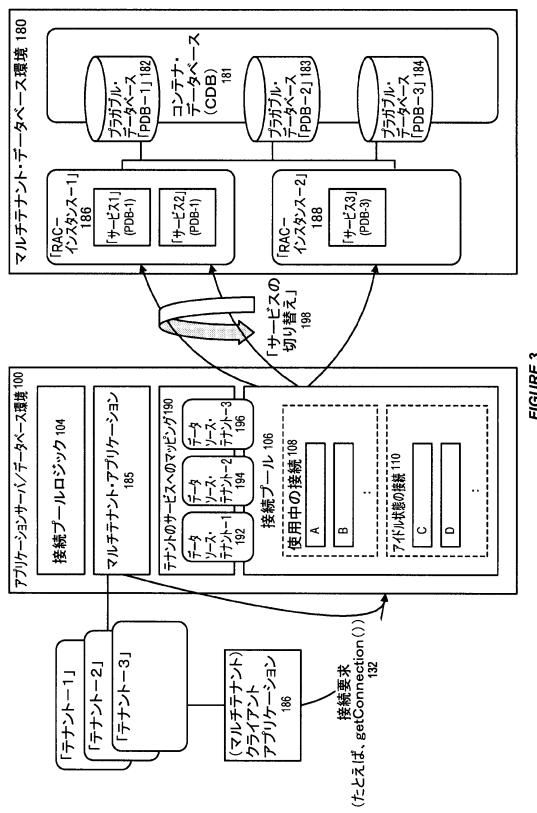


FIGURE 3

【図4】

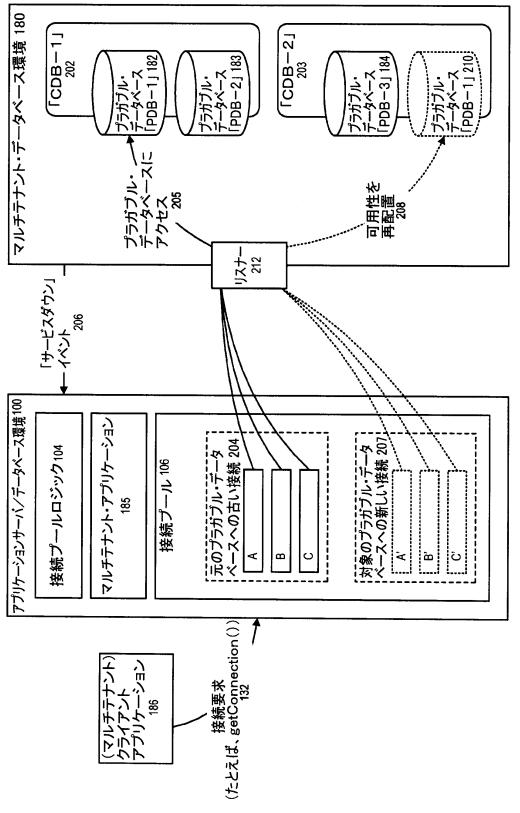


FIGURE 4

【図 5】

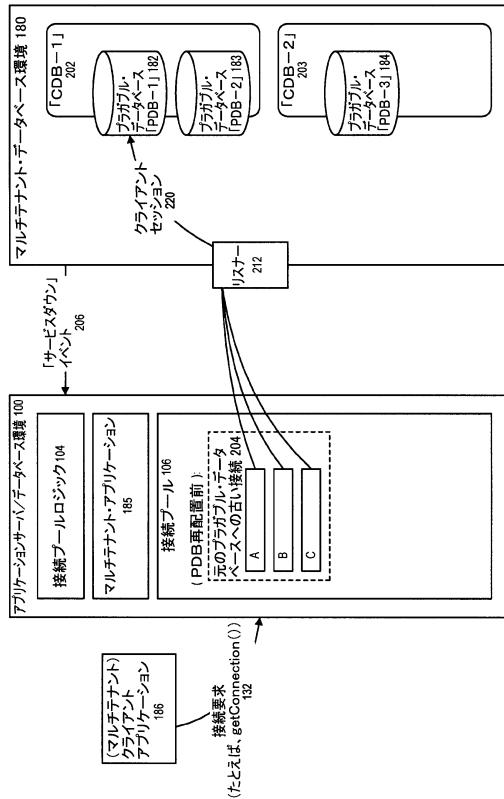


FIGURE 5

【図 6】

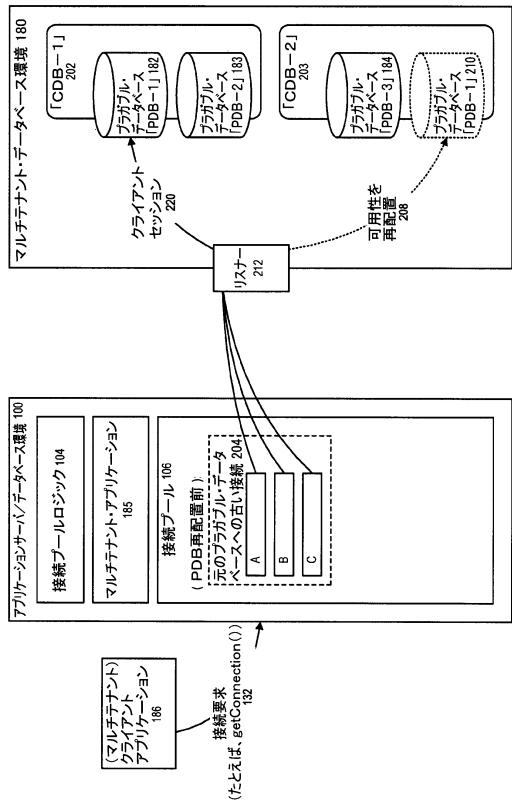


FIGURE 6

【図 7】

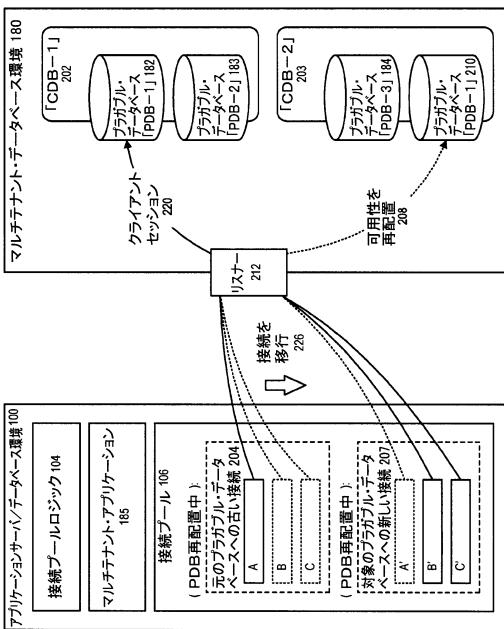


FIGURE 7

【図 8】

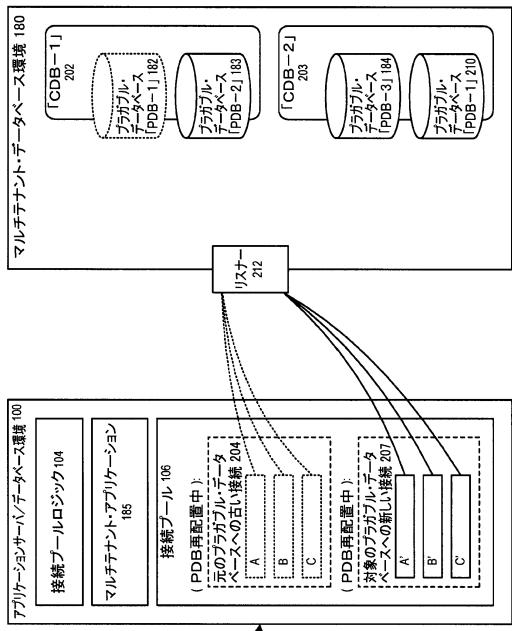
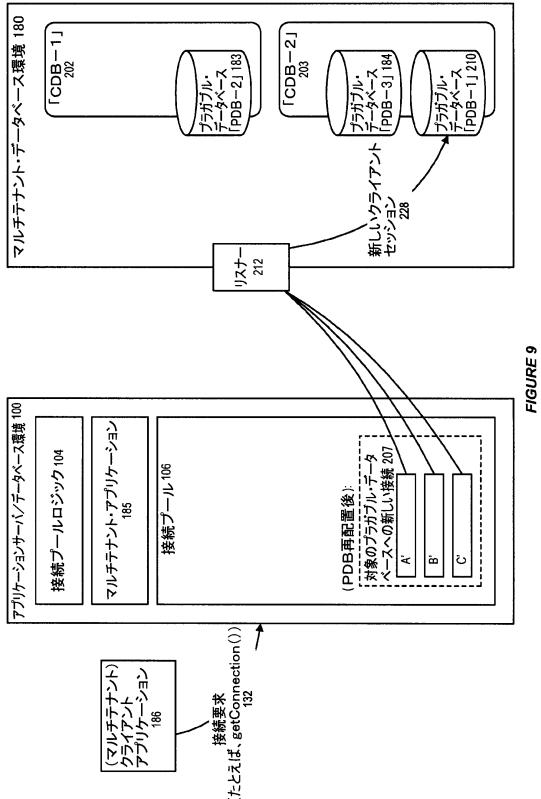
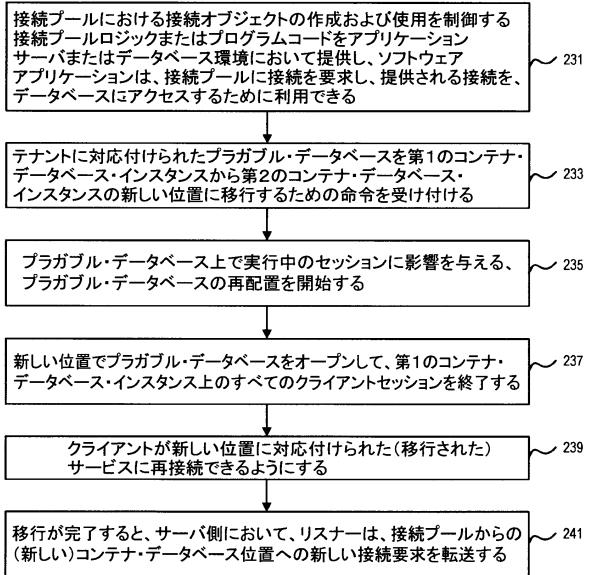


FIGURE 8

【図9】



【図10】



10

20

30

40

50

フロントページの続き

・ロード、ナンバー・18、セント・ジョンズ・ウッズ、プレスティージ・ストリート、レキシントン・タワー、オラクル・インディア・プライベート・リミテッド内

(72)発明者 ヘグド, ビディヤ

インド、560 029 バンガロール、チッカ・アダゴディ、セカンド・クロス・ロード、ナンバー・18、セント・ジョンズ・ウッズ、プレスティージ・ストリート、レキシントン・タワー、オラクル・インディア・プライベート・リミテッド内

(72)発明者 マヒダラ, チャンドラ・セカール・クリシュナ

インド、560 029 バンガロール、チッカ・アダゴディ、セカンド・クロス・ロード、ナンバー・18、セント・ジョンズ・ウッズ、プレスティージ・ストリート、レキシントン・タワー、オラクル・インディア・プライベート・リミテッド内

(72)発明者 ナマチバヤム, アラムバラータナタン

インド、560 029 バンガロール、チッカ・アダゴディ、セカンド・クロス・ロード、ナンバー・18、セント・ジョンズ・ウッズ、プレスティージ・ストリート、レキシントン・タワー、オラクル・インディア・プライベート・リミテッド内

審査官 甲斐 哲雄

(56)参考文献 米国特許出願公開第2014/0324911(US, A1)

特開2015-121924(JP, A)

Nicolas Michael, Yixiao Shen, Database Live Migration with Oracle Multitenant and the Oracle Universal Connection Pool(UCP) on Oracle Real Application Clusters(RAC), An Oracle White Paper, 米国, 2014年10月, <http://www.administrative-affairs-division.first-patent-examination-department.jpo.go.jp/joho/npl/index2.html>

(58)調査した分野 (Int.Cl., DB名)

G 06 F 16/00 - 16/958