(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7075888号
(P7075888)

(45)発行日　令和4年5月26日(2022.5.26)

(24)登録日　令和4年5月18日(2022.5.18)

(51)国際特許分類
Ｇ０６Ｆ　16/176(2019.01)

ＦＩ
Ｇ０６Ｆ　16/176

請求項の数　9　（全20頁）

(21)出願番号　特願2018-533871(P2018-533871)
(86)(22)出願日　平成29年8月3日(2017.8.3)
(65)公表番号　特表2019-528490(P2019-528490A)
(43)公表日　令和1年10月10日(2019.10.10)
(86)国際出願番号　PCT/US2017/045288
(87)国際公開番号　WO2018/027028
(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4月1日(2020.4.1)
(31)優先権主張番号　15/227,899
(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)

(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９４０６５レッドウッド・シティー，オラクル・パークウェイ５００
(74)代理人　110001195
特許業務法人深見特許事務所
(72)発明者　ドゥ・ラバルネ，ジャン
フランス、７８０００ ベルサイユ、リュ・デュ・マル・ドゥ・ラトル・ドゥ・タシニー、１６
(72)発明者　ドルゴブ，ユーリ
アメリカ合衆国、９４０６５ カリフォルニア州、レッドウッド・ショアーズ、
最終頁に続く

(54)【発明の名称】　マルチテナント・データベース環境における接続の効率的な転用のためのシステム、コンピュータで実施する方法、コンピュータプログラムおよび装置

(57)【特許請求の範囲】
【請求項１】
接続の効率的な転用のサポートを含む、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、
接続プールとを備え、前記接続プールは、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、
前記システムは、接続待機期間の値を決定し、前記接続待機期間よりも短く、最大値を条件とし、同じ所望の属性を有する接続に対するペンディング状態の要求の量に比例する、ダブル待機期間の値を動的に計算し、前記所望の属性を有する接続に対する要求を受け付けることに応答して、
前記要求に関連付けられた前記ダブル待機期間中、ダブル待機モードで動作し、前記ダブル待機モードは、
前記所望の属性を有する特定の接続が、前記接続プール内にすでに存在するが、前記要求時に借りられているかどうかを判定することと、
前記所望の属性を有する特定の接続が、前記接続プール内にすでに存在するが、前記要求時に借りられている場合、前記接続待機期間よりも短い前記ダブル待機期間中、前記所望の属性を有する特定の接続が再利用のために利用可能になるのを、待機およびポーリング

し、前記ダブル待機期間中に、前記所望の属性を有する特定の接続が再利用のために利用可能にならない場合、前記要求に関連付けられた前記ダブル待機期間が終了すると、残りの前記接続待機期間中に、前記所望の属性を有するように別の接続を作成または転用するかどうかを判定する一方で、待機とポーリングを続行することを含む、シングル待機モードに復帰することとを含む、システム。

【請求項２】

接続の効率的な転用のサポートを含む、データベースへのアクセスを提供するためのコンピュータで実施する方法であって、

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、

前記システムの制御方法は、接続待機期間の値を決定し、前記接続待機期間よりも短く、最大値を条件とし、同じ所望の属性を有する接続に対するペンディング状態の要求の量に比例する、ダブル待機期間の値を動的に計算し、前記所望の属性を有する接続に対する要求を受け付けることに応答して、

前記要求に関連付けられた前記ダブル待機期間中、ダブル待機モードで動作し、前記ダブル待機モードは、

前記所望の属性を有する特定の接続が、前記接続プール内にすでに存在するが、前記要求時に借りられているかどうかを判定することと、

前記所望の属性を有する特定の接続が、前記接続プール内にすでに存在するが、前記要求時に借りられている場合、前記接続待機期間よりも短い前記ダブル待機期間中、前記所望の属性を有する特定の接続が再利用のために利用可能になるのを、待機およびポーリングし、前記ダブル待機期間中に、前記所望の属性を有する特定の接続が再利用のために利用可能にならない場合、前記要求に関連付けられた前記ダブル待機期間が終了すると、残りの前記接続待機期間中に、前記所望の属性を有するように別の接続を作成または転用するかどうかを判定する一方で、待機とポーリングを続行することを含む、シングル待機モードに復帰することとを含む、コンピュータで実施する方法。

【請求項３】

前記特定の接続が前記ダブル待機期間内に利用可能にならなかった場合、前記接続プールは、その他の接続を転用するかどうかを判定することを含む通常の動作を再開する、請求項２に記載のコンピュータで実施する方法。

【請求項４】

シングル待機モードおよびダブル待機モードを含む互いに異なるモードを利用して、前記接続プールをポーリングすることができる、請求項２または３に記載のコンピュータで実施する方法。

【請求項５】

前記接続プールは、前記ダブル待機期間の値を決定する自動チューニング処理をサポートする、請求項２～４のいずれか１項に記載のコンピュータで実施する方法。

【請求項６】

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する、請求項２～５のいずれか１項に記載のコンピュータで実施する方法。

【請求項７】

各ソフトウェアアプリケーションは、１つ以上のテナントに対応付けられる、請求項２～６のいずれか１項に記載のコンピュータで実施する方法。

【請求項８】

コンピュータシステムに、請求項２～７のいずれか１項に記載のコンピュータで実施する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９】

請求項２～７のいずれか１項に記載のコンピュータで実施する方法を実行するための手段を備える、装置。
【発明の詳細な説明】
【技術分野】
【０００１】
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。
【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 ＥＦＦＩＣＩＥＮＴ ＲＥＰＵＲＰＯＳＩＮＧ ＯＦ ＣＯＮＮＥＣＴＩＯＮＳ ＩＮ Ａ ＭＵＬＴＩ－ＴＥＮＡＮＴ ＤＡＴＡＢＡＳＥ ＥＮＶＩＲＯＮＭＥＮＴ」と題された米国特許出願第１５／２２７，８９９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、接続プールの利用を含む、接続の効率的な転用のサポートを有する、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。
【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。
【課題を解決するための手段】
【０００５】
概要
本明細書において、接続プールの利用を含む、接続の効率的な転用のサポートを有する、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、データベースへのアクセスを可能にするために、ソフトウェアアプリケーションは、接続を提供するように要求できる。要求を受け付けることに応答して、接続プールは、まず、プール内に、要求時に借りられているまさに所望の属性を有する特定の接続がすでに存在しているかどうかを判定できる。このような接続が存在する場合、接続プールは、その特定の接続が利用可能になるまでの時間、待機できる。これを、本明細書において、ダブル待機（ｄｏｕｂｌｅ－ｗａｉｔ）と称する。次に、特定の接続がダブル待機時間内に利用可能にならなかった場合、接続プールは、たとえば、その他の接続を転用することによって、通常の動作を再開する。

【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いて接続の効率的な転用のサポートを説明する図である。
【図５】実施形態に係る、接続プール環境において接続の効率的な転用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いて接続の効率的な転用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いて接続の効率的な転用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いて接続の効率的な転用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いて接続の効率的な転用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いて接続の効率的な転用のサポートを提供する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】
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。
【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。
【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 １１２およびＢ １１４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することができ、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することができ、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。
【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 １１６、Ｄ １１８、Ｅ １２０、Ｆ １２２、Ｇ １２４、およびＮ １２６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）とラベル付けすることができ、ここでも、これらのラベル／色は、説明のために提供される。

【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、「ｇｅｔＣｏｎｎｅｃｔｉｏｎ（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　１３４（赤）と示される、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。

【００１９】

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。

【００２０】

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ことのサポート 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、ＵＣＰ）およびデータベース・ドライバ（たとえば、ＪＤＢＣ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ンク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。

【００２１】

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。

【００２２】

実施形態によると、たとえば、特定のシャード内で各ロー（ｒｏｗ）が格納される場所を決定する１つ以上のカラム（ｃｏｌｕｍｎ）としてシャード・キー（ＳＨＡＲＤ＿ＫＥＹ）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（ＣＯＮＮＥＣＴ＿ＤＡＴＡ）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ＶＡＲＣＨＡＲ２、ＣＨＡＲ、ＤＡＴＥ、ＮＵＭＢＥＲ、またはＴＩＭＥＳＴＡＭＰ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。

【００２３】

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に、各シャードは、小さな部分またはチャンクにさらに分割できる。各チャンクは、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、チャンクは、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。

【００２４】

たとえば、各チャンクをシャード・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・キーを特定のチャンクにマッピングし、そのチャンク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ンク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ンクの集まりである。

【００２５】

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。

【００２６】

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ンク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ンク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「ＤＢ　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、チャンク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、チャンク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。

【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、チャンク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、チャンク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。

【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。

【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ンクへの接続を、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。

【００３２】
たとえば、図２に示すように、実施形態によると、特定のチャンク（たとえば、チャンクＡ１）への接続は、そのチャンクへ接続する（１７４）ために使用できる。特定のシャードまたはチャンク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ンクに転用し、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ンク間のデータ分散は、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、チャンク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】
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Ｃ２、に接続要求を直接転送できる。
【００３４】
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ナーに転送できる。
【００３５】
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。
【００３６】
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。
【００３７】
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。
【００３８】
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。
【００３９】
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。
【００４０】
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１８０は、たとえば、コンテナ・データベース（ＣＤＢ）１８１、ならびに、ここでは、「ＰＤＢ－１」１８２、「ＰＤＢ－２」１８３、および「ＰＤＢ－３」１８４と示される１つ以上のプラガブル・データベース（ＰＤＢ）を備えることができる。
【００４１】
実施形態によると、各ＰＤＢは、アプリケーションサーバもしくはデータベース環境１００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として提供されるマルチテナント・アプリケーション１８５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ＰＤＢは、この例において「ＲＡＣ－インスタンス－１」および「ＲＡＣ－インスタンス－２」を含む１つ以上のＲＡＣ（Ｏｒａｃｌｅ　Ｒｅａｌ　Ａｐｐｌｉｃａｔｉｏｎ　Ｃｌｕｓｔｅｒ）インスタンス１８６、１８８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１９０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。

【００４２】

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１９２、１９４、１９６に対応付け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ＲＡＣインスタンスまたはＰＤＢへの接続を利用するために、必要であれば、サービスを切り替える（１９８）ことができる。

【００４３】

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、ＤＲＣＰ（Ｏｒａｃｌｅ　Ｄａｔａｂａｓｅ　Ｒｅｓｉｄｅｎｔ　Ｃｏｎｎｅｃｔｉｏｎ　Ｐｏｏｌ）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。

【００４４】

実施形態によると、１つのタグのみが接続ごとに対応付けられる。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、Ｂｏｏｌｅａｎ値として）。

【００４５】

プールにおける接続の効率的な転用

実施形態によると、システムは、接続の効率的な転用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、データベースへのアクセスを可能にするために、接続を提供するように要求できる。要求を受け付けることに応答して、接続プールは、まず、プール内に、要求時に借りられているまさに所望の属性を有する特定の接続がすでに存在しているかどうかを判定する。このような接続が存在する場合、接続プールは、その特定の接続が利用可能になるまでの時間、待機できる。これを、本明細書において、ダブル待機と称する。続いて、特定の接続がダブル待機時間内で利用可能にならなかった場合、接続プールは、たとえば、その他の接続を転用することによって、通常の動作を再開する。

【００４６】

従来の接続プールでは、ソフトウェアアプリケーションが（たとえば、特定の接続ラベルを有する、または、特定のセッション状態の）特定の所望の属性を有する接続を要求するたびに、接続プールの通常の処理手順では、（１）所望の属性を有する利用可能な接続があるかどうかを判定し、ない場合、（２）異なる属性を有する接続であるが、要求に応えるために、その属性を変更することによって転用できる利用可能な接続があるかどうかを判定する、または、（３）所望の属性を有する完全に新しい接続を作成しようとする。

【００４７】

許可された接続の最大数に達した場合、接続プールは、一般的に、存在しているが使用中である接続が開放されるまで待機し、開放された接続を転用する。

【００４８】

このような環境において、接続プールは、シングル待機機能をサポートする、つまり、適した接続が利用可能になるまで待機し、その後、適宜動作する。

【００４９】

しかしながら、実施形態によると、システムは、ダブル待機機能を備えることができる。ダブル待機機能は、上述のシングル待機機能の代わりにまたはそれに加えて、接続プールが、まず、要求時に借りられているまさに所望の属性を有する特定の接続がすでに存在し

ているかどうかを判定し、そうであった場合、上述した通常の処理手順を再開する前に、その接続が利用可能になるまでの時間（本明細書において、接続ダブル待機タイムアウト（Δｔｃｄｗ）と称する）、待機できるようにする。

【００５０】

ダブル待機機能では、接続を転用しなくても、または接続の属性を変更しなくても、これらの接続を再利用する可能性が増える。接続属性変更は、ラウンドトリップ／レイテンシー、およびサーバ側ＣＰＵ使用率の観点から計算コストの高い動作であるため、ダブル待機の利用は、要求に応答して接続を提供および／または転用する効率的な手段を提供する。

【００５１】

たとえば、実施形態によると、システムは、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備える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を含む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。

【００５２】

このような環境において、所定のテナントが利用可能な接続がプールにない場合、別のテナントに対応付けられた利用可能な接続を転用する（計算コストの高い動作であろう）代わりに、接続プールは、現在借りられている所定のテナント用の接続（つまり、まさに所望の属性を有する特定の接続）をプールがすでに含んでいるかどうかをまず判定することを含むダブル待機を行うことができる。そうである場合、接続プールは、その特定の（つまり、その所定のテナントの）接続が利用可能になるまでの時間、待機でき、結果、属性を変更する必要なく接続を提供できる。接続プールは、まさに所望の属性を有する接続が利用可能にならなかった場合にのみ、別のテナントの接続を転用する。

【００５３】

図４～図８は、実施形態に係る、接続プール環境において接続の効率的な転用のサポートを説明する図である。

【００５４】

上述したように、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が接続プールによって提供されるように要求できる。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。たとえば、ソフトウェアアプリケーションが、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、接続プールロジックは、新しい（赤）接続を作成する、または、存在しているアイドル状態の接続を、たとえば、（青または緑）から（赤）に転用することができる。

【００５５】

しかしながら、接続プールが行う、借りる（ボロー）動作は、通常、複雑であり、いくつかの入念にまとめられた接続選択機構、たとえば、ＲＡＣの負荷分散、高い可用性、アフィニティ、接続妥当性、ラベル付け処理、ハーベスト（ｈａｒｖｅｓｔｉｎｇ）、またはさまざまなタイムアウト処理を含む。上記例では、たとえば、（青）接続の（赤）接続への変形を転用が含み得ることを説明するために互いに異なる色を使用したが、実際行われているところでは、この変形は、一方のＲＡＣインスタンス上の接続をクローズして他方のＲＡＣインスタンス上で代用の接続を作成することを必要とし得、このような動作を実装するために必要なさまざまな処理手順もあわせて必要とし得る。

【００５６】

計算コストが高くない属性について、これは、比較的簡単な処理かもしれない。しかしながら、計算コストの高い属性変更、たとえば、マルチテナント環境において実装され得るようなコンテナ・データベース内のプラガブルデータベースの変更は、より熟考された手法を必要とする。

【００５７】

図４に示すように、実施形態によると、ダブル待機機能２５０は、接続プールが待機を行い（２５２）、その後、適宜、待機後に接続の転用を行う（２５４）ために使用することができる。この処理については、さらに詳細に後述する。

【００５８】

実施形態によると、上述の手法は、ボロー要求に一致した接続があるかもしれず、且つ、借りているスレッドがこの接続をダブル待機中であっても、接続ダブル待機タイムアウト（Δｔｃｄｗ）内に接続が返される保証がないと認識している。

【００５９】

これに対処するために、実施形態によると、接続ダブル待機タイムアウトは、接続待機タイムアウト（Δｔｃｗ）よりも短く設定でき、結果、ダブル待機が失敗した場合にシステムが通常の処理手順、つまり、シングル待機のボローイング・モードに復帰できる。

【００６０】

たとえば、実施形態によると、（合計の）接続待機タイムアウトまたは時間（Δｔｃｗ）は５００ミリ秒として設定されるとともに、接続ダブル待機タイムアウト（Δｔｃｄｗ）は、５０ミリ秒に設定できる。

【００６１】

これに加えて、実施形態によると、上述の手法は、一般に、借りられている接続がいつプールに返却されるかをシステムが予測できないだろうと認識している。しかしながら、このような情報のためのシグナル・イベントを当てにすることは、性能面でコストが高くなるだろう。たとえば、返却された接続ごとにシステムがシグナルＡＰＩを待つ予定である場合、これによって、接続を借りる／返却する機構の性能が落ちてしまうだろう。

【００６２】

これに対処するために、実施形態によると、接続プールは、少ない数のシグナルＡＰＩ待機呼び出しをプールのビジーウェイト（ｂｕｓｙ－ｗａｉｔｉｎｇ）ポーリングと組み合わせた、必要な接続を獲得するためのボロー動作をサポートする。一般的に、合図が送られることをボロー動作が待機する唯一の状況は、そのときに借りられる接続が何もないときである。妥当な利用可能な接続の出現が、現在借りているスレッドのすべてがポーリングを開始することをトリガーする。

【００６３】

出来る限り速やかに適切な接続を獲得しようとするために、一般的に、スレッドは、短い（たとえば、ナノ秒、またはゼロ時間）間隔でプールをポーリングすべきである。しかしながら、実際には、ゼロ時間のポーリング間隔では、膨大なＣＰＵ使用率になり得る。これに対処するために、実施形態によると、接続プールは、ＣＰＵ負荷とこれらの間隔の持続時間との間のバランスを取るための、さらに後述するが、自動チューニング処理をサポートする。

【００６４】

たとえば、実施形態によると、ポーリング間のスリープ間隔は、以下を用いて演算できる。

【００６５】

【数１】

$$\Delta t_{sleep} = \Delta t_0(\left|CPU_{expected} - CPU_{current}\right|)$$

【００６６】

Δｔ０は、経験的一定間隔であり、ＣＰＵｅｘｐｅｃｔｅｄは、予想されるＣＰＵ使用率であり、ＣＰＵｃｕｒｒｅｎｔは、現在のＣＰＵ使用率である。実施形態によると、ＣＰＵ使用率の値は、すべての参加スレッドのＣＰＵ使用率の積分値である。

【００６７】

シングル待機およびダブル待機モード

実施形態によると、システムは、２つの異なるモードを利用して接続プールをポーリングするように構成される。これらのモードは、本明細書において、全体的に、接続シングル

待機モードおよび接続ダブル待機モードと称する。

【００６８】

接続シングル待機モードにおいて、接続プールロジックは、利用可能になる接続のすべてが予約され（たとえば、（赤）、（青）、または（緑）接続のうちのいずれか。特定の接続の種類であるかは問わない）、必要に応じて転用され（たとえば、（赤）接続が（緑）に変形される）、そして借りられることをボロー要求が期待すると想定する。このシングル待機モードは、接続待機タイムアウト（Δｔｃｗ）の間、継続し、構成可能な接続プールプロパティとして定義できる。

【００６９】

接続ダブル待機モードにおいて、接続プールは、ボロー要求が、要求された特性または種類の接続、つまり、まさに所望の属性（たとえば、（赤）接続のみ）を有する接続しか期待しないと想定し、結果、転用は必要とされない。このダブル待機モードは、接続ダブル待機タイムアウト（Δｔｃｄｗ）の間、継続し、構成可能な接続プールプロパティと定義することができるまたはシステムによって自動的に／動的に算出される。

【００７０】

定義上、接続待機タイムアウト（Δｔｃｗ）は、借りているスレッドが接続を求めてプールをポーリングする最長期間である。

【００７１】

実施形態によると、システムは、まずダブル待機モードでポーリングし、その後、シングル待機モードでポーリングするように構成できる。ダブル待機判定が、Δｔｃｄｗ時間内に接続が開放されると保証しない場合、（Δｔｃｄｗ＝Δｔｃｗ）と設定することは、多くの状況において、転用せざるを得なくなるまでに何も得られないまま時間がかかってしまうことになるため、あまり有用ではない。逆に、Δｔｃｄｗの値＝０に設定することは、ダブル待機モードをオフにすることを意味するだろう。

【００７２】

上記は、実施形態によると、たとえば、以下のように、ダブル待機タイムアウトΔｔｃｄｗの時間が、概ね、ある範囲の間で変動すべきであることを示している。

【００７３】

【数２】

$$0 \leq \Delta t_{cdw} \leq \Delta t_{cw}$$

【００７４】

Δｔｃｄｗの値≦Δｔｃｗであると、任意の接続を求めてシングル待機し、ダブル待機が失敗した場合、その接続を転用するために費やせる、ゼロ以外の期間がある。

【００７５】

実施形態によると、ダブル待機タイムアウトΔｔｃｄｗの値は、構成可能なプロパティとして定義できる。上述したように、別の手法は、自動チューニング処理を利用することである。たとえば、ダブル待機期間を、統計的な平均ボロー（「ａｂ」）期間Δｔａｂに比例するようにできる。これは、以下を用いて演算できる。

【００７６】

【数３】

$$\Delta t_{ab} = \frac{\sum_{i=0}^{i=n} \Delta t_i}{n}$$

【００７７】

ｎは、ある長さの期間の直近のΔｔｉープールにある必要な種類の接続のすべてについての個々の（ｉ）ボロー期間（つまり、接続を借りてからプールへ返却するまでの間に経過した時間）である。Δｔａｂの値は、ペンティング状態のボロー要求（Ｎｃ）に当てはま

る、借りられている接続に反比例し、同様のペンティング状態のボロー要求、つまり、同じ種類（Ｎｂｒ）の接続を求める要求の数に比例する。

【００７８】

実施形態によると、ダブル待機タイムアウト（Δｔｃｄｗ）の値は、自動チューニング処理に応じて、システムによって、以下のように、自動的に／動的に算出できる。

【００７９】

【数４】

$$\Delta t_{cdw} = \min\left(F\Delta t_{cw}, \frac{N_{br}\Delta t_{ab}}{N_c}\right)$$

【００８０】

Ｆは、［０．．１］の範囲内のΔｔｃｄｗ／Δｔｃｗの最大許容値である。

各ポーリングの前にΔｔｃｄｗの値をチューニングすることも可能であるが、実際には、この手法では、必要な種類の借りられている接続の数を判定するための分析を必要とするだろうし、いくらかのリソースを必要とするだろう。代わりに、システムは、特定のボロー動作を行うたびにΔｔｃｄｗを１回演算して、その特定のボロー動作用の定数として利用できる。

【００８１】

たとえば、図５に示すように、実施形態によると、接続プールは、特定の種類の（たとえば、（青）ラベルの）接続の要求を受け付ける（２６０）ことができる。

【００８２】

図６に示すように、実施形態によると、システムは、ダブル待機機能を備えることができる。ダブル待機機能は、上述のシングル待機機能の代わりにまたはそれに加えて、接続プールが、要求時に借りられているまさに所望の属性を有する特定の接続がすでに存在しているかどうかをまず判定し、そうであった場合、その接続が利用可能になるまでの時間、待機できるようにする。ダブル待機タイムアウト（Δｔｃｄｗ）内に所望の接続が利用可能になると、使用のために所望の接続がクライアントアプリケーションに返される（２６２）。

【００８３】

図７に示すように、実施形態によると、ダブル待機タイムアウト（Δｔｃｄｗ）内に適した接続が利用可能にならなかった場合、接続プールは、シングル待機モードの間、通常の処理手順を再開できる。

【００８４】

図８に示すように、実施形態によると、（残りの、またはシングル待機）接続タイムアウト（Δｔｃｗ）の間、接続プールは、存在しているアイドル状態の接続を、この例において、（青）接続２６６に転用し（２６４）、当該接続をクライアントアプリケーションに返す（２６８）かどうかを判定できる。

【００８５】

以下の追加例によって、上記を説明する。

例１：Δｔｃｗが１０秒であり、２種類の接続、５０個の（緑）接続および５０個の（青）接続がある、１００個の接続を有するプールを考える。現在、５０個すべての（緑）接続と、２５個の（青）接続とが借りられており、２５個の（青）接続が利用可能であり、１５０個の（緑）を借りる要求がペンディング状態である。Δｔａｂは、現在、１秒である。Ｆが０．５である例を考える。すると、Δｔｃｄｗ＝ｍｉｎ（（０．５×１０），（（１５０＋１）／５０））、つまり、３秒に等しい。ボロー要求は、まさに必要な種類の接続を期待して３秒間ダブル待機して、任意の接続を獲得するために別の７秒間待機し、当該任意の接続を転用する。

【００８６】

例２：１５０個ではなく３００個のボローイング要求がある、同じ入力を用いた上述の例

を考える。すると、Δｔｃｄｗ＝ｍｉｎ（（０．５×１０），（（３００＋１）／５０））、つまり、５秒に等しい。これは、（緑）接続を借りるために、借りているスレッドが５秒間ダブル待機し、借りられていない場合、必要であれば、転用のために別の５秒間シングル待機することを意味する。

【００８７】

例３：３００個ではなく５０００個のボローイング要求がある、同じ入力を用いた上述の例を考える。Δｔｃｄｗ＝５秒。これは、例２と同じ挙動になる。

【００８８】

接続ボロー処理

図９は、実施形態に係る、接続プール環境において接続の効率的な転用のサポートをさらに説明する図である。

【００８９】

図９に示すように、実施形態によると、接続ボロー動作は、ダブル待機ポーリング準備ステップ２８０から、ダブル待機ポーリング周期２８２へ、シングル待機ポーリング準備ステップ２９０へ、シングル待機ポーリング周期２９２へ、１つ以上のボロー後動作ステップ３００へと続く、と見ることができる。各々については、さらに後述する。

【００９０】

ダブル待機ポーリングの準備

実施形態によると、ダブル待機ポーリングの準備は、以下を含み得る。

【００９１】

ａ．Δｔｃｄｗを演算する。

ｂ．現在のタイムスタンプを保存して、ダブル待機ポーリングを開始する。

【００９２】

ダブル待機ポーリング周期

実施形態によると、ダブル待機ポーリング周期は、以下を含み得る。

【００９３】

ａ．ダブル待機ポーリングのための期間として、Δｔｃｄｗを決定する。この期間がすでに経過しているかどうかをチェックする。経過している場合、ダブル待機ポーリング周期を出て、シングル待機ポーリングの準備を開始する。

【００９４】

ｂ．適切な種類の利用可能な妥当な接続を求めてポーリングする。適切な種類の利用可能な妥当な接続が見つかった場合、ボロー後動作へジャンプし、ボロー要求は、終了する。見つからなかった場合、次のステップに移動する。

【００９５】

ｃ．Δｔｓｌｅｅｐを演算する。

ｄ．スリープして、（ａ）に移動する。

【００９６】

シングル待機ポーリングの準備

実施形態によると、シングル待機ポーリングの準備は、以下を含み得る。

【００９７】

ａ．Δｔｃｗ－Δｔｃｄｗを演算する。

ｂ．現在のタイムスタンプを保存して、シングル待機ポーリングを開始する。

【００９８】

シングル待機ポーリング周期

実施形態によると、シングル待機ポーリング周期は、以下を含み得る。

【００９９】

ａ．Δｔｃｗ－Δｔｃｄｗは、シングル待機ポーリングのための期間である。この期間がすでに経過しているかどうかをチェックする。経過している場合、シングル待機周期を出て、ボロー後動作の実行を開始する。

【０１００】

ｂ．あらゆる種類の利用可能な妥当な接続を求めてポーリングする。任意の種類の利用可能な妥当な接続が見つかった場合、ボロー後動作へジャンプする。ボロー要求は、終了する。見つからなかった場合、次のステップに移動する。

【０１０１】

ｃ．接続プールを大きくできるスペースがある場合、必要な種類の接続とともに接続プールを大きくする。新しい接続ができると、それを取得して、ボロー後動作へジャンプする。ボロー要求は終了する。そうでない場合、次のステップに移動する。

【０１０２】

ｄ．Δｔｓｌｅｅｐを演算する。

ｅ．スリープして、（ａ）に移動する。

【０１０３】

ボロー後動作

実施形態によると、ボロー後動作は、以下を含み得る。

【０１０４】

ａ．接続に対してボロー後動作を行う。

ｂ．接続またはｎｕｌｌを返す。

【０１０５】

実施形態によると、ダブル待機機構のための接続返却動作について、通常の接続返却ロジックが必要に応じで動作し、接続はもはや必要でないため、アプリケーションから返却される。返却は、利用可能な接続の数をインクリメントし、プールにおいて接続が利用可能であると印をつけ、プールが少なくとも１つの利用可能な接続を求めて待機している場合、プールに信号を送る。

【０１０６】

図１０は、実施形態に係る、接続プール環境において接続の効率的な転用のサポートを提供する方法を説明する図である。

【０１０７】

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。

【０１０８】

ステップ３１２において、接続プールと共に使用するためのダブル待機機能が提供され、所望の属性を有する特定の接続の要求に応答して、接続プールは、まず、接続プール内に要求時に借りられているまさに所望の属性を有する接続がすでに存在しているかどうかを判定し、そうであった場合、接続プールは、ダブル待機タイムアウトまで、または、その接続が利用可能になるまでの時間、待機する。

【０１０９】

ステップ３１４において、ダブル待機タイムアウトまたは時間が経過すると、接続プールは、異なる属性を有する利用可能な接続であるが、その属性を変更することによって要求に応えるために転用できる接続があるかどうかを判定するために、または、所望の属性を有する完全に新しい接続を作成しようとして、シングル待機モードに復帰する。

【０１１０】

ステップ３１６において、接続プールは、まさに所望の属性を有する接続、転用された接続、新しい接続、のうちの１つを提供する、または、ｎｕｌｌもしくは他の値を返す。

【０１１１】

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なように、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。

【０１１２】

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ＩＣＳ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。

【０１１３】

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。

【図面】

【図１】

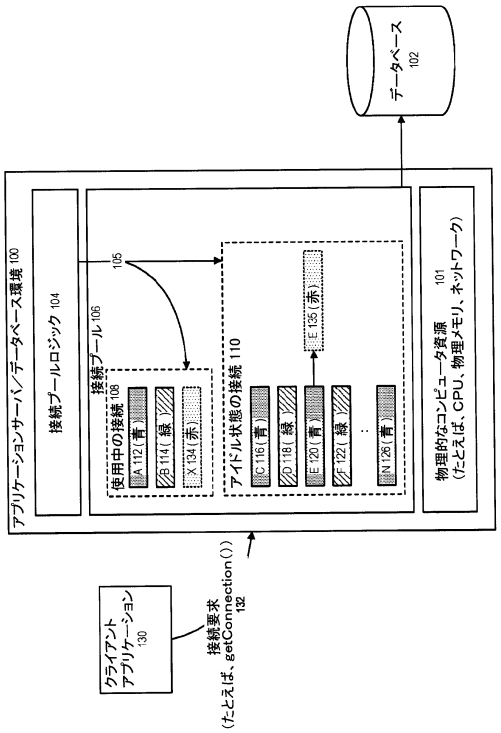

FIGURE 1

【図２】

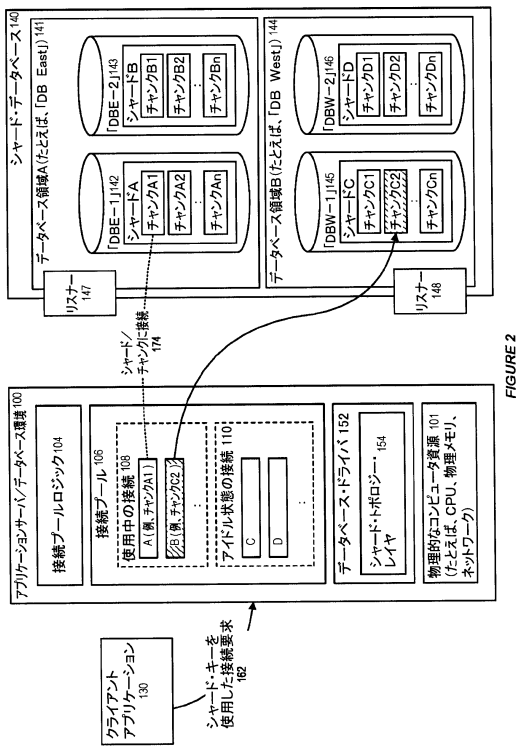

FIGURE 2

【図３】

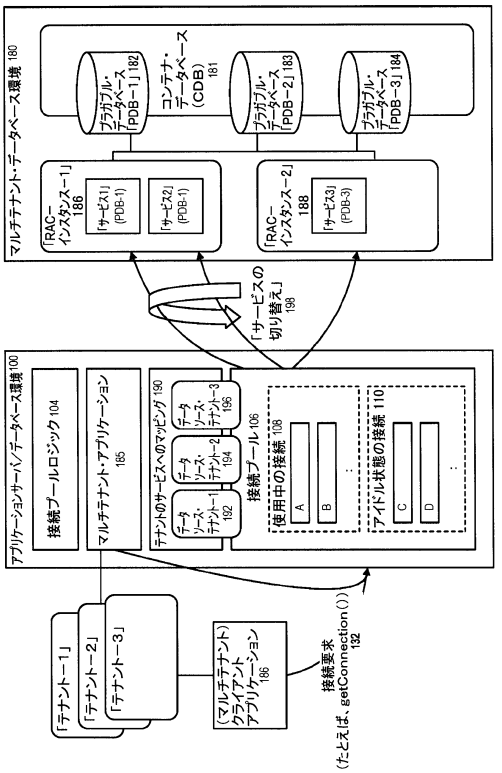

FIGURE 3

【図４】

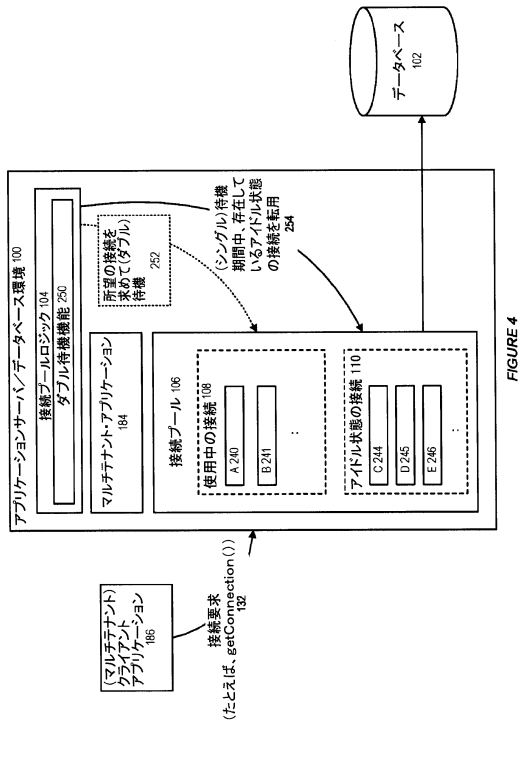

FIGURE 4

【図５】

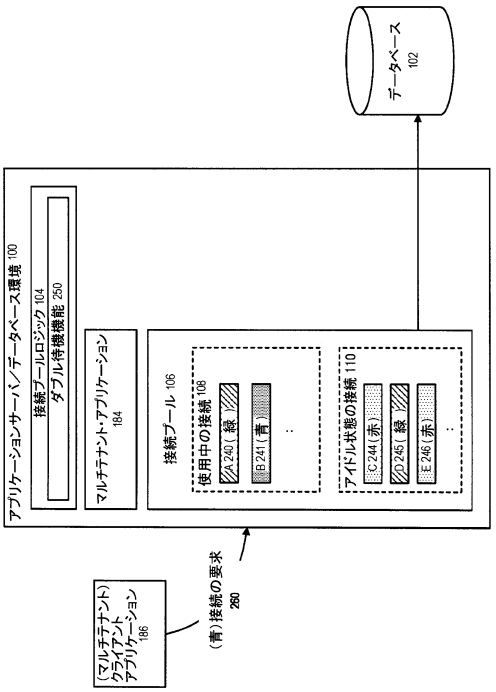

FIGURE 5

【図６】

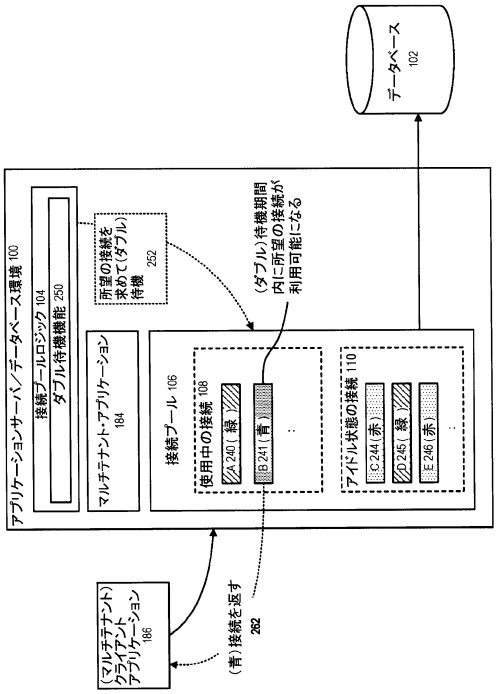

FIGURE 6

【図７】

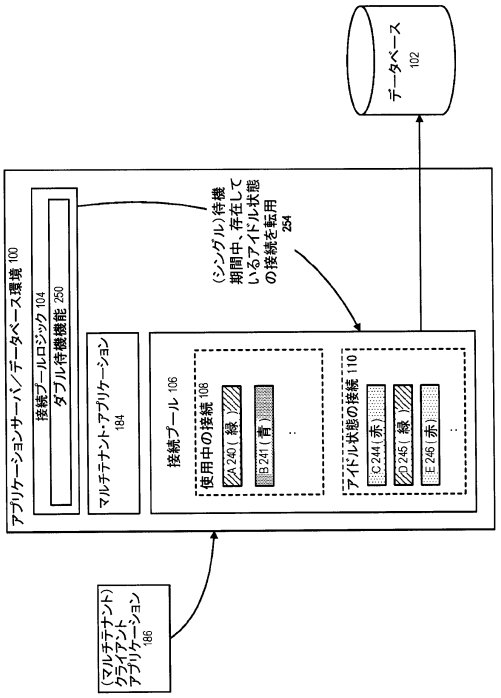

FIGURE 7

【図８】

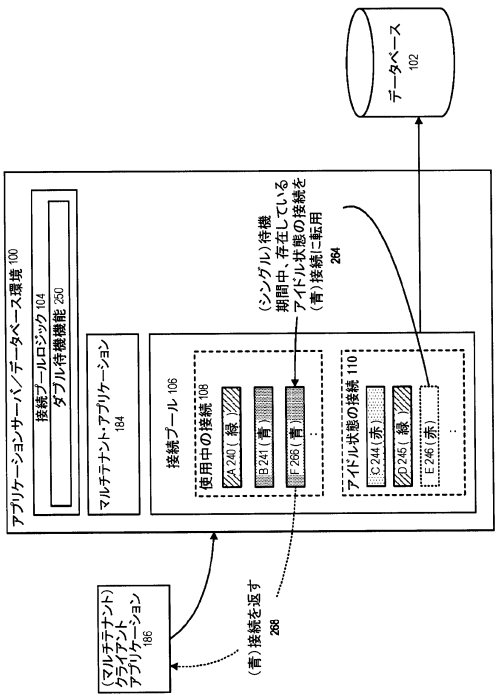

FIGURE 8

【図９】

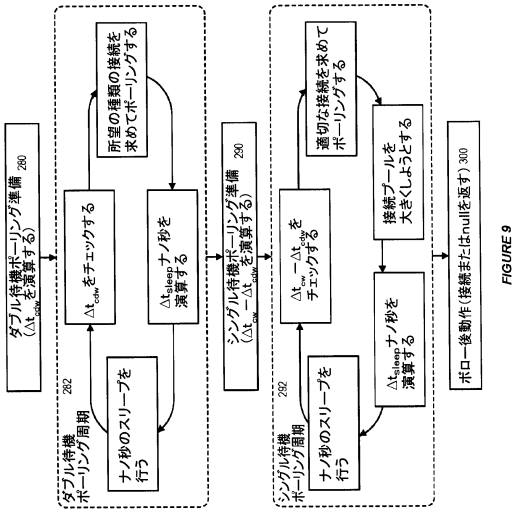

FIGURE 9

【図１０】

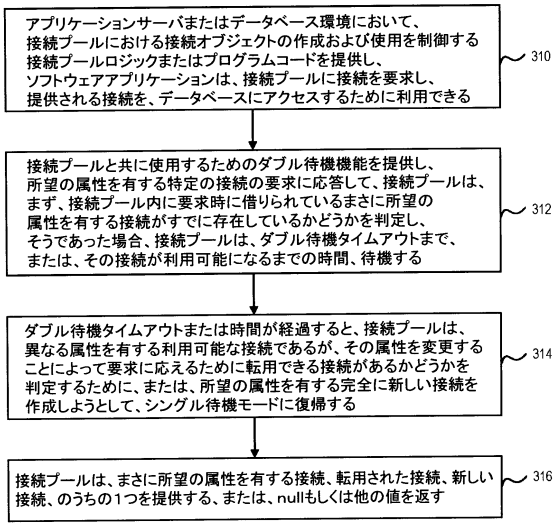

FIGURE 10

フロントページの続き

オラクル・パークウェイ、５００、エム／エス・５・オゥ・ピィ・７
(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　バーマ，サウラブ
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　木村 大吾
(56)参考文献　米国特許出願公開第２０１４／０３２４９１１（ＵＳ，Ａ１）
米国特許出願公開第２００８／０２５０４１９（ＵＳ，Ａ１）
特開２０１３－２３９１９９（ＪＰ，Ａ）
(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１６／００－１６／９５８