

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号
特許第5380540号
(P5380540)

(45) 発行日 平成26年1月8日(2014.1.8)

(24) 登録日 平成25年10月4日(2013.10.4)

(51) Int.Cl.

F I

G O 6 F 9/445 (2006.01)

G O 6 F 9/44 (2006.01)

G O 6 F 11/34 (2006.01)

G O 6 F 9/06 6 1 O L

G O 6 F 9/06 6 2 O A

G O 6 F 11/34 B

請求項の数 22 (全 35 頁)

(21) 出願番号	特願2011-529112 (P2011-529112)	(73) 特許権者	500046438
(86) (22) 出願日	平成21年9月15日 (2009.9.15)		マイクロソフト コーポレーション
(65) 公表番号	特表2012-503827 (P2012-503827A)		アメリカ合衆国 ワシントン州 9805
(43) 公表日	平成24年2月9日 (2012.2.9)		2-6399 レッドモンド ワン マイ
(86) 国際出願番号	PCT/US2009/057042		クロソフト ウェイ
(87) 国際公開番号	W02010/036539	(74) 代理人	100140109
(87) 国際公開日	平成22年4月1日 (2010.4.1)		弁理士 小野 新次郎
審査請求日	平成24年9月4日 (2012.9.4)	(74) 代理人	100075270
(31) 優先権主張番号	12/239, 184		弁理士 小林 泰
(32) 優先日	平成20年9月26日 (2008.9.26)	(74) 代理人	100101373
(33) 優先権主張国	米国 (US)		弁理士 竹内 茂雄
		(74) 代理人	100118902
			弁理士 山本 修
		(74) 代理人	100153028
			弁理士 上田 忠

最終頁に続く

(54) 【発明の名称】 データ層アプリケーションコンポーネント

(57) 【特許請求の範囲】

【請求項 1】

コンピュータに、データ層アプリケーションコンポーネント(DAC)を配置するための方法を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体であって、前記方法は、

コンパイル可能なDACモデルに基づいてDACパッケージをビルドするステップであって、前記DACパッケージをビルドすることは、前記DACモデルをコンパイルすることを含む、ステップと、

前記DACパッケージからサーバファブリックに複数のDAC型をインポートするステップであって、前記複数のDAC型のそれぞれは、前記DACの論理実装を定義し、前記サーバファブリックは、少なくとも1つのDAC、1または複数のデータベースサーバランタイム、および少なくとも1つのハードウェアリソースをモデル化する、ステップと、

前記サーバファブリックにおいて、前記複数のDAC型のうち1つのDAC型を、前記1または複数のデータベースサーバランタイムに配置して、DACインスタンスを作成するステップであって、前記DACインスタンスの作成は、少なくとも前記DACインスタンスの2以上のプロパティおよび配置に対して特有のオブジェクトを定義する配置スキーマを含む情報を保存することを含み、前記2以上のプロパティは、一意な識別子プロパティおよび配置名称プロパティを含む、ステップと

を含むことを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項 2】

前記 D A C 型は、少なくともプロパティ、アプリケーションスキーマおよび配置ポリシーを含むことを特徴とする請求項 1 に記載のコンピュータ読み取り可能な記録媒体。

【請求項 3】

前記 D A C 型は、ユーザが、不変条件、要求、および意図される前記 D A C の使用を定義するための機構を提供することを特徴とする請求項 2 に記載のコンピュータ読み取り可能な記録媒体。

【請求項 4】

前記 D A C インスタンスが、配置ポリシー、ファブリックマッピング、およびデータをさらに含むことを特徴とする請求項 1 に記載のコンピュータ読み取り可能な記録媒体。

10

【請求項 5】

データベース要素を一組の支援コンピュータリソースに関連付けるためのコンピュータ実装システムであって、少なくとも 1 つのプロセッサ、および前記プロセッサにデータ層アプリケーションコンポーネント (D A C) を配置するための方法を実行させるためのプログラムを記録したメモリを備え、前記方法は、

データベース要素のコレクションを含むコンパイル可能な D A C モデルを作成するステップであって、前記データベース要素のコレクションは、

1 または複数のデータベースオブジェクトに関するメタデータであって、前記メタデータは、各前記データベースオブジェクトの識別子を含む、メタデータと、

前記データベースオブジェクトを操作するよう動作可能なデータベースランタイムリソースのデータベースランタイムリソース識別子と、

20

前記データベースランタイムリソースの実行に関するポリシーを含む、ステップと、

前記 D A C モデルに基づいて D A C パッケージをビルドするステップであって、前記 D A C パッケージをビルドすることは、前記 D A C モデルをコンパイルすることを含む、ステップと、

前記 D A C パッケージからサーバファブリックに複数の D A C 型をインポートするステップであって、前記複数の D A C 型のそれぞれは、前記 D A C の論理実装を定義し、前記サーバファブリックは、少なくとも 1 つの D A C 、 1 または複数のデータベースサーバランタイム、および少なくとも 1 つのハードウェアリソースをモデル化する、ステップと

30

、
前記サーバファブリックにおいて、前記複数の D A C 型のうち 1 つの D A C 型を、前記 1 または複数のデータベースサーバランタイムに配置して、D A C インスタンスを作成するステップであって、前記 D A C インスタンスの作成は、少なくとも前記 D A C インスタンスの 2 以上のプロパティおよび配置に対して特有のオブジェクトを定義する配置スキーマを含む情報を保存することを含み、前記 2 以上のプロパティは、一意な識別子プロパティおよび配置名称プロパティを含む、ステップと

を含むことを特徴とするシステム。

【請求項 6】

前記データベース要素のコレクション内の 1 または複数のデータベース要素は修正されるが、D A C 名は不変であることを特徴とする請求項 5 に記載のシステム。

40

【請求項 7】

前記データベース要素のコレクション、または、1 もしくは複数のランタイムリソースが変更可能であり、前記データベース要素のコレクションに関連付けられる D A C 名は不変であり、1 または複数のデータベースアプリケーションのアプリケーションの接続文字列を、前記データベース要素のコレクションに対する変更に応答して記録する必要が無いことを特徴とする請求項 5 に記載のシステム。

【請求項 8】

前記 D A C 名を使用する前記データベースアプリケーションが、異なるネットワークコンピュータ上に置かれることを特徴とする請求項 7 に記載のシステム。

50

【請求項 9】

前記データベースランタイムリソースの実行に関するポリシーが、前記データベース要素のコレクションに関連付けられるDAC名に適用され、および、前記データベース要素のコレクション、または、1もしくは複数のランタイムリソースが変更される場合に、前記ポリシーは継続して適用されることを特徴とする請求項5に記載のシステム。

【請求項 10】

前記データベース要素のコレクションとインターフェースをとるAPIをさらに備え、前記APIは、前記データベース要素のコレクション内の1または複数の要素の利用を測定することができることを特徴とする請求項5に記載のシステム。

【請求項 11】

データ層アプリケーションコンポーネント(DAC)を配置するための方法であって、データベース要素のコレクションを含むコンパイル可能なDACモデルを作成するステップであって、前記データベース要素のコレクションは、

1または複数のデータベースオブジェクトに関するメタデータであって、前記メタデータは、各前記データベースオブジェクトの識別子を含む、メタデータと、

前記データベースオブジェクトを操作するよう動作可能なデータベースランタイムリソースのデータベースランタイムリソース識別子と、

前記データベースランタイムリソースの実行に関するポリシーを含む、ステップと、

前記DACモデルに基づいてDACパッケージをビルドするステップであって、前記DACパッケージをビルドすることは、前記DACモデルをコンパイルすることを含む、ステップと、

前記DACパッケージからサーバファブリックに複数のDAC型をインポートするステップであって、前記複数のDAC型のそれぞれは、前記DACの論理実装を定義し、前記サーバファブリックは、少なくとも1つのDAC、1または複数のデータベースサーバランタイム、および少なくとも1つのハードウェアリソースをモデル化する、ステップと、

前記サーバファブリックにおいて、前記複数のDAC型のうち1つのDAC型を、前記1または複数のデータベースサーバランタイムに配置して、DACインスタンスを作成するステップであって、前記DACインスタンスの作成は、少なくとも前記DACインスタンスの2以上のプロパティおよび配置に対して特有のオブジェクトを定義する配置スキーマを含む情報を保存することを含み、前記2以上のプロパティは、一意な識別子プロパティおよび配置名称プロパティを含む、ステップと

を含むことを特徴とする方法。

【請求項 12】

前記DACモデルを表現するものをDACプロジェクトとして保存するステップと、

前記DACプロジェクトをソースコントロールシステムまたはソースコントロールデータベースにチェックインして、前記DACプロジェクトを記憶するステップと

をさらに含むことを特徴とする請求項11に記載の方法。

【請求項 13】

前記DACパッケージをビルドする前に、前記DACモデルをエディットし、前記DACモデルを認証するステップと、

前記DACインスタンスを配置する前に前記DAC型を構成するステップと、

前記DACインスタンスを特定の組の物理的コンピュータリソースに対してインストールするステップと

をさらに含むことを特徴とする請求項11に記載の方法。

【請求項 14】

データベースオブジェクトに関するメタデータをパッケージ化する、前記DACの管理の一単位が、第1の物理的コンピュータリソースに対してマッピングされることを特徴とする請求項11に記載の方法。

【請求項 15】

前記第1の物理的コンピュータリソースが、プロセッサ、メモリストレージデバイス、物理的ストレージデバイス、ハードディスク、コンパクトディスク、ハードドライブ、およびレジスタの内の1つまたは複数を含むことを特徴とする請求項14に記載の方法。

【請求項 16】

前記管理の一単位が、前記第1の物理的コンピュータリソースから第2の物理的コンピュータリソースに再マッピングされることを特徴とする請求項14に記載の方法。

【請求項 17】

前記データベースランタイムリソースの実行に関するポリシーは、リソース制限ポリシー、または、ソフトウェアリソースもしくはハードウェアリソースの要求に関するポリシーであることを特徴とする請求項11に記載の方法。

10

【請求項 18】

前記管理の一単位は、関連するDAC名により識別され、前記DAC名はデータベースアプリケーションによる使用のためにアクセス可能であることを特徴とする請求項14に記載の方法。

【請求項 19】

前記データベースオブジェクトが、スキーマ、テーブル、ビュー、ストアドプロシージャ、関数、トリガ、データ型、インデックス、ログイン、ユーザ、許可、ポリシー、統計値、主キー、外部キー、デフォルト、およびチェックの制約の内の1つまたは複数を含むことを特徴とする請求項11に記載の方法。

20

【請求項 20】

前記管理の一単位は、DAC型、DACインスタンスを含み、前記DAC型は、少なくともプロパティ、アプリケーションスキーマ、および配置ポリシーを含むことを特徴とする請求項14に記載の方法。

【請求項 21】

前記DAC型は、ユーザが、不変条件、要求、および意図される前記DACの使用を定義するための機構を提供することを特徴とする請求項20に記載の方法。

【請求項 22】

前記DACインスタンスが、プロパティ、配置スキーマ、配置ポリシー、ファブリックマッピング、およびデータを含むことを特徴とする請求項20に記載の方法。

30

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、データ層アプリケーションコンポーネントに関する。

【背景技術】**【0002】**

アプリケーションのオーサリング、インストールおよび実行時を含むアプリケーションのライフサイクル中、一般的に、多くの異なるオブジェクトが作成され、あるいは、アプリケーションに関連付けられる。例えば、プログラムコード、実行コード、ソフトウェアリソース、および、ハードウェアリソースが、アプリケーションに関連付けられ、また、このアイテムそれぞれが、種々のファイル名およびファイル型およびファイル形式に割り当てられる。データベースアプリケーションを、プログラムコード、実行コードおよびリソースに関連付ける他に、ログイン、テーブル、ストアドプロシージャおよびデータベースポリシー等の追加のオブジェクトに関連付けることができる。

40

【0003】

トラッキングおよび維持の対象となるオブジェクトが多いと、データベースアプリケーションにそのライフサイクルを通して関連付けられる全てのアイテムを、データベース管理者が効果的に管理することは、難しくかつ時間がかかる場合が多い。例えば、データベース管理者が、データベースアプリケーションにより使用されるハードウェアまたは他のアイテムに変更を加えたい場合、または、データベースアプリケーションがインストール

50

されるデータベースサーバインスタンスを追加または変更したい場合、所望の変更を加える処理は、時間がかかり、かつ高コストなものになってしまう。オブジェクトは、コード化することができ、または、オブジェクトが常駐するハードウェアリソースもしくはそこで動作するハードウェアリソースに直接リンクさせることができる。また、そのようなオブジェクトを他の場所に移動させることは、多大な再プログラミングを伴うことになる。アプリケーションオブジェクトが複数の場所に記憶される場合、または、単一のデータベースが複数のアプリケーションにより使用されるオブジェクトを記憶する場合、オブジェクトの移動または追加のインストールに伴う労力が増大する。

【発明の概要】

【0004】

データベースアプリケーションのライフサイクルの段階での高度な管理の生産性を達成するために、データベースアプリケーションを構成するアイテム、またはアプリケーションにより使用されるアイテムを関連付ける、グルーピング抽象化を有することは有益である。ソフトウェアのグルーピング抽象化が無いと、データベースアプリケーションの識別、概念化、移動、または複製、あるいは管理は非常に難しい。本明細書に開示される実施形態は、例えば、データベースアプリケーションを全体として移動または複製することを許可する、データベースオブジェクトのメタデータに関連するコンテナ内の、データベースアプリケーションの要素を関連付ける。コンテナは、ランタイムリソース識別子をデータベース要素に関連付け、このランタイムリソース識別子により、ハードウェアもしくは他のランタイムリソースの変更を可能にし、または、データベースアプリケーションの要素を選択して、要素を再コード化することなく異なるリソースを使用することを可能にする。

【0005】

加えて、複数のデータベースアプリケーションコンテナは自身を、全体として移動、複製、または管理することを可能にする単一のグルーピングまたはファブリックに、関連付けることができる。ファブリックは、データベースアプリケーションの要素により使用されるリソース識別子を参照して、複数のデータベースアプリケーションの管理を可能にする。

【0006】

特定の実施形態に従って、コンピュータ可読媒体が、データベースオブジェクトに関するメタデータをパッケージ化する管理の一単位を含む、DAC (data-tier application component: データ層アプリケーションコンポーネント) を記憶する。データベースオブジェクトに関するメタデータには、データベースオブジェクトの識別子、データベースオブジェクトを操作するよう動作可能なデータベースランタイムリソースのデータベースランタイムリソース識別子、および、データベースランタイムリソースの実行に関するポリシーが含まれる。

【0007】

本概要は、以下の「発明を実施するための形態」でさらに説明される概念から選択したものを簡略化した形式で紹介するために提供するものである。本概要は、特許請求の範囲に記載されている主題の重要な特徴または本質的な特徴を特定することを意図するものでも、特許請求の範囲に記載されている主題の範囲を決定する際の一助として使用することを意図するものでもない。

【図面の簡単な説明】

【0008】

【図1】DACを使用するデータベース開発ツールを例示するブロック図である。

【図2】DACおよびDACパッケージを使用してDACインスタンスをインストールする方法を例示するフロー図である。

【図3】DACの特定の実施形態を例示する概略図である。

【図4】関連するDAC名を有するDACの第2の実施形態を例示するブロック図である。

【図 5】一組の D A C を含むサーバファブリックの特定の実施形態を例示するブロック図である。

【図 6】一組の D A C を含むサーバファブリックの第 2 の実施形態を例示するブロック図である。

【図 7】サーバファブリックのエンティティにおいてデータベース管理アクションおよびポリシーを適用する方法を例示するフロー図である。

【図 8】サーバファブリックのエンティティの関係を例示するフロー図である。

【図 9】データベースアプリケーションを管理するシステムの実施形態のブロック図である。

【図 10】サーバファブリック要素を処理する方法を例示するフロー図である。

10

【図 11】S Q L インスタンスを管理するためのシステムの第 1 の特定の実施形態を例示するブロック図である。

【図 12】S Q L インスタンスを管理するためのシステムの第 2 の特定の実施形態を例示するブロック図である。

【図 13】S Q L インスタンスを管理するためのシステムの第 3 の特定の実施形態を例示するブロック図である。

【図 14】S Q L インスタンスを管理するためのシステムの第 4 の特定の実施形態を例示するブロック図である。

【図 15】S Q L インスタンスを管理するためのシステムの第 5 の特定の実施形態を例示するブロック図である。

20

【図 16】S Q L インスタンスを管理するためのシステムの第 6 の特定の実施形態を例示するブロック図である。

【発明を実施するための形態】

【0009】

特定の実施形態により、データベースアプリケーションの要素を識別し、かつ、データベースアプリケーションのための管理の一単位を提供する、D A C (d a t a - t i e r a p p l i c a t i o n c o m p o n e n t : データ層アプリケーションコンポーネント)の使用について記載する。D A C により識別されるデータベースアプリケーションは、データベースオブジェクト識別子および/またはデータベースオブジェクトのメタデータに関連付けられ、これらを介してデータベースアプリケーションを操作することが可能である。加えて、データベースランタイム識別子は、データベースアプリケーションの要素により使用されるランタイムリソースの内の 1 つまたは複数と関連付けられる。データベースランタイムリソースの実行に関する 1 つまたは複数のポリシーはまた、D A C により識別される。その結果、データベースオブジェクト識別子により D A C を参照して、データベースアプリケーションを変更、移動または複製することが可能である。同様に、それぞれの要素に関連付けられるデータベースランタイム識別子を変更することにより、ランタイムリソースを変更することができ、かつ、データベースアプリケーションを複製することができる。その結果、データベース管理者は、データベースアプリケーションのインスタンスの編成の側面を管理することに時間と注意を費やすことなく、データベースアプリケーションのコンテンツの管理に集中することができる。

30

40

【0010】

特定の実施形態では、コンピュータ可読媒体として実装することができる D A C (d a t a - t i e r a p p l i c a t i o n c o m p o n e n t : データ層アプリケーションコンポーネント)を開示する。D A C には、データベースオブジェクトに関するメタデータをパッケージ化する管理の一単位が含まれる。データベースオブジェクトに関するメタデータには、データベースオブジェクトの識別子が含まれる。メタデータにはまた、データベースオブジェクトを操作するよう動作可能なデータベースランタイムリソースのデータベースランタイムリソース識別子が含まれる。メタデータには、データベースランタイムリソースの実行に関するポリシーも含まれる。加えて、上述の管理の一単位が、複数のデータベースオブジェクト、複数のランタイムリソース、およびデータベースランタ

50

イムリソースの実行に関する複数のポリシーのメタデータを含むことができることを理解されたい。さらに、データベースオブジェクトのメタデータには、データベースオブジェクト識別子、データベースオブジェクトのスキーマ（例えば、テーブルの列スキーマ、ストアドプロシージャおよび/またはストアドファンクションのテキストボディ）、データベースオブジェクトに関する許可、および/または他のメタデータのエントリであって、データベースサーバ内のオブジェクトを作成または変更するために、および/または、データベースオブジェクトについてリーズニングするために使用することが可能な他のメタデータのエントリを含むことができる。

【0011】

別の特定の実施形態では、データベース要素を一組の支援コンピュータリソースに関連付けるための、コンピュータ実装システムを開示する。システムには、1つまたは複数のデータベースオブジェクトに関するメタデータを含むデータベース要素のコレクションを含む、DAC (data-tier application component: データ層アプリケーションコンポーネント) が含まれる。データベース要素のコレクションにはまた、データベースオブジェクトを操作するよう動作可能なデータベースランタイムリソースのデータベースランタイムリソース識別子、および、データベースランタイムリソースの実行に関するポリシーが含まれる。DACは、データベース要素のコレクションに関連付けられるDAC名を有する。DAC名は、データベースアプリケーションによる使用のためにアクセス可能である。データベース要素のコレクションは、一組のユーザ選択可能な支援コンピュータリソースに、プログラム可能に関連付けられる。特定の実施形態では、DAC名が不変のままとしながら、支援コンピュータリソースを変更することができる。

【0012】

別の特定の実施形態では、DAC (data-tier application component: データ層アプリケーションコンポーネント) を配置する方法を開示する。方法は、DACモデルを作成するステップを含む。DACモデルには、1つまたは複数のデータベースオブジェクトに関するメタデータを含むデータベース要素のコレクションが含まれる。1つまたは複数のデータベースオブジェクトに関するメタデータには、各データベースオブジェクトの識別子、データベースオブジェクトを操作するよう動作可能なデータベースランタイムリソースのデータベースランタイムリソース識別子、および、データベースランタイムリソースの実行に関するポリシーが含まれる。方法は、DACモデルに基づきDACパッケージをビルドするステップ、および、DAC型をDACパッケージからサーバファブリックにインポートするステップを含む。方法はさらに、DAC型を配置して、配置されるDACインスタンスを作成するステップを含む。

【0013】

開示される主題には、データベースアプリケーションを構成するアイテムを関連付ける、ソフトウェアのグルーピング抽象化が含まれる。ソフトウェアのグルーピング抽象化により、データベースアプリケーションの識別、概念化、および管理が可能となる。本開示には、データベースアプリケーションに関連付けられるデータベースオブジェクト、ポリシー、およびランタイムリソースを効果的にモデル化する、DAC (data-tier application component: データ層アプリケーションコンポーネント) と呼ばれる抽象化が含まれる。

【0014】

DACは、バージョニングを含む、アプリケーションのライフサイクル全体の管理の単位を形成する。例えば、HR (Human Resources: 人的資源) DACは、HR DACが支援するHRアプリケーションコードと併せて開発することができる。HRアプリケーション実行コードがアプリケーションサーバに配置されるため、HR DACは、Microsoft (登録商標) SQL Serverのインスタンスに配置される。そして、HR DACを監視および管理することが可能である。例えば、HR DACへの予測される負荷が、HR DACを保持するカレントサーバの容量を超える恐

10

20

30

40

50

れがある場合、H R D A Cを、新しいサーバに移動させることができる。移動は、D A Cに関連付けられるデータベースオブジェクト識別子によってデータベースアプリケーションを参照することにより、かつ、データベース要素に関連付けられるデータベースランタイム識別子を変更して新しいサーバのランタイムリソースを参照することにより達成することができる。D A Cは、一般には複数のタイプのデータベースランタイム環境に、および/または、特にMicrosoft SQL Serverランタイム環境にまで亘り、これらには、Microsoft SQL Server、Microsoft SQL Serverデータベース、Microsoft SQL Serverデータベースエンジン、Microsoft SQL Server Analysis Services、Microsoft SQL Server Reporting Services、および、データベースサーバおよび/または異なるベンダおよびタイプのデータベースのインスタンスが含まれる。

10

【0015】

加えて、本開示では、データベースサーバおよび/またはMicrosoft SQL Server環境を全体として管理することに効果をもたらすことができ、D A Cの管理により重きをおき、個々のコンピュータ、データベースインスタンス、およびデータベースの管理にはそれほど着目しない。本開示は、このことに対して「サーバファブリック」の概念（「サーバファブリック」とも称する）を介して対処する。「サーバファブリック」の概念は、編成のD A C、Microsoft SQL Serverランタイム、およびハードウェアリソースの全てをモデル化する。サーバファブリックは、D A C、ランタイム、およびリソース層におけるエンティティおよびコンテナの統一されたビューを提供し、かつ、エンティティ間のより動的なマッピングを可能にする。サーバファブリックにはまた、管理アクションを宣言および起動するための中央管理ポイント、中央ポリシー管理者、および影響、what-if（仮定）、もしくは配置の分析等の分析に使用されるモデルまたはデータを含有する中央リーズニング（動作推定）ポイントが含まれる。これらの宣言的、モデル駆動の特徴により、データベース管理者および開発者は、グローバルな管理アクションを定義することができる。グローバルな管理アクションは、編成の規則を実施し、D A Cに必要な全ての特徴を有するランタイムに対してD A Cが配置されることを確実にし、および、サーバファブリック内のエンティティが、最小限の手動の介入で環境の変化に対応することを可能とする。

20

30

【0016】

データベース管理者および開発者にとって、データ層アプリケーションコンポーネントの開発、配置および管理が著しく容易になるよう、サーバファブリック内のエンティティの状態についてのより深い知見により駆動されるこれらの自己管理能力が求められる。本開示は、データベース管理者および開発者が、そのライフサイクルを通して自分のアプリケーション管理において、より生産的であることを可能にする。これにより、データベース管理者および開発者は、追加のデータプラットフォーム技術およびストレージプラットフォーム技術をそのシステムに組み込むことに、より時間を費やすことができる。

【0017】

本開示では、着目点を従来のMicrosoft SQL Serverのインスタンスおよびコンピュータから、データベースアプリケーションに移す。D A Cは、データベースアプリケーションのインスタンスおよびデータベースコンテナの両方の上にある新しい高レベルの抽象化である。D A Cには、従来の3層アプリケーションのデータレイヤサポートを形成する一組のデータベースエンティティが含まれる。現代のシステムでは、データベースアプリケーションは、複数のデータベースに及ぶことが多く、また、システムレベルのアプリケーション情報が「master」および「msdb」等のシステムデータベース内に記憶される。

40

【0018】

より具体的には、本開示では、D A Cと呼ばれるコンテナ内にデータベースアプリケーションをモデル化する。D A Cは、データベースアプリケーションのアーチファクト、関

50

連するポリシー、および参照されるランタイムリソースの全てをアプリケーションに対してまとめる論理的表現である。各DACは、管理の一単位として機能する。ポリシーを、DACに対して適用することができ、DACを、全体として移動することができる。ポリシーを使用して、必要とされるまたは好ましい、成果または検討事項の開発者または配置の意図を明らかにすることができる。例えば、開発者は、DACが64ビットランタイムで配置可能であるだけである、というポリシーを指定することにより必要な意図を宣言することができる。

【0019】

サーバファブリックは、一組のDACをモデル化し、それらを、下層のMicrosoft SQL Serverランタイム（インスタンスまたはマトリクスなど）、および、ランタイムにより使用されるハードウェアリソースにマッピングする。サーバファブリックは、一組のDACが機能するのに必要とする全てのランタイムおよびハードウェアリソースのモデルを提供する。サーバファブリックはまた、他の技術を使用すると非常に複雑になるアクションを簡略化させる中央集中管理の特性を提供する。例えば、各DACは、サーバファブリックにより管理される1つまたは複数の周知の名称（「エンドポイント名」）を有する。エンドポイント名は、DACにより使用されているハードウェアリソースまたはインスタンスとは無関係である。その結果、アプリケーションは、コンピュータ名およびインスタンス名の代わりにエンドポイント名を使用して接続する。そのためアプリケーションの変更を要求せずに、DACをインスタンス間で移動させることができる。

【0020】

全ての利用可能なDAC、ランタイム、およびリソースのモデルを、サーバファブリック内の中央管理リーズニング能力と組み合わせることによって、いくつかのシナリオが可能になる。例えば、サーバファブリック内の全てのDACのリソースの使用を、監視することができる。別の例として、中央リーズニングエンジンは、ホストのランタイムに割り当てられるリソースに対するDACの将来の作業負荷を予測することができる。別の例として、予測されるリソースの不足に応じてアクションを起こすことができる。例えば、システムは、ランタイムに利用可能なリソースを動的に統括し、余剰能力を有するランタイムを発見し、アプリケーションの接続に影響が無いようにDACをそのライフタイムに移動させる。ツールと能力の組は、サーバファブリックにおいてDACをそのライフサイクル（開発、配置および管理）を通して管理するものとして記載される。

【0021】

特定の実施形態では、DACは、データベースオブジェクト、データベースオブジェクトを操作するよう動作可能なデータベースランタイムリソースのデータベースランタイムリソース識別子、およびデータベースランタイムリソースの実行に関するポリシーのメタデータをパッケージ化する管理の一単位である。図1に例示するように、異なるアクションを、ライフサイクルの状態に応じてDACに適用することが可能である。

【0022】

例えば、図1を参照すると、ユーザ100（例えば、開発者）は、DACモデル102を作成することができる。DACモデル102には、DAC（data-tier application component：データ層アプリケーションコンポーネント）104が含まれる。DAC104は、DACコンテナ106に含むことができる。DAC104には、データベース要素のコレクションが含まれる。図1に示すように、DAC104には、第1の要素108、第2の要素110、および第3の要素112が含まれる。第1の要素108は、データベースオブジェクト識別子を含むデータベースオブジェクトのメタデータを含み得る。第2の要素110は、データベースオブジェクトを操作するよう動作可能なデータベースランタイムリソースを含み得る。第3の要素112は、データベースランタイムリソースの実行に関するポリシーを含み得る。DACモデル102で表されるデータベースアプリケーションの開発中に、ユーザ100は、DACモデル102の作成、エディット、認証をすることができ、かつ、DACモデル102を表現するものをDACプロジェクト114として保存することができる。DACプロジェクト114に

は、1つまたは複数のファイル118を含むことができる。次にDACプロジェクト114を、データベースアプリケーションの開発中に、ソースコントロールシステムまたはソースコントロールデータベース116にチェックインおよびソースコントロールシステムまたはソースコントロールデータベース116からチェックアウトすることができる。

【0023】

ユーザ100（例えば、開発者）が、DACモデル102で表されるデータベースアプリケーションの開発を完了すると、DACモデル102は、DACモデル102に基づいてコンパイル、またはビルドされ、完成したDACパッケージ120にまとめられる。DACパッケージ120は、所望の通りに複製および配置することができるDACモデル102の概念化されたバージョンを表すDAC型122として、サーバファブリック（例えば、サーバファブリック126）にインポートすることができる。DAC型122を配置して、配置されるDACインスタンス124を作成することができる。DACインスタンス124は、DACモデル102として開発されたデータベースアプリケーションの実用化インストールを表す。従って、開発者は、DACモデル102を開発しつつ、データベースアプリケーションを再コード化せずに複製することができるポータブルなデータベースアプリケーションを作成したことになる。開発者はまた、DACインスタンス124を配置する前にDAC型122を構成することができる。さらに、DACインスタンス124は、特定の組の物理的コンピュータリソースに対してインストールすることができる。

【0024】

図2を参照すると、データ層アプリケーションコンポーネントを配置する特定の方法的フロー図が例示される。方法は、202にて、1つまたは複数のデータベースオブジェクトのメタデータを含むデータベース要素のコレクションを含有するDACモデルを作成するステップを含む。DACモデルにはさらに、データベースオブジェクトを操作するよう動作可能なデータベースランタイムリソースのデータベースランタイムリソース識別子が含まれる。DACモデルにはさらに、データベースランタイムリソースの実行に関するポリシーが含まれる。特定の実施形態では、204にて、方法は、DACモデルを表現するものをDACプロジェクトとして保存するステップ、および、選択的に、DACプロジェクトをソースコントロールシステムまたはソースコントロールデータベースにチェックインしてDACプロジェクトを記憶するステップを含む。別の特定の実施形態では、206にて、方法は、DACパッケージをビルドする前に、DACモデルをエディットし、DACモデルを認証するステップと、DACインスタンスを配置する前にDAC型を構成するステップと、DACインスタンスを特定の組の物理的コンピュータリソースに対してインストールするステップとを含む。208に移って、方法は、DACモデルに基づきDACパッケージをビルドするステップを含む。210に移って、方法は、DAC型をDACパッケージからサーバファブリックにインポートするステップを含む。212に移って、方法は、DAC型を配置して、配置されるDACインスタンスを作成するステップを含む。

【0025】

DACが、サーバファブリック内にDAC型とDACインスタンスという2つの主要な細分割部を有することは強調すべきことである。DAC型は、DACがサーバファブリックにインポートされる時に構築される。DAC型は、DACにおいてテーブルおよびキューブ等のオブジェクトの論理実装を定義し、どのようにDACが使用されることを開発者が意図するかを定義するポリシーを含む。DACインスタンスは、サーバファブリックにおいて、1つまたは複数のデータベースサーバランタイム（例えば、Microsoft SQL Serverランタイム）にDAC型が配置される時に、構築される。DACインスタンスは、DAC型のエンティティがそれらのサーバランタイム層におけるホストにどのようにマッピングされるかを記録する。DACインスタンスには、どのようにDACがサーバファブリック内で動作することを管理者が意図するかを定義するポリシーを含む。

【0026】

図3を参照すると、DAC (data-tier application comp

10

20

30

40

50

onent : データ層アプリケーションコンポーネント) のコンテンツの高レベルの図が例示される。この図は、開発者の意図 (DAC 型) と配置の意図 (DAC インスタンス) との概念的な分離を例示する。図 3 に示すように、DAC は、DAC 型 304 と DAC インスタンス 306 とに概念的に分割することができる。DAC 型 304 には、構造、ロジック、ポリシーおよび DAC 定義を備える他のエンティティが含まれる。DAC 型 304 により、開発者が、不変条件、要求、ならびに、意図される DAC 302 の使用および解釈を正式に定義するための機構を提供することもできる。DAC 型 304 を備えるエンティティは、一単位として管理されることが意図される。情報は、DAC 302 の特定のバージョンの全ての配置に対してグローバルである。従って、DAC 型 304 は、開発者の意図を表現すること、バージョンングすること、特定のバージョンに対して不変であることができ、また、一意的であることが保証される識別子を有する。編成の目的で、DAC 型 304 の 3 つの主要な部分は、プロパティ 308、アプリケーションスキーマ 310、開発者の意図とポリシー 312 として示される。

10

【0027】

DAC 型 304 のプロパティ 308 は、DAC 型 304 に特有の特徴または属性である。例えば、プロパティ 308 は、人が読める名称および記述を含むことができる。更なる例として、プロパティ 308 は、最大バージョン番号および最小バージョン番号等の 1 つまたは複数のバージョン識別子を含むことができる。更なる例として、プロパティ 308 は、DAC 型 304 に対して一意的であることが保証される識別子を含むことができる。更なる例として、プロパティ 308 は、周知のエンドポイント名を含むことができる。更なる例として、プロパティ 308 は、他の DAC 型への依存関係を含むことができる。

20

【0028】

DAC 型 304 のアプリケーションスキーマ 310 は、DAC 302 を備える Microsoft SQL Server オブジェクトを定義する。DAC 型 304 のこの部分におけるオブジェクト定義は、DAC 302 に固有のものであり、任意の変更により DAC 型の新しいバージョンを生成することができる。アプリケーションスキーマ 310 は、本質的には、DAC 302 が「何」から成るのかを定義する。例えば、アプリケーションスキーマ 310 は、テーブル、キューブおよびデータマイニングモデル等のアプリケーション構造を含むことができる。さらなる例として、アプリケーションスキーマ 310 は、ストアドプロシージャ、ユーザ定義の関数、およびトリガ等のアプリケーションロジックを含むことができる。さらなる例として、アプリケーションスキーマ 310 は、アプリケーション定義の一部としてみなされる参照データを含むことができる。例えば、米国の郵便番号および他の郵便番号、または国番号および州番号等のルックアップデータが参照され得る。

30

【0029】

DAC 型 304 の開発者の意図とポリシー 312 により、どのように DAC 302 が使用または解釈されるよう意図されているかについての追加の情報が提供される。例えば、開発者の意図とポリシー 312 は、DAC 302 を配置するために利用可能でなければならないサーバファブリックリソース要求を含むことができる。例えば、DAC 302 は、Microsoft SQL Server 2008 に要求して、空間データ型と CLR 統合、および 80 GB のディスク空き容量のサポートを可能にすることができる。更なる例として、開発者の意図とポリシー 312 は、DAC 302 内に組み込まれるコンフィギュアビリティ (設定可能性) を定義する構成の意図 (パラメータ化) を含むことができる。これにより、DAC のバージョンに影響を与えない、配置に特有の構成が可能にされ、この構成には配置時に提供可能なまたは提供しなければならない情報が含まれる。例えば、構成のパラメータは、配置の処理が、ファイルグループおよび Microsoft SQL Server のインスタンスの名称など、サーバファブリックリソースの特定の型へのマッピングを提供しなければならないことを示すことができる。さらなる例として、開発者の意図とポリシー 312 は、DAC 302 のどの部分を外部に公開するかを指定するインターフェース定義を含むことができる。これにより、公のインターフェースと D

40

50

A C 3 0 2 の内部の実装の詳細とを分離することができる。例えば、インターフェース定義では、どのエンティティがクライアントアプリケーションまたは別の D A C により参照可能であるかを指定する。さらなる例として、開発者の意図とポリシー 3 1 2 は、この D A C 3 0 2 に対して実行されるクエリ等の作業負荷のプロファイル情報を含むことができる。

【 0 0 3 0 】

開発者の意図とポリシーの例には、D A C を配置するために利用可能でなければならないサーバファブリックリソース要求が含まれる。例えば、D A C は、M i c r o s o f t S Q L S e r v e r 2 0 0 8 に要求して、空間データ型と C L R 統合および 8 0 G B のディスク空き容量のサポートを可能にすることができる。開発者の意図とポリシーの別の例には、D A C 内に組み込まれるコンフィギュアビリティを定義する構成の意図（パラメータ化）が含まれる。これにより、D A C のバージョンに影響を与えない、配置に特有の構成が可能にされ、この構成には配置時に提供可能なまたは提供しなければならない情報が含まれる。例えば、構成のパラメータは、配置の処理が、ファイルグループおよび M i c r o s o f t S Q L S e r v e r のインスタンスの名称など、サーバファブリックリソースの特定の型へのマッピングを提供しなければならないことを示すことができる。開発者の意図とポリシーの別の例には、D A C のどの部分を外部に公開するかを指定するインターフェース定義が含まれ得る。これにより、公のインターフェースと D A C の内部の実装の詳細とを分離することができる。例えば、インターフェース定義では、どのエンティティがクライアントアプリケーションまたは別の D A C により参照可能であるかを指定する。開発者の意図とポリシーの別の例には、D A C に対して実行されるクエリ等の作業負荷のプロファイル情報が含まれ得る。

【 0 0 3 1 】

図 3 に示すように、D A C インスタンス 3 0 6 は、実際のインスタンスまたは配置に特有の情報を保存する。この情報は、特定の配置に対して一意的である。D A C インスタンス 3 0 6 は、配置の意図を表現する。D A C インスタンス 3 0 6 は、D A C 型 3 0 4 に関連付けられる。複数の D A C インスタンスが、各 D A C 型 3 0 4 に対して存在することが可能である。D A C インスタンス 3 0 6 は、サーバファブリック内において一意的であることが保証される識別子を有する。D A C インスタンス 3 0 6 は、サーバファブリックマッピングを含有する。

【 0 0 3 2 】

D A C インスタンス 3 0 6 のプロパティ 3 1 4 は、D A C 3 0 2 の配置インスタンスに特有の特徴または属性である。プロパティ 3 1 4 には、D A C インスタンス 3 0 6 識別子および配置の名称等のアイテムが含まれる。

【 0 0 3 3 】

D A C インスタンス 3 0 6 の配置スキーマ 3 1 6 は、特定の配置に対して特有の S Q L オブジェクトを定義する D A C インスタンス 3 0 6 の一部である。配置スキーマ 3 1 6 には、プリンシパルおよびインデックス等のオブジェクトが含まれる。

【 0 0 3 4 】

D A C インスタンス 3 0 6 の配置の意図とポリシー 3 1 8 は、D A C 3 0 2 が「どのように」配置または解釈されることが意図されるかについての情報を提供する。配置の意図とポリシー 3 1 8 には、特定の配置に対して特有の意図されるサーバファブリックマッピングおよび意図されるポリシー等のアイテムが含まれる。

【 0 0 3 5 】

ファブリックマッピング 3 2 0 は、D A C インスタンス 3 0 6 と下層のリソースとの間のマッピングを定義する D A C インスタンス 3 0 6 の一部である。ファブリックマッピング 3 2 0 のコンテンツの例には、インスタンスまたはマトリクス等の M i c r o s o f t S Q L S e r v e r ランタイムマッピング、および、ディスクストレージおよびコンピュータ等のハードウェアリソースが含まれる。

【 0 0 3 6 】

データ 3 2 2 は、特定の D A C インスタンスに関連するデータを表す D A C インスタンス 3 0 6 の一部である。データ 3 2 2 には、アプリケーションのユーザにより作成されるアプリケーションデータと、D A C インスタンス 3 0 6 またはアプリケーションに対して特有の性能データ等の実用的データとの両方が含まれる。

【 0 0 3 7 】

図 4 を参照すると、4 0 0 にて、D A C (d a t a - t i e r a p p l i c a t i o n c o m p o n e n t : データ層アプリケーションコンポーネント) の特定の実施形態が例示される。D A C 4 0 2 には、データベース要素のコレクション 4 0 4 が含まれる。データベース要素のコレクション 4 0 4 には、データベースオブジェクトのデータベースオブジェクト識別子 4 0 6 が含まれる。データベース要素のコレクション 4 0 4 にはまた、データベースオブジェクトを操作するよう動作可能なデータベースランタイムリソースのデータベースランタイムリソース識別子 4 0 8 が含まれる。データベース要素のコレクション 4 0 4 にはまた、データベースランタイムリソースの実行に関するポリシー 4 1 0 が含まれる。D A C 4 0 2 にはまた、データベース要素のコレクション 4 0 4 に関連付けられる D A C 名 4 1 2 が含まれる。D A C 名 4 1 2 は、データベースアプリケーション 4 1 4 による使用のためにアクセス可能である。データベース要素のコレクション 4 0 4 は、4 1 6 にて示すように、一組のユーザ選択可能な支援コンピュータリソース 4 1 8 に、プログラム可能に関連付けられる。

【 0 0 3 8 】

D A C 名 4 1 2 (例えば、エンドポイント名) は、ドメイン名サーバ (D N S) 名に若干類似して動作する。アプリケーション実行ファイルは、D A C 名に接続し、D A C 4 0 2 をホストする特定のコンピュータまたはインスタンスには接続しない。D A C 4 0 2 の特定のコンピュータまたはインスタンスへの関連付けは、D A C 4 0 2 をコンピュータおよびインスタンスにマッピングすることにより、サーバファブリック内に確立される。D A C 4 0 2 は、アプリケーションがデータベースサーバに接続する方法に影響を与えることなく (例えば、接続文字列に影響を与えることなく)、M i c r o s o f t S Q L S e r v e r の新しいインスタンスに D A C 4 0 2 を単に再マッピングすることにより、新しいインスタンスに移動させることが可能である。変更の必要があるアプリケーションは無く、全て同じ D A C 名に接続している。例えば、給与という D A C 名の D A C を、コンピュータ 1 上のデフォルトのインスタンスからコンピュータ 2 上の別のインスタンスに移動することができる。全てのアプリケーションは、周知の D A C 名である給与を参照するため、同じ D A C 4 0 2 に接続している。

【 0 0 3 9 】

特定の実施形態では、一組のユーザ選択可能な支援コンピュータリソース 4 1 8 は、自動化ツール (例えば、ソフトウェアアプリケーション、ユーザ制御のアプリケーションなどであり、M i c r o s o f t S Q L S e r v e r M a n a g e m e n t S t u d i o、V i s u a l S t u d i o、ラインコマンド、P o w e r s h e l l、等がある) を使用して変更することが可能である。別の特定の実施形態では、コレクションに関連付けられる D A C 名 4 1 2 は、複数のデータベースアプリケーションからアクセス可能である。

【 0 0 4 0 】

別の特定の実施形態では、一組のユーザ選択可能な支援コンピュータリソース 4 1 8 は修正されるが、D A C 名 4 1 2 は不変である。

【 0 0 4 1 】

別の特定の実施形態では、データベース要素のコレクション 4 0 4 内のデータベース要素は修正されるが、D A C 名 4 1 2 は不変である。

【 0 0 4 2 】

別の特定の実施形態では、データベースランタイムリソースは、M i c r o s o f t S Q L S e r v e r のインスタンスであり、M i c r o s o f t S Q L S e r v e r のインスタンスが第 1 の M i c r o s o f t S Q L S e r v e r インスタンスから

10

20

30

40

50

第2のMicrosoft SQL Serverインスタンスに修正されるが、DAC名412は不変である。

【0043】

別の特定の実施形態では、データベース要素のコレクション404は変更することができ、または、一組のユーザ選択可能な支援コンピュタリソース418は変更することができ、かつ、DAC名412は不変であるため、データベース要素のコレクション404が変更される時、または、一組のユーザ選択可能な支援コンピュタリソース418が変更される時は、1つまたは複数のデータベースアプリケーション（例えば、データベースアプリケーション414）のアプリケーションの接続文字列を、記録（例えば、変更、更新、アプリケーションの再コンパイルの要求、アプリケーションのバイナリおよび/または実行ファイルの再作成の要求）する必要が無い。

10

【0044】

別の特定の実施形態では、DAC名412を使用するデータベースアプリケーションは、異なるネットワークコンピュータ上に置かれる。

【0045】

別の特定の実施形態では、データベースランタイムリソースの実行に関するポリシー410が、DAC名412に適用され、データベース要素のコレクション404が変更される時、または、一組のユーザ選択可能な支援コンピュタリソース418が変更される時、ポリシー410は、適用が継続される。

【0046】

20

別の特定の実施形態では、DAC402は、データベース要素のコレクション404とインターフェースをとるAPI(application program interface:アプリケーションプログラムインターフェース)をさらに備える。例えば、APIは、データベース要素のコレクションの要素の利用を測定することができる。例えば、APIにより、開発者は図1のようなデータベース要素のコレクション404とインターフェースをとることができる。

【0047】

特定の実施形態では、DAC402は、コンピュータ可読媒体に記憶される。コンピュータ可読媒体は、データベースオブジェクト、データベースオブジェクトを操作するよう動作可能なデータベースランタイムリソースのデータベースランタイムリソース識別子、およびデータベースランタイムリソースの実行に関するポリシーのメタデータをパッケージ化する管理の一単位を含むことができる。一度に1つずつであるように記載したが、管理の一単位が、複数のデータベースオブジェクト、データベースオブジェクトを操作するよう動作可能なデータベースランタイムリソースの複数のデータベースランタイムリソース識別子、およびデータベースランタイムリソースの実行に関する複数のポリシーのメタデータをパッケージ化することが可能であることを理解されたい。加えて、複数のデータベースオブジェクトは、スキーマ、テーブル、ビュー、ストアドプロシージャ、関数、トリガ、データ型、インデックス、ログイン、ユーザ、許可、ポリシー、統計値、主キー、外部キー、デフォルト、チェックの制約、および/または、データベースサーバまたはデータベースにおいて作成可能な任意の他のオブジェクトを含むことができる。

30

40

【0048】

管理の一単位を、第1の物理的コンピュタリソースに対してマッピングすることができる。管理の一単位を、第1の物理的コンピュタリソースから第2の物理的コンピュタリソースに再マッピングすることができる。第1の物理的コンピュタリソースの例には、プロセッサまたはメモリストレージデバイスがある。特定の実施形態では、管理の一単位は、関連するDAC名により識別される。ここで、DAC名はデータベースアプリケーションによる使用のためにアクセス可能である。

【0049】

データベースオブジェクトを操作するよう動作可能なデータベースランタイムリソースは、SQL(Structured Query Language)サーバのインスタ

50

ンスである。Microsoft SQL Serverのインスタンスには、Microsoft SQL Serverデータベースエンジンのインスタンス、Microsoft SQL Server Analysis Servicesのインスタンス、もしくは、SQL Reporting Servicesのインスタンス、または、他のデータベースのサーバタイプおよびベンダのインスタンスを含むことができる。

【0050】

ポリシーは、リソース制限ポリシー、または、ソフトウェアリソースもしくはハードウェアリソースの要求に関するプロシージャを含むデータベースランタイムリソースの実行に関するものとすることができる。

【0051】

図5を参照すると、500にて、システムの特定の実施形態が例示される。システム500には、一組のDAC502が含まれる。図5に示す実施形態では、第1のDAC504、第2のDAC506、および第3のDAC508が、一組のDAC502に含まれる。あるいは、一組のDAC502には、任意の数のDACを含むことができる。DAC504、506、および508のそれぞれには、データベース要素のコレクション510、512、および514の論理的表現が含まれる。

【0052】

システム500には、一組のDAC502をホストする一組のデータベースランタイムリソース516がさらに含まれる。図5に示す実施形態では、一組のデータベースランタイムリソース516には、第1のデータベースランタイムリソース518、第2のデータベースランタイムリソース520、および第3のデータベースランタイムリソース522が含まれる。あるいは、一組のデータベースランタイムリソース516には、任意の数のデータベースランタイムリソースを含むことができる。

【0053】

システム500には、一組のデータベースランタイムリソース516により使用されて一組のDAC502をホストするコンピュータリソース524がさらに含まれる。図5に示す実施形態では、コンピュータリソース524には、第1のコンピュータリソース526、第2のコンピュータリソース528、および第3のコンピュータリソース530が含まれる。あるいは、コンピュータリソース524には、任意の数のコンピュータリソースを含むことができる。

【0054】

コンピュータリソース524は、1つまたは複数のコンピュータサーバリソース、1つまたは複数のコンピュータデータストレージリソース、または、1つまたは複数の仮想マシンリソースを含むことができる。1つまたは複数のコンピュータサーバリソースのそれぞれは、プロセッサ、メモリ、またはネットワーク要素を含むことができる。1つまたは複数のコンピュータデータストレージリソースの少なくとも1つを、複数のファイルを含む記憶ボリュームとすることができる。1つまたは複数のコンピュータサーバリソースの少なくとも1つを、最初に第1のSAN(storage area network: ストレージエリアネットワーク)にマッピングし、続いて、第2のSANにマッピングすることができる。

【0055】

一実施形態では、一組のDAC502内のDAC504、506、および508のそれぞれには、DACコンテナが含まれる。各DACコンテナは、データベースオブジェクト、データベースオブジェクトを操作するよう動作可能なデータベースランタイムリソースのデータベースランタイムリソース識別子、および、データベースランタイムリソースの実行に関するポリシーのメタデータをパッケージ化する管理の一単位を含むことができる。

【0056】

一組のDAC502内の第1のDAC504は、一組のDAC502内の第2のDAC506に依存することとしても良い。

10

20

30

40

50

【 0 0 5 7 】

一組のDAC 502内のDAC 504、506、および508のそれぞれは、サーバファブリック501のDAC層により表すことができる。コンピュータリソース526、528、および530を含むコンピュータリソース524は、サーバファブリック501のハードウェアリソース層により表すことができる。一組のデータベースランタイムリソース516内のデータベースランタイムリソース518、520、および522のそれぞれは、サーバファブリック501のSQL (Structured Query Language) サーバランタイム層により表すことができる。

【 0 0 5 8 】

一組のデータベースランタイムリソース516は、Microsoft SQL Server サービスを提供する1つまたは複数のMicrosoft SQL Server ランタイムインスタンスを含むことができる。1つまたは複数のMicrosoft SQL Server ランタイムインスタンスは、Microsoft SQL Server、Microsoft SQL Server Analysis Services、およびMicrosoft SQL Server Reporting Servicesの1つまたは複数のインスタンスを含むことができる。さらに、1つまたは複数のMicrosoft SQL Server ランタイムインスタンスは、スキーマおよびビューを含むことができる。

10

【 0 0 5 9 】

DAC 504、506、および508のそれぞれを、データベースランタイムリソース518、520、および522の1つにマッピングすることができ、かつ、各データベースランタイムリソース518、520、および522を、1つまたは複数の物理的コンピュータリソース526、528、および530にマッピングすることができる。データベースランタイムリソースを、第1のコンピュータから第2のコンピュータに再マッピングすることができる。

20

【 0 0 6 0 】

DAC 504、506、および508のそれぞれは、関連するDAC名を含むことができる。DAC名は、データベースアプリケーションによる使用のためにアクセス可能である。

【 0 0 6 1 】

図6を参照すると、一組のDACを含むサーバファブリックの第2の実施形態を例示するブロック図が、600にて示される。サーバファブリック608は、一組のDAC (複数のDAC)、DACをホストするMicrosoft SQL Server ランタイム、および、それらのランタイムに使用されてDACをホストするリソースのモデルに基づく表現である。サーバファブリックのモデルは、図6に示すような3つの層を有し、すなわち、DAC層610、サーバランタイム層612、およびホストリソース層614である。モデル内のオブジェクトの全ては、一組の一般的モデル化概念 (noun) を使用して記述され、かつ、それらのオブジェクトに対して実行される一連のアクション (verb) を有する。

30

【 0 0 6 2 】

DAC層610は、DACのサーバファブリック定義をホストする。示される実施形態では、DAC層610には、第1のDAC 622、第2のDAC 624、および第3のDAC 626が含まれる。あるいは、DAC層610には、任意の数のDACを含むことができる。サーバランタイム層612は、サーバファブリック608内にデータベースランタイムリソースのビュー (例えば、Microsoft SQL Server サービス) を提供する。示される実施形態では、サーバランタイム層612には、第1のデータベースランタイムリソース616、第2のデータベースランタイムリソース618、および第3のデータベースランタイムリソース620が含まれる。あるいは、サーバランタイム層612には、任意の数のデータベースランタイムリソースを含むことができる。ホストリソース層614は、サーバファブリック608内の要素により使用されるコンピューテ

40

50

イングとストレージの容量を表す。示される実施形態では、ホストリソース層 6 1 4 には、第 1 のサーバ 6 3 6 および第 2 のサーバ 6 3 8 を含むサーバリソース 6 3 0 が含まれる。さらに、ホストリソース層 6 1 4 には、第 1 のストレージリソース 6 4 0 および第 2 のストレージリソース 6 4 2 を含むデータリソース 6 3 2 が含まれる。さらに、ホストリソース層 6 1 4 には、第 1 の仮想リソース 6 4 4 および第 2 の仮想リソース 6 4 6 を含む仮想リソース 6 3 4 が含まれる。あるいは、ホストリソース層 6 1 4 は、任意の数のサーバリソース、データリソース、または仮想リソースを含むことができる。

【 0 0 6 3 】

サーバファブリック 6 0 8 は、D A C、一組の D A C をホストするデータベースランタイムリソース、および、データベースランタイムリソースにより使用され D A C をホストするコンピュータリソースを識別する。各 D A C 6 1 0 には、データベース要素のコレクションの論理的表現が含まれる。中央リーズニングポイント 6 0 4 は、サーバファブリック 6 0 8 のエンティティを監視する。中央ポリシー 6 0 6 は、サーバファブリック 6 0 8 により識別された影響を受けるエンティティにおいて、受け取ったデータベース管理アクションおよびポリシーを自動的に適用するために使用される。

【 0 0 6 4 】

サーバファブリック 6 0 8 内の各層は、4 つのタイプのエンティティ、すなわち、コンテナ、要素、リソース、およびマッピングを含有する。コンテナは、サーバファブリック 6 0 8 の所与の層内の最高レベルのエンティティであり、管理および識別の直観的な単位を提供する。コンテナの例には、D A C、コンピュータ、仮想マシン、ストレージ、Microsoft SQL Server のインスタンス、Microsoft SQL Server Analysis Services、および Microsoft SQL Server Reporting Services が含まれる。要素は、コンテナ内に存在するサーバファブリック 6 0 8 内の一単位である。例えば、テーブルおよびストアドプロシージャは、D A C 内に存在する要素である。リソースは、有限の容量を有する特定のタイプの要素であり、かつ、その容量を決めるプロパティを有する。プロセッサデバイスは、粒度の細かいリソースの例であり、そのクロック速度は、容量を決めるプロパティの一例である。ストレージデバイスは、物理ディスク等のより粒度の細かいリソースを含有するリソースの一例である。ストレージデバイスの容量は、ストレージデバイスが含有する物理的ディスクのプロパティのみにより決められるのではなく、RAID レベル等の構成のプロパティによっても決められる。マッピングは、サーバファブリック 6 0 8 の 2 つの部分の間の関係を表す。マッピングは、サーバファブリック 6 0 8 を Microsoft SQL Server 環境の実際のモデルに形成する、一般的モデル化概念である。マッピングは、コンテナとコンテナ、コンテナと要素、コンテナとリソース、要素と要素、要素とリソース、およびリソースとリソース間で行うことができる。

【 0 0 6 5 】

サーバファブリック 6 0 8 における各層 6 1 0、6 1 2、および 6 1 4 の構成は、上述の一般的モデル化概念を適用することにより定義することができる。各層は、1 つまたは複数のコンテナ、それらのコンテナ内の要素およびリソース、ならびに、サーバファブリック 6 0 8 の近接する層内のコンテナ間または要素間のマッピングから成る。

【 0 0 6 6 】

D A C 層 6 1 0 は、D A C のサーバファブリック定義をホストする。各 D A C は、D A C コンテナを使用して定義される。各 D A C コンテナは、構造、コード、バージョン、およびポリシー等のアプリケーションエンティティを表す多数の要素で構成される。

【 0 0 6 7 】

D A C 層 6 1 0 におけるマッピングにより、D A C と Microsoft SQL Server ランタイム (例えば、Microsoft SQL Server のインスタンス) との間の関係を記述する。例えば、D A C 層 6 1 0 内の D A C を、サーバランタイム層 6 1 2 内の Microsoft SQL Server のインスタンスにマッピングすることが可能である。そして、D A C をインスタンス間に再マッピングすることにより

10

20

30

40

50

、1つのインスタンスから別のインスタンスにDACを移動させることができる。例として、第1のDAC622は、第1のデータベースランタイムリソース616（例えば、Microsoft SQL Serverの第1のインスタンス）にマッピングされ、また、次に第1のDAC622を第2のデータベースランタイムリソース618に再マッピングすることにより、第1のDAC622を、第2のデータベースランタイムリソース618（例えば、Microsoft SQL Serverの第2のインスタンス）に移動させることができる。

【0068】

マッピングではまた、DAC間の関係について記述することもできる。DACは、共通の要素の再利用を可能にするために別のDACへの依存関係を表現することができる。例えば、消耗品の注文のための企業のポリシーを実施する一組のストアドプロシージャ等の、いくつかのアプリケーションにより共有される要素を有する「ベースの」DACを、定義することが可能である。それらの要素を必要とする任意の他のDACは、「ベースの」DACにマッピングすることによりそれらを参照することができる。例えば、第1のDAC622は、第2のDAC624に依存することとしても良い。

【0069】

サーバランタイム層612は、サーバファブリック内にMicrosoft SQL Serverサービスを提供する。Microsoft SQL Serverランタイムコンテナには、Microsoft SQL Serverのインスタンス、Microsoft SQL Server Analysis Services、およびMicrosoft SQL Server Reporting Servicesのインスタンスが含まれる。各ランタイムコンテナは、1つまたは複数のデータベース要素を有することができる。データベース要素は、スキーマ、テーブル、ビュー、ストアドプロシージャ、関数、データ型、インデックス、ログイン、ユーザ、ポリシー、統計値、主キー、外部キー、デフォルト、チェックの制約、および/または、データベースサーバまたはデータベースにおいて作成可能な任意の他のオブジェクトを含むことができる。

【0070】

サーバランタイム層612におけるマッピングにより、特定のSQLランタイム（例えば、Microsoft SQL Serverのインスタンス）と、そこに割り当てられる物理的リソースとの関係を記述する。例えば、サーバランタイム層612に存在するインスタンスが、ハードウェアリソース層614内の物理的コンピュータにマッピングされる。インスタンスは、インスタンスをコンピュータ間で再マッピングすることにより、物理的コンピュータ間で移動させることができる。例として、第1のデータベースリソース616を、第1のサーバ636にマッピングすることができ、かつ、第1のデータベースリソース616は、第1のデータベースリソース616を第1のサーバ636から第2のサーバ638に再マッピングすることにより、第2のサーバ638に移動させることができる。

【0071】

ホストリソース層614は、サーバファブリック608内の要素により使用されるコンピューティングとストレージの容量を表す。ホストリソース層614におけるコンテナは、物理的コンピュータ、仮想マシン、および記憶ボリュームが含まれる。リソースコンテナはまた、サブ要素を含有することができる。例えば、物理的サーバは、プロセッサ、メモリ、およびネットワーク要素を含有する。記憶ボリュームは、ファイル要素とともに、言及したアーチファクトの記憶に関する他の要素を含有することができる。

【0072】

ホストリソース層614におけるマッピングにより、物理的リソース間の関係を記述する。例えば、ホストリソース層のマッピングは、どのSAN LUN（Logical Unit Number：論理ユニット番号）が、各物理的サーバからアクセス可能であるのかを表現する。物理的サーバとSAN LUNとの間のマッピングを変更することにより、SAN LUNは、物理的サーバ間で移動させることができる。

10

20

30

40

50

【 0 0 7 3 】

サーバファブリック 6 0 8 は、サーバファブリック 6 0 8 全体に亘ってエンティティに適用することが可能な管理アクションおよびポリシーの中央宣言を支援する。サーバファブリック 6 0 8 の中央管理ポイント 6 0 2 は、データベース管理アクションおよびポリシーを受け取ることができる。サーバファブリック 6 0 8 はまた、サーバファブリックのエンティティを監視すること、かつ、管理アクションおよびポリシーを自動的に適用することが可能であるリーズニングエンジンを提供する。サーバファブリック 6 0 8 の中央管理機構が図 6 に例示される。

【 0 0 7 4 】

中央管理機構は全て、コンテナの関連する詳細、関係、および制約をサーバファブリック 6 0 8 に記録するモデルを参照する。

10

【 0 0 7 5 】

サーバファブリック 6 0 8 内の `Microsoft SQL Server` のインスタンスの 1 つは、中央管理ポイント 6 0 2 として指名され、中央集中管理の能力（ディスカバリ（発見）、ポリシー定義および監視等）を全てのサーバファブリック層に提供する。中央管理ポイント 6 0 2 は、全てのコンテナのアクションがトリガ（例えば、インポート、移動、作成、および配置）される場所である。中央管理ポイント 6 0 2 を、サーバファブリック 6 0 8 により識別されるデータベースランタイムリソースの 1 つにおいて、ホストすることができる。例えば、中央管理ポイント 6 0 2 を、第 1 のデータベースランタイムリソース 6 1 6、第 2 のデータベースランタイムリソース 6 1 8、または第 3 のデータベースランタイムリソース 6 2 0 においてホストすることができる。中央管理ポイント 6 0 2 は、サーバファブリック層 6 1 0、6 1 2、および 6 1 4 に対してディスカバリ、ポリシー定義、および監視を提供することができる。

20

【 0 0 7 6 】

中央リーズニングポイント 6 0 4 は、サーバファブリック 6 0 8 内の要素のデータモデルを供給する。モデルは、リソース使用の傾向およびサーバファブリック 6 0 8 全体に亘る可能性のあるアクションの影響を予測するために、分析エンジンにより使用される。中央リーズニングポイント 6 0 4 を使用して、`SQL` サーバファブリック 6 0 8 により識別されるエンティティに対して中央リーズニングアクションを行うことができる。中央リーズニングアクションには、例えば、他の `DAC` が第 1 の `DAC` に依存しているような場合に、第 1 の `DAC` をアンインストールすることの影響を推定する、第 1 のアクションを含むことができる。中央リーズニングアクションにはまた、例えば、コンピュートリソースの内の 1 つのメモリリソースを修正することの 1 つまたは複数の `DAC` への影響を推定する第 2 のアクションを含むことができる。中央リーズニングアクションにはまた、例えば、`SQL` サーバファブリック内の `Microsoft SQL Server` のインスタンスであって、配置対象の `DAC` をホストするよう構成されるインスタンスのリストを提示する、第 3 のアクションを含むことができる。

30

【 0 0 7 7 】

中央ポリシー 6 0 6 には、挙動またはアクションを導くよう定義される正式な原則であるポリシーを含むことができる。ここで、ポリシーは、`SQL` サーバファブリック 6 0 8 により識別される一組のエンティティに適用することができる一組の条件を指定する。例えば、一組のエンティティは、1 つまたは複数の `Microsoft SQL Server` ランタイムインスタンスを含むことができ、かつ、ポリシーは、`Microsoft SQL Server` のソフトウェアバージョンに基づき、1 つまたは複数の `Microsoft SQL Server` ランタイムインスタンスの `SQL` サーバファブリック 6 0 8 への組み込みを制限することができる。例として、`Microsoft SQL Server 2008` のインスタンスのみを、サーバランタイム層 6 1 2 に組み込むことができる。更なる例として、一組のエンティティは、1 つまたは複数のコンピュートリソースを含むことができ、かつ、ポリシーは、オペレーティングシステムまたはメモリ容量に基づき、1 つまたは複数のコンピュートリソースの `SQL` サーバファブリック 6 0 8 へ

40

50

の組み込みを制限することができる。例として、64ビットバージョンのWindows Server 2008で、少なくとも4ギガバイト(4GB)のメモリを有するサーバのみを、ホストリソース層614に組み込むことができる。

【0078】

中央管理ポイント602は、SQLサーバファブリックのアクションをトリガすることができる。例えば、SQLサーバファブリックのアクションは、インポートアクション、エクスポートアクション、配置アクション、コピーアクション、認証アクション、インストールアクション、アンインストールアクション、追加アクション、削除アクション、入手アクション、設定アクション、列挙アクション、移動アクション、作成アクション、保存アクション、アップグレードアクション、開始アクション、終了アクション、再起動アクション、実行アクション、一時停止アクション、再開アクション、無効化アクション、有効化アクション、測定アクション、監視アクション、予測アクション、統括アクション、マップアクション、アンマップアクション、ディスクバクション、比較アクション、マージアクション、ブロックアクション、アンブロックアクション、承諾アクション、または、取り消しアクションを含むことができる。

10

【0079】

ライフサイクルの一連のアクションにより、エンティティを、作成またはサーバファブリック608へのインポートの時間から、サーバファブリック608からの削除の時間まで管理するための一組のアクションを記述する。例えば、ライフサイクルの一連のアクションは、所与のエンティティの作成、修正、および消去のようなアクションを対象とする。いくつかのライフサイクルの一連のアクションは、1つまたは複数の層610、612、および614に適用することができる。ライフサイクルの一連のアクションには、インポートアクション、エクスポートアクション、配置アクション、コピーアクション、認証アクション、インストールアクション、アンインストールアクション、追加アクション、削除アクション、入手アクション、設定アクション、列挙アクション、移動アクション、作成アクション、保存アクション、およびアップグレードアクションが含まれる。

20

【0080】

インポートアクションでは、新しいコンテナおよび関連する要素を、外部の永続的なデータストアから作成する。インポートアクションは、DAC層610に適用することができる。例えば、インポートアクションにより、DACをDACパッケージファイルからサーバファブリック608にインポートすることができる。エクスポートアクションは、永続的な外部のデータストア(例えば、ファイル)を、コンテナおよびそれに含まれる要素から作成する。エクスポートアクションは、DAC層610に適用することができる。例えば、エクスポートアクションは、DACをファイルにエクスポートすることができる。

30

【0081】

配置アクションでは、コンテナおよびそのオブジェクトを意図される場所に置く。配置アクションは、DAC層610に適用することができる。例えば、配置アクションは、第1のDAC622をサーバランタイム層612(例えば、第1のインスタンス616)内のインスタンスに配置することができる。

【0082】

コピーアクションでは、一組の要素を1つのコンテナから別のコンテナにコピーする。コピーアクションは、DAC層610およびサーバランタイム層612に適用することができる。例えば、コピーアクションにより、データベースをMicrosoft SQL Serverの1つのインスタンスからサーバランタイム層612の別のインスタンスに(例えば、第1のインスタンス616から第2のインスタンス618に)コピーすることができる。

40

【0083】

認証アクションでは、対象のコンテナ、要素、リソース、またはマッピングが指定された基準でコンパイルしているかどうかをチェックする。認証アクションは、DAC層610、サーバランタイム層612、およびホストリソース層614に適用することができる

50

。例えば、認証アクションにより、Microsoft SQL ServerのインスタンスがCLR統合を有効にしたことを認証することができる。

【0084】

インストールアクションでは、サーバファブリック608においてDACを動作可能に構成する。インストールアクションは、DAC層610およびサーバランタイム層612に適用することができる。例えば、インストールアクションにより、DACを第1のインスタンス616にインストールすることができる。アンインストールアクションでは、コンテナおよびその要素を指定される場所から削除する。アンインストールアクションは、DAC層610およびサーバランタイム層612に適用することができる。例えば、アンインストールアクションにより、DACを第1のインスタンス616からアンインストールすることができる。

10

【0085】

追加アクションでは、外部で作成されたコンテナ、要素、またはリソース、を表現するものをサーバファブリックに組み込む。追加のアクションは、サーバランタイム層612およびホストリソース層614に適用することができる。例えば、追加アクションにより、第1のMicrosoft SQL Serverインスタンス616をサーバランタイム層612に追加することができる。削除アクションでは、外部で作成されたコンテナ、要素、またはリソースをサーバファブリック608との関連から切り離す。削除アクションは、サーバランタイム層612およびホストリソース層614に適用することができる。例えば、削除アクションにより、Microsoft SQL Serverインスタンス616をサーバランタイム層612から削除することができる。

20

【0086】

入手アクションでは、コンテナ、要素、リソース、またはマッピング、の現在の構成を取り出す。入手アクションは、DAC層610、サーバランタイム層612、およびホストリソース層614に適用することができる。例えば、入手アクションにより、第1のMicrosoft SQL Serverインスタンス616にインストールされたDACのバージョン番号を入手することができる。

【0087】

設定アクションでは、コンテナ、要素、リソース、またはマッピング、の現在の構成を更新する。設定アクションは、DAC層610、サーバランタイム層612、およびホストリソース層614に適用することができる。例えば、設定アクションにより、第1のMicrosoft SQL Serverインスタンス616にインストールされたDACのバージョン番号を設定することができる。

30

【0088】

列挙アクションでは、コンテナ内の一組の要素またはリソースを入手する。列挙アクションは、DAC層610、サーバランタイム層612、およびホストリソース層614に適用することができる。例えば、列挙アクションにより、第1のサーバ636にリソースを列挙することができる。

【0089】

移動アクションでは、一組の要素またはリソースを1つのコンテナ、要素、またはリソースから、別のコンテナ、要素、またはリソースに移動させる。移動アクションは、DAC層610、サーバランタイム層612、およびホストリソース層614に適用することができる。例えば、移動アクションにより、データベースをMicrosoft SQL Serverの1つインスタンスからサーバランタイム層612内の別のインスタンスに移動させることができる。

40

【0090】

作成アクションでは、新しいコンテナまたは要素を作成する。作成アクションは、DAC層610およびサーバランタイム層612に適用することができる。例えば、作成アクションにより、DACがMicrosoft SQL Server 2008を要求することを指定する新しいポリシーの制約を作成することができる。

50

【0091】

保存アクションでは、コンテナ、要素、リソース、またはマッピングの現在の構成を継続させる。保存アクションは、DAC層610、サーバランタイム層612、およびホストリソース層614に適用することができる。例えば、保存アクションにより、第1のMicrosoft SQL Serverインスタンス616にインストールされたDACのバージョン番号を保存することができる。

【0092】

アップグレードアクションでは、コンテナまたは要素の既存の構成を新しい定義に変換する。アップグレードアクションは、DAC層610およびサーバランタイム層612に適用することができる。例えば、アップグレードアクションにより、第1のMicrosoft SQL Serverインスタンス616にインストールされたOrdersテーブルの定義を第1のバージョンから第2のバージョンにアップグレードすることができる。

10

【0093】

サーバファブリック608は、監視性能、照合変更、およびポリシー順守の認証等のアクションを実行するいくつかの処理を有する。以下の一組の一連のアクションを使用して、処理を管理することができる。その他の一連のアクションの全てが、サーバファブリック608の全ての層（例えば、DAC層610、サーバランタイム層612、およびホストリソース層614）に適用される。処理制御の一連のアクションには、開始アクション、終了アクション、再起動アクション、実行アクション、一時停止アクション、再開アクション、無効化アクション、および有効化アクションが含まれる。

20

【0094】

開始アクションでは、処理を開始し、処理は終了されるまで継続される。例えば、開始アクションは、第1のMicrosoft SQL Serverインスタンス616の照合を開始する。終了アクションでは、処理の実行を中止する。例えば、終了アクションにより、第1のMicrosoft SQL Serverインスタンス616の照合を終了することができる。再起動アクションでは、実行中の処理を終了させ、それを再度開始させる。例えば、再起動アクションにより、第1のMicrosoft SQL Serverインスタンス616の照合を再起動することができる。実行アクションでは、実行されて完了時に自動的に終了する処理を、開始する。例えば、実行アクションにより、第1のMicrosoft SQL Serverインスタンス616上でディスカバリを実行することができる。一時停止アクションでは、処理を中断し、処理は再開されるまで中断されたままにされる。例えば、一時停止アクションにより、第1のMicrosoft SQL Serverインスタンス616上でディスカバリを一時停止させることができる。再開アクションでは、一時停止された処理を継続させる。例えば、再開アクションにより、第1のMicrosoft SQL Serverインスタンス616上のディスカバリを再開することができる。無効化アクションでは、処理を利用できないように構成し、処理の開始を不可能にする。処理は、有効にされるまで無効のままになる。例えば、無効化アクションにより、第1のMicrosoft SQL Serverインスタンス616の照合を無効にすることができる。有効化アクションでは、無効にされた処理が開始されるように構成する。例えば、有効化アクションにより、第1のMicrosoft SQL Serverインスタンス616の照合を有効にすることができる。

30

40

【0095】

性能管理の一連のアクションを使用して、サーバファブリックリソースの現在の利用について、現在のまたは予測されるサーバファブリック608の健全性を分析する。これらのverbはまた、サーバファブリック608内の種々のコンテナおよび要素によるリソースの使用を統括する手段も提供する。性能管理の一連のアクションは、測定アクション、監視アクション、予測アクション、および統括アクションを含むことができる。

【0096】

測定アクションでは、リソース、要素、またはコンテナに関する性能統計値を収集する

50

。例えば、測定アクションにより、第1のサーバ636のボリュームE上のデータベースファイルFinance.mdbにより使用されるスペースを測定することができる。監視アクションでは、リソース、要素、またはコンテナに関する性能統計値を、ポリシーに対して評価する。例えば、監視アクションにより、容量の80%を超える利用の発生時について、第1のサーバ636のボリュームE上のデータベースファイルFinance.mdbにより使用されるスペースを監視することができる。予測アクションでは、履歴データ、リソース変更、または、配置もしくは作成等のサーバファブリックアクションの関数として、コンテナ、要素、またはリソースの性能統計値を予想する。例えば、予測アクションにより、増大した会計のDACを収容するために追加のストレージがいつ必要になるかを予測することができる。更なる例として、予測アクションは、全体的に増大したサーバファブリックを収容するために追加のリソースがいつ必要になるかを予測する。統括アクションでは、コンテナまたは要素に割り当てられるリソースの最小量または最大量を指定する。例えば、統括アクションにより、第1のMicrosoft SQL Serverインスタンス616のCPUの利用を、第2のサーバ638のCPUの容量の50%に制限することができる。

【0097】

マッピングの一連のアクションでは、サーバファブリック608内の層の間の関係を管理する簡素な手段を提供する。マッピングの一連のアクションは、主として、近接する2つの層の間の関係を確立するために使用される。一連のアクションを使用して全ての関係が管理されるわけではない。例えば、コンピュータとそのストレージとの関係は、コンピュータを物理的に操作することにより制御される。マッピングの一連のアクションは、マップアクションおよびアンマップアクションを含むことができる。概念的には、これらの一連のアクションは、サーバファブリック608に対して行うことが可能な最高レベルのアクションであるとして考えることができ、このアクションの結果、一連の他のライフサイクルアクションが実行されることになる。例えば、DACをMicrosoft SQL Serverのインスタンスにマッピングすることで、DACがそのインスタンス上で利用可能であることをシステムに知らせることになる。これにより、そのDACを、指定されるインスタンス上で利用可能にすることが求められる、配置またはインストール等の全ての一連のアクションが呼び出される。

【0098】

マップアクションでは、コンテナ、要素、またはリソースの間の関係を作成する。例えば、マップアクションにより、第1のDAC622を第1のMicrosoft SQL Serverインスタンス616に結びつけることができる。アンマップアクションでは、コンテナ、要素、またはリソースの間のマッピングを削除する。例えば、アンマップアクションにより、第1のDAC622を第1のMicrosoft SQL Serverインスタンス616から分離させることができる。

【0099】

照合は、サーバファブリック内の、所定の定義から外れたエンティティを識別するための処理である。例えば、照合により、プロダクションサーバ内のテーブルの定義が、DACにおいて定義されたものから変更されているかどうかを検出する。3つのサーバファブリックの一連のアクションが、照合を実装する。照合の一連のアクションは、ディスカバクション、比較アクション、およびマージアクションを含むことができる。

【0100】

ディスカバクションでは、新しいアイテムを見つけ、かつ、新しいアイテムに生産的な見識を与える動作が定義される。例えば、ディスカバクションにより、第1のMicrosoft SQL Serverインスタンス616内のEmployeeテーブル内に新しい列を発見することができる。比較アクションでは、コンテナ、要素、またはリソースの現在の構成を、予め定義された構成とマッチさせ、任意の相違を報告する。例えば、比較アクションにより、第1のMicrosoft SQL Serverインスタンス616に存在するEmployeeテーブルを、DAC内のテーブル定義比較するこ

とができ、かつ、DAC内に定義されていない列を有する、第1のMicrosoft SQL Serverインスタンス616内のテーブルのバージョンを見つける。マージアクションでは、複数のインスタンスから、コンテナ、要素、またはリソースの単一の構成を作成する。例えば、マージアクションにより、Employeeテーブル内の新しい列の定義をDAC定義にマージすることができる。

【0101】

アクセスの一連のアクションでは、サーバファブリックモデルの一部へのアクセスを制御する手段を提供して、長期間実行される処理を支援する。例えば、ファイルを1つのボリュームから別のボリュームに移動させる処理は、移動先のボリュームへのアクセスを制御して、複数の処理が相容れないスペースの割り当てを行わないようにしなければならない。セキュリティの一連のアクションでは、サーバファブリック608内のコンテナおよび要素へのアクセスを、認可されたユーザに限定する。本開示では、新しい認可の基礎構造を提供しないが、むしろ、これらの一連のアクションにより、サーバファブリック608全体に亘る認可の構成についての統一されたインターフェースを表す。従って、本開示では、Microsoft SQL Serverには既に整っている種々の下層にある認可の機構を活用する。アクセスおよびセキュリティの一連のアクションは、ブロックアクション、アンブロックアクション、承諾アクション、および、取り消しアクションを含むことができる。

【0102】

ブロックアクションでは、コンテナ、要素、またはリソースの構成への発行処理排他的アクセスを提供し、他の処理がその構成にアクセスすることを阻止する。この一連のアクションは、実際のエンティティ自体ではなく、サーバファブリック608のモデル内のデータに作用する。例えば、ブロックアクションにより、第1のサーバ636の記憶ボリュームEへのアクセスをブロックすることができ、ファイル移動を実行するためにスペースを確保する。アンブロックアクションは、コンテナ、要素、またはリソースの構成への排他的アクセスを解除する。この一連のアクションは、実際のエンティティ自体ではなく、サーバファブリック608のモデル内のデータに作用する。例えば、ファイル移動が完了した後、アンブロックアクションにより、第1のサーバ636の記憶ボリュームEへのアクセスをアンブロックすることができる。承諾アクションでは、コンテナ、要素、またはリソースにアクセスするための主要な許可を与える。例えば、承諾アクションにより、DACへの主要なJanet書き込みアクセスを承諾することができる。取り消しアクションでは、コンテナまたは要素にアクセスするための主要な許可を拒否する。例えば、取り消しアクションにより、主要なJanetからDACへの書き込みアクセスを取り消すことができる。

【0103】

図7を参照すると、データベース管理アクションおよびポリシーを自動的に適用する方法の特定の実施形態が例示される。702にて、方法は、SQLサーバファブリック(例えば、図6のサーバファブリック608の中央管理ポイント602)の中央管理ポイントにおいて、データベース管理アクションおよびポリシーを受け取るステップを含む。SQLサーバファブリックは、DAC、一組のDACをホストするデータベースランタイムリソース、およびDACをホストするためにデータベースランタイムリソースにより使用されるコンピュータリソースを識別する。DACのそれぞれには、データベース要素のコレクションの論理的表現が含まれる。704に移ると、方法はまた、中央リーズニングポイント(例えば、図6のサーバファブリック608の中央リーズニングポイント604)において、SQLサーバファブリックのエンティティを監視するステップを含む。706に移ると、方法はまた、SQLサーバファブリックにより(例えば、図6の中央ポリシー606を使用して)識別される影響を受けるエンティティにおいて、受け取ったデータベース管理アクションおよびポリシーを自動的に適用するステップを含む。

【0104】

図8を参照すると、ERD(entity relationship diagram)

10

20

30

40

50

m：エンティティ関係図）により、サーバファブリックがデータ層のコンポーネントを記述するモデルをどのように表すかについて、および、そのコンポーネントがどのように関係するかを例示する。図8のERDでは、このモデルがどのような形になってサーバファブリックの3つの層の構造とセマンティック（意味）を支援するのかを例示する。

【0105】

各層は、コンテナ、要素、およびリソースと同様にモデル内の正規のエンティティとして表される。モデルはまた、これらのエンティティ間の関係セマンティックを定義し、その結果サーバファブリックの構造が定義される。この構造では、本開示の実施形態によるサーバファブリックに対するリーズニングおよびアクションを起こすことを可能にすることにより、次のセクションで説明する機構が容易にされる。

10

【0106】

図9を参照すると、900にて、システムの特定の例示の実施形態が例示される。CMS（Central Management Server：中央管理サーバ）902により、サーバファブリックに中央集中管理ポイントが提供される。CMS902は、CMR（Central Management Repository：中央管理リポジトリ）904に接続する。サーバファブリック内のMicrosoft SQL Serverの管理されるインスタンスは、CMS902に接続し、情報をCMR904にアップロードする。CMS902は、ネットワーク上のSQLサービスのインスタンスをディスカバリするアクション、DACをプロビジョンするアクション、DAC要素を所望の場所に置くアクション、DACを基準に対してチェックするアクション、および、サーバファ

20

【0107】

CMR904は、サーバファブリックのモデル（例えば、全ての関連する詳細、関係、および制約）を含有する。CMS902は、配置分析、影響分析、およびwhat-if分析のための中央リーズニングポイントであり、コアデータ型、詳細データ型、およびリンクデータ型を含有する。この型の内の2つにより、IMR（例えば、IMR910および914）からCMR904への複製の頻度に基づく分類を用いて、CMR904に含有されるデータを表す。頻繁に複製されるデータをコアデータと称する。それほど頻繁に複製されないデータを詳細データと称する。リンクデータ型は、CMR904に記憶されないデータに対するインデックスまたはポインタとして機能する。

30

【0108】

CMDW（Central Management Data Warehouse：中央管理データウェアハウス）906により、一組の物理的サーバおよびインスタンス全体に亘る性能統計値履歴を中央に集中してレポートすることを可能にする。各管理されるSQLサーバランタイムインスタンス上に存在するデータは、収集されて、監視およびレポートの目的でCMDW906のデータベースの配置場所にアップロードされる。

【0109】

CFS（Central Fabric Service：中央ファブリックサービス）908は、コアのCMS902のアクションのためのAPIを提供する。これらのアクションには、ネットワーク上のSQLサービスのインスタンスをディスカバリするアクション、DACをプロビジョンするアクション、DAC要素を所望の場所に置くアクション、DACを基準に対してチェックするアクション、および、サーバファブリック内の要素間の相違を照合するアクションが含まれる。

40

【0110】

IMR（Instance Management Repository：インスタンス管理リポジトリ）は、Microsoft SQL Serverの各インスタンス内に存在するデータベースである。例えば、第1のIMR910は、第1のMicrosoft SQL Serverインスタンス912内に存在する。第2のIMR914は、第2のMicrosoft SQL Serverインスタンス916内に存在する。IMRは、全ての配置またはサーバファブリックの修正の最初の対象である。「コレクシ

50

ョンセット」として既知の機構により、IMR情報をコピーし、CMR904内にアップロードしてサーバファブリックの集約レベルのビューを提供する。

【0111】

IMDW (Instance Management Data Warehouse : インスタンス管理データウェアハウス) は、Microsoft SQL Server の各インスタンス内に存在する管理データウェアハウスである。例えば、第1のIMDW918は、第1のMicrosoft SQL Server インスタンス912内に存在する。第2のIMDW920は、第2のMicrosoft SQL Server インスタンス916内に存在する。IMDWは、インスタンスのデータの性能 (例えば、レポートおよび/または監視) の最初の対象であり、後で、CMDW906内にアップロードされてサーバファブリックの集約的なビューを提供する。

10

【0112】

IFS (Instance Fabric Server : インスタンスファブリックサーバ) は、コアの管理アクションのためのAPIを提供する。例えば、第1のIFS922は、第1のMicrosoft SQL Server インスタンス912内に存在する。第2のIFS924は、第2のMicrosoft SQL Server インスタンス916内に存在する。これらのアクションには、ローカルなインスタンス上にDACインスタンスを認証してインストールするアクション、外れたインストールをディスカバリするアクション、および照合動作を提供するアクションが含まれる。IFSはまた、インスタンスの性能を測定することができ、CMS902から送られるアクションを実行する。

20

【0113】

図10を参照すると、1000にて、エンド・ツー・エンド管理の処理の例示の図が例示される。本開示は、開発から配置までの完全なソフトウェアライフサイクル全体に亘ってMicrosoft SQL Server データベースアプリケーションを管理するための、ツールおよび基礎構造を動作に提供する。このセクションでは、本開示のいくつかの標準的なワークフローを前のセクションで説明された主要なアーキテクチャのコンポーネントに対して用いて、核となる設計の原則に見識を提供する。

【0114】

1002にて、システムがインストールされ、サーバファブリックが構成される。1004にて、DACが、既存の配置から作成され、サーバファブリックにインポートされる。1006にて、サーバファブリックの利用が監視および予測される。監視と予測に基づき、サーバファブリックリソースを調節することができる。例えば、1008にて、サーバファブリックリソースが統括される。1010にて、サーバファブリック要素が移動される。

30

【0115】

図11を参照すると、インストール処理例示が示される。インストールパッケージは、3つのコンポーネントで構成される。すなわち、ツール拡張機能、サーバファブリックサービス、および管理リポジトリスキーマである。ツール拡張機能は、Visual Studio およびMicrosoft SQL Server Management Studio に適用される。サーバファブリックサービスおよび管理リポジトリスキーマは、中央管理サーバおよび管理されるインスタンス内にインストールされる。

40

【0116】

1102にて、インストールパッケージが取得される。1104にて、ツール拡張機能 (Visual Studio およびMicrosoft SQL Server Management Studio) がインストールされる。1106にて、中央管理サーバ (CMS) および中央ファブリックサービス (CFS) が、インストールされ構成される。1108にて、中央管理リポジトリ (CMR) スキーマおよび中央管理データウェアハウス (CMDW) スキーマがインストールされる。1110にて、全てのランタイムインスタンスがディスカバリされ、サーバファブリックにおいて管理されるインスタンスと

50

なるインスタンスがディスカバリされる。1112にて、サーバファブリックインストールのジョブが、各管理されるインスタンス内に作成される。1114にて、インスタンス管理リポジトリ (IMR) スキーマおよびインスタンス管理データウェアハウス (IMDW) スキーマが、各管理されるインスタンス内に作成される。1116にて、サーバファブリックIMDWのCMDWへのデータアップロード機構がインストールされ、1118にて開始される。

【0117】

図12を参照すると、既存の配置からDACを作成し、サーバファブリック内にインポートする説明に役立つ実例が示される。DACは、サーバファブリック内のコアの論理的な管理の単位である。サーバファブリックはDACを管理して、最適なリソース利用を確保にする。DACは、新しいDACプロジェクトを作成することにより、Visual StudioまたはMicrosoft SQL Server Management Studioのどちらかにおいて認可される。加えて、アプリケーションデータベースオブジェクトを選択することにより、新しいDACを、以前のデータベースの配置からリバースエンジニアすることが可能である。

【0118】

1202にて、DACが作成される。1204にて、インスタンスオブジェクトが、DACに追加される。1206にて、インスタンスの設定およびポリシーがDACに追加される。1208にて、配置の設定およびポリシーがDACに追加される。1210にて、DACの設定およびポリシーが、IMRに対して認証される。1212にて、DACはIMRに保存される。1214にて、DACはIMRからCMRにコピーされる。選択的に、1216にて、DACパッケージバイナリファイルが、Visual Studioにおける後の開発での使用のために作成される。

【0119】

図13を参照すると、サーバファブリックの利用を監視および予測する説明に役立つ実例が示される。リソース利用のポリシーにより、サーバファブリックおよびDAC全体に亘って設定および評価が可能であり、リソース利用の過不足を防ぐことができる。

【0120】

1302にて、グローバルリソースの監視ポリシーがCMRにおいて設定される。1304にて、統計値がIMDWに追加される。1306にて、データベースリソースの監視ポリシーがIMRにおいて設定される。1308にて、リソースの利用の統計値がIMDWにおいて監視される。1310にて、IMDWリソースの利用の統計値がCMDWにエクスポートされる。1312にて、現在のリソースの利用がCMDWにおいて監視される。1314にて、予測されるリソースの利用がCMDWにおいて監視される。

【0121】

図14を参照すると、サーバファブリックリソースを調節する説明に役立つ実例が示される。リソース割り当てのポリシーにより、サーバファブリックおよびDAC全体に亘って設定および評価が可能であり、DACが利用可能なリソースのバランスを保つことができる。リソースの調節は、配置済みのリソースの統括、および、ボリューム間でのファイルの移動またはサーバ間のデータベースアプリケーションの移動といったサーバファブリック要素の移動、等のアクションを介して達成することが可能である。サーバファブリックリソースは、ポリシーを統括する新しいリソースを実装することにより調節することができる。

【0122】

1402にて、リソースのバランスを保つオプションが、対象のDACに関するDACに対して予測される。1404にて、新しいリソースのポリシーがCMRにおいて設定される。1406にて、新しいリソースのポリシーは、各IMRにコピーされる。1408にて、リソース統括の規則が各IMRに実装される。

【0123】

図15を参照すると、サーバファブリックリソースを調節する別の説明に役立つ実例が

10

20

30

40

50

示される。図 15 において、DAC を 1 つのファイルから別のファイルに移動させる処理が例示される。

【0124】

1502 にて、DAC が CMR から取得される。1504 にて、可能性のある対象ボリュームが予測される。1506 にて、DAC が別のボリューム（ボリューム 2）にマッピングされる。1508 にて、DAC は CMR から IMR にコピーされる。1510 にて、DAC は、IMR において認証される。1512 にて、DAC はインストールされる（データベースを変更、ファイルをコピー、...）。1514 にて、DAC は CMR において認証される。

【0125】

図 16 を参照すると、サーバファブリックリソースを調節する別の説明に役立つ実例が示される。図 16 において、DAC を Microsoft SQL Server の 1 つのインスタンスから別のそれに移動させる処理が例示される。

【0126】

1602 にて、DAC が CMR から取得される。1604 にて、可能性のある対象インスタンスが予測される。1606 にて、DAC はランタイムインスタンス 2 にマッピングされる。1608 にて、DAC は、CMR からインスタンス 2 IMR にコピーされる。1610 にて、DAC は、インスタンス 2 IMR において認証される。1612 にて、DAC は、インスタンス 2 IMR にインストールされる。1614 にて、DAC のエンドポイント名が、CMR において移動される。1616 にて、DAC は CMR において認証される。

【0127】

本明細書に記載される実施形態の例示では、種々の実施形態の構造についての全体的な理解を提供することが意図される。例示では、本明細書に記載される構造または方法を利用する装置およびシステムの要素および特徴の全てを完全に説明することは意図されない。本開示を精査すれば、多くの他の実施形態が当業者には明らかであろう。本開示から、他の実施形態が利用されかつ導かれ、構造的および論理的な置き換えおよび変更が、本開示の範囲から逸脱することなくなされ得るであろう。従って、本開示および図面は、制限するものではなく例示として見なされるべきである。

【0128】

当業者はさらに理解するであろうが、本明細書に開示される実施形態と併せて記載される、種々の例示の論理的なブロック、構成、モジュール、回路、およびアルゴリズムステップは、電子機器、コンピュータソフトウェア、またはその組み合わせとして実装できる。ハードウェアとソフトウェアのこの互換性を明確に例示するために、種々の例示のコンポーネント、ブロック、構想、モジュール、回路、またはステップが、その機能性の観点から概して記載された。そのような機能性がハードウェアまたはソフトウェアとして実装されるかどうかは、特定のアプリケーション、およびシステム全体に課される設計上の制約に依存する。当業者は、記載される機能性を、各特定のアプリケーションに対して色々な方法で実装することができるが、そのような実装の決定を、本開示の範囲から逸脱する原因として解釈すべきではない。

【0129】

本明細書に開示される実施形態と併せて記載される方法のステップは、直接ハードウェアにおいて、プロセッサにより実行されるソフトウェアモジュールにおいて、または、その 2 つの組み合わせにおいて、具現化することができる。ソフトウェアモジュールは、RAM (random access memory)、フラッシュメモリ、ROM (read only memory)、レジスタ、ハードディスク、着脱可能ディスク、CD-ROM、または従来技術において既知の任意の他の形式の記憶媒体、等のコンピュータ可読媒体に存在させることができる。例示の記憶媒体は、プロセッサに連結され、プロセッサが記憶媒体から情報を読み書きできるようにされる。代替えとして、記憶媒体はプロセッサに統合することができ、または、プロセッサと記憶媒体を、コンピュータデバイス

10

20

30

40

50

またはコンピュータシステム内で離散したコンポーネントとして存在させることができる。

【 0 1 3 0 】

本明細書において特定の実施形態が例示されかつ記載されたが、同一または同様の目的を達成するために設計される、任意の後に続く構成は、示される特定の実施形態と置き換えることができることを理解されたい。本開示では、種々の実施形態の、任意のおよび全ての後に続く適合または変動を対象とすることが意図される。

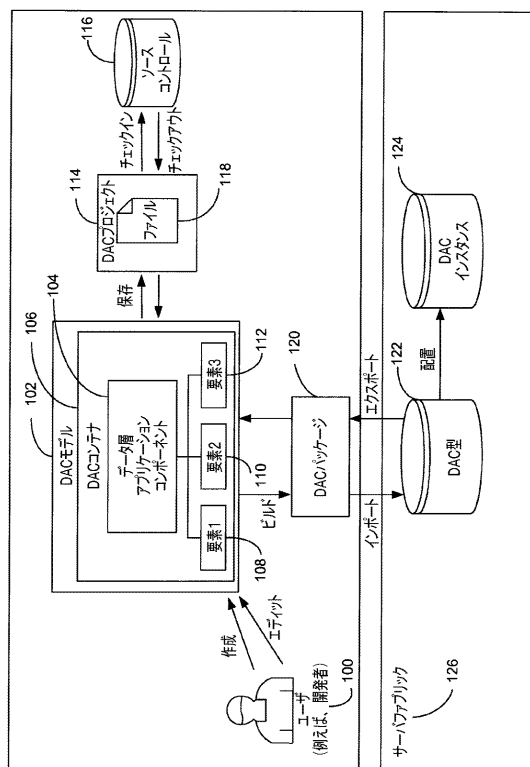
【 0 1 3 1 】

本開示の要約書は、37 C. F. R. § 1.72 (b) に従って提供され、かつ、請求項の範囲または意味を解釈または制限するために使用されないという理解のもとに提出される。加えて、前述の「発明を実施するための形態」において、本開示を簡素化する目的で、種々の特徴が単一の実施形態でまとめられ、または、単一の実施形態で記載される。本開示は、請求される実施形態が、各請求項に明示的に列挙されるものより多くの特徴を要求するという意図を反映するものとして解釈されるべきではない。むしろ、以下の請求項が反映するように、発明の主題は、開示される実施形態の内のいずれかの全て特徴のより少ないものを対象とする。

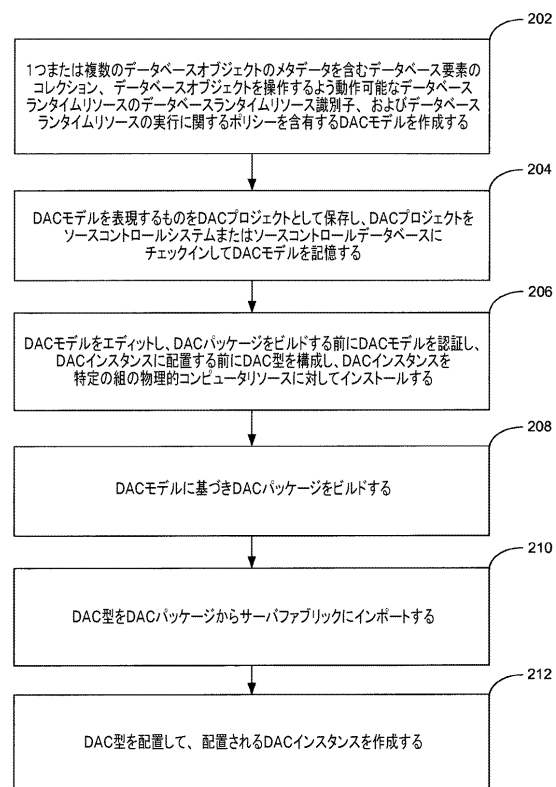
【 0 1 3 2 】

開示される実施形態の上述の記載が提供されて、任意の当業者が開示される実施形態を形成または使用することが可能にされる。これらの実施形態に対する種々の修正が、当業者には容易に明らかにされるであろう、また、本明細書に定義される一般的原理が、本開示の範囲から逸脱することなく他の実施形態に適用される。従って、本開示は、本明細書に示される実施形態に限定されることは意図されず、以下の請求項に定義される原理および新規の特徴に整合する可能性のある一番広い範囲に一致することが意図される。

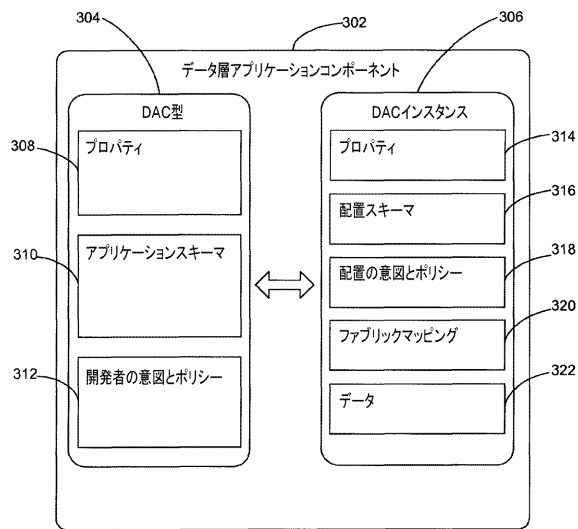
【 図 1 】



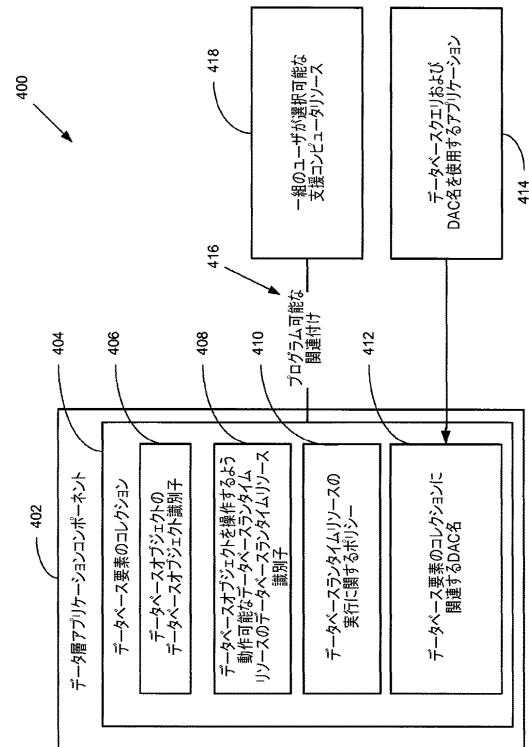
【 図 2 】



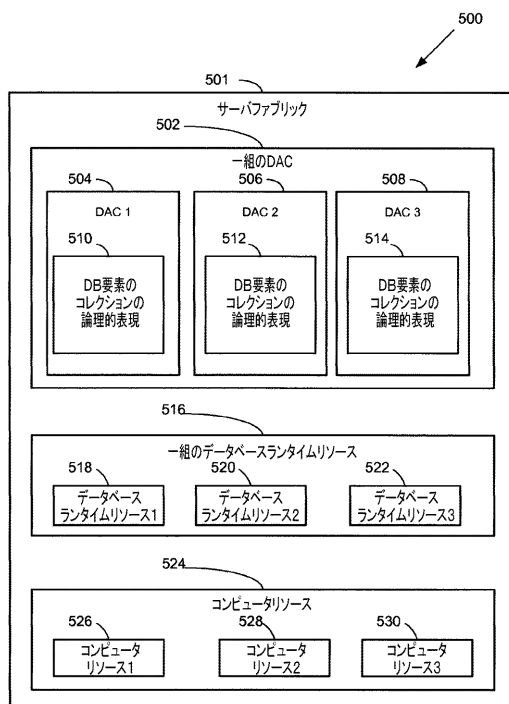
【図 3】



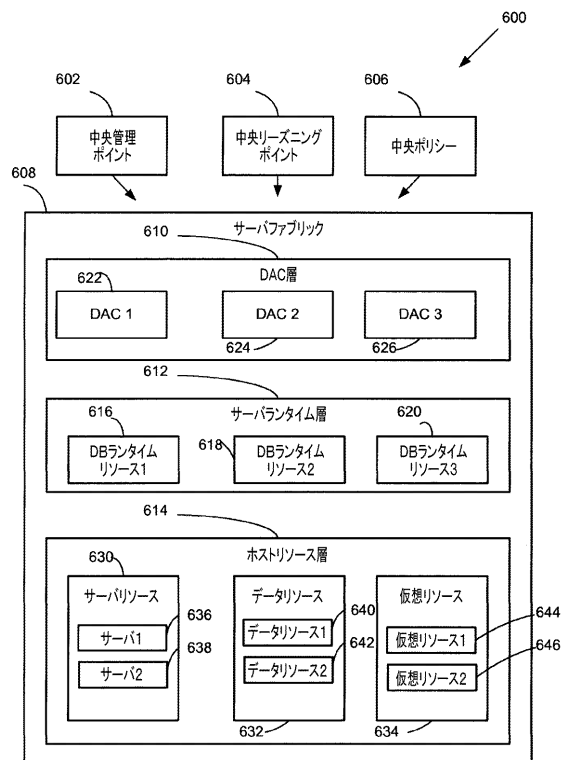
【図 4】



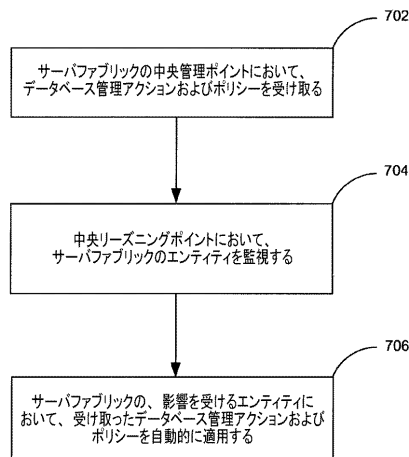
【図 5】



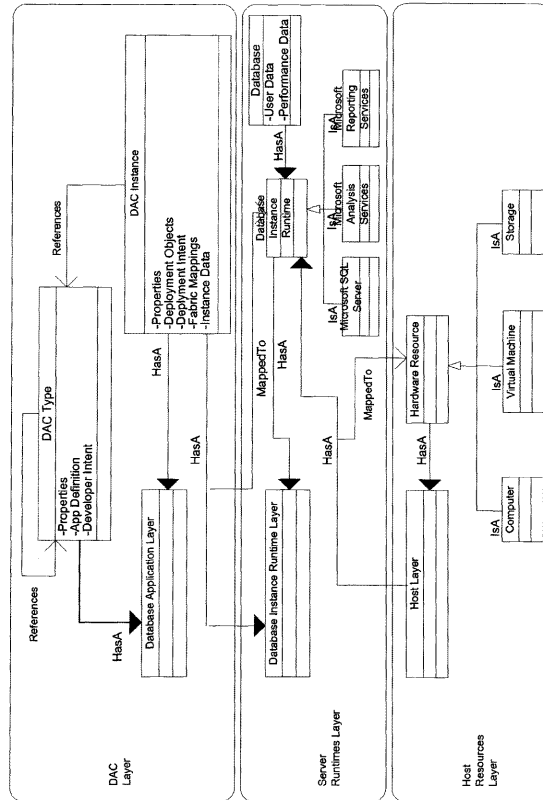
【図 6】



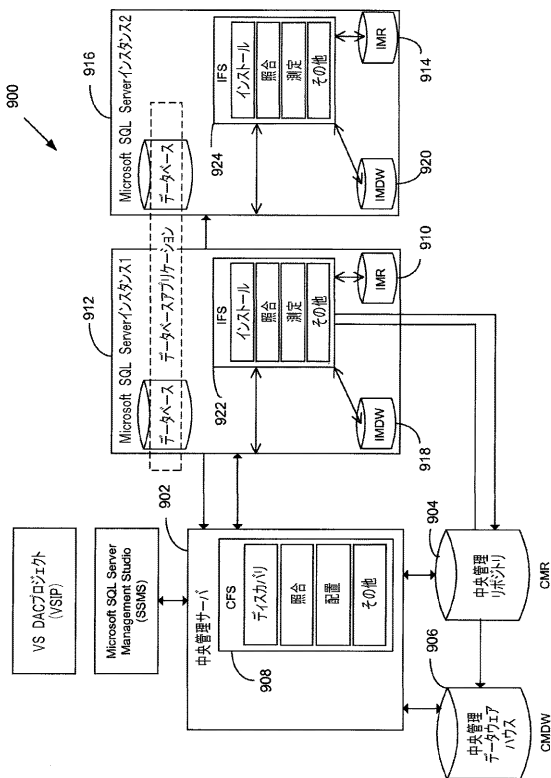
【図 7】



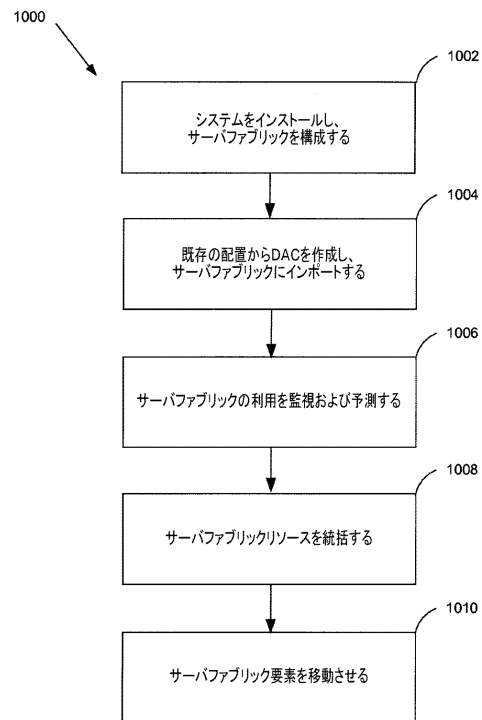
【図 8】



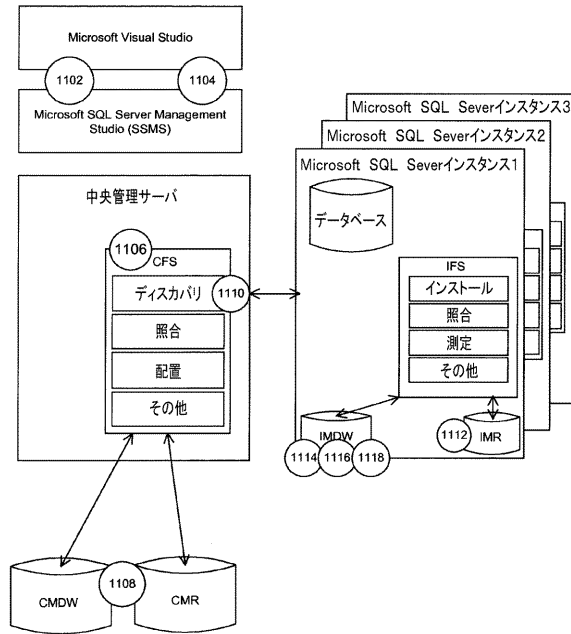
【図 9】



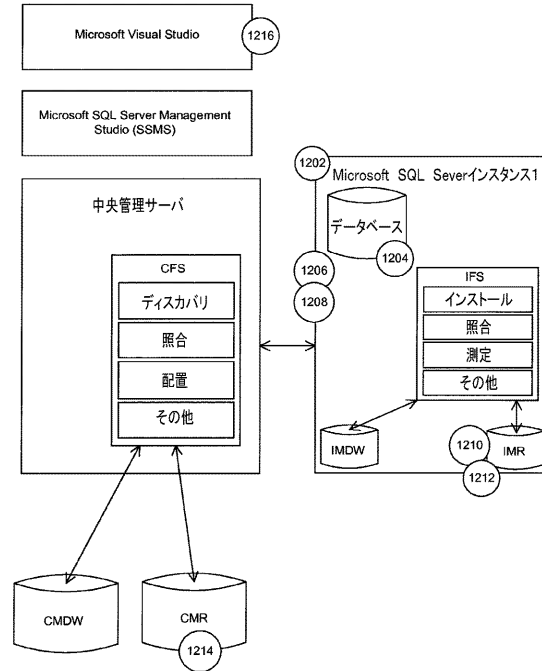
【図 10】



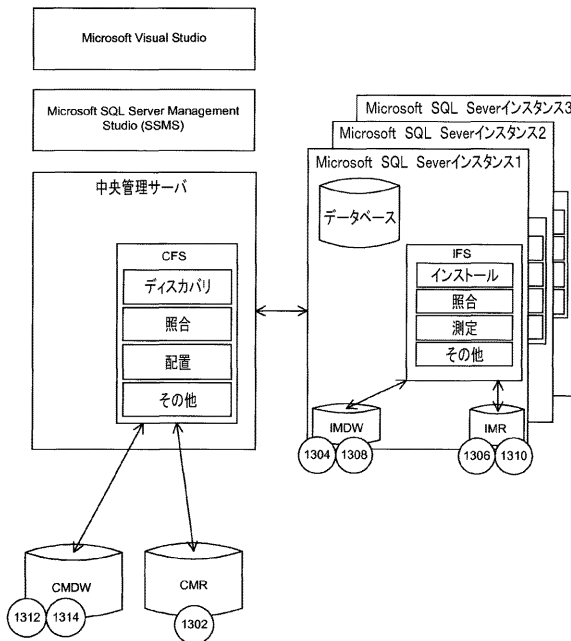
【図 1 1】



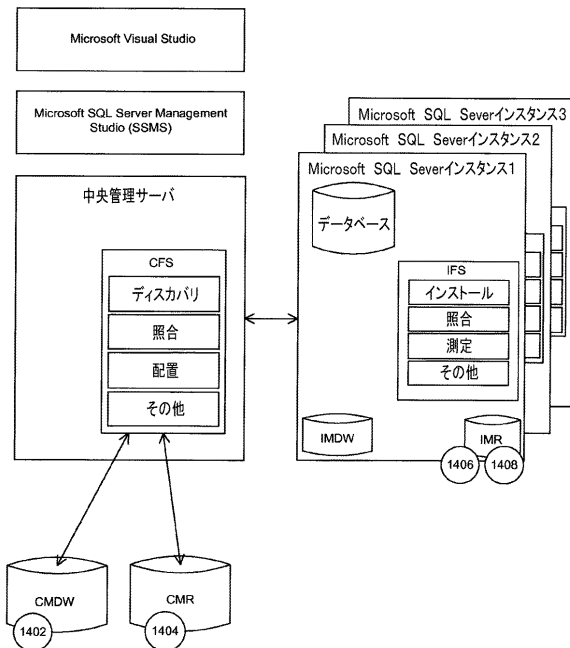
【図 1 2】



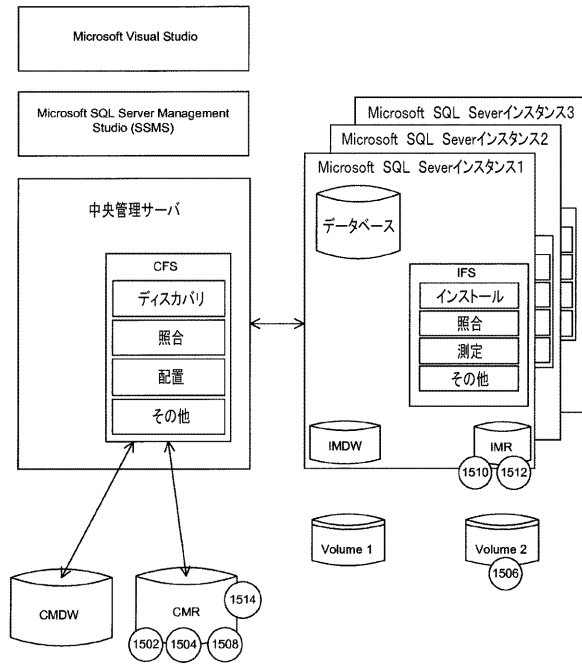
【図 1 3】



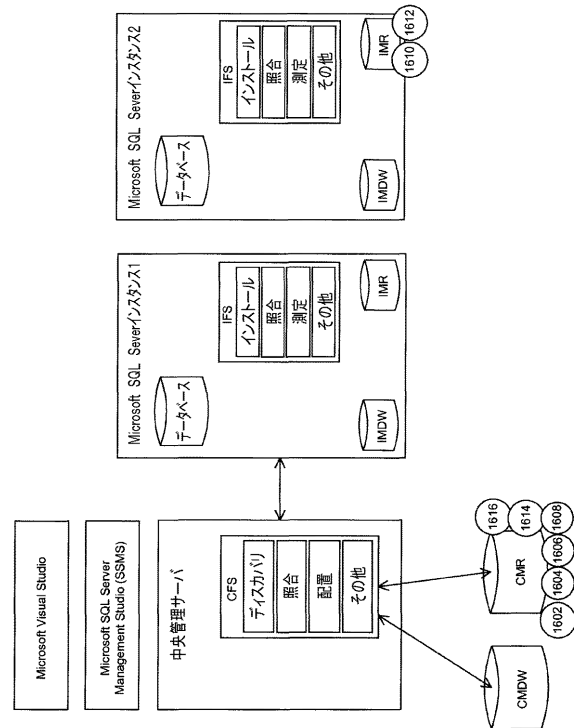
【図 1 4】



【図 15】



【図 16】



フロントページの続き

- (74)代理人 100120112
弁理士 中西 基晴
- (74)代理人 100147991
弁理士 鳥居 健一
- (74)代理人 100119781
弁理士 中村 彰吾
- (74)代理人 100162846
弁理士 大牧 綾子
- (74)代理人 100173565
弁理士 末松 亮太
- (74)代理人 100138759
弁理士 大房 直樹
- (74)代理人 100091063
弁理士 田中 英夫
- (72)発明者 オムリ バハット
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション エルシーエー - インターナショナル パテンツ内
- (72)発明者 ショーン バイス
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション エルシーエー - インターナショナル パテンツ内
- (72)発明者 サーメル ボシュラ
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション エルシーエー - インターナショナル パテンツ内
- (72)発明者 チャーリー ディー . カーソン
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション エルシーエー - インターナショナル パテンツ内
- (72)発明者 マット ホリングスワース
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション エルシーエー - インターナショナル パテンツ内
- (72)発明者 ロバート ハッチンソン
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション エルシーエー - インターナショナル パテンツ内
- (72)発明者 プラヴィーン セシャドリ
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション エルシーエー - インターナショナル パテンツ内
- (72)発明者 ケン ヴァン ハイニング
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション エルシーエー - インターナショナル パテンツ内
- (72)発明者 サミア ヴェルケダカール
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション エルシーエー - インターナショナル パテンツ内

審査官 塚田 肇

- (56)参考文献 国際公開第2007/145680(WO, A1)
特開2004-272908(JP, A)
特表2004-536391(JP, A)

(58)調査した分野(Int.Cl. , D B 名)

G 0 6 F 9 / 4 4 5

G 0 6 F 9 / 4 4

G 0 6 F 1 1 / 3 4