

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5711130号
(P5711130)

(45) 発行日 平成27年4月30日(2015. 4. 30)

(24) 登録日 平成27年3月13日(2015. 3. 13)

(51) Int.Cl.

F I

G 0 6 F 9/50 (2006.01)

G 0 6 F 9/46 4 6 5 D

請求項の数 14 (全 32 頁)

| | | | |
|---------------|-------------------------------|-----------|---------------------|
| (21) 出願番号 | 特願2011-530096 (P2011-530096) | (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2012-504297 (P2012-504297A) | | アメリカ合衆国 ワシントン州 9805 |
| (43) 公表日 | 平成24年2月16日 (2012. 2. 16) | | 2-6399 レッドモンド ワン マイ |
| (86) 国際出願番号 | PCT/US2009/057050 | | クロソフト ウェイ |
| (87) 国際公開番号 | W02010/039428 | (74) 代理人 | 100140109 |
| (87) 国際公開日 | 平成22年4月8日 (2010. 4. 8) | | 弁理士 小野 新次郎 |
| 審査請求日 | 平成24年9月7日 (2012. 9. 7) | (74) 代理人 | 100075270 |
| (31) 優先権主張番号 | 12/242, 808 | | 弁理士 小林 泰 |
| (32) 優先日 | 平成20年9月30日 (2008. 9. 30) | (74) 代理人 | 100101373 |
| (33) 優先権主張国 | 米国 (US) | | 弁理士 竹内 茂雄 |
| 前置審査 | | (74) 代理人 | 100118902 |
| | | | 弁理士 山本 修 |
| | | (74) 代理人 | 100162846 |
| | | | 弁理士 大牧 綾子 |

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ファブリック管理

(57) 【特許請求の範囲】

【請求項 1】

データベースエレメント(510、512、514)のコレクション及びデータ層アプリケーションコンポーネントのポリシー(104)を各々が含むデータ層アプリケーションコンポーネントのセット(502)、

前記データ層アプリケーションコンポーネントのセット(502)をホストするデータベースランタイムリソースのセット(516)、及び

前記データ層アプリケーションコンポーネントのセット(502)をホストするために前記データベースランタイムリソースのセットによって使用されるコンピューティングリソース(524)、

を識別する、ファブリック(501、1106)を含むコンピュータ読み取り可能なデータストレージデバイスと、及び

1又は複数のファブリックポリシのうち特定のファブリックポリシを受信する管理ポイント(602)と、を含み、

前記ファブリックポリシは、少なくとも1つのデータベースランタイムリソースの実行に関連しており、

前記ファブリックポリシは、前記データ層アプリケーションコンポーネントのセット、前記データベースランタイムリソースのセット、及びコンピューティングリソースのいずれかに関連する最適値、又は限界値を規定しており、

前記ファブリックは、前記データ層アプリケーションコンポーネントのセット、前記デ

データベースランタイムリソースのセット、及び前記コンピューティングリソースに関連づけられており、前記ファブリックがこれらの関連づけの変更を識別すると、前記特定のファブリックポリシーに関連する1または複数の動作が、前記ファブリック(501、1106)によって識別された前記データ層アプリケーションコンポーネントのセット、前記データベースランタイムリソースのセット、前記コンピューティングリソースに関連するエンティティに適用されることを特徴とするシステム。

【請求項2】

前記1または複数の動作が、

前記コンピューティングリソースのファイルシステムが増大するかまたはリソース限界に達すると予測された場合に、前記ファイルシステムの1または複数のファイルを他のコンピューティングリソースのオペレーティングシステムボリュームの第2のファイルシステムに移動させる動作と、

前記コンピューティングリソースのプロセッサが高負荷を受けた場合に、ソフトウェアアプリケーションを低プロセッサ負荷の他のコンピューティングリソースに移動する動作を含むことを特徴とする請求項1に記載のシステム。

【請求項3】

前記コンピューティングリソースが、1もしくは複数のコンピュータリソース、1もしく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もしくは複数の仮想マシンリソースを含むことを特徴とする請求項1に記載のシステム。

【請求項4】

前記データ層アプリケーションコンポーネントの各々が、ファブリックのデータ層アプリケーションコンポーネント層に関連し、前記コンピューティングリソースの各々が、前記ファブリックのハードウェアリソース層に関連し、前記データベースランタイムリソースの各々が、前記ファブリックのランタイム層に関連していることを特徴とする請求項1に記載のシステム。

【請求項5】

前記データ層アプリケーションコンポーネントの各々が、前記ファブリックによって前記データベースランタイムリソースの1つにマッピングされ、データベースランタイムリソースの各々が、前記ファブリックによって前記コンピューティングリソースにマッピングされることを特徴とする請求項2に記載のシステム。

【請求項6】

前記コンピューティングリソースは複数のコンピュータを含み、前記データベースランタイムリソースの前記1つは、前記ファブリックポリシーに従って前記複数のコンピュータのうち第1のコンピュータから前記複数のコンピュータのうち第2のコンピュータにマッピングが変更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7】

前記コンピューティングリソースは複数のオペレーティングシステムボリュームを含み、前記データベースランタイムリソースの前記1つは、前記ファブリックポリシーに従って前記複数のオペレーティングシステムボリュームのうち第1のオペレーティングシステムボリュームから前記複数のオペレーティングシステムボリュームのうち第2のオペレーティングシステムボリュームにマッピングが変更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8】

前記特定のファブリックポリシーの前記1または複数の動作は、1または複数のリソースを特定のアプリケーションに割り当てる動作、並びにソフトウェアアプリケーション及び前記ソフトウェアアプリケーションに関連付けられているファイルを前記コンピューティングリソースの第1のコンピュータシステムから第2のコンピュータシステムに移動させる動作のうちの1または複数を含むことを特徴とする請求項1に記載のシステム。

【請求項9】

1または複数のファブリックポリシーを適用する、コンピュータによって実行される方法

10

20

30

40

50

であって、該ファブリックポリシは少なくとも1つのデータベースランタイムリソースの実行に関連しており、該方法は

サーバファブリック(501、1106)の管理ポイント(602)において前記1または複数のファブリックポリシのうち特定のファブリックポリシを受信するステップ(702)であって、該サーバファブリック(501、1106)はコンピュータ読み取り可能データストレージデバイス(902、904、906)に記憶されており、該サーバファブリックは、データ層アプリケーションコンポーネント、データベースランタイムリソース、及びコンピューティングリソースを関連づけている、ステップと、

各々がデータベースエレメントのコレクションおよびデータ層アプリケーションコンポーネントポリシ(104)を含む前記データ層アプリケーションコンポーネント(502)、

前記データ層アプリケーションコンポーネント(502)をホストする前記データベースランタイムリソース(516)、及び

前記データ層アプリケーションコンポーネント(502)をホストするために前記データベースランタイムリソース(516)によって使用される前記コンピューティングリソース(524)

を識別するステップと、

前記サーバファブリック(501、1106)のエンティティを監視するステップ(704)と、および

前記サーバファブリックが前記データ層アプリケーションコンポーネント、前記データベースランタイムリソース、及び前記コンピューティングリソースの関連づけの変更を識別すると、前記サーバファブリック(501、1106)によって識別された前記データ層アプリケーションコンポーネント、前記データベースランタイムリソース、前記コンピューティングリソースに関連するエンティティに前記特定のファブリックポリシに関連する1または複数の動作を適用するステップと、

を含み、前記ファブリックポリシはデータ層アプリケーションコンポーネント、前記データベースランタイムリソース、及びコンピューティングリソースのいずれかに関連する最適値、又は限界値を規定することを特徴とする方法。

【請求項10】

前記1または複数の動作を適用するステップが、前記特定のファブリックポリシを取り出すステップと、1または複数の対応する動作を生成して前記特定のファブリックポリシを適用するステップと、を含むことを特徴とする請求項9に記載の方法。

【請求項11】

前記特定のファブリックポリシに従って、前記コンピュータは前記サーバファブリックのリソースの現在の利用を監視して、該リソースの利用がリソースの能力の一定値を超えているかを判定することを特徴とする請求項9に記載の方法。

【請求項12】

前記1または複数の動作はリソース使用統計を含むことを特徴とする請求項9に記載の方法。

【請求項13】

前記特定のファブリックポリシは活動または動作を誘導するように定義された権限のある方針であり、前記特定のファブリックポリシは、前記サーバファブリックに関連するソフトウェアのバージョンに基づいて、前記エンティティのセットの前記ファブリックへの組み込みを制限することを特徴とする請求項9に記載の方法。

【請求項14】

前記エンティティのセットが1または複数のデータベースランタイムインスタンスを含み、前記特定のファブリックポリシは、前記1または複数のデータベースランタイムインスタンスに適用される条件に基づいて、前記1または複数のデータベースランタイムインスタンスの前記サーバファブリックへの組み込みを制限することを特徴とする請求項13に記載の方法。

10

20

30

40

50

【発明の詳細な説明】

【背景技術】

【0001】

アプリケーションのオーサリング (authoring)、インストール及び実行時を含むアプリケーションのライフサイクルの間、通常は、多くの異なったオブジェクトが形成されるかまたはアプリケーションと関連付けられる。例えば、プログラミングコード、実行可能コード、ソフトウェアリソース、及びハードウェアリソースは、アプリケーションと関連付けられてもよく、これらのアイテム各々は、様々なファイル名及びファイルタイプを割り当てられる。データベースアプリケーションは、プログラミングコード、実行可能コード及びリソースと関連付けられるのに加え、ポリシ、表、保存された処理、及びデータベースポリシ等の追加のアイテムと関連付けられてもよい。

10

【発明の概要】

【発明が解決しようとする課題】

【0002】

追跡及び維持のために、データベース管理者にとって、データベースのライフサイクルに亘ってデータベースに関連付けられている全てのアイテムを効率的に管理することは、しばしば困難でありかつ多大な時間を必要とするものである。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って使用されているハードウェアもしくは他のアイテムの変更を行いたい、またはデータベースアプリケーションがインストールされているデータベースサーバインスタンスを追加もしくは変更したいと望む場合、所望の変更を行うプロセスは、多大な時間を必要としかつコストがかかる。オブジェクトは、コード化されてもよいし、当該オブジェクトが存在して動作するハードウェアリソースに直接リンクされてもよく、アイテムとハードウェアリソースとの間の関連付けの変更は、著しい再プログラミングを伴う。アイテムの変更に関する試みは、アプリケーションオブジェクトが複数の場所に保存されているか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使用されるオブジェクトを保存している場合に増大する。

20

【課題を解決するための手段】

【0003】

データベースのライフサイクル段階において高い管理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によって生成されるかまたは使用されるアイテムに関連付けられているグルーピング抽象概念 (grouping abstraction) を有することが有効である。ソフトウェアグルーピング抽象概念無しに、データベースアプリケーションを識別、概念化、移動もしくは再生成、またはその他管理をすることは、非常に困難である。本明細書に開示されている実施形態は、データベースアプリケーションが全体として、例えば、移動されるかまたは再生成されることを可能にする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ているコンテナ内のデータベースアプリケーションのエLEMENT (element) に関する。コンテナは、ハードウェア又は他のランタイムリソースを変更可能であるかまたはデータベースアプリケーションのエLEMENTを選択して当該ELEMENTを記録すること無しに異なったリソースを使用することが可能なデータベースELEMENTと、ランタイムリソース (runtime resource) 識別子を関連付ける。

30

40

【0004】

さらに、複数のデータベースアプリケーションコンテナは、アプリケーションコンテナが全体として、移動させられるか、再生成されるかまたは他の管理がされることを可能とする単一のグルーピングまたはファブリック (fabric) と関連付けられてもよい。このファブリックは、データベースアプリケーションのエLEMENTによって使用されるリソース識別子を参照し、複数のデータベースアプリケーションの管理を可能とする。

【0005】

特定の実施形態によれば、1のシステムは、データ層アプリケーションコンポーネント (data-tier application component) (DAC) のセットを識別するファブリックを含む。データ層アプリケーションコンポーネントの各々は、データベースのコレクション (

50

collection) の論理表示 (logical representation)、データ層アプリケーションコンポーネントをホストするデータベースランタイムリソースのセット、及びデータ層アプリケーションコンポーネントをホストするためにデータベースランタイムリソースのセットによって使用されるコンピューティングリソース (computing resource) を含む。システムは、ファブリックポリシーを受信するための管理ポイント (management point) も含む。ファブリックポリシーの 1 または複数の動作は、当該ファブリックによって識別されている影響を受けたエンティティに自動的に適用されて、当該ファブリックポリシーにファブリックエレメントを準拠させる。

【0006】

この発明の概要は、発明を実施するための形態において以下でさらに説明されるいくつかの概念を、単純な形式で紹介するために提供されている。この発明の概要は、本発明の重要な特徴または本質的な特徴を特定するためのものではなく、本発明の範囲を限定するためのものでもない。

【図面の簡単な説明】

【0007】

【図1】データ層アプリケーションコンポーネント (DAC) を使用するデータベース開発ツールを示すブロック図である。

【図2】DAC 及び 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示すフロー図である。

【図3】DAC の特定の実施形態を示す全体図である。

【図4】関連付けられている DAC 名を有している DAC の第2の実施形態を示すブロック図である。

【図5】DAC のセットを含むサーバファブリックの特定の実施形態を示すブロック図である。

【図6】DAC のセットを含むサーバファブリックの第2の実施形態を示すブロック図である。

【図7】データベース管理動作の適用及びサーバファブリックのエンティティにおけるポリシーの適用の方法を示すフロー図である。

【図8】サーバファブリックのエンティティ関係を示すフロー図である。

【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エレメントの処理方法を示すフロー図である。

【図11】SQL インスタンスを管理するシステムの第1の特定の実施形態を示すブロック図である。

【図12】SQL インスタンスを管理するシステムの第2の特定の実施形態を示すブロック図である。

【図13】SQL インスタンスを管理するシステムの第3の特定の実施形態を示すブロック図である。

【図14】SQL インスタンスを管理するシステムの第4の特定の実施形態を示すブロック図である。

【図15】SQL インスタンスを管理するシステムの第5の特定の実施形態を示すブロック図である。

【図16】SQL インスタンスを管理するシステムの第6の特定の実施形態を示すブロック図である。

【発明を実施するための形態】

【0008】

特定の実施形態は、データベースアプリケーションのエレメントを識別してデータベースアプリケーションの管理の一単位を提供するデータ層アプリケーションコンポーネント (DAC) の使用を説明する。DAC によって識別されてい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及び / またはデータベースオブジェクトのメタ

10

20

30

40

50

データに関連付けられる。データベースオブジェクト識別子及び/またはデータベースオブジェクトのメタデータを介して、データベースアプリケーションが操作される。さらに、データベースランタイム識別子が、データベースアプリケーションのエLEMENTに使用されている1または複数のランタイムリソースに関連付けられている。データベースランタイムリソースの実行に関連している1または複数のポリシもDACによって識別される。結果として、DACは、データベースオブジェクト識別子によって参照されて、データベースアプリケーションが変更、移動または複製され得る。同様に、ランタイムリソースが変更されてもよいし、またはデータベースアプリケーションが、各々のELEMENTに関連付けられているデータベースランタイム識別子を変更することによって複製されてもよい。結果として、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組織特性の管理に時間を充てて留意することなく、データベースアプリケーションのコンテンツの管理及びポリシによって特定される様な目的に集中することが可能である。

10

【0009】

特定の実施形態において、データ層アプリケーションコンポーネントのセットを識別するファブリックを含むシステムが開示される。データ層アプリケーションコンポーネントは、データベースELEMENTのコレクションの論理表示を含む。ファブリックは、データ層アプリケーションコンポーネントのセットをホストするデータベースランタイムリソースのセットも識別する。ファブリックは、データ層アプリケーションコンポーネントのセットをホストするデータベースランタイムリソースのセットによって使用されるコンピューティングリソースを識別する。このシステムは、ファブリックポリシを受信するために管理ポイントも含む。ファブリックポリシの1または複数の動作は、当該ファブリックによって識別されている影響を受けたエンティティに自動的に適用されて、当該ファブリックポリシにファブリックELEMENTを準拠させる。

20

【0010】

他の特定の実施形態において、ファブリックポリシを自動的に適用する方法が開示される。この方法は、サーバファブリックの管理ポイントでのファブリックポリシの受信を含む。サーバファブリックは、1または複数のデータ層アプリケーションコンポーネントを識別する。データ層アプリケーションコンポーネントの各々は、データベースELEMENTのコレクションの論理表示を含む。サーバファブリックは、データ層アプリケーションコンポーネントのセットをホストするデータベースランタイムリソースを識別する。このサーバファブリックは、データ層アプリケーションコンポーネントをホストするデータベースランタイムリソースによって使用されるコンピューティングリソースを識別する。この方法は、サーバファブリックのエンティティを監視すること、及びサーバファブリックによって識別されて影響を受けたエンティティにおいてファブリックポリシの1または複数のアクションを自動的に適用して、サーバファブリックに関連付けられているELEMENTをファブリックポリシに準拠させることを含む。例えば、ファブリックポリシの1または複数の動作を自動的に適用することには、ファブリックポリシを検索すること、目標のセットに亘ってファブリックポリシを評価すること、1または複数の対応する動作を生成してファブリックポリシを実施すること、及び目標のセットに対して対応する動作を実行することを含む。目標物のセットは、コンピューティングシステムの1または複数の物理コンピューティングリソースまたはコンピューティングシステムの全てのコンピューティングリソースを含んでもよい。

30

40

【0011】

開示される本発明は、データベースアプリケーションを構成するアイテムに関連しているソフトウェアグルーピング抽象概念を含む。ソフトウェアグルーピング抽象概念は、データベースアプリケーションの識別、概念化、及び管理を可能にする。本開示は、データベースアプリケーションに関連付けられているデータベースオブジェクト、ポリシ、及びランタイムリソースを効率的にモデリングする、データ層アプリケーションコンポーネント(DAC)と称される抽象概念を含む。

【0012】

50

DACは、アプリケーションの全ライフサイクルに関する、バージョンング(versioning)を含む管理の一単位を形成する。例えば、ヒューマンリソース(HR)DACは、これをサポートしているHRアプリケーションコードとともに開発されてもよい。HRアプリケーション実行可能ファイルがアプリケーションサーバに配置されると、HR DACは、マイクロソフトSQLサーバのインスタンスに配置される。その後、HR DACは監視されかつ管理されることが可能である。例えば、HR DACにおいて予測される負荷が、HR DACが維持される現在のコンピュータまたはデータベースインスタンスの能力を超えている場合、HR DACは、新しいデータベースインスタンスに移動させられてもよい。移動は、DACに関連付けられてい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して、データベースエレメントに関連付けられているデータベースランタイム識別子を新しいデータベースインスタンスの参照ランタイムリソースに変更することによって完遂される。DACは、全体として複数のタイプのデータベースランタイム環境、並びに/または特に、マイクロソフトSQLサーバインスタンス、マイクロソフトSQLデータベースエンジンマイクロソフトSQLサーバアナリティクスサービスインスタンス、マイクロソフトSQLサーバレポーティングサービスインスタンス、データベースサーバ及び/もしくは他のベンダー及び他のタイプのデータベースを含むマイクロソフトSQLサーバランタイム環境に亘り得る。

【0013】

さらに、本開示は、全体としてデータベース環境の管理に利益をもたらし、DACの管理に更に集中することを可能とし、個人のコンピュータ、データベースインスタンス及びデータベースの管理への注意を軽減することを可能とする。本開示は、この問題を「ファブリック」(「マイクロソフトSQLサーバファブリック」とも称される)の概念によって解決する。この概念は、組織のDACs、データベースインスタンス(例えば、マイクロソフトSQLサーバ)ランタイム、及びハードウェアリソースの全てをモデリングする。ファブリックは、DAC、ランタイム及びリソース層内のエンティティ及びコンテナの統合表示を提供し、エンティティ間の更に動的なマッピングを可能にする。ファブリックは、管理動作を宣言及び起動する中央管理ポイント、中央ポリシー管理者、並びに影響分析(impact analysis)、what-if分析または配置分析等の分析に使用されるモデル及びデータを含む中央推論ポイントも含む。これらの宣言型のモデル駆動の特徴は、組織ルールを実施する(enforce)全体的な管理動作を規定することを管理者及び開発者に可能にし、DACによって要求される全ての特徴を有するランタイムにおいてDACが配置されることを保証し、サーバファブリック内のエンティティが最小の手作業の介入で環境の変化に対応することを可能にする。

【0014】

ファブリック内のエンティティの状態のより深い知識によって駆動されるこれらの自己管理能力は、データ層アプリケーションコンポーネントの開発、配置及び管理を、管理者及び開発者に対して著しく容易にするために必要とされる。本開示によって、データベース管理者及び開発者が、アプリケーションのライフサイクルを通して彼らのアプリケーションを管理することにおいて更に生産的になり、彼らのシステムの中に追加のデータ及び記憶プラットフォーム技術を包含させる更なる時間を彼らに与える。

【0015】

本開示は、データベースインスタンス(例えば、マイクロソフトSQLサーバインスタンス)、データベース、コンピュータ、記憶装置及びネットワークリソースへの従来の注意の集中から、データベースアプリケーションへ注意の集中を移す。DACは、データベースインスタンス及びデータベースの両方の上に存在する新しい更に高いレベルの抽象概念である。DACは、従来の三階層のアプリケーションへのデータ層サポートを形成するデータベースインスタンスエンティティのセットを含む。近年のシステムにおいて、データベースアプリケーション及びシステムレベルアプリケーション情報は、しばしば複数のデータベース及びデータベースインスタンスに亘る。例えば、マイクロソフトSQLサーバシステムレベルアプリケーション情報は、「マスタ」及び「msdb」等のシステムデ

10

20

30

40

50

データベース内に保存されてもよい。

【0016】

さらに具体的には、本開示は、DACと称される、コンテナ内のデータベースアプリケーションをモデリングする。DACは、データベースアプリケーションアーティファクト(database application artifact)、関連するポリシ及びアプリケーションに関して参照されるランタイムリソースの全てをまとめる論理表示である。DACの各々は、管理の一単位として提供される。ポリシは、DACに対して適用されても良く、DACは、全体として管理される。ポリシは、要求されるかまたは好ましい結果または判断の、開発者意図または配置意図を明示するために使用されてもよい。例えば、開発者は、DACが64ビットランタイムにおいてのみ配置可能であるというポリシを特定することによって要求された意図を宣言することが可能である。

10

【0017】

ファブリックは、DACのセットをモデリングし、これらを下層のデータベースインスタンスランタイム(例えば、マイクロソフトSQLサーバランタイム、マイクロソフトSQLサーバインスタンス)及び当該ランタイムによって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、DACが機能するために必要とするランタイム及びハードウェアリソースの全てのモデルを提供する。ファブリックは、他の技術を使用すると非常に複雑な動作を単純化する中央管理特性を提供する。例えば、DACの各々は、ファブリックによって管理される1または複数の既知の名前(「エンドポイント名」)を有する。このエンドポイント名は、DACによって使用されるハードウェアリソースまたはインスタンスから独立している。結果として、アプリケーションは、コンピュータ名及びインスタンス名の代わりにエンドポイント名を用いて接続をし、アプリケーションの変更の要求無しにインスタンス間でDACが移動されることが可能になる。

20

【0018】

利用可能なDAC、ランタイム及びリソースの、ファブリック内の中央管理機能及び推測機能を用いた結合は、様々な計画を可能とする。例えば、サーバファブリック内のDACのリソース使用が監視されてもよい。他の例において、中央推論エンジンは、DACの将来の作業負荷を、これらのホストランタイムに割り当てられるリソースに対して予測してもよい。他の例において、動作は、予測されたリソース不足に対応して行われてもよい。例えば、システムは、ランタイムに使用可能なリソースを動的に管理して、能力超過のランタイムを発見して、実行されているアプリケーションへの影響を最小化するか少なくとも低減させる態様でDACを当該ランタイムに移動してもよい。ツール及び機能のセットは、ファブリック内のライフサイクル(開発、配置及び管理)の間にDACを管理するために記述される。

30

【0019】

特定の実施形態において、DACは、データベースオブジェクトのメタデータ、データベースオブジェクトを操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、及びデータベースランタイムリソースの実行に関連するポリシをパッケージングする管理の一単位である。図1に示しているように、異なった動作が、ライフサイクルにおけるDACの状態に基づいてDACに適用され得る。

40

【0020】

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を生成してもよい。DACモデル102は、データ層アプリケーションコンポーネント(DAC)104を含む。DAC104は、DACコンテナ106内に含まれていてもよい。DAC104は、データベースエレメントのコレクションを含む。図1に示されているように、DAC104は、第1のエレメント108、第2のエレメント110、及び第3のエレメント112を含む。第1のエレメント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んでもよい。第2のエレメント110は、データベースオブジェクトを操作するために動作可能なデータベースランタイムリソースを含んでもよい。第3のエレメント112は、データベースランタイムリソースの

50

実行に関連するポリシを含んでもよい。DACモデル102内に表示されたデータベースアプリケーションの開発の間、ユーザ100は、DACモデルプロジェクト114として、DACモデル102を生成、編集及び有効化してもよい。DACプロジェクト114は、1または複数のファイル118を含んでもよい。その後、DACプロジェクト114は、データベースアプリケーションが開発下にある間に、ソース制御システムまたはソース制御データベース116にチェックイン (checked into) 及びチェックアウト (checked out) させられてもよい。

【0021】

一度、ユーザ100 (例えば、開発者) がDACモデル102内に示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に基づいた完全なDACパッケージ120にコンパイルされるかまたは組み込まれる。DACパッケージ120は、必要に応じて再生成されて配置されてもよいDACモデル102の概念化バージョンを示すDACタイプ122として、ファブリック126の中に取り込まれる。DACタイプ122は、DACモデル102として開発されているデータベースアプリケーションの操作的インストール (operational installation) を示す、配置されたDACインスタンス124を生成するために配置されてもよい。従って、DACモデル102を開発する際に、開発者は、データベースアプリケーションの記録無しに複製され得るポータブルデータベースアプリケーションを生成する。開発者は、DACインスタンス124の配置前に、DACタイプ122を設定してもよい。さらに、DACインスタンス124は、物理コンピューティングリソースの特定のセットに関してインストールされてもよい。

【0022】

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の配置の特定の方法のフロー図が示されている。この方法は、202において、1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エレメントのコレクションを包含するDACモデルを生成するステップを含む。DACモデルは、データベースオブジェクトを操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を更に含む。DACモデルは、データベースランタイムリソースの実行に関連するポリシを更に含む。特定の実施形態において、204にて、この方法は、DACプロジェクトとしてのDACモデルの表示を保存するステップと、DACプロジェクトを保存するために、DACプロジェクト内で、ソース制御システム内またはソース制御データベース内に任意的にチェックインするステップと、を含んでいる。他の特定の実施形態において、206にて、この方法は、DACモデルを編集して、DACパッケージの構成の前にDACモデルを有効にするステップと、DACインスタンスを配置する前のDACタイプを設定するステップと、物理コンピューティングリソースの特定のセットに関連してDACインスタンスをインストールするステップと、を含む。208に移動すると、この方法は、DACモデルに基づいてDACパッケージを構成するステップを含む。210に移動すると、この方法は、DACタイプを、DACパッケージからサーバファブリック内に取り込むステップを含んでいる。212に移動すると、この方法は、DACタイプを配置して、配置されたDACインスタンスを生成するステップを含む。

【0023】

DACが、DACタイプ及びDACインスタンスとの間のサーバファブリック内において、2つのメイン下位区分を有することが強調されるべきである。DACタイプは、DACがサーバファブリックに取り込まれた場合に構成される。DACタイプは、表及びキューブ (cube) 等の、DAC内のオブジェクトの論理実装を定義し、かつ使用されるDACに対して開発者がどのように意図しているかを定義するポリシを含む。DACインスタンスは、DACタイプがサーバファブリック内に1または複数のデータベースサーバランタイム (例えば、マイクロソフトサーバランタイム) に配置される際に構築される。DACインスタンスは、DACタイプのエンティティがどのようにサーバランタイム層内のホストに位置するかを記録する。DACインスタンスは、DACに関して管理者がどのよう

10

20

30

40

50

に操作することを意図しているかを定義するポリシを含む。

【 0 0 2 4 】

図 3 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 (D A C) のコンテンツのハイレベル表示が示されている。この図は、開発者の意図 (D A C タイプ) の、配置意図 (D A C インスタンス) からの概念的分離を示している。図 3 に示されているように、D A C は、D A C タイプ 3 0 4 及び D A C インスタンス 3 0 6 に概念的に分割されている。D A C タイプ 3 0 4 は、構造、ロジック、ポリシ、及び D A C 定義を含む他のエンティティを含む。D A C タイプ 3 0 4 は、開発者のための仕組みも提供して、不変条件、要求、並びに D A C 3 0 2 の意図された使用法及び解釈を形式的に定義する。D A C タイプ 3 0 4 を含むエンティティは、一単位として管理されることが意図されている。この情報は、D A C 3 0 2 の特定のバージョンの全ての配置に対する全体的な情報である。従って、D A C タイプ 3 0 4 は、開発者の意図を表してもよく、バージョン管理されてもよく、特定のバージョンに対して不変であり、かつ、唯一であると保証される識別子を有している。D A C タイプ 3 0 4 の 3 つの主セクションは、構成の目的で、プロパティ 3 0 8 、アプリケーションスキーマ 3 1 0 、並びに開発者意図及びポリシ 3 1 2 と示される。

10

【 0 0 2 5 】

D A C タイプ 3 0 4 のプロパティ 3 0 8 は、D A C タイプ 3 0 4 に特有の特徴または特性である。例えば、プロパティ 3 0 8 は、人間が可読な名前及び説明を含んでもよい。他の例として、プロパティ 3 0 8 は、メジャーバージョン番号及びマイナーバージョン番号等の 1 または複数のバージョン識別子を含んでもよい。他の例として、プロパティ 3 0 8 は、D A C タイプ 3 0 4 にとって唯一であると保証されている識別子を含んでもよい。他の例として、プロパティ 3 0 8 は、既知のエンドポイント名を含んでもよい。他の例として、プロパティ 3 0 8 は、他の D A C タイプへの依存関係を含んでもよい。

20

【 0 0 2 6 】

D A C タイプ 3 0 4 のアプリケーションスキーマ 3 1 0 は、D A C 3 0 2 を含むデータベースインスタンス及びデータベースオブジェクトを定義する。D A C タイプ 3 0 4 のこの部分のオブジェクト定義は、D A C 3 0 2 に固有のものであり、任意の変更は、D A C タイプの新しいバージョンを生成可能である。アプリケーションスキーマ 3 1 0 は、D A C 3 0 2 が「何」で構成されているかを本質的に定義する。例えば、アプリケーションスキーマ 3 1 0 は、表、キューブ、及びデータマイニング (mining) モデル等のアプリケーション構造を含んでもよい。他の例において、アプリケーションスキーマ 3 1 0 は、保存された手順、ユーザ定義の機能、及びトリガ等のアプリケーションロジックを含んでもよい。他の例として、アプリケーションスキーマ 3 1 0 は、アプリケーション定義の一部と考えられる参照データを含んでもよい。例えば、ジップコード及び他の郵便番号、または州及び地域コード等の参照データが参照されてもよい。

30

【 0 0 2 7 】

D A C タイプ 3 0 4 の開発者意図及びポリシ 3 1 2 は、D A C 3 0 2 がどのように使用するかまたは解釈することを目的としているかについての追加情報を提供する。例えば、開発者意図及びポリシ 3 1 2 は、D A C 3 0 2 を配置することが可能であるべきファブリックリソース要求を含んでもよい。例えば、D A C 3 0 2 は、マイクロソフト S Q L サーバ 2 0 0 8 を要求して、空間データタイプ及び C L R 統合に対するサポートを可能とし、かつ 8 0 G B のディスクスペースを要求してもよい。他の例として、開発者意図及びポリシ 3 1 2 は、D A C 3 0 2 に組み込まれる設定可能性を定義する設定意図 (パラメータ化) を含んでもよい。ポリシに関する他の例は、利用可能性要求の詳細であり、例えば、D A C は、クラスタリングが可能である S Q L サーバインスタンスにおける配置を要求する。このことは、配置時間において供給され得るまたはされるべき情報を含む、D A C バージョンに影響を与えない配置特有の設定を可能にする。例えば、設定パラメータは、配置プロセスが、ファイルグループ及びマイクロソフト S Q L サーバのインスタンスの名前等のファブリックリソースの特定のタイプへのマッピングステップを提供すべきことを示すことが可能である。他の例として、開発者意図及びポリシ 3 1 2 は、D A C 3 0 2 の

40

50

どの部分が外部に曝されるのかを特定するインタフェース定義を含んでもよい。これは、パブリックインタフェースとDAC302内部の実施詳細との分離を可能にする。例えば、インタフェース定義は、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他のDACによって参照され得るかを特定し得る。他の例として、開発者意図及びポリシ312は、このDAC302に対して実行されるクエリ等の作業負荷特性情報を含んでもよい。

【0028】

図3に示されているように、DACインスタンス306は、実際のインスタンスまたは配置特有の情報を捕集する。この情報は、特定の配置に特有である。DACインスタンス306は、配置意図を表現する。DACインスタンス306は、DACタイプ304に関連付けられている。複数のDACインスタンスは、DACタイプ304の各々に対して存在している。DACインスタンス306は、サーバファブリック内で唯一であると保証されている識別子を有している。DACインスタンス306は、サーバファブリックマッピングを含む。

【0029】

DACインスタンス306のプロパティ314は、DAC302の配置されたインスタンスに特有の特徴または特性である。プロパティ314は、DACインスタンス306識別子及び配置名等のアイテムを含む。

【0030】

DACインスタンス306の配置スキーマ316は、DACインスタンス306の一部であり、特定の配置に特有のSQLオブジェクトを定義する。配置スキーマ316は、認証プリンシパル及びインデックス等のオブジェクトを含む。

【0031】

DACインスタンス306の配置意図及びポリシ318は、どのようにDAC302がどのように配置又は解釈されることが意図されているかについての情報を提供する。配置意図及びポリシ318は、特定の配置に特有の意図されたファブリックマッピング及び意図されたポリシ等のアイテムを含む。

【0032】

ファブリックマッピング320は、DACインスタンス306の一部であり、DACインスタンス306と下層にあるリソースとの間のマッピングを定義する。ファブリックマッピング320のコンテンツの例は、関係データベースインスタンスまたは解析サーバインスタンス等のマイクロソフトSQLサーバランタイムマッピング、及びディスクストレージ及びコンピュータ等のハードウェアリソースを含む。

【0033】

データ322は、DACインスタンス306の一部であり、特定のDACインスタンスに関連付けられているデータを示す。データ322は、DACインスタンス306またはアプリケーションに特有な動作データ等のアプリケーションデータ及び動作データの両方のユーザによって生成されるアプリケーションデータを含む。

【0034】

図4を参考にすると、データ層アプリケーションコンポーネント(DAC)の特定の実施形態が、400として示されている。DAC402は、データベースエレメント404のコレクションを含む。データベースエレメント404のコレクションは、データベースオブジェクト406のデータベースオブジェクト識別子を含む。データベースエレメント404のコレクションは、データベースオブジェクト408を操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を含む。データベースエレメント404は、データベースランタイムリソースの実行に関連するポリシ410を含む。DAC402は、データベースエレメント404のコレクションに関連付けられているDAC名412も含む。DAC名412は、データベースアプリケーション414による使用のためにアクセス可能である。データベースエレメント404のコレクションは、416に示されているように、ユーザ選択可能サポートコンピュータリソース418のセット

10

20

30

40

50

にプログラム的に関連付けられている。

【 0 0 3 5 】

D A C 名 4 1 2 (エンドポイント名) は、ドメインネームサービス (D N S) 名といくらから同じように機能する。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 をホスティングしてい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。特定のコンピュータまたはインスタンスと D A C 4 0 2 との関連付けは、D A C 4 0 2 をコンピュータ及びデータベースインスタンスにマッピングすることによってファブリック内で確立される。D A C 4 0 2 は、アプリケーションがデータベースに接続する接続路に影響を与えずに (例えば、接続ストリングに影響を与えずに)、例えば、D A C 4 0 2 をマイクロソフト S Q L サーバの新しいインスタンスに単純に再マッピングすることによって新しいインスタンスに移動させられ得る。アプリケーションは変更される必要は無い。これらは全て、同一の D A C エンドポイント名に接続され続けている。例えば、ペイロールの D A C エンドポイント名を有している D A C は、コンピュータ 1 のデフォルトのデータベースインスタンスからコンピュータ 2 の他のインスタンスに移動させられ得る。全てのアプリケーションが既知の D A C エンドポイント名ペイロールを参照する故、全てのアプリケーションは同一の D A C 4 0 2 に接続しているままである。

10

【 0 0 3 6 】

特定の実施形態において、ユーザ選択可能サポートコンピュタリソース 4 1 8 のセットは、自動化されたツール (例えば、ソフトウェアアプリケーション、マイクロソフト S Q L サーバマネジメントスタジオ、マイクロソフトビジュアルスタジオ、コマンドライン、パワースhell等のユーザコントロールされているアプリケーション) を使用して変更可能である。他の特定の実施形態において、コレクションに関連付けられている 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。

20

【 0 0 3 7 】

他の特定の実施形態において、D A C 名 4 1 2 は、ユーザ選択可能サポートコンピュタリソース 4 1 8 のセットが変更されても変化しない。

【 0 0 3 8 】

他の特定の実施形態において、D A C 名 4 1 2 は、データベースエレメント 4 0 4 のコレクション内のデータベースエレメントが変更されても変化しない。

【 0 0 3 9 】

他の特定の実施形態において、データベースランタイムリソースはマイクロソフト S Q L サーバのインスタンスであり、D A C 名 4 1 2 は、マイクロソフト S Q L サーバのインスタンスが、第 1 のマイクロソフト S Q L サーバインスタンスから第 2 のマイクロソフト S Q L サーバインスタンスに変更されても変化しない。

30

【 0 0 4 0 】

他の特定の実施形態において、データベースエレメント 4 0 4 のコレクションが変化可能か、またはユーザ選択可能サポートコンピュタリソース 4 1 8 のセットが変化可能であり、D A C エンドポイント名 4 1 2 は変化しないので、データベースエレメント 4 0 4 のコレクションが変更された場合またはユーザ選択可能サポートコンピュタリソース 4 1 8 が変更された場合に、1 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 4 1 4) のアプリケーション接続ストリングは、記録 (例えば、代替、更新、アプリケーションの再コンパイルの必要、アプリケーションバイナリ及び / または実行ファイルの再生成の必要) される必要が無い。

40

【 0 0 4 1 】

他の特定の実施形態において、D A C 名 4 1 2 を使用するデータベースアプリケーションは、異なったネットワークコンピュータに配されている。

【 0 0 4 2 】

他の特定の実施形態において、データベースランタイムリソースの実行に関連しているポリシー 4 1 0 は D A C 名 4 1 2 に適用され、ポリシー 4 1 0 は、データベースエレメント 4 0 4 のコレクションが変更された場合またはユーザ選択可能サポートコンピュタリソー

50

ス 4 1 8 のセットが変更される場合に、適用され続ける。

【 0 0 4 3 】

他の特定の実施形態において、D A C 4 0 2 は、アプリケーションプログラムインタフェース (A P I) をさらに含んで、データベースエレメント 4 0 4 のコレクションとインタフェースをとる。例えば、A P I は、データベースエレメントのコレクションのエレメントの利用の評価が可能である。例えば、図 1 のように、A P I は、開発者がデータベースエレメント 4 0 4 のコレクションとインタフェースをとることを可能にしてもよい。

【 0 0 4 4 】

特定の実施形態において、D A C 4 0 2 は、コンピュータ可読媒体内に保存される。コンピュータ可読媒体は、データベースオブジェクトのメタデータ、データベースオブジェクトを操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、及びデータベースランタイムリソースの実行に関連しているポリシをパッケージングする管理の一単位を含んでもよい。一度説明したが、管理の一単位は、複数のデータベースオブジェクトのメタデータ、データベースオブジェクトを操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、及びデータベースランタイムリソースの実行に関連している複数のポリシをパッケージング可能である。さらに、複数のデータベースオブジェクトは、スキーマ、表、表示、保存されている手順、機能、トリガ、データタイプ、ログイン、ユーザ、許可、ポリシ、統計データ、プライマリーキー、デフォルト、チェック、制約、及び/またはデータベースインスタンスもしくはデータベース内で生成可能な任意の他のオブジェクトを含んでもよい。

【 0 0 4 5 】

管理の一単位は、第 1 の物理コンピューティングリソースにマッピングされてもよい。管理の一単位は、第 1 の物理コンピューティングリソースから第 2 の物理コンピューティングに再マッピングされてもよい。第 1 の物理コンピューティングリソースの例は、プロセッサ、ネットワークデバイスまたはメモリ記憶デバイスである。特定の実施形態において、管理の一単位は、関連付けられた D A C 名によって識別され、D A C 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 4 6 】

データベースオブジェクトを操作可能なデータベースランタイムリソースは、データベースインスタンスであってもよい。データベースインスタンスは、マイクロソフト S Q L サーバデータベースエンジンのインスタンス、マイクロソフト S Q L サーバアナリシスサービスのインスタンス、もしくは S Q L サーバレポーティングサービスのインスタンス、または他のデータベースタイプ及び他のベンダーのインスタンスを含んでもよい。

【 0 0 4 7 】

ポリシは、データベースランタイムリソースの実行に関連していてもよく、リソース制限ポリシ、またはソフトウェアもしくはハードウェアリソース要求に関連している手順を含む。

【 0 0 4 8 】

図 5 を参照すると、システムの特定の実施形態が 5 0 0 で示されている。システム 5 0 0 は、D A C のセット 5 0 2 を含んでいる。データベースによる D A C のホストに関連して、複数の D A C が単一のデータベース内に存在してもよいが、または単一の D A C が複数のデータベースに亘っていてもよい。図 5 に示されている実施形態において、第 1 の D A C 5 0 4 、第 2 の D A C 5 0 6 、及び第 3 の D A C 5 0 8 は、D A C のセット 5 0 2 の中に含まれている。代替的に、D A C のセット 5 0 2 は、多数の D A C を含んでもよい。D A C 5 0 4 、5 0 6 及び 5 0 8 の各々は、データベースエレメント 5 1 0 、5 1 2 及び 5 1 4 のコレクションの論理表示を含む。

【 0 0 4 9 】

システム 5 0 0 は、D A C のセット 5 0 2 をホストするデータベースランタイムリソースのセット 5 1 6 を更に含む。図 5 に示されている実施形態において、データベースランタイムリソースのセット 5 1 6 は、第 1 のデータベースランタイムリソース 5 1 8 、第 2

10

20

30

40

50

のデータベースランタイムリソース 5 2 0、及び第 3 のデータベースランタイムリソース 5 2 2 を含む。代替的に、データベースランタイムリソースのセット 5 1 6 は、多数のデータベースランタイムリソースを含んでもよい。

【 0 0 5 0 】

システム 5 0 0 は、データベースランタイムリソースのセット 5 1 6 によって D A C のセット 5 0 2 をホストするために使用されるコンピューティングリソース 5 2 4 を更に含む。図 5 に示されている実施形態において、コンピューティングリソース 5 2 4 は、第 1 のコンピューティングリソース 5 2 6、第 2 のコンピューティングリソース 5 2 8、及び第 3 のコンピューティングリソース 5 3 0 を含む。代替的に、コンピューティングリソース 5 2 4 は、多数のコンピューティングリソースを含んでもよい。

10

【 0 0 5 1 】

コンピューティングリソース 5 2 4 は、1 または複数のサーバリソース、1 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 または複数の仮想マシンリソースを含んでもよい。1 または複数のコンピュータサーバリソースの各々は、プロセッサ、メモリ、またはネットワークエレメントを含んでもよい。1 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 つは、複数のファイルを含むオペレーティングシステムストレージボリュームであってもよい。1 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 つは、第 1 のストレージエリアネットワーク (S A N) に最初にマッピングされて、その後、第 2 の S A N にマッピングされてもよい。

【 0 0 5 2 】

20

1 つの実施形態において、D A C のセット 5 0 2 のセット内の D A C 5 0 4、5 0 6 及び 5 0 8 の各々は、D A C コンテナを含む。D A C コンテナの各々は、データベースオブジェクトのメタデータ、データベースオブジェクトを操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、及びデータベースランタイムリソースの実行に関連するポリシをパッケージングする管理の一単位を含んでもよい。

【 0 0 5 3 】

D A C のセット 5 0 2 の第 1 の D A C 5 0 4 は、D A C のセット 5 0 2 内の第 2 の D A C 5 0 6 に依存していてもよい。

【 0 0 5 4 】

D A C のセット 5 0 2 内の D A C 5 0 4、5 0 6 及び 5 0 8 の各々は、ファブリック 5 0 1 の D A C 層によって表されていてもよい。コンピューティングリソース 5 2 6、5 2 8 及び 5 3 0 を含むコンピューティングリソース 5 2 4 は、ファブリック 5 0 1 のハードウェアリソース層によって表されていてもよい。データベースランタイムリソースのセット 5 1 6 内のデータベースランタイムリソース 5 1 8、5 2 0 及び 5 2 2 の各々は、ファブリック 5 0 1 の S Q L (Structured Query Language) サーバランタイム層によって表されてもよい。

30

【 0 0 5 5 】

データベースランタイムリソース 5 1 6 のセットは、マイクロソフト S Q L サーバサービスを提供する 1 または複数のランタイムインスタンスを含んでもよい。この 1 または複数のランタイムインスタンスは、マイクロソフト S Q L サーバ、マイクロソフト S Q L サーバアナリシスサービス、及びマイクロソフト S Q L サーバレポーティングサービスの 1 または複数のインスタンスを含んでもよい。さらに、1 または複数のマイクロソフト S Q L サーバランタイムインスタンスは、スキーマ及び表示を含んでもよい。

40

【 0 0 5 6 】

D A C 5 0 4、5 0 6 及び 5 0 8 の各々は、データベースランタイムリソース 5 1 8、5 2 0 及び 5 2 2 の 1 または複数にマッピングされてもよく、データベースランタイムリソース 5 1 8、5 2 0 及び 5 2 2 の各々は、物理コンピューティングリソース 5 2 6、5 2 8 及び 5 3 0 の 1 または複数にマッピングされてもよい。データベースランタイムリソースは、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されてもよい。

【 0 0 5 7 】

50

D A C 5 0 4、5 0 6 及び 5 0 8 の各々は、関連付けられている D A C エンドポイント名を含んでもよい。D A C エンドポイント名は、データベースアプリケーションが使用するためにアクセス可能である。

【 0 0 5 8 】

図 6 を参照すると、D A C のセットを含むファブリックの第 2 の実施形態を示しているブロック図が 6 0 0 で示されている。ファブリック 6 0 8 は、D A C のセット (D A C s)、D A C をホストしているマイクロソフト S Q L サーバランタイム、及び D A C をホストするためにこれらのランタイムによって使用されるリソースのモデルベース表示である。ファブリックモデルは、図 6 に示されているように 3 つの層を有する。すなわち、D A C 層 6 1 0、ランタイム層 6 1 2、及びホストリソース層 6 1 4 である。このモデル内の全てのオブジェクトは、一般的なモデリング概念 (名詞) のセットを使用して記載され、これらのオブジェクトにおいて実行され得る一連の動作 (動詞) を有する。

【 0 0 5 9 】

例えば、動作は、ファイルシステムが急激に大きくなるかまたはリソースの限界を上回ると予測された際に、ファイルシステムの 1 または複数のファイルを自動的に他のオペレーティングシステムボリュームの第 2 のファイルシステムに移動することを含んでもよい。他の例として、動作は、ソフトウェアアプリケーションを実行しているプロセッサが高負荷になった場合に、当該ソフトウェアアプリケーションを比較的低いプロセッサ負荷の他のコンピュータに自動的に移動することを含んでもよい。更に他の例として、動作は、1 または複数の動作を自動的に実行して、評価されたサービスメトリック (service metric) の質 (例えば、約束されたパフォーマンスのサービスの質 (Q o s) レベル) に従ってパフォーマンス (例えば、アプリケーションのパフォーマンス) を維持することを含んでもよい。

【 0 0 6 0 】

D A C 層 6 1 0 は、D A C のファブリック定義をホスティングする。示された実施形態において、D A C 層 6 1 0 は、第 1 の D A C 6 2 2、第 2 の D A C 6 2 4、及び第 3 の D A C 6 2 6 を含む。代替的に、D A C 層 6 1 0 は、多数の D A C を含んでもよい。ランタイム層 6 1 2 は、ファブリック 6 0 8 内のデータベースランタイムリソース (例えば、マイクロソフト S Q L サーバサービス) の表示を提供する。示された実施例において、ランタイム層 6 1 2 は、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、及び第 3 のデータベースランタイムリソース 6 2 0 を含む。代替的に、ランタイム層 6 1 2 は、多数のデータベースランタイムリソースを含んでもよい。ホストリソース層 6 1 4 は、コンピュータリソース 6 3 0 を含み、第 1 のコンピュータ 6 3 6 及びコンピュータサーバ 6 3 8 を含む。さらに、ホストリソース層 6 1 4 は、データリソース 6 3 2 を含み、第 1 のストレージリソース 6 4 0 及び第 2 のストレージリソース 6 4 2 を含む。さらに、ホストリソース層 6 1 4 は、仮想リソースを含み、第 1 の仮想リソース 6 4 4 及び第 2 の仮想リソース 6 4 6 を含む。代替的に、ホストリソース層 6 1 4 は、多数のコンピュータリソース、データリソースまたは仮想リソースを含んでもよい。

【 0 0 6 1 】

ファブリック 6 0 8 は D A C を識別し、データベースランタイムリソースは D A C のセットをホストし、コンピューティングリソースは、D A C 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って使用される。D A C 6 1 0 の各々は、データベースエレメントのコレクションの論理表示を含む。推論ポイント (例えば、中央推論ポイント 6 0 4) は、ファブリック 6 0 8 のエンティティを監視する。1 または複数のポリシー (例えば、中央ポリシー 6 0 6) が使用されて、受信されたデータベース管理動作、及びファブリック 6 0 8 によって識別された影響を受けているエンティティにおけるポリシーが自動的に適用される。

【 0 0 6 2 】

ファブリック 6 0 8 内の層の各々は、4 つのタイプのエンティティを含んでいる。すな

わち、コンテナ、エレメント、リソース及びマッピングである。コンテナは、ファブリック 608 の所定の層内のトップレベルエンティティであり、管理及び同一性 (identity) の直感的な単位を提供する。コンテナの例には、DAC、コンピュータ、仮想マシン、ストレージデバイス、マイクロソフト SQL サーバのインスタンス、マイクロソフト SQL サーバアナリシスサービスのインスタンス、及びマイクロソフト SQL サーバレポートサービスインスタンスが含まれる。エレメントは、コンテナ内に存在するファブリック 608 内の単位である。例えば、表及び保存されている手順は、DAC 内に存在するエレメントである。リソースは、有限の能力を有するエレメントの特別なタイプであり、自身の能力を決定する特性を有する。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、プロセッサデバイスのクロック速度は、その能力に関連する特性の例である。ストレージデバイスは、物理ディスク等の更に粒度の細かいリソースを含むリソースの例である。ストレージデバイスの能力は、ストレージデバイスが含む物理ディスクの特性によってだけでなく、RAID レベル等の設定特性によっても判定される。マッピングは、ファブリック 608 の 2 つの部分間の関係を表す。マッピングは、ファブリック 608 をマイクロソフト SQL サーバ環境の実際のモデルにする名詞である。マッピングは、コンテナ - コンテナ間、コンテナ - エレメント間、コンテナ - リソース間、エレメント - エレメント間、エレメント - リソース間、及びリソース - リソース間であり得る。

10

【0063】

ファブリック 608 内の層 610、612 及び 614 の各々の構成は、上述の全体的なモデリング概念を適用することによって定義され得る。各々の層は、これらのコンテナ内の 1 または複数のコンテナ、エレメント及びリソース、並びにファブリック 608 の隣り合う層内のコンテナもしくはエレメント間のマッピングからなっている。

20

【0064】

DAC 層 610 は、DAC のファブリック定義をホストする。DAC の各々は、DAC コンテナを使用して定義されてもよい。DAC コンテナの各々は、構造、コード、バージョン及びポリシ等のアプリケーションエンティティを表す多数のエレメントによって構成されている。

【0065】

DAC 層 610 におけるマッピングは、DAC とマイクロソフト SQL サーバランタイム (例えば、マイクロソフト SQL サーバのインスタンス) との間の関係を表す。例えば、DAC 層 610 内の DAC は、サーバランタイム層 612 内のマイクロソフト SQL サーバのインスタンスにマッピングされ得る。その後、DAC は、一方のインスタンスから他方のインスタンスまで、インスタンス間の DAC の再マッピングによって移動させられ得る。1 つの例として、第 1 の DAC 622 は、第 1 のデータベースランタイムリソース 616 (例えば、マイクロソフト SQL サーバの第 1 のインスタンス) にマッピングされてもよく、その後、第 1 の DAC 622 は、第 1 の DAC 622 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って、第 2 のデータベースランタイムリソース 618 (例えば、マイクロソフト SQL サーバの第 2 のインスタンス) も移動させられ得る。

30

【0066】

マッピングは、DAC 間の関係も表し得る。DAC は、共通のエレメントの再利用を可能にするために、他の DAC への依存関係を表し得る。例えば、注文の供給のための企業ポリシを実施する保存された手順のセット等のいくつかのアプリケーションによって共有されるエレメントを有する「ベース」DAC が定義され得る。これらのエレメントを必要とする任意の他の DAC は、「ベース」DAC にマッピングされることで当該エレメントを参照する。例えば、第 1 の DAC 622 は、第 2 の DAC 624 に依存している。

40

【0067】

ランタイム層 612 は、サーバファブリック内のマイクロソフト SQL サーバサービスの表示を提供する。マイクロソフト SQL サーバランタイムコンテナは、マイクロソフト

50

S Q Lサーバ、マイクロソフトS Q Lサーバアナリシスサービス及びマイクロソフトS Q Lサーバレポーティングサービスのインスタンスを含む。ランタイムコンテナの各々は、1または複数のデータベースエレメントを有する。データベースエレメントは、スキーマ、表、表示、保存されている手順、機能、データタイプ、インデックス、ログイン、ユーザ、ポリシ、統計データ、プライマリキー、外部キー、デフォルト、チェック制約、及び/または任意の他のオブジェクトであってデータベースインスタンスまたはデータベース内で生成され得るオブジェクトを含んでもよい。

【0068】

ランタイム層612におけるマッピングは、特定のデータベースインスタンスランタイム（例えば、マイクロソフトS Q Lサーバインスタンスのインスタンス）とこれに割り当てられている物理リソースとの間の関係を表す。例えば、ランタイム層612内に存在するデータベースインスタンスは、ハードウェアリソース層614内の物理コンピュータにマッピングされる。データベースインスタンスは、コンピュータ間でのインスタンスの再マッピングによって物理コンピュータ間を移動させられ得る。1つの例として、第1のデータベースリソース616は、第1のサーバ636にマッピングされ得、第1のデータベースリソース616は、第1のサーバ636から第2のサーバ638への第1のデータベースリソース616の再マッピングによって、第2のサーバ638に移動させられ得る。

【0069】

ホストリソース層614は、サーバファブリック608内のエレメントによって利用されるコンピューティング能力及びストレージ能力を示す。ホストリソース層614におけるコンテナは、物理コンピュータ、仮想マシン、及びオペレーティングシステムストレージボリュームを含む。リソースコンテナは、サブエレメントも含み得る。例えば、物理サーバは、プロセッサ、メモリ及びネットワークエレメントを含む。オペレーティングシステムストレージボリュームは、ファイルエレメント及び言及されるアーティファクトの保存に関する他のエレメントを含んでもよい。

【0070】

ホストリソース層614におけるマッピングは、物理リソース間の関係を表す。例えば、ホストリソース層マッピングは、どのS A N論理装置番号（L U N）が各々の物理コンピュータからアクセス可能かを表す。S U N L U Nは、物理コンピュータとS A N L U Nとの間のマッピングを変更することによって、物理コンピュータ間を移動させられ得る。

【0071】

ファブリック608は、ファブリック608全体でエンティティに適用され得る管理動作及びポリシの中央宣言（central declaration）をサポートする。ファブリック608の管理ポイント（例えば、中央管理ポイント602）は、データベース管理動作及びポリシを受信してもよい。ファブリック608は、ファブリックエンティティを監視して管理動作及びポリシを自動的に適用可能な推論エンジンも提供する。ファブリック608の中央管理機構は、図6に示されている。

【0072】

中央管理機構は、ファブリック608内のコンテナの関連する詳細、関係及び制約を記録するモデルを全て参照する。

【0073】

ファブリック608内のマイクロソフトS Q Lサーバインスタンスの1つは、中央管理ポイント602として指名され得、全てのファブリック層に対して中央管理能力（発見、ポリシ定義、及び監視）を提供する。中央管理ポイント602は、全てのコンテナの動作がトリガ（例えば、取り込み、移動、生成、及び配置）がされる場所である。中央管理ポイント602は、ファブリック608によって識別されるデータベースランタイムリソースの1つにおいてホストされてもよい。例えば、中央管理ポイント602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、または第3のデータベースランタイムリソース620においてホストされてもよい。中央管

10

20

30

40

50

理ポイント602は、サーバファブリック層610、612及び614に関する発見、ポリシ定義、及び監視を提供してもよい。

【0074】

中央推論ポイント604は、ファブリック608内のエレメントのデータモデルを提供する。このモデルは、ファブリック608全体のリソース使用傾向及び可能な動作の影響を予測するために解析エンジンによって使用される。中央推論ポイント604は、ファブリック608によって識別されるエンティティにおける中央推論動作を行うために使用されてもよい。中央推論動作は、例えば、他のDACが第1のDACに対する依存関係を有し得る場合に、第1のDACのアンインストールの予測される影響を評価する第1の動作を含んでもよい。中央推論動作は、例えば、1または複数のDACにおけるコンピューティングリソースの1または複数の変更の予測される影響を評価する第2の動作も含んでもよい。中央推論動作は、例えば、配置されるべきDACホストするように構成されているファブリック内のマイクロソフトSQLサーバインスタンスのリストを提供する第3の動作を含んでもよい。従って、ファブリック608が使用されて、推奨される動作及び推奨される動作の予測される影響が判定される。ユーザインタフェースは、推奨される動作の実行の予測される影響に基づいて、ファブリック608の関連するハードウェアリソース及びソフトウェアアプリケーションの予測される利用表示を示してもよい。予測される利用の視点で推奨される動作の実行の予測される影響の表示は、推奨される動作を実行するかを判断することを管理者に可能にさせ得る。管理者が関連するハードウェアリソース及びソフトウェアアプリケーションにおける予測影響が受け入れられないと考えた場合、管理者は、推奨される動作を実行しないことを選択してもよい。管理者が、予測される影響が受け入れられると判断した場合、管理者は推奨される動作を実行することを選択してもよい。

【0075】

中央ポリシ606は、活動または動作を誘導するために定義された権限のある方針であるポリシを含んでもよく、ポリシは、ファブリック608によって識別されたエンティティのセットに適用され得る条件のセットを特定する。例えば、エンティティのセットは、1または複数のマイクロソフトSQLサーバランタイムインスタンスを含んでもよく、ポリシは、マイクロソフトSQLサーバインスタンスのソフトウェアバージョンに基づいて、ファブリック608内への1または複数のマイクロソフトSQLサーバランタイムインスタンスの組み込みを制限してもよい。例として、マイクロソフトSQLサーバ2008のみがランタイム層612内に組み込まれ得る。更なる例として、エンティティのセットは、1または複数のコンピューティングリソースを含んでもよく、ポリシは、オペレーティングシステムのバージョンまたはメモリの量に基づいて、ファブリック608内への1または複数のコンピューティングリソースの組み込みを制限してもよい。例として、Windows 2008の64ビットバージョン及び少なくとも4ギガバイト(4GB)のメモリを含むコンピュータのみが、ホストリソース層614内に組み込まれ得る。説明されたポリシは、ファブリック608内へのランタイムインスタンス及びハードウェアリソースの組み込み制限するポリシの例である。代替的に、ポリシは、ファブリック608内へのDACの取り込みを制限する。例えば、ポリシは、最良の実行に適合しているクエリを有するDACのみをファブリック608内に受け入れてもよい。最良の実行は、プログラマ的に検証可能なルールのセットとしてファブリックモデル内に表されてもよい。最良の実行の例は、データベースクエリ内において「SELECT*」命令文の使用を回避するものである。他の例は、保存されている全ての手順が特定の名前付け慣例に適合することを要求するものでもよい。

【0076】

中央管理ポイント602は、ファブリック動作をトリガしてもよい。例えば、ファブリック動作は、取り込み動作、取り出し動作、配置動作、コピー動作、検証動作、インストール動作、アンインストール動作、追加動作、除去動作、取得動作、設定動作、列挙動作、移動動作、生成動作、保存動作、アップグレード動作、開始動作、停止動作、削除動作

10

20

30

40

50

、再開動作、実行動作、一時停止動作、レジューム動作、無効化動作、有効化動作、評価動作、監視動作、予測動作、管理（govern）動作、マッピング動作、アンマッピング（unmap）動作、発見動作、比較動作、統合動作、遮断動作、遮断解除動作、許可動作、または取り消し動作を含んでもよい。サーバファブリック動作は、最適化動作を含んでもよい。リソース利用度が高すぎる場合に考慮する軽減動作は、データベース設定を最適化することである。データベース最適化例は、インデックスの追加またはデータ圧縮を含む。

【0077】

ライフサイクル動詞は、生成またはファブリック 608 への取り込みの時点から、ファブリック 608 からの除去の時点までのエンティティの管理のための動作のセットを記述する。例えば、ライフサイクル動詞は、所定のエンティティの生成、変更及び削除等の動作を含む。いくつかのライフサイクル動詞は、層 610、612 及び 614 の 1 または複数に適用されてもよい。ライフサイクル動詞は、取り込み動作、取り出し動作、配置動作、コピー動作、検証動作、インストール動作、アンインストール動作、追加動作、除去動作、取得動作、設定動作、列挙動作、移動動作、生成動作、保存動作、及びアップグレード動作を含む。

10

【0078】

取り込み動作は、外部、持続性のデータストアから新しいコンテナ及び関連するエレメントを生成する。取り込み動作は、DAC 層 610 に適用されてもよい。例えば、取り込み動作は、DAC ファイルを DAC パッケージファイルからファブリック 608 に取り込んでもよい。取り出し動作は、持続的外部データストア（例えば、ファイル）をコンテナ及びそれに含まれるエレメントから生成する。取り出し動作は、DAC 層 610 に適用されてもよい。例えば、取り出し動作は、DAC をファイルに取り出す。

20

【0079】

配置動作は、コンテナ及びそのオブジェクトを意図された場所に配置する。配置動作は、DAC 層 610 に適用されてもよい。例えば、配置動作は、第 1 の DAC 622 をランタイム層 612 内のインスタンス（例えば、第 1 のインスタンス 616）に配置してもよい。

【0080】

コピー動作は、一方のコンテナから他方のコンテナにエレメントのセットをコピーする。コピー動作は、DAC 層 610 及びランタイム層 612 に適用してもよい。例えば、コピー動作は、マイクロソフト SQL サーバインスタンスからランタイム層 612 内の他のインスタンス（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 DAC をコピーしてもよい。

30

【0081】

検証動作は、目標のコンテナ、エレメント、リソース、またはマッピングが特定の基準に適合しているかをチェックする。検証動作は、DAC 層 610、サーバランタイム層 612 及びホストリソース層 614 に適用してもよい。例えば、検証動作は、マイクロソフト SQL サーバのインスタンスが、使用可能な CLR 統合を有していることを検証してもよい。

【0082】

インストール動作は、DAC をファブリック 608 内で動作可能に設定する。インストール動作は、DAC 層 610 及びランタイム層 612 に適用してもよい。例えば、インストール動作は、DAC を第 1 のデータベースインスタンス 616 にインストールしてもよい。アンインストール動作は、コンテナ及びそのエレメントを特定の場所から除去する。アンインストール動作は、DAC 層 610 及びランタイム層 612 に適用してもよい。例えば、アンインストール動作は、第 1 のデータベースインスタンスが 616 から DAC をアンインストールしてもよい。

40

【0083】

追加動作は、外部に生成されたコンテナ、エレメント、またはリソースの表示をファブリック内に組み込む。追加動作は、ランタイム層 612 及びホストリソース層 614 に適

50

用してもよい。例えば、追加動作は、第1のマイクロソフトSQLサーバインスタンス616をランタイム層612に追加する。除去動作は、外部に生成されたコンテナ、エレメント、またはリソースの、ファブリック608との関連付けを絶つ。除去動作は、ランタイム層612及びホストリソース層614に適用してもよい。例えば、除去動作は、マイクロソフトSQLサーバインスタンス616をランタイム層612から除去してもよい。

【0084】

取得動作は、コンテナ、エレメント、リソースまたはマッピングの現在の設定を取得する。取得動作は、DAC層610、ランタイム層612及びホストリソース層614に適用してもよい。例えば、取得動作は、第1のマイクロソフトSQLサーバインスタンス616にインストールされているDACのバージョン番号を取得してもよい。

10

【0085】

設定動作は、コンテナ、エレメント、リソースまたはマッピングの現在の設定を更新する。設定動作は、DAC層610、ランタイム層612、及びホストリソース層614に適用してもよい。例えば、設定動作は、第1のマイクロソフトSQLサーバインスタンス616にインストールされているDACのバージョン番号を設定してもよい。

【0086】

列挙動作は、コンテナ内のエレメントまたはリソースのセットを取得する。列挙動作は、DAC層610、ランタイム層612、及びホストリソース層614に適用してもよい。例えば、列挙動作は、第1のサーバ636におけるリソースを列挙する。

【0087】

20

移動動作は、一方のコンテナ、エレメント、またはリソースから他方にエレメントまたはリソースのセットを移動する。移動動作は、DAC層610、ランタイム層612、及びホストリソース層614に適用してもよい。例えば、移動動作は、マイクロソフトSQLサーバの1のインスタンスから他のインスタンスにデータベースを移動してもよい。他の例にとして、移動動作は、データベースインスタンス(SQLサーバインスタンス)を移動してもよい、または一方のコンピュータシステムから他方のコンピュータシステムまでデータベースファイルを移動してもよい。

【0088】

生成動作は、新しいコンテナまたはエレメントを生成する。生成動作は、DAC層610及びランタイム層612に適用してもよい。例えば、生成動作は、DACがマイクロソフトSQLサーバ2008インスタンスにおける配置を要求していることを特定する新しいポリシ制約を生成してもよい。

30

【0089】

保存動作は、コンテナ、エレメント、リソース、またはマッピングの現在の設定を持続させる。保存動作は、DAC層610、ランタイム層612、及びホストリソース層614に適用してもよい。例えば、保存動作は、マイクロソフトSQLサーバインスタンス616にインストールされているDACのバージョン番号を保存してもよい。

【0090】

アップグレード動作は、コンテナまたはエレメント現在の設定を新しい定義に変更する。アップグレード動作は、DAC層610及びランタイム層612に適用してもよい。例えば、アップグレード動作は、第1のマイクロソフトSQLサーバインスタンス616にインストールされているオーダー表(order table)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してもよい。

40

【0091】

ファブリック608は、ポリシに従って、監視動作、変更の調整、及び検証等の動作を実行するさまざまなプロセスを有する。動詞のセットは、プロセスの管理のために使用されてもよい。残りの動詞の全ては、ファブリック608の全て層(例えば、DAC層610、ランタイム層612、及びホストリソース層614)に適用される。プロセス制御動詞は、開始動作、停止動作、再開動作、実行動作、一時停止動作、レジューム動作、無効化動作、及び有効化動作を含む。

50

【 0 0 9 2 】

開始動作は、停止されるまで実行し続けるプロセスを開始する。例えば、開始動作は、第1のマイクロソフトSQLサーバインスタンス616における調整を開始する。停止動作は、プロセスの実行を中止する。例えば、停止動作は、第1のマイクロソフトSQLサーバインスタンス616における調整を停止してもよい。再開動作は、プロセスの実行を停止させて、その後、プロセスの実行を再度開始する。例えば、再開動作は、第1のマイクロソフトSQLサーバインスタンス616における調整を再開してもよい。実行動作は、実行されて、完了すると自動的に停止するプロセスを開始する。例えば、実行動作は、第1のマイクロソフトSQLサーバインスタンス616における発見を実行してもよい。一時停止動作は、プロセスを一時停止し、プロセスは、レジュームされるまで一時停止したままである。例えば、一時停止動作は、第1のマイクロソフトSQLサーバインスタンス616における発見を一時停止してもよい。レジューム動作は、停止されたプロセスを継続する。例えば、レジューム動作は、第1のマイクロソフトSQLサーバインスタンス616における発見をレジュームする。無効化動作は、利用できないので開始することができないプロセスを設定する。このプロセスは、有効化されるまで無効化されたままである。例えば、無効化動作は、マイクロソフトSQLサーバインスタンス616における調整を無効にしてもよい。有効化動作は、無効化されているプロセスを開始可能なように設定する。例えば、有効化動作は、各々、第1のマイクロソフトSQLサーバインスタンス616における調整を可能にしてもよい。

10

【 0 0 9 3 】

パフォーマンス管理動詞は、ファブリックリソースの現在の利用に関してファブリック608の現在または予測される健全度を解析するために使用される。例えば、クエリ等のDAC動作の評価された待ち時間統計データが監視されて、現在の健全度状態が判定されかつ将来の健全度状態が予測されてもよい。これらの動詞は、ファブリック608内の様々なコンテナ及びエレメントによるリソースの使用を管理するための手段も提供する。パフォーマンス管理動詞は、評価動作、監視動作、予測動作、及び管理動作を含んでもよい。

20

【 0 0 9 4 】

評価動作は、リソース、エレメント、またはコンテナに関するパフォーマンス統計データを収集する。例えば、評価動作は、第1のサーバ636のボリュームEにあるデータベースファイルFinance.mdbによって使用されるスペースを評価してもよい。監視動作は、リソース、エレメント、またはコンテナに関連するパフォーマンス統計データをポリシーに関して評価する。例えば、監視動作は、使用が能力の80%を超えている場合の事象に対し、第1のサーバ636のボリュームEにあるデータベースファイルFinance.mdbによって使用されるスペースを監視してもよい。予測動作は、履歴データ、リソース変更、または配置もしくは移動等のファブリック動作に応じた、コンテナ、エレメントまたはリソースのパフォーマンス統計データを予測する。例えば、予測動作は、財務(Finance)DACの増大に対応するために追加のストレージが必要とされるのがいつかを予測してもよい。さらなる例として、予測動作は、ファブリック全体の増大に対応するために追加のリソースが必要とされるのがいつかを予測する。管理動作は、コンテナまたはエレメントに割り当てられるリソースの最小または最大量を特定する。例えば、管理動作は、第1のSQLサーバインスタンス616のCPU使用率を第2のサーバ638のCPU能力の50%に制限してもよい。リソース管理は、コンテナ及び/またはエレメントのコレクションに適用されてもよい。

30

40

【 0 0 9 5 】

マッピング動詞は、ファブリック608内の層間の関係を管理する単純な手段を提供する。マッピング動詞は、2つの隣接する層の間の関係を構築するために主に使用される。マッピング動詞は、マッピング動作及びアンマッピング動作を含んでもよい。概念的には、これらの動詞は、一連の実行されている他のライフサイクル動作を生む、ファブリック608に対して行われ得るトップレベル動作であるとも考えられる。例えば、マイクロソ

50

フトSQLサーバのインスタンスにDACをマッピングすることは、システムに、DACがそのインスタンスにおいて利用可能であるべきことを伝える。これは、特定のインスタンスにおいてこのDACを利用可能にするために必要とされる、配置またはインストール等の動詞の全てを呼び出す。

【0096】

マッピング動作は、コンテナ、エレメントまたはリソースの間の関係を生成する。例えば、マッピング動作は、第1のDAC622を第1のマイクロソフトSQLサーバインスタンス616に加えてもよい。アンマッピング動作は、コンテナ、エレメントまたはリソースの間のマッピングを除去する。例えば、アンマッピング動作は、第1のマイクロソフトSQLサーバインスタンス616から第1のDAC622を切り離す。

10

【0097】

調整は、規定されている定義からずれてしまったファブリック内のエンティティを識別するプロセスである。例えば、調整は、プロダクションサーバ内の表の定義が、DACにおいてどのように定義されるかから変更されているかを検知する。3つのサーバファブリック動詞は、調整を実装する。調整動詞は、発見動作、比較動作、及び統合動作を含んでもよい。

【0098】

発見動作は、新しいアイテムが発見されかつ新しいアイテムに生産的な理解 (productive insight) が与えられる操作を定義する。例えば、発見動作は、第1のマイクロソフトSQLサーバインスタンス616内の従業員表 (employee table) 内で新しい列を発見してもよい。比較動作は、コンテナ、エレメントまたはリソースの現在の設定と、予め規定されている設定とをマッチングさせて、差異を報告する。例えば、比較動作は、第1のマイクロソフトSQLサーバインスタンス616にある従業員表をDAC内の表定義と比較して、第1のマイクロソフトSQLサーバインスタンス616内の表のバージョンがDAC内で定義されていない列を有していることを発見してもよい。統合動作は、複数のインスタンスからのコンテナ、エレメントまたはリソースの単一の設定を生成する。例えば、統合動作は、従業員表内の新しい列の定義をDAC定義内に統合してもよい。

20

【0099】

アクセス動詞は、ファブリックモデルの一部へのアクセスを制御して、長期間のプロセスをサポートする手段を提供する。例えば、ファイルを一方のボリュームから他方のボリュームに移動するプロセスは、宛先ボリュームへのアクセスを制御して、複数のプロセスがスペース割り当ての競合を生成しないようにしなければならない。セキュリティ動詞は、ファブリック608内のコンテナ及びエレメントへのアクセスを承認されたユーザに限定する。本開示は、新しい承認の基礎構造を提供するのではなく、これらの動詞は、ファブリック608に亘る承認設定の統一されたインタフェースを表す。アクセス及びセキュリティ動詞は、遮断動作、遮断解除動作、許可動作、及び取り消し動作を含んでもよい。

30

【0100】

遮断動作は、コンテナ、エレメント、またはリソースの設定に発行プロセス排他的アクセスを提供し、他のプロセスが設定にアクセスするのを防止する。この動詞は、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おいて動作する。例えば、遮断動作は、ファイルの移動を実行するためのスペースが確保されるように、第1のサーバ636のボリュームEへのアクセスを遮断してもよい。遮断解除動作は、コンテナ、エレメントおまたはリソースへの排他的アクセスを解除する。この動詞は、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おいて動作する。例えば、ファイルの移動が完了した後、遮断解除動作は、第1のサーバ636のストレージボリュームEへのアクセスの遮断を解除してもよい。許可動作は、コンテナ、エレメントまたはリソースへのアクセスのためにプリンシパル許可を与える。例えば、許可動作は、プリンシパルであるジャネットのDACへの書き込みアクセスを許可してもよい。取り消し動作は、コンテナまたはエレメントへのアクセスのためのプリンシパル許可を拒否する。例えば、取り消し動作は、プリンシパルであるジャネットからDACへの書

40

50

き込みアクセスを取り消してもよい。

【 0 1 0 1 】

図 7 を参照すると、データベース管理動作及びポリシを自動的に適用する方法の特定の実施例が示されている。7 0 2 において、この方法は、ファブリックの中央管理ポイント（例えば、図 6 内のサーバファブリック 6 0 8 の中央管理ポイント 6 0 2 ）においてファブリックポリシを受信するステップを含む。ファブリックは、D A C、D A C のセットをホストするデータベースランタイムリソース、及び D A C 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って使用されるコンピューティングリソースを識別する。D A C の各々は、データベースのコレクションの論理表示を含む。7 0 4 に移動すると、この方法は、中央管理ポイント（例えば、図 6 内のサーバファブリック 6 0 8 の中央管理ポイント 6 0 2 ）においてファブリックのエンティティを監視するステップも含む。7 0 6 に移動すると、この方法は、ファブリックエレメントをファブリックポリシ（例えば、図 6 内の中央ポリシ 6 0 6 を使用）に準拠させるためにファブリックによって識別された影響をうけるエンティティにおいて、ファブリックポリシの 1 または複数の動作を自動的に適用するステップも含む。

10

【 0 1 0 2 】

図 8 を参照すると、エンティティ関係図（E R D）が、データ層のコンポーネントを表しているモデルをファブリックがどのように表しているか、及びこれらがどのように関連しているかを示している。図 8 のエンティティ関係図（E R D）は、ファブリックの 3 つの層の構造及びセマンティクス（semantics）をサポートするために、このモデルがどのような形態をとっているかを示している。

20

【 0 1 0 3 】

各々の層は、コンテナ、エレメント及びリソースと同様にモデル内の形式的なエンティティとして示されている。モデルは、ファブリックの定義された構造をもたらししているエンティティ間の関係セマンティクスも定義する。この構造は、ファブリックに亘って判断を行ってファブリックにおいて動作を行うことを本開示の実施形態に可能にすることによって、次のセクションにおいて説明される機構を容易にする。

【 0 1 0 4 】

図 9 を参照すると、システムの特定の例示的实施形態が、9 0 0 で示されている。中央管理サーバ（C M S）9 0 2 は、ファブリックに中央管理ポイントを提供する。C M S 9 0 2 は、中央管理リポジトリ（C M R）9 0 4 に接続している。ファブリック内のデータベースインスタンスの管理されているインスタンス（例えば、マイクロソフト S Q L サーバ）は、C M S 9 0 2 に接続しており、C M R 9 0 4 内に情報をアップロードする。C M S 9 0 2 は、ネットワーク上の S Q L サービスインスタンスを発見し、D A C を用意し、所望の場所に D A C エレメントを配置し、基準に対して D A C をチェックし、かつファブリック内のエレメント間の差異を調整するための動作を提供する、

30

【 0 1 0 5 】

C M R 9 0 4 は、ファブリックのモデル（例えば、関連する詳細、関係及び制約）を含む。C M R 9 0 2 は、配置分析、影響分析、及び what-if 分析のための中央推論ポイントであり、コアデータタイプ、詳細データタイプ、及びリンクデータタイプを含む。2 つのタイプは、I M R（例えば、I M R 9 1 0 及び 9 1 4）から C M R 9 0 4 への応答の頻度に基づいている分類を用いて、C M R 内に含まれているデータを表示する。高頻度応答データは、コアデータと称される。それよりも応答頻度の低いデータは、詳細データと称される。リンクデータタイプは、インデックスまたはデータへのポインタとして提供され、これらは C M R 9 0 4 内に保存されない。

40

【 0 1 0 6 】

中央管理データウェアハウス（C M D W）9 0 6 は、物理コンピュータ及びインスタンスのセットに亘る履歴パフォーマンス統計データの集中報告を可能にする。管理されている S Q L サーバランタイムインスタンスの各々にあるデータは、監視及び報告のために C M D W 9 0 6 データベースの場所に収集されかつアップロードされる。

50

【 0 1 0 7 】

中央ファブリックサービス (C F S) 9 0 8 は、コア C M S 9 0 2 動作に関する A P I を提供する。これらの動作は、ネットワーク上の S Q L サービスインスタンスの発見、D A C の用意、所望の場所への D A C エLEMENT の配置、基準に照らした D A C のチェック、及びファブリック内のELEMENT間の差異の調整を含む。

【 0 1 0 8 】

インスタンス管理リポジトリ (I M R) は、データベースインスタンスの各々内に存在するデータベースである。例えば、第 1 の I M R 9 1 0 は、第 1 のマイクロソフト S Q L サーバインスタンス 9 1 2 内に存在する。第 2 の I M R 9 1 4 は、第 2 のマイクロソフト S Q L サーバインスタンス 9 1 6 内に存在する。I M R は、ファブリックの全ての配置または変更の最初の目標である。「コレクションセット」として知られる機構は、I M R 情報をコピーし、それを C M R 9 0 4 内にアップロードしてファブリックの総合レベル (aggregate-level) の表示を提供する。

10

【 0 1 0 9 】

インスタンス管理データウェアハウス (I M D W) は、データベースインスタンスの各々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 I M D W 9 1 8 は、第 1 のマイクロソフト S Q L サーバインスタンス 9 1 2 内に存在する。第 2 の I M D W 9 2 0 は、第 2 のマイクロソフト S Q L サーバインスタンス 9 1 6 内に存在する。I M D W は、インスタンス及びホストコンピュータに関するパフォーマンスデータ (例えば、報告及び/または監視) 及び設定データの最初の目標であり、これらは、サーバファブリックの総合表示を提供するために、後に C M D W 9 0 6 内にアップロードされる。

20

【 0 1 1 0 】

インスタンスファブリックサーバ (I F S) は、コア管理動作のための A P I を提供する。例えば、第 1 の I F S 9 2 2 は、第 1 のマイクロソフト S Q L サーバインスタンス 9 1 2 内に存在する。第 2 の I F S 9 2 4 は、第 2 のマイクロソフト S Q L サーバインスタンス 9 1 6 内に存在する。これらの動作は、ローカルインスタンスにおける D A C インスタンスの有効化及びインストール、インストール動向 (install drift) 発見、及び調整操作の提供を含む。I F S は、インスタンス及び D A C のパフォーマンスを評価して、C M S 9 0 2 から送信される動作を実行する。

【 0 1 1 1 】

図 1 0 を参照すると、エンド間管理プロセスの例示が参照符号 1 0 0 0 で示されている。本開示は、開発から工程への配置までのソフトウェアライフサイクル全体に亘るデータベースアプリケーションの管理のためのツール及び基礎構造を提供する。このセクションにおいて、本開示のいくつかの標準のワークフローは、コアデザイン法則の理解をもたらすために、以前のセクションにおいて説明された重要なアーキテクチャのコンポーネントに亘って実行される。

30

【 0 1 1 2 】

1 0 0 2 において、システムがインストールされてファブリックが設定される。1 0 0 4 において、D A C が現在の配置から生成されて、ファブリック内に取り込まれる。1 0 0 6 において、ファブリックの使用が監視されて予測される。この監視及び予測に基づいて、ファブリックリソースが調整されてもよい。例えば、1 0 0 8 において、ファブリックリソースが管理される。1 0 1 0 において、ファブリックELEMENTが移動させられる。

40

【 0 1 1 3 】

図 1 1 を参照すると、インストールプロセスの例が示されている。インストールパッケージは、3つのコンポーネントからなっている。すなわち、ツール拡張部、ファブリックサービス、及び管理リポジトリスキーマである。ツール拡張部は、マイクロソフトビジュアルスタジオまたはマイクロソフト S Q L サーバマネジメントスタジオに適用され、追加のツールに関しても拡張され得る。ファブリックサービス及び管理リポジトリスキーマは、中央管理サーバ及び管理されているインスタンス内にインストールされる。

50

【 0 1 1 4 】

1 1 0 2において、インストールパッケージが取得される。1 1 0 4において、ツール拡張部（例えば、マイクロソフトビジュアルスタジオ及びマイクロソフトSQLサーバマネジメントスタジオに関する）がインストールされる。1 1 0 6において、中央管理サーバ（CMS）及び中央ファブリックサービス（CFS）がインストールされて設定される。1 1 0 8において、中央管理リポジトリ（CMR）及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。1 1 1 0において、ランタイムインスタンスの全てが発見され、ファブリック内で管理されたインスタンスになるインスタンスが発見される。1 1 1 2において、ファブリックインストールジョブが管理されているインスタンスの各々の中に生成される。1 1 1 4において、インスタンス管理リポジトリ（IMR）及びインスタンス管理データウェアハウス（IMDW）スキーマが管理されているインスタンスの各々の中にインストールされる。1 1 1 6において、ファブリックIMDRからCMDWへのデータアップロード機構がインストールされ、これらは1 1 1 8で開始される。

10

【 0 1 1 5 】

図12を参照すると、現在の配置からDACを生成して、それをファブリック内に取り込むことの例が示されている。DACは、ファブリック内の管理のコア論理単位である。ファブリックはDACを管理して、最適なリソース利用を保証する。DACは、新しいDACプロジェクトを生成することによって、マイクロソフトビジュアルスタジオまたはマイクロソフトSQLサーバマネジメントスタジオの中に加えられる。さらに、新しいDACは、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って、以前のデータベース配置を解析して模倣する。

20

【 0 1 1 6 】

1 2 0 2において、DACが生成される。1 2 0 4において、インスタンスオブジェクトがDACに追加される。1 2 0 6において、インスタンス設定及びポリシがDACに追加される。1 2 0 8において、配置設定及びポリシがDACに追加される。1 2 1 0において、DAC設定及びポリシがIMRに対して有効化される。1 2 1 2において、DACがIMRに保存される。1 2 1 4において、DACがIMRからCMRにコピーされる。1 2 1 6において、任意的に、ビジュアルスタジオにおける後の開発用途のためにDACパッケージバイナリファイルが生成される。

30

【 0 1 1 7 】

図13を参照すると、サーバファブリックの使用の監視及び予測の例が示されている。リソース利用ポリシが、サーバファブリック及びDACに亘って設定されて評価され得、リソースの過剰使用及び不十分な使用が回避される。

【 0 1 1 8 】

1 3 0 2において、全体リソース監視ポリシがCMRにおいて設定される。1 3 0 4において、統計データがIMDWに追加される。1 3 0 6において、データベースリソース監視ポリシがIMRにおいて設定される。1 3 0 8において、リソース使用統計データが、IMDWにおいて監視される。代替的に、他のパフォーマンス健全性統計データ（例えば、クエリ等のDAC動作の待ち時間統計データ）が監視されかつ予測される。1 3 1 0において、IMDWリソース使用統計データがCMDWに出力される。1 3 1 2において、現在のリソース使用が、CMDWにおいて監視される。1 3 1 4において、予測されるリソース使用がCMDWにおいて監視される。

40

【 0 1 1 9 】

図14を参照すると、サーバファブリックリソースの調整の例が示されている。リソース割り当てポリシは、ファブリック及びDACに亘って設定されて評価され得、DACが利用可能なリソースがバランスされる。リソースの調整は、インプレースのリソース管理、及びボリューム間のファイルの移動もしくはサーバ間のデータベースアプリケーションの移動のようなファブリックエレメントの移動等の動作を介して達成され得る。インプレースのリソース管理の例には、1または複数のリソース管理ルールに基づいた、CPU、

50

メモリまたはストレージスペースリソースの、データベースインスタンスまたはDACへの再割り当てが含まれる。ファブリックリソースは、新しいリソース管理ポリシーを実装することによって調整され得る。

【0120】

1402において、リソースバランスオプションが、目標DACに関連するDACに関して予測される。1404において、新しいリソースポリシーがCMRにおいて設定される。1406において、新しいリソースポリシーがIMRの各々にコピーされる。1408において、リソース管理ルールがIMRの各々に実装される。

【0121】

図15を参照すると、ファブリックリソースの調整の他の例が示されている。図15において、一方のファイルから他方のファイルまでの移動のプロセスが示されている。

10

【0122】

1502において、DACがCMRから取得される。1504において、潜在的な目標ボリュームが予測される。1506において、DACが他のボリューム（ボリューム2）にマッピングされる。1508において、ADCがCMRからIMRにコピーされる。1510において、DACがIMRにおいて有効化される。1512において、DACがインストールされる（データベース、コピーファイル等を変更する）。1514において、DACがCMRにおいて有効化される。

【0123】

図16を参照すると、ファブリックリソースの調整の他の例が示されている。図16において、一方のマイクロソフトSQLサーバインスタンスから他方のマイクロソフトSQLサーバインスタンスにDACを移動するプロセスが示されている。

20

【0124】

1602において、DACがCMRから取得される。1604において、潜在的な目標インスタンスが予測される。1606において、DACがランタイムインスタンス2にマッピングされる。1608において、DACがCMRからインスタンス2IMRにコピーされる。1610において、DACがインスタンス2IMRにおいて有効化される。1612において、DACがインスタンス2IMRにインストールされる。1614において、DACエンドポイント名がCMRにおいて移動される。1616において、CMRにおいてDACが有効化される。

30

【0125】

本明細書に記載されている例示の実施形態は、様々な実施形態の構造の全般的な理解をもたらすことを目的としている。これらの例は、本明細書に記載されている構造又は方法を使用する装置及びシステムの全ての要素及び特徴の完全な説明として提供することは意図していない。多くの他の実施形態が、本開示を読んだ当業者に明らかである。他の実施形態が、本開示から使用され導き出されてもよく、構造的かつ論理的置換及び変更が、本開示の範囲から逸脱することなくなされ得る。従って、本開示及び図は、限定的ではなく例示的なものと見なされるべきである。

【0126】

当業者は、本明細書に開示されている実施形態に関連して説明されている様々な例示的論理ブロック、構成、モジュール、回路、及びアルゴリズムステップが、電子ハードウェア、コンピュータソフトウェア、またはこれらの組み合わせとして実装されてもよいことをさらに理解するだろう。ハードウェア及びソフトウェアのこの置換可能性を明確に説明するために、様々な例示のコンポーネント、ブロック、構成、モジュール、回路またはステップが、これらの機能に関して一般的に説明されている。これらの機能がハードウェアまたはソフトウェアのどちらとして実装されるかは、システム全体に与えられている特定の用途及びデザイン制約に依存する。当業者は、特定の用途に対して様々な方法でこの説明されてきた機能を実装してもよいが、このような実装判断は、本開示の範囲からの逸脱をもたらすと解釈されるべきではない。

40

【0127】

50

本明細書において開示されている実施形態に関連して説明されている方法のステップは、ハードウェア内、プロセッサによって実行されるソフトウェアモジュール内、またはこれら2つの組み合わせ内に直接的に具体化されていてもよい。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、リムーバブルディスク、CD-ROM、または当技術分野で知られている任意の他の記憶媒体等の、コンピュータ可読媒体内に存在してもよい。例示の記憶媒体はプロセッサと組み合わせられ、プロセッサは記憶媒体から情報を読み込み、情報を記憶媒体に書き込むことが可能である。代替例において、記憶媒体はプロセッサに統合されているか、またはプロセッサ及び記憶媒体は、コンピューティングデバイスまたはコンピュータシステム内に個別的なコンポーネントとして存在していてもよい。

10

【0128】

特定の実施形態が本明細書内で説明及び記載されてきたが、同一または同様の目的を達成するためにデザインされる後日の任意の構成が、示されている特定の実施形態に置き換えられ得ることが理解されるべきである。この開示は、様々な実施形態の任意の及び全ての後日の適合又は変更を包含することを目的としている。

【0129】

本開示の要約は、本願の特許請求の範囲の請求項の範囲のまたは意味を解釈または限定するために使用されないという理解と共にもたらされている。さらに、上述の発明を実施するための形態において、様々な特徴が、開示の簡素化のためにグループ化されるかまたは単一の実施形態において説明されていてもよい。本開示は、特許請求の範囲に記載されている実施形態が、これらの特許請求の範囲に明示的に列挙されているよりも多くの特徴を要求する意図を反映しているとして解釈されるべきではない。むしろ、特許請求の範囲を検討する場合、本発明は、いくつかの開示された実施形態の特徴の全てよりも少ない特徴を対称にしているともよい。

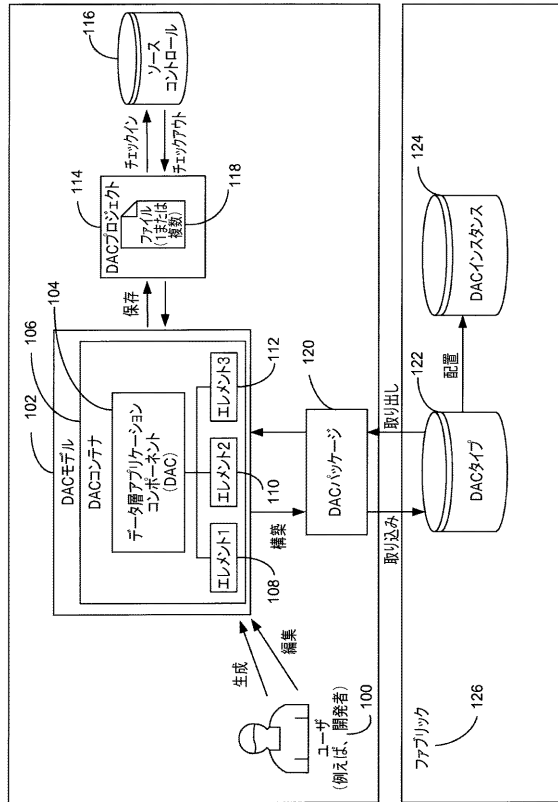
20

【0130】

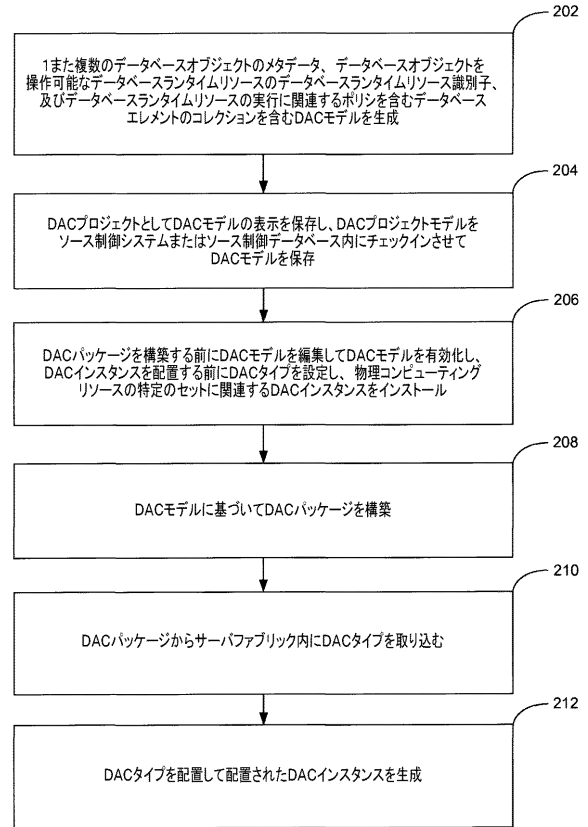
開示された実施形態の上記説明は、本技術分野の当業者が、開示された実施形態を構成または使用することを可能にするために提供されている。これらの実施形態への様々な変更が本技術分野の当業者に容易に理解され、本明細書内に定義されている一般的な原理は、本開示の範囲を逸脱することなく他の実施形態に適用されてもよい。従って、本開示は、本明細書に示されている実施形態に限定されるものではなく、本開示には、添付の特許請求の範囲によって定義される原理及び新規な特徴と一致する可能な限り最も広い範囲が認められる。

30

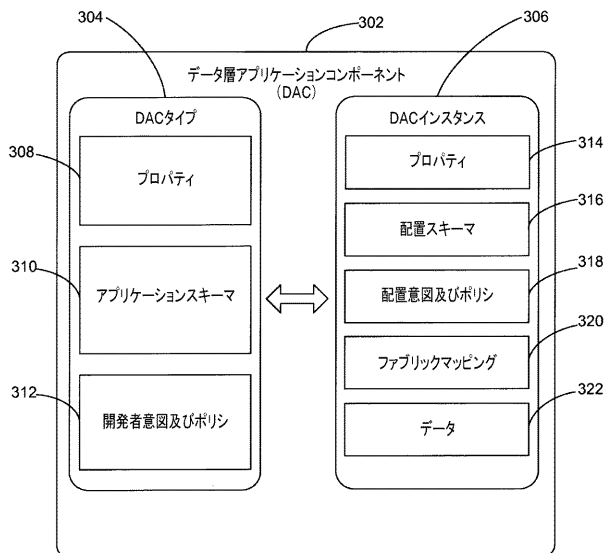
【図 1】



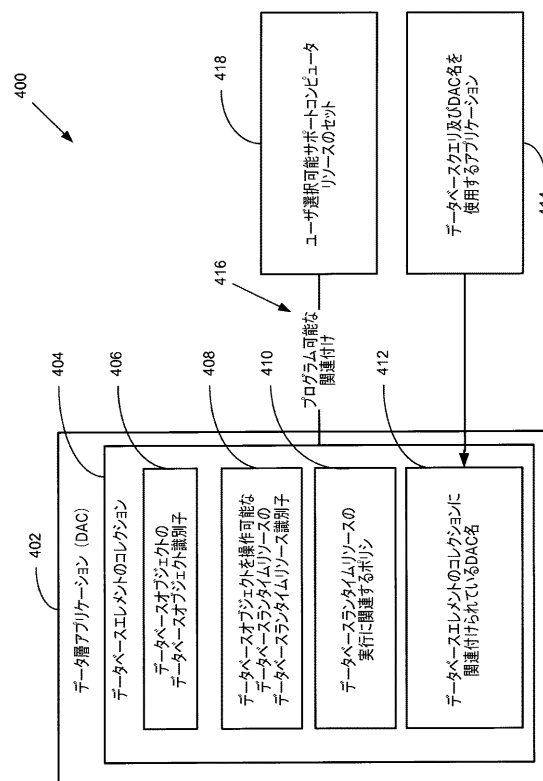
【図 2】



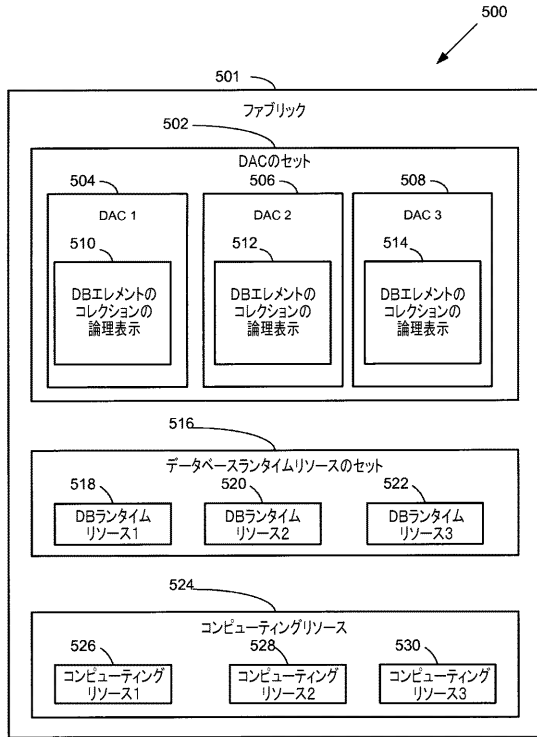
【図 3】



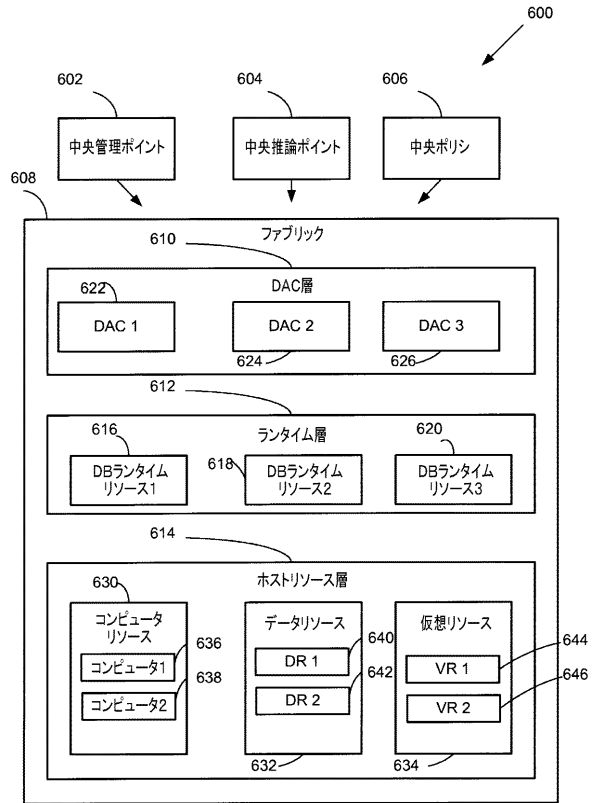
【図 4】



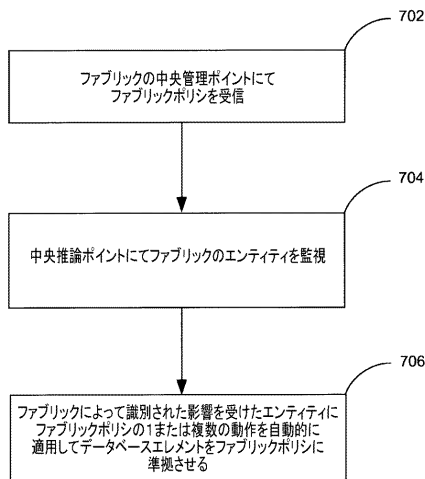
【図 5】



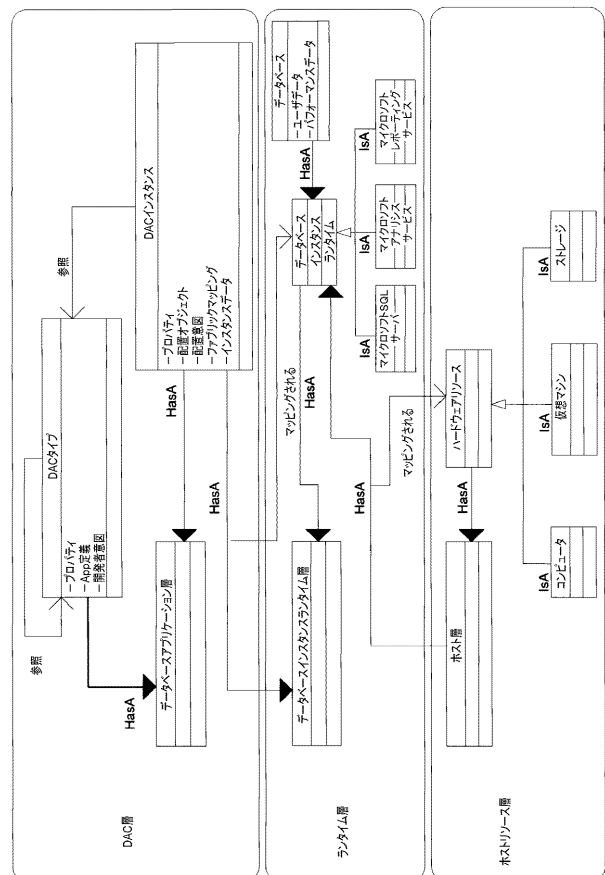
【図 6】



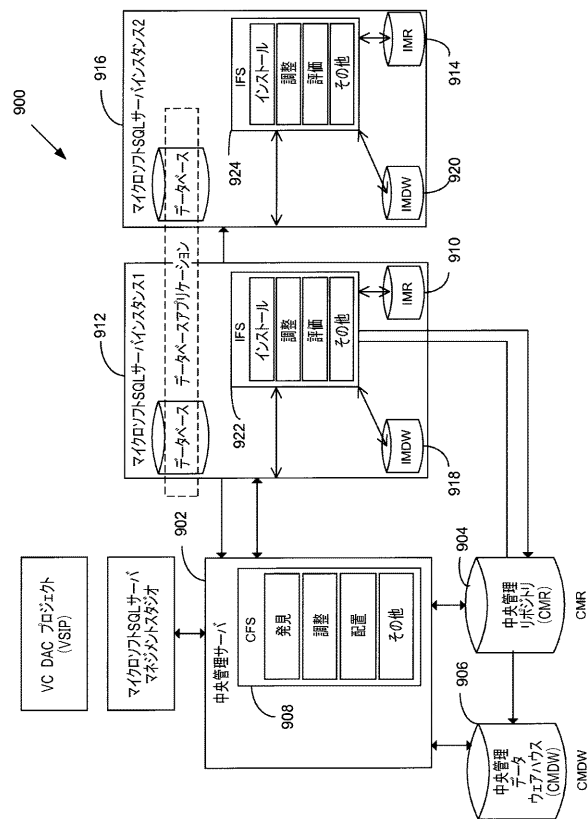
【図 7】



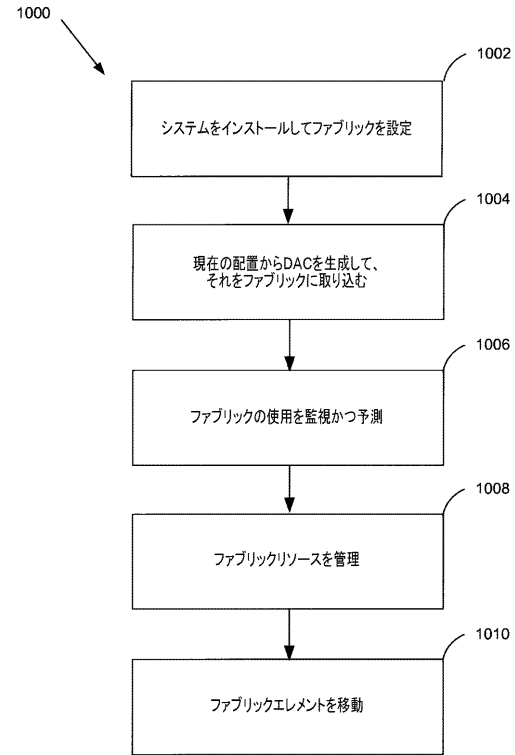
【図 8】



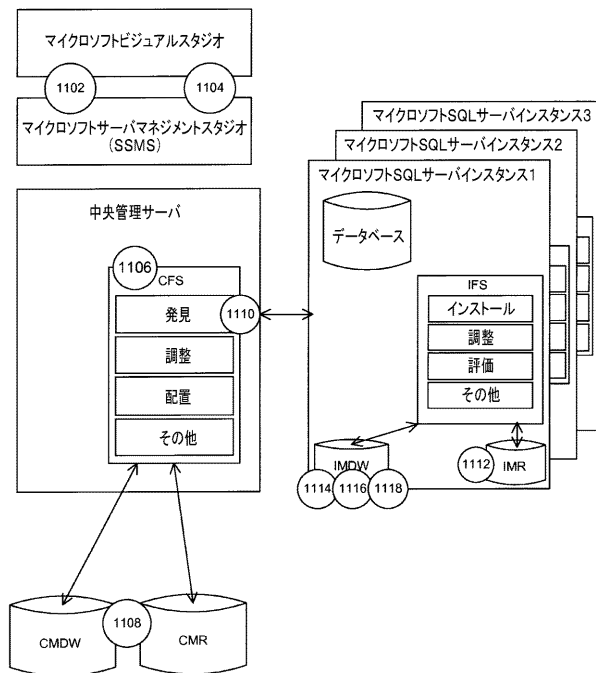
【図 9】



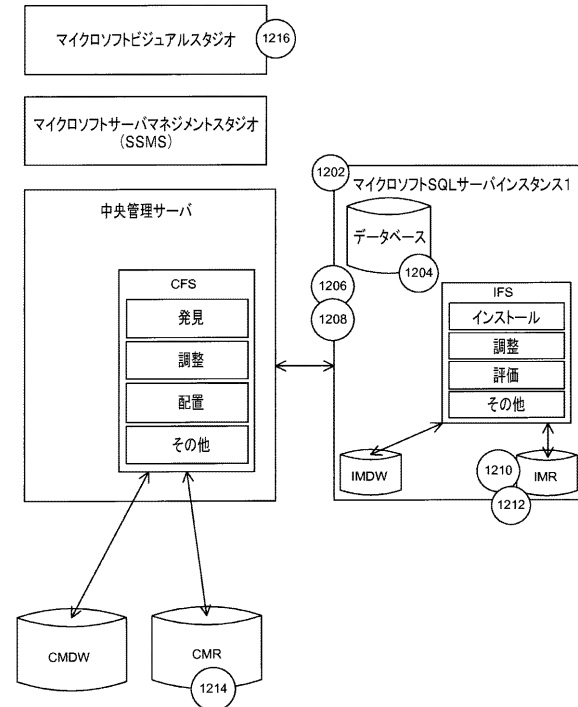
【図 10】



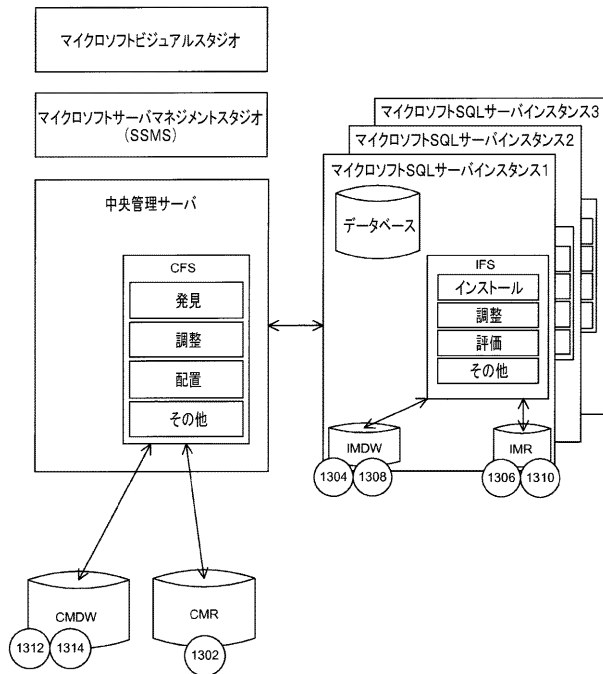
【図 11】



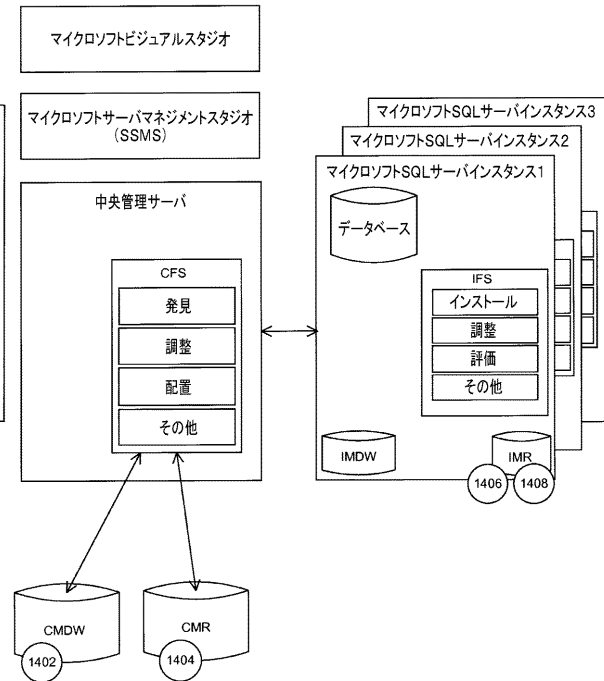
【図 12】



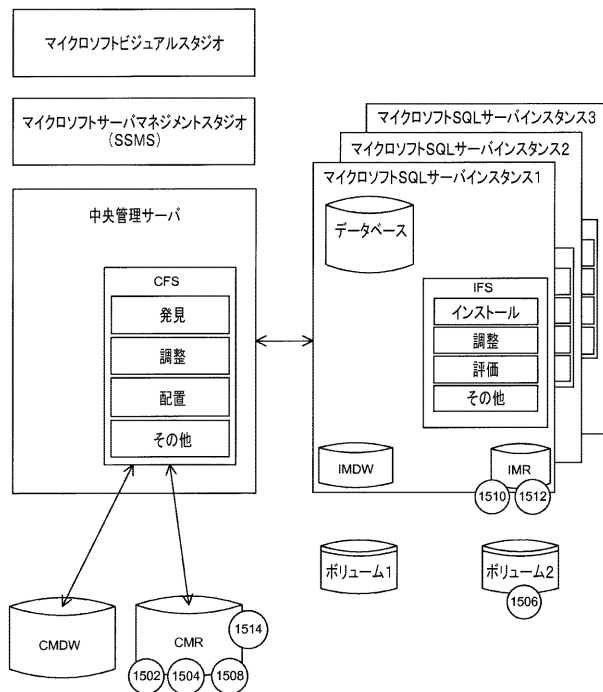
【図 13】



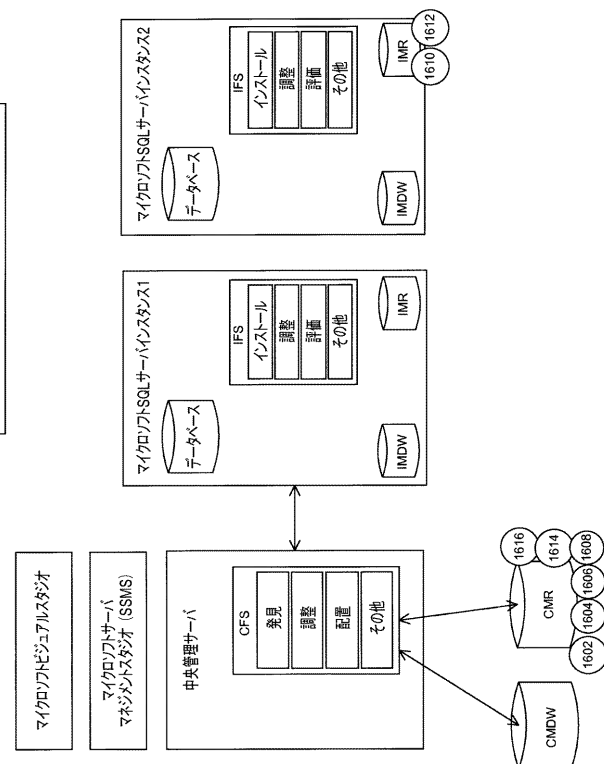
【図 14】



【図 15】



【図 16】



フロントページの続き

- (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 ジョン エム・オズレイク
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 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 ジン フェン
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 シシャードリー
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 井上 宏一

- (56)参考文献 特開2005 - 32242 (JP, A)
特開2008 - 40645 (JP, A)
特開2004 - 234114 (JP, A)

- (58)調査した分野(Int.Cl., DB名)
G06F 9/46 - 9/54