

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5591938号
(P5591938)

(45) 発行日 平成26年9月17日(2014.9.17)

(24) 登録日 平成26年8月8日(2014.8.8)

(51) Int.Cl.

F I

G 0 6 F 12/00 (2006.01)

G 0 6 F 12/00 5 1 3 D

G 0 6 F 12/00 5 1 7

請求項の数 57 (全 48 頁)

(21) 出願番号 特願2012-535259 (P2012-535259)
 (86) (22) 出願日 平成22年10月15日 (2010.10.15)
 (65) 公表番号 特表2013-508857 (P2013-508857A)
 (43) 公表日 平成25年3月7日 (2013.3.7)
 (86) 国際出願番号 PCT/US2010/052960
 (87) 国際公開番号 W02011/049839
 (87) 国際公開日 平成23年4月28日 (2011.4.28)
 審査請求日 平成25年10月15日 (2013.10.15)
 (31) 優先権主張番号 12/603,541
 (32) 優先日 平成21年10月21日 (2009.10.21)
 (33) 優先権主張国 米国 (US)

早期審査対象出願

(73) 特許権者 512105152
 デルフィクス コーポレーション
 アメリカ合衆国 カリフォルニア 940
 25, メンロー パーク, ミドルフィールド
 ド ロード 275, スイート 50
 (74) 代理人 100077539
 弁理士 飯塚 義仁
 (74) 代理人 100114742
 弁理士 林 秀男
 (74) 代理人 100125265
 弁理士 貝塚 亮平

最終頁に続く

(54) 【発明の名称】 仮想データベースシステム

(57) 【特許請求の範囲】

【請求項 1】

仮想データベースシステムを作成する方法であって、
 複数のデータベースブロックを備えているソースデータベースの異なる時点のコピーで
 ある種々の時点コピーを受け取ることと、

前記ソースデータベースの複数の種々の時点コピーについての複数のデータベースブ
 ロックをストレージシステムに記憶することであって、該記憶した複数のデータベースブ
 ロックの少なくとも一部が前記ソースデータベースの複数の時点コピーに関連していること
 と、

1つの仮想データベースのために、複数ファイルからなるファイルセットを作成すること
 であって、該ファイルセット内の各ファイルが、前記ソースデータベースの1つの時点
 コピーに関連した前記ストレージシステム内の前記データベースブロックにそれぞれリン
 クされており、前記1つの仮想データベースの1以上のデータベースブロックが別の仮想
 データベースと共有されることと、

前記1つの仮想データベースに関連する前記ファイルセットを、該ファイルセットに対
 する読み出し及び書き込みが可能なようにデータベースサーバにマウントすること、
 を含む方法。

【請求項 2】

前記仮想データベースに関連する前記ファイルセット内のデータの読み出しリクエスト
 を、前記データベースサーバから受信することと、

10

20

前記ファイルセット内のファイルに関連する少なくとも１つのデータベースブロック内のデータにアクセスすることと、

前記読み出しリクエストに応答して前記データを送信することと、をさらに含む、請求項１に記載の方法。

【請求項３】

前記ソースデータベースの時点コピーを受け取るためのリクエストを送信することをさらに含む、請求項１に記載の方法。

【請求項４】

前記ソースデータベースは生産データベースのストレージレベルのスナップショットである、請求項１に記載の方法。

10

【請求項５】

前記ソースデータベースは生産データベースを複製する予備データベースである、請求項１に記載の方法。

【請求項６】

前記仮想データベースは第１の仮想データベースであり、前記ソースデータベースは第２の仮想データベースである、請求項１に記載の方法。

【請求項７】

前記ソースデータベースと、前記ソースデータベースの時点コピーを管理するための既定のポリシーとを関連付けることをさらに含む、請求項１に記載の方法。

【請求項８】

20

前記既定のポリシーは、前記ソースデータベースの時点コピーを受け取るスケジュールを指定する、請求項７に記載の方法。

【請求項９】

前記スケジュールは、前記ソースデータベースの時点コピーを受け取る暦日を指定する、請求項８に記載の方法。

【請求項１０】

前記既定のポリシーは、保存期間後に前記時点コピーを消去するスケジュールを指定する、請求項７に記載の方法。

【請求項１１】

前記既定のポリシーは、前記ストレージシステムの利用可能スペースに基づいて、前記時点コピーの消去を指定する、請求項７に記載の方法。

30

【請求項１２】

前記ストレージシステムは、バーチャルマシンにおいて稼働する、請求項１に記載の方法。

【請求項１３】

前記ソースデータベースの時点コピーを送信するように構成されたプログラムコードを、前記ソースデータベースに関連する生産データベースシステムに送信することをさらに含む、請求項１に記載の方法。

【請求項１４】

以前の時点コピー受け取り後の前記ソースデータベースにおける変化を示す取引ログを表す情報を、前記ソースデータベースから受け取ることをさらに含む、請求項１に記載の方法。

40

【請求項１５】

取引ログを表す情報の以前の受け取り後の前記ソースデータベースにおける変化を示す取引ログを表す情報を、前記ソースデータベースから受け取ることをさらに含む、請求項１に記載の方法。

【請求項１６】

既定のポリシーに基づいて前記ソースデータベースに関連する取引ログを管理することをさらに含む、請求項１５に記載の方法。

【請求項１７】

50

既定のポリシーは、前記ソースデータベースの取引ログを受け取るスケジュールを指定する、請求項 15 に記載の方法。

【請求項 18】

前記スケジュールは、前記ソースデータベースの取引ログを受け取る暦日を指定する、請求項 8 に記載の方法。

【請求項 19】

前記既定のポリシーは、保存期間後に取引ログを消去するスケジュールを指定する、請求項 7 に記載の方法。

【請求項 20】

前記既定のポリシーは、前記ストレージシステムの利用可能スペースに基づいて、取引ログの消去を指定する、請求項 7 に記載の方法。

10

【請求項 21】

前記データベースサーバは第 1 のデータベースサーバであり、前記ソースデータベースは第 2 のデータベースサーバであり、前記第 1 のデータベースサーバのバージョンは前記第 2 のデータベースサーバのバージョンとは異なる、請求項 1 に記載の方法。

【請求項 22】

前記データベースサーバは第 1 のデータベースサーバであり、前記ソースデータベースは第 2 のデータベースサーバであり、前記第 1 のデータベースサーバは、前記第 2 のデータベースサーバが稼働しているするオペレーティングシステムとは異なるオペレーティングシステムで稼働している、請求項 1 に記載の方法。

20

【請求項 23】

前記ファイルセットは第 1 のファイルセットであり、前記仮想データベースは第 1 の仮想データベースであり、前記データベースサーバは第 1 のデータベースサーバであり、前記方法は、

前記ソースデータベースの時点コピーに関連する、前記ストレージシステムにおける前記データベースブロックにその各ファイルがリンクされ、第 2 の仮想データベースのための第 2 のファイルセットを作成することと、

前記第 2 の仮想データベースに関連する前記第 2 のファイルセットを、前記第 2 のファイルセットからの読み出し、およびそれへの書き込みが可能になるように第 2 のデータベースサーバにマウントすることと、をさらに含む、請求項 1 に記載の方法。

30

【請求項 24】

前記第 1 のファイルセットに関連する前記記憶したデータベースブロックの少なくとも一部は、前記第 2 のファイルセットにも関連する、請求項 23 に記載の方法。

【請求項 25】

前記第 1 の仮想データベースへのデータの書き込みリクエストを、前記データベースサーバから受信することと、

前記第 1 の仮想データベースに関連する前記第 1 のファイルセット内のファイルに関連するデータベースブロックを特定することと、をさらに含む、請求項 23 に記載の方法。

【請求項 26】

前記ファイルセットは第 1 のファイルセットであり、前記仮想データベースは第 1 の仮想データベースであり、前記データベースサーバは第 1 のデータベースサーバであり、前記時点コピーは第 1 の時点コピーであり、前記方法は、

40

前記ソースデータベースの第 2 の時点コピーに関連する、前記ストレージシステムにおける前記データベースブロックにその各ファイルがリンクされている、第 2 の仮想データベースに関する第 2 のファイルセットを作成することと、

前記第 2 の仮想データベースに関連する前記第 2 のファイルセットを、前記第 2 のファイルセットからの読み出し、およびそれへの書き込みが可能になるように第 2 のデータベースサーバにマウントすることと、をさらに含む、請求項 1 に記載の方法。

【請求項 27】

前記第 1 のファイルセットに関連する前記記憶したデータベースブロックの少なくとも

50

一部は、前記第2のファイルセットにも関連し、

前記第2の仮想データベースに関連する前記第2のファイルセットにも関連する前記データベースブロックに応答して、前記データベースブロックをコピーすることと、

前記コピーしたデータベースブロックと前記ファイルとをリンクして、前記データを前記コピーしたデータベースブロックに書き込むことと、をさらに含む、請求項25に記載の方法。

【請求項28】

前記時点コピーの受け取りは、前記ソースデータベースに関連するデータベースブロックが提供するデータを含む、前記時点コピーに対応するデータストリームの受け取りを含む、請求項1に記載の方法。

10

【請求項29】

前記受け取ったデータストリームを解析して、データベースブロックを特定することと、前記特定したデータベースブロックを記憶することと、をさらに含む、請求項28に記載の方法。

【請求項30】

前記時点コピーに対応するデータストリームは、以前の時点コピーの受け取り後に変化した前記ソースデータベース内のデータベースブロックを含む、請求項28に記載の方法。

【請求項31】

20

前記受け取ったデータストリームを解析してデータベースブロックを特定することと、前記データストリームを処理するために、前記データベースブロックのメタデータを解析して前記データベースブロックの全長を決定することと、をさらに含む、請求項28に記載の方法。

【請求項32】

前記受け取ったデータストリームを解析してデータベースブロックを特定することと、前記データベースブロックのメタデータを解析して前記データベースブロックを記憶するか否かを決定することと、をさらに含む、請求項28に記載の方法。

【請求項33】

前記受け取ったデータストリームを解析してデータベースブロックを特定することと、一時データベースブロック、空データベースブロック、または前記ソースデータベースの時点コピーの以前の取得後から変化しなかったデータベースブロックのいずれかである前記データベースブロックを記憶しないことを決定することと、をさらに含む、請求項28に記載の方法。

30

【請求項34】

階層メモリストレージデバイス内に前記取引ログを表す前記情報を記憶することをさらに含む、請求項15に記載の方法。

【請求項35】

前記ファイルセットのマウンティングは、ストレージプロトコルに依存しない、請求項1に記載の方法。

40

【請求項36】

前記ストレージシステムに記憶する前に、前記データベースブロックを圧縮することをさらに含む、請求項1に記載の方法。

【請求項37】

前記ソースデータベースは、生産データベースの一部である、請求項1に記載の方法。

【請求項38】

前記生産データベースの前記一部は、テーブルスペースを備えている、請求項37に記載の方法。

【請求項39】

前記生産データベースの前記一部は、データベーステーブルを少なくとも備えている、

50

請求項 37 に記載の方法。

【請求項 40】

前記ファイルセットにリンクされている前記データベースブロックは、前記ソースデータベースの一部を備えている、請求項 1 に記載の方法。

【請求項 41】

前記ファイルセットにリンクされている前記データベースブロックは、前記ソースデータベースに関連するデータベーステーブルを少なくとも備えている、請求項 1 に記載の方法。

【請求項 42】

前記仮想データベースは、全ての情報にアクセス可能な権限を有する仮想データベースである、請求項 1 に記載の方法。

10

【請求項 43】

前記仮想データベースは、秘密情報ではないと見なされるサブセット情報にアクセス可能な、権限を有さない仮想データベースである、請求項 1 に記載の方法。

【請求項 44】

前記仮想データベースは、秘密情報を隠す、権限を有さない仮想データベースである、請求項 43 に記載の方法。

【請求項 45】

時点コピーとブックマークトークンとを関連付けることと、

前記ブックマークトークンを指定して仮想データベースの作成に用いられる、前記ソースデータベースの前記時点コピーを指定することと、をさらに含む、請求項 1 に記載の方法。

20

【請求項 46】

ブックマークトークンを記憶することをさらに含む、請求項 44 に記載の方法。

【請求項 47】

前記仮想データベースと、所定の権限を有するユーザへの情報のアクセスを指定する 1 つ以上の権限とを関連づけることをさらに含む、請求項 1 に記載の方法。

【請求項 48】

権限は、ポリシー管理を許可する管理者権限、VDB のプロビジョニングを許可する所有者権限、および VDB に関連する情報の確認を許可する監査員権限のいずれかである、請求項 1 に記載の方法。

30

【請求項 49】

仮想データベースを作成するためのコンピュータ実装システムであって、

コンピュータプロセッサと、前記コンピュータプロセッサを実行するように構成されたコンピュータプログラムモジュールを記憶するコンピュータ可読ストレージ媒体とを備えており、前記コンピュータプログラムモジュールが、

複数のデータベースブロックを備えているソースデータベースの異なる時点のコピーである種々の時点コピーを受け取るように構成された時点コピーマネージャモジュールと、

前記ソースデータベースの複数の種々の時点コピーについての複数のデータベースブロックをストレージシステムに記憶するように構成されたストレージ割当マネージャモジュールであって、該記憶した複数のデータベースブロックの少なくとも一部が前記ソースデータベースの複数の時点コピーに関連している、前記ストレージ割当マネージャモジュールと、

40

1 つの仮想データベースのために、複数ファイルからなるファイルセットを作成するように構成された仮想データベースマネージャモジュールであって、該ファイルセット内の各ファイルが、前記ソースデータベースの 1 つの時点コピーに関連した前記ストレージシステム内の前記データベースブロックにそれぞれリンクされており、前記 1 つの仮想データベースの 1 以上のデータベースブロックが別の仮想データベースと共有される、前記仮想データベースマネージャモジュールと、

前記 1 つの仮想データベースに関連する前記ファイルセットを、該ファイルセットに対

50

する読み出し及び書き込みが可能なようにデータベースサーバにマウントするように構成されたファイル共有マネージャモジュールと、
を備えるシステム。

【請求項 5 0】

前記仮想データベースマネージャモジュールは、

前記仮想データベースに関連する前記ファイルセット内のデータの読み出しリクエストを、前記データベースサーバから受信し、

前記ファイルセット内のファイルに関連する少なくとも 1 つのデータベースブロック内のデータにアクセスし、かつ、

前記読み出しリクエストに応答して前記データを送信するようにさらに構成されている、請求項 4 9 に記載のシステム。

10

【請求項 5 1】

前記仮想データベースマネージャモジュールは、

第 1 の仮想データベースへのデータの書き込みリクエストを、前記データベースサーバから受け取り、

前記第 1 の仮想データベースに関連する第 1 のファイルセット内のファイルに関連するデータベースブロックを特定し、

第 2 の仮想データベースに関連する第 2 のファイルセットにも関連する前記データベースブロックに응答して、前記データベースブロックをコピーし、かつ、

前記コピーしたデータベースブロックと前記ファイルとをリンクして、前記データを前記コピーしたデータベースブロックに書き込むようにさらに構成されている、請求項 4 9 に記載のシステム。

20

【請求項 5 2】

前記時点コピーマネージャモジュールは、前記ソースデータベースの時点コピーを受け取るためのリクエストを送信するようにさらに構成されている、請求項 4 9 に記載のシステム。

【請求項 5 3】

前記時点コピーマネージャモジュールは、前記ソースデータベースのデータベースブロックが提供するデータを含むデータストリームを含む時点コピーを受け取る、請求項 4 9 に記載のシステム。

30

【請求項 5 4】

前記時点コピーマネージャモジュールは、前記受け取ったデータストリームを解析して記憶するデータベースブロックを特定するようにさらに構成されている、請求項 4 9 に記載のシステム。

【請求項 5 5】

前記データストリームは、以前の時点コピーの受け取り後に変化した前記ソースデータベース内のデータベースブロックを含む、請求項 5 3 に記載のシステム。

【請求項 5 6】

仮想データベースを作成するためのコンピュータ実行可能コードを記憶するコンピュータ可読記憶媒体であって、前記コードは、

40

複数のデータベースブロックを備えているソースデータベースの異なる時点のコピーである種々の時点コピーを受け取るように構成された時点コピーマネージャモジュールと、

前記ソースデータベースの複数の種々の時点コピーについての複数のデータベースブロックをストレージシステムに記憶するように構成されたストレージ割当マネージャモジュールであって、該記憶した複数のデータベースブロックの少なくとも一部が前記ソースデータベースの複数の時点コピーに関連している、前記ストレージ割当マネージャモジュールと、

1 つの仮想データベースのために、複数ファイルからなるファイルセットを作成するように構成された仮想データベースマネージャモジュールであって、該ファイルセット内の各ファイルが、前記ソースデータベースの 1 つの時点コピーに関連した前記ストレージシ

50

ステム内の前記データベースブロックにそれぞれリンクされており、前記１つの仮想データベースの１以上のデータベースブロックが別の仮想データベースと共有される、前記仮想データベースマネージャモジュールと、

前記１つの仮想データベースに関連する前記ファイルセットを、該ファイルセットに対する読み出し及び書き込みが可能ないようにデータベースサーバにマウントするように構成されたファイル共有マネージャモジュールと、を備えている、コンピュータ可読記憶媒体。

【請求項５７】

前記ソースデータベースは第１のソースデータベースであり、前記複数のデータベースブロックは第１の複数のデータベースブロックであり、前記複数ファイルからなるファイルセットは第１のファイルセットであり、前記仮想データベースは第１の仮想データベースであり、前記データベースサーバは第１のデータベースサーバであり、

第２の複数のデータベースブロックを備えている第２のソースデータベースの時点コピーを受け取ることと、

前記第２のソースデータベースの前記時点コピーについての複数のデータベースブロックを前記ストレージシステムに記憶することであって、該記憶した複数のデータベースブロックの少なくとも一部が前記第２のソースデータベースの複数の時点コピーに関連していることと、

１つの第２の仮想データベースのために、複数ファイルからなる第２のファイルセットを作成することであって、該複数ファイルからなる第２のファイルセットが、前記第２のソースデータベースの前記時点コピーに関連した前記ストレージシステム内の前記データベースブロックにリンクされていることと、

前記第２の仮想データベースに関連する前記第２のファイルセットを、該第２のファイルセットに対する読み出し及び書き込みが可能ないように第２のデータベースサーバにマウントすること、

を含む、請求項１に記載の方法。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、一般的にデータベースに関するものであり、特に、データベースおよびデータベースに基づくライフサイクルワークフローを管理するためのストレージ効率システムに関するものである。

【背景技術】

【０００２】

データベースは、組織に不可欠なデータを記憶するため、組織のＩＴインフラストラクチャの重要な部分となっている。組織が利用する情報の増大に伴い、情報を記憶するデータベースの管理に要求されるインフラストラクチャの複雑性も増す。インフラストラクチャの複雑性が増すと、データベースおよびデータベースに応じて決まるアプリケーションの管理に要求される資源も増大する。これにより費用が増大する。この費用には、データベースを管理するハードウェアに関する費用に加えて、ハードウェアの維持に必要な補助人員の人件費も含まれ得る。インフラストラクチャの複雑性が増すと、データベースに関連するメンテナンス操作にも影響を及ぼし、例えば、バックアップおよび復旧操作が顕著に長くなる。

【０００３】

組織の標準的なインフラストラクチャ環境では、生産データベースサーバが、組織の日常取引を管理するアプリケーションを稼働する。データベースのコピーにより、生産データベースまたは生産データベースに応じて決まるアプリケーションにおける変化を検証して、生産環境を保護する。生産データベースのコピーは、生産データベースおよび生産データベースに応じて決まるアプリケーションに関連するライフサイクルワークフローのいくつかの段階において要求され得る。例えば、生産データベースに組み込まれる、変更可

10

20

30

40

50

能なライフサイクルにおける段階は、開発段階、同調段階、検証段階、品質保証段階、認証段階、訓練段階、および段階分け段階を含み得る。段階ごとの生産データベースのコピーには、余分かつ高価なハードウェアインフラストラクチャに加えて、データコピーに必要な何日または何週間という時間に伴う費用が必要となる。ハードウェアの追加にはさらに、床面積の要求並びに電力および冷却に伴う費用などの、ハードウェアの物理的記憶に伴う追加費用が必要となる。余分なハードウェアはまた、通常、利用可能な資源の使用を非効率的にする。

【 0 0 0 4 】

ライフサイクルワークフローは複雑な場合があり、多くの場合に複数のチーム間の連携が必要となる。それ故、ライフサイクルにおける特定段階のサポートなどの特定の目的に使用可能なデータベースの構成には、データベースに関連するさらなる処理が必要となることがある。例えば、データベースは多くの場合に重大な秘密情報を含み、環境管理データベースにとって安全性および完全性が重視すべき事柄となる。この結果、大抵は、異なる段階を操作する異なるチームに要求されるアクセス許可はそれぞれ異なる。例えば、生産データベースサーバを管理する人員がアクセス可能なデータは、多くの場合に、ライフサイクルの検証段階の操作者がアクセス可能なデータとは異なる。これにより、データベースに関連する任意のライフサイクルワークフローの種々の段階における許可の管理がさらに複雑になる。

【 発明の概要 】

【 課題を解決するための手段 】

【 0 0 0 5 】

既存のデータベース技術では解決できない要求に対処するために、本発明の実施形態は、ストレージおよび他のコンピュータ資源を効率的に用いる仮想データベースを実現する。仮想データベース（VDB）の作成方法は、ソースデータベースから種々の時点コピーを読み出すことを含む。これらの実施形態では、「ソースデータベース」は、企業内のデータベース、例えば、生産データベース、予備データベース、およびライフサイクルにおける任意の他のデータベースの物理的コピーを含む。以下の本発明の詳細な説明では、「生産データベース」と「ソースデータベース」は同一のものを意味し、互いに置換可能である。複数のデータベースブロックがソースデータベースから読み出され、ストレージシステムに記憶される。データベースブロックはデータベースに用いられるデータユニットであり、ストレージに記憶される一定のバイト数を有する。データベースブロックはページとも呼ばれ得る。データベースブロックの一部は、データベースブロックに関するメタデータを記憶する。データベースブロックのメタデータに記憶され得る情報の例は、データベースブロックに記憶されるデータに関連する情報、データベースの一部であるデータベースブロックのオブジェクトに関する情報、またはデータベースブロック内のデータが更新された時を示す情報を含む。データベースブロックが更新された時を示す情報は、それらの更新時刻に基づいてデータベースブロックの相対順序を決定するために利用され得る。ソースデータベースから取得され、ストレージシステムが記憶したデータベースブロックは、ソースデータベースの種々の時点コピーに対応し、データベースブロックの少なくとも一部は、ソースデータベースの複数の時点コピーに関連する。仮想データベース用のファイルセットを作成する。VDBにおいて作成されたファイルセット内の各ファイルは、ソースデータベースの時点コピーに関連するストレージシステムのデータベースブロックにリンクされている。VDBに関連するファイルセットがデータベースサーバに組み込まれると、データベースサーバはファイルセットからの読み出し、およびそれへの書き込みが可能になる。一実施形態では、仮想データベースは、別の仮想データベースの時点コピーに基づいて作成され得る。

【 0 0 0 6 】

一実施形態では、ソースデータベースの同一の時点コピーに関連するデータベースブロックに基づいて複数のVDBを作成し得る。代替的に、ソースデータベースの種々の時点コピーに関連する2つのVDBを作成してもよい。ストレージシステムが記憶したデータ

10

20

30

40

50

ベースブロックを、互いに異なるVDBに関連するファイルセット間で共有し得る。ストレージシステムが記憶したデータベースブロックは、ソースデータベースの同一の時点コピーに関連するVDB間、またはソースデータベースの種々の時点コピーに関連するVDB間で共有され得る。一実施形態では、リンク操作、ローディング、またはプロビジョニング操作の前にスクリプト前操作が実行されてもよい。スクリプト前操作により、ソースデータベースに記憶されない処理情報などのユーザ指定操作が実行可能になる。同様に、リンク操作、ローディング、またはプロビジョニング操作後に、スクリプト後操作が実行され得る。スクリプト後操作はスクリプト前操作に関連し得、スクリプト前操作における処理情報をさらに処理する。

【0007】

10

一実施形態では、VDBに記憶したデータを読み出すための、データベースサーバからのリクエストを受信できる。リクエストされたデータはデータベースブロックからアクセスされ、読み出しリクエストに回答してデータベースサーバに送信される。別の実施形態では、データベースサーバからのリクエストを受信し得、そのデータをVDBに書き込む。VDBに関連するファイルセット内のファイルに関連するデータベースブロックは、書き込みリクエストと共に送られたデータを書き込むために特定される。特定されたデータベースブロックが第2のVDBにも関連する場合には、データベースブロックをコピーする。コピーしたデータベースブロックをファイルにリンクし、書き込みリクエストに関するデータを、コピーしたデータベースブロックに書き込む。コピーされる元のデータベースブロックが第2のVDBとの関連性を維持するため、第2のVDBは書き込み操作でも

20

変化しなかったデータを確認できる。

【0008】

本要約および以下の詳細な説明に記述する特徴および利点を全て含む訳ではない。多くのさらなる特徴および利点が、図面、明細書および請求項を確認した当業者に明らかとなる。

【図面の簡単な説明】

【0009】

【図1】本発明の実施形態に従う、生産データベースからデータベースストレージシステムに情報をコピーする方法、およびファイル共有システムを利用してプロビジョニングした仮想データベースを示す図である。

30

【0010】

【図2a】本発明の実施形態に従う、仮想データベースのソースである生産データベースシステムのデータベースサーバのバージョンとは異なるバージョンの仮想データベースシステムが稼働する方法を示す図である。

【0011】

【図2b】本発明の実施形態に従う、仮想データベースのソースである生産データベースシステムのデータベースサーバを実行するオペレーティングシステムとは異なるオペレーティングシステムにより実行されるデータベースサーバを用いて仮想データベースシステムが稼働し得る方法を示す図である。

【0012】

40

【図3】本発明の実施形態に従う、生産データベースが提供する情報のストレージ効率のよいコピー、および仮想データベースのプロビジョニングを実行するシステム構築を示す概略図である。

【0013】

【図4】本発明の実施形態に従う、データベースストレージシステムにおける生産データベースのストレージ効率のよいコピーを実現するための、データベースストレージシステムの構成要素と、生産データベースシステムの構成要素とのやりとりを示す図である。

【0014】

【図5】本発明の実施形態に従う、生産データベースシステムから受け取ったデータストリームをデータベースストレージシステムにおいて処理して、ストレージ効率のよい方法

50

でデータを保存する方法を説明するプロセスフローチャートである。

【0015】

【図6】本発明の実施形態に従う、生産データベースシステムからデータベースストレージシステムに取引ログファイルをコピーして、所定の時点における仮想データベースのプロビジョニングを可能にする方法を説明するプロセスフローチャートである。

【0016】

【図7】本発明の実施形態に従う、生産データベースシステムと比較したデータベースストレージシステム内の取引ログの記憶に用いられるファイルを示す図である。

【0017】

【図8】本発明の実施形態に従う、データベースストレージシステムの様々な時点においてデータベースに関するデータを保持する方法を示す図である。

10

【0018】

【図9】本発明の実施形態に従う、所定の時点において仮想データベースを作成する方法を説明するプロセスフローチャートである。

【0019】

【図10】本発明の実施形態に従う、所定の時点におけるデータベースの読み出し - 書き込みコピーを作成した、仮想データベースのプロビジョニングを示す図である。

【0020】

【図11】本発明の実施形態に従う、図10とは異なる時点におけるデータベースの読み出し - 書き込みコピーを作成した、仮想データベースのプロビジョニングを示す図である。

20

【0021】

【図12】本発明の実施形態に従う、ストレージシステムデータストアが記憶したデータベースブロックが、異なるVDBにおいて作成されたファイル構造を共有する方法を示す図である。

【0022】

【図13】本発明の実施形態に従う、データベースの読み出し - 書き込みコピーを作成した、生産データベースシステムからコピーした取引ログに基づく仮想データベースのプロビジョニングを示す図である。

【0023】

【図14】ある例示の環境において、データベースまたはデータベースに応じて決まるアプリケーションを変化させる、データベースのライフサイクルワークフローを示す図である。

30

【0024】

【図15】従来方法を用いる、データベースおよびデータベースアプリケーションに関連するプログラムコードの検証および開発のためのワークフローを実施するシステム環境を示す図である。

【0025】

【図16】本発明の実施形態に従う、VDBを用いたデータベースおよびデータベースアプリケーションに関連するプログラムコードの検証および開発のためのワークフローを実施するシステム環境を示す図である。

40

【0026】

【図17】本発明の実施形態に従う、VDBを用いたデータベースおよびデータベースアプリケーションに関連するプログラムコードの複数の場所における検証および開発のためのワークフローを実施するシステム環境を示す図である。

【0027】

【図18a】従来方法における、データベースのバックアップおよび復旧のためのワークフローを実施するシステム環境を示す図である。

【0028】

【図18b】本発明の実施形態に従う、VDBを用いたデータベースのバックアップおよ

50

び復旧のためのワークフローを実施するシステム環境を示す図である。

【0029】

【図19】従来方法における、ある機械から別の機械へのデータベース内の情報のコピーに要求される一般的シナリオ用のワークフローを実施するシステム環境を示す図である。

【0030】

【図20】本発明の実施形態に従う、ある機械から別の機械へのデータベース内の情報のコピーに要求される一般的シナリオ用のVDBに基づいてワークフローを実施するシステム環境を示す図である。

【0031】

【図21】本発明の別の実施形態に従う、ある機械から別の機械へのデータベース内の情報のコピーに要求されるシナリオ用のVDBに基づいてワークフローを実施するシステム環境を示す図である。

10

【0032】

【図22】本発明の実施形態に従う、ある機械から、生産データベースシステムとは異なる別の機械へのデータベース内の情報のコピーに要求される一般的シナリオ用のVDBに基づいてワークフローを実施するシステム環境を示す図である。

【0033】

【図23】従来方法を用いる、データベースからデータウェアハウスおよびデータマートを作成するシナリオ用のワークフローを実施するシステム環境を示す図である。

【0034】

20

【図24】本発明の実施形態に従う、データベースからデータウェアハウスおよびデータマートを作成するシナリオ用のワークフローを実施する、VDBに基づくシステム環境を示す図である。

【0035】

【図25】機械可読媒体から命令を読み出し得、プロセッサまたはコントローラによりその命令を実行し得るコンピューティングマシンの実施形態を示す図である。

【発明を実施するための形態】

【0036】

図面は、説明のみを目的として本発明の種々の実施形態を示す。下の記述から、本明細書に記載された本発明の原理から逸脱することなく、本明細書に記述する構造および方法の代替的な実施形態が用いられ得ることを当業者は理解する。

30

【0037】

仮想データベースシステム

本発明の特定の実施形態では、特定の時点において、生産データベースまたは仮想データベースの状態に基づいて1つ以上の仮想データベースを作成し、その後、所望に応じて、仮想データベースに個別にアクセスおよびそれを修正することができる。データベースは、コンピュータが実施するアプリケーションに利用される、コンピュータにおいて記憶されるデータを含む。データベースサーバは、データベースと相互に通信できるコンピュータプログラムであり、データベースに記憶されたデータへのアクセスなどのデータベースサービスを提供する。データベースサーバは、ORACLE、SYBASE、MICROSOFT SQL SERVER、IBMのDB2、MYSQLなどが提供するデータベース管理システムなどの市販のプログラムを含む。データベースは、関係モード、オブジェクトモデル、階層モード、またはネットワークモデルなどのデータベースモデルを用いて実施され得る。「生産データベース」という用語は特定例では、この技術の有用な用途を説明するために用いられるが、生産データベースとしてデータベースを用いるかどうかに関わらず、開示する技術は任意のデータベースに用いられ得ることが理解される。実施形態はさらに、生の生産データベースではなく、生産データベースのストレージレベルスナップショット、または生産データベースのクローンを用いて仮想データベースを作成してもよい。仮想データベースは、データベースファイルの物理的实施が、データベースサーバによるデータベースファイルの論理的な使用と分離しているという意味で「仮想」で

40

50

ある。

【 0 0 3 8 】

一実施形態では、生産データベースが提供する情報を、定期的に等、様々な時間においてストレージシステムにそれぞれコピーする。これらのコピーにより、これら種々の時点（複数の異なる時点）における生産データベースに関連するデータベースファイルをそれぞれ復元できる。この情報は必要な場合にのみコピーされるように、効率的な方法でストレージシステムが管理し得る。例えば、データベースの一部が以前にコピーしたバージョンから変化していない場合には、変化していない部分はコピーする必要がない。時点間において作成した仮想データベースは、その時点で利用可能なデータベース情報を含むファイルセットとして記憶される。各ファイルはデータベースブロックセットおよびデータベースブロックを参照するためのデータ構造を含む。一部の実施形態では、データベースブロックは効率的な記憶のために圧縮され得る。一部の実施形態では、データベースブロックは記憶するデータの安全性を高めるために暗号化された形式でストレージシステムデータストア 390 に記憶されてもよい。データベースサーバに関する要求に応じて、以前の時点における生産データベースの状態に対応する生産データベースに関するデータベースファイルを作成することにより、データベースサーバに仮想データベースを作成できる。仮想データベースに対応するファイルは、ファイル共有機構を用いてデータベースサーバが使用できるようにされ、ファイル共有機構は仮想データベースとストレージシステムが記憶した適切なデータベースブロックとをリンクする。仮想データベースをデータベースサーバに対して利用可能にする処理は、仮想データベースの「プロビジョニング」とも呼ばれる。一部の実施形態では、仮想データベースのプロビジョニングは、仮想データベースに基づいて稼働データベースサーバを構築する処理の管理を含む。複数の V D B が、同一の時点における生産データベースの状態に基づいてプロビジョニングされてもよい。これに対して、種々の V D B が、同一の生産データベースまたは互いに異なる生産データベースの異なる時点における状態に基づいてプロビジョニングされてもよい。一部の実施形態では、プロビジョニングされたデータベースの健全性およびユーザ動作をモニタする。データベースストレージシステム 100 には、これらのイベントが通知される。データベースストレージシステム 100 は、既定の規則またはユーザ指定規則のいずれかに基づいてこれらのイベントに対処する。例えば、ユーザ動作が仮想データベースの使用可能性に影響を及ぼす場合には、モニタリングコンソールに警告メッセージを表示するか、ユーザに電子メールを送信する。仮想データベースがプロビジョニングされたデータベースサーバは、その後、ストレージシステムが記憶したファイルを読み出すか、そこに書き込むことができる。データベースブロックは、種々の異なる V D B にその各々が関連する異なるファイル間で共有され得る。特に、対応する仮想データベースシステム 130 がデータベースブロック内の情報を読み出すのみで、データベースブロックに書き込まない場合に、データベースブロックは共有される。一実施形態では、仮想データベースマネージャ 375 が、必要な場合にのみデータベースブロックをコピーする。例えば、特定のデータベースブロックが、同一のデータベースブロックを読み出す複数の V D B 間で共有され得る。ただし、仮想データベースシステム 130 の 1 つがデータベースブロックの書き込みを試みる場合には、書き込む操作が、その仮想データベースシステム 130 に対応する V D B に対するデータベースブロックと、他の V D B に対するデータベースブロックとが異なるようにさせるので、データベースブロックは個々にコピーされる。

【 0 0 3 9 】

図 1 は情報がどのように生産データベースからデータベースストレージシステムにコピーされるか、および情報がファイル共有システムを用いてどのように仮想データベースとしてプロビジョニングされるかの実施形態を示す図である。生産データベースシステム 110 は、組織のデータを管理する。一部の実施形態では、生の生産データベースではなく、生産データベースのストレージレベルスナップショット、または生産データベースのクローンから情報がコピーされ得る。データベースストレージシステム 100 は、以下に詳細に記載するように、1 つ以上の生産データベースシステム 110 からデータベースに関

10

20

30

40

50

連するデータを取得し、効率的な方法でデータを記憶する。データベース管理ユーザインターフェース 140 は、データベース管理者がデータベースストレージシステム 100 がサポートする種々の機能を実行できるようにする。

【0040】

管理システム 140 からのリクエストに応答するか、所定のスケジュールに基づいて、データベースストレージシステム 100 はデータに関連するリクエスト 150 を生産データベースシステム 110 に送信し得る。生産データベースシステム 110 はこのリクエストに応答して、生産データベースが記憶している情報をデータストリーム 160 として送信する。リクエスト 150 は定期的に送られ、生産データベースシステム 110 はこのリクエストに応答して、前回の応答 160 の送信後に、生産データベースに記憶したデータの変化を示す情報を送信する。データベースストレージシステム 100 は、生産データベースシステム 110 が送信したデータ 160 を受け取って、そのデータを記憶する。データベースストレージシステム 100 は、受け取ったデータ 160 を解析して、その情報を記憶するか、その情報が以前の時点におけるデータベースの復旧に有用でない場合にはそれをスキップするか否かを決定できる。データベースストレージシステム 100 は、例えば、変化したデータベースブロックのバージョンを保持し、変化していないデータベースブロックを再利用することにより、情報を効率的に記憶する。一実施形態では、データベースストレージシステム 100 は階層キャッシュシステムを利用し、そこではキャッシュ読み出し操作、および磁気ディスクへの書き込み操作のログの保持のために、高速の半導体ドライブ (SSD) または同等のストレージデバイスが構成される。

【0041】

仮想データベースを作成するために、データベースストレージシステム 100 は、所定の時点における生産データベースシステム 110 に対応する情報を示すファイルを作成する。データベースストレージシステム 100 は、ファイル共有システム 120 を利用して仮想データベースシステム 130 に対して対応するファイルを公開 170 する。仮想データベースシステム 130 は、データベースストレージシステム 100 が公開 170 したファイルで動作し得るデータベースサーバを稼働する。それ故、ストレージ効率のよい方法で、生産データベースの仮想コピーを所定の時点に対する仮想データベースシステム 130 のために作成する。

【0042】

図 2 a、図 2 b は仮想データベースのソースである生産データベースシステム 110 とは異なるデータベースサーバのバージョンおよび / またはオペレーティングシステムを、仮想データベースシステム 130 が稼働し得ることを示す図である。データベースストレージシステム 100 に記憶される仮想データベースファイルは、データベースサーバ 230 がデータベースサーバ 205 とは異なるバージョンであり、かつ / またはオペレーティングシステム 240 がオペレーティングシステム 210 とは異なる場合でも、仮想データベースシステム 130 がファイルを動作するように、適切に修正される。図 2 a に示すように、仮想データベースシステム 130 において稼働するデータベースサーバ 230 のバージョン V_y は、生産データベースシステム 110 において稼働するデータベースサーバ 205 のバージョン V_x とは異なる。同様に、図 2 b に示すように、仮想データベースシステム 130 において稼働するオペレーティングシステム 240 は OS_y であり、生産データベースシステム 110 において稼働するオペレーティングシステム OS_x とは異なる。一実施形態では、サーバ 230 および 205 は、異種のデータベースソフトウェアプログラムを稼働できる。これにより、データベースを稼働するために異なるオペレーティングシステムまたはデータベースサーバのバージョンを試みる能力が与えられる。この能力により、データベースおよび / またはアプリケーションのアップグレード、パッチング、またはデータ移動において、生産システムにいかなる影響をも与えることなく、操作を容易に検証できる。操作は、その後、生産システムに配置される前に、隔離環境で認証され得る。一部の実施形態では、データベースストレージシステム 100 は、ホストコンピュータにおいて複数のオペレーティングシステムを同時に稼働可能であるプラットフォーム

仮想ソフトウェアまたはサーバ仮想ソフトウェアにより提供されるバーチャルマシンにおいて実行され得る。

【0043】

システム構築

図3は生産データベースから情報のストレージ効率のよいコピーを作成すること、およびその情報を利用して1つ以上の仮想データベースのプロビジョニングの実行に適したシステム環境を示すハイレベルブロック図である。システム環境は、1つ以上の生産データベースシステム110、データベースストレージシステム100、管理システム140、および1つ以上の仮想データベースシステム130を備えている。図3に示すシステムは、必要に応じて、ネットワーク上で互いに通信できる。

10

【0044】

生産データベースシステム110は、通常、組織により利用され、その日常取引を保持する。例えば、オンライン書籍販売サイトは、生産システム110内の書籍注文、書籍返却、または在庫管理に関する全ての進行中の取引を保存できる。生産システム110は、データベースサーバ345、生産DBデータストア350、ベンダーインターフェースモジュール335、および生産システムライブラリ385を含む。代替的な構成では、生産データベースシステム110は、異なるおよび/または追加のモジュールを含んでもよい。

【0045】

生産DBデータストア350は、企業の日常取引を示す情報などに相当し得る、データベースに関連するデータを記憶する。データベースサーバ345は、データベースサーバおよびアプリケーションプログラミングインターフェース(API)を提供するコンピュータプログラムであり、生産DBデータストア350が記憶したデータを管理する。生産システムライブラリ385は、生産データベースシステム110から情報を抽出するのに有用なAPIを備える。ベンダーインターフェースモジュール335は、ベンダーが規定するAPI、例えば、以前の時点から変化したデータベースブロックを取得するAPIに相当し、データベースサーバ345が提供する機能をカスタマイズする。ベンダーインターフェースモジュールの例には、RMAN APIを実施するベンダーであるORACLEが提供するデータベースサーバのプログラムコードがある。MICROSOFTのSQL SERVERまたはIBMのDB2などの他のベンダーが提供するデータベースサーバも、類似のAPIを有する。一実施形態では、ベンダーインターフェースモジュール335は、ファイル共有システム120に類似するファイル共有システムを用いてデータベースストレージシステム100において生産データベースシステム110の生産DBデータストア350をマウントする(利用可能に取り付ける)。データベースストレージシステム100に生産DBデータストア350をマウントすることにより、生産データベースシステム110が記憶した情報を、データベースストレージシステム100に転送できる。

20

30

【0046】

生産システムライブラリ385は、ベンダーインターフェースモジュール335の要求に応じて、種々の方法により実装され得る。一実施形態では、ベンダーインターフェースモジュール335は、生産システムライブラリ385に実装された機能呼び出すために生産システムライブラリ385をロードする。例えば、生産システムライブラリ385は、C/C++実行可能プログラム、またはC/C++プログラムが生成した2進コードとやりとりするJAVA(登録商標) NATIVE INTERFACEを利用するJAVA(登録商標)プログラムにより呼び出され得る実行可能プログラムコードを含む「.so」または「.dll」ファイル拡張子を有する共有のオブジェクトファイルである。代替的に、生産システムライブラリ385は、JAVA(登録商標)プログラミング言語を用いて実装され得、「.jar」拡張子を有するファイルとして生産データベースシステム110にインストールされ得る。JAVA(登録商標)プログラムは、実行用に生産データベースシステム110において稼働するJAVA(登録商標) VIRTUAL

40

50

MACHINEを要求する。別の実施形態では、生産システムライブラリ385の一部は、実行可能な「.so」の共有オブジェクトファイルとして実装され得、生産システムライブラリ385の別の部分は、「.jar」ファイルとしてインストールされたJAV A（登録商標）プログラムとして実装され得る。

【0047】

ベンダーインターフェースモジュール335は、データベースストレージシステム100からのリクエストに回答して、リクエストされた情報を生産DBデータストア350から収集し、収集した情報をデータベースストレージシステム100に返す。ベンダーインターフェースモジュール335は、生産DBデータストア350から情報を取得するためのリクエストをデータベースサーバ345に送信できる。ベンダーインターフェースモジュール335は、生産システムライブラリ385のプログラムコードをロードして、それを起動してそのデータストリームを、さらなる処理のためにデータベースストレージシステム100に送信する。一部の実施形態では、ベンダーインターフェースモジュール335は、必要なデータベースブロックを取得するためのリクエストをデータベースサーバ345に送信する代わりに、生産DBデータストア350と直接通信してもよい。他の実施形態では、ベンダーインターフェースモジュール335は、生の生産データベースではなく、生産データベースのストレージレベルスナップショットまたは生産データベースのクローンから必要なデータベースブロックを取得できる。

【0048】

データベースストレージシステム100は、生産データベースシステム110に利用可能な情報を取得してそれを記憶する。取得された情報は、データベースが記憶しているデータを含むデータベースブロック、取引ログ情報、データベースに関連するメタデータ情報、データベースのユーザに関連する情報などを含む。取得情報はさらに、データベースに関連する環境設定ファイルを含み得る。例えば、データベースはベンダー固有の環境設定ファイルを利用して、データベースに関連する初期化パラメータを含む種々の設定パラメータを指定できる。環境設定ファイルのコピーにより、ソース生産データベースに類似する設定パラメータをVDBに設定可能になる。一部の実施形態では、データベース管理者はユーザインターフェース395を用いて設定パラメータファイルを変更し、特定の使用シナリオ用の設定にVDBをカスタマイズできる。例えば、生産データベースは、特定のキャッシュサイズを用いるデータベースサーバ345によりアクセスされ得、一方、対応するVDBは、異なるキャッシュサイズを用いるデータベースサーバ360によりアクセスされ得る。

【0049】

取得情報はさらに、データベースを用いるアプリケーションに関連する情報を含み得、例えば、統合業務ソフト（ERP）アプリケーションはデータベースを用いて、ERPアプリケーションに特有のデータを有し得る。ERPアプリケーションデータを取得することにより、生産データベースシステムに基づいて作成されたVDBにおいて、類似のERPアプリケーションが実行可能になる。これは検証および開発などを目的として、生産環境に類似する環境用のVDBを作成する場合の使用シナリオにおいて有益となる。データベース管理者はユーザインターフェース395を用いて、生産環境に特有の情報をコピーするための論理に加えて、仮想データベースシステム130として用いるVDBに、情報を適切にインストールするための論理を指定できる。

【0050】

一部の実施形態では、管理者権限を有するユーザなどの生産データベースのユーザに関する情報は、固有のAPIを用いるか、生産データベースにおける固有のスクリプトを実行することにより取得できる。ユーザに関する情報は、システム内のVDBのライフサイクル管理を容易にするために用いられ得る。一実施形態では、データベース管理者はユーザインターフェース395を用いて、ユーザアカウントの作成、およびそのアクセス許可に関する情報を指定できる。例えば、VDBがテスト目的で作成される場合には、テスト組織用のVDBにおいてテストユーザが作成され得、一方、VDBが生産データベース

の予備として作成される場合には、生産サポート規則を有するユーザのみがアクセスできるようにすべきである。一部の実施形態では、ユーザが権限を有するVDBをプロビジョニングできるか否かはアクセス許可により決定され得る。権限を有するVDBの一例は、社会保障番号またはクレジットカード情報などの非公開情報（権限を有さないユーザがアクセスできない情報）へのフルアクセスを有するVDBである。これに対応する権限が不要なVDBは、秘密または暗号化した非公開情報を有するVDBである。権限を有するVDBの別の例は、ユーザが意識せずにアクセス可能な極秘データを有するVDBである。対応する権限が不要なVDBは、暗号化された秘密情報を有するVDBである。

【0051】

一部の実施形態では、アクセス権は、管理者、所有者および監査員の3レベルに簡易化される。管理者は、データベースおよびホストを含む全ての管理オブジェクトを完全に制御できる。管理者に利用可能な制御にはポリシー管理が含まれる。所有者は資源を利用するためのアクセスを有し、例えば、VDBをプロビジョニングできる。監査員は、ログは確認できるが、システム資源を消費する権限はない場合がある。

【0052】

ストレージシステムデータストア390が記憶したデータは、仮想データベースシステム130に公開され得、仮想データベースシステム130が、生産データベースシステム110が記憶した生産データベースのコピーとしてデータを処理することを可能にする。データベースストレージシステム100は、時点コピーマネージャ310、取引ログマネージャ320、インターフェースマネージャ330、システム構成マネージャ315、ストレージ割当マネージャ365、ファイル共有マネージャ370、仮想データベースマネージャ375、およびストレージシステムデータストア390を含む。代替的な構成では、データベースストレージシステム100は、異なるおよび/または追加のモジュールを含んでもよい。

【0053】

時点コピーマネージャ310は、ベンダーインターフェースモジュール335にリクエストを送信することにより、生産データベースシステム110と相互に通信し、生産DBデータストア350が記憶したデータベースの時点コピー（「PITコピーとも呼ばれる」）を示す情報を取得できる。時点コピーマネージャ310は、生産データベースシステム110から取得したデータをストレージシステムデータストア390に記憶する。時点コピーマネージャ310が取得したデータは、生産DBデータストア350からコピーしたデータベースのデータベースブロック（またはページ）に対応する。生産DBデータストア350から情報を取得する第1のPITコピーのリクエスト後のPITコピーリクエストは、データベース内の以前のリクエストから変化したデータのみを取得してもよい。第1のリクエストにより収集されたデータは、第2のリクエストにおいて収集されたデータと組み合わせさせて、第2のリクエスト用の生産DBデータストア350からデータを取得した時点に対応するデータベースのコピーを再構成する。

【0054】

取引ログマネージャ320は、生産データベースシステム110が記憶した取引ログの一部を取得するためのリクエストを生産データベースシステム110に送信する。一部の実施形態では、取引ログマネージャ320からのリクエストは、ベンダーインターフェースモジュール335に送信される。ベンダーインターフェースモジュール335からの、取引ログマネージャ320が取得したデータは、ストレージシステムデータストア390に記憶される。一実施形態では、取引ログに関するリクエストは、取引ログに関する以前のリクエストの処理後に変化した、生産データベースシステム110内の取引ログのみを取得する。時点コピーマネージャ310が取得したデータベースブロックは、取引ログマネージャ320が取得した取引ログと組み合わせられて、種々の時点のコピー（以下、「時点コピー」という）が作成された種々の時のうちのいずれかである或る過去の時に対応する、生産システム110内のデータベースのコピーを再構成するために用いられ得る。

【0055】

ストレージ割当マネージャ 365 は、生産データベースシステム 110 から取得したデータを保存する機能を有する。例えば、時点コピーマネージャ 310 は、ストレージ割当マネージャの API を呼び出して、生産データベースシステム 110 から取得したデータブロックを保存できる。ストレージ割当マネージャ 365 は、生産データベースシステム 110 から取得し得るデータブロック各々の種々のバージョンを追跡する。所定の時点においてストレージ割当マネージャ 365 がリクエストされ、所定の時点前に取得したデータブロックの最新バージョンが提供され得る。ストレージ割当マネージャ 365 は、データブロックのコピーを構成するためにも用いられ得る。データブロックが読み出しのみを目的としてコピーされる場合には、ストレージ割当マネージャ 365 は、完了したデータブロックを参照するポインタを維持するのに十分なストレージのみを割り当てる。しかしながら、コピーされたデータブロックも書き込む場合には、ストレージ割当マネージャ 365 は、データブロックを実際にコピーするのに十分なストレージを割り当て、元のデータブロックの更新を回避する。

10

【0056】

ファイル共有マネージャ 370 は、ストレージシステムデータストア 390 が記憶したファイルを、データベースストレージシステム 100 と接続可能なネットワーク上のコンピュータ全体で共有できるようにする。ファイル共有マネージャ 370 は共有ファイル用のファイル共有システム 120 を利用する。共有ファイル用のシステムの例には、ネットワークファイルシステム (NFS) がある。共有ファイル用のシステムは、ファイバチャネルストレージエリアネットワーク (FC-SAN)、ネットワーク接続ストレージ (NAS)、またはそれらの組み合わせおよび変形例を利用できる。共有ファイル用のシステムは、小型コンピュータ用周辺機器インターフェース (SCSI) プロトコル、インターネット小型コンピュータ用周辺機器インターフェース (iSCSI) プロトコル、ファイバチャネルプロトコル、または他の類似のおよび関連するプロトコルに基づくものでもよい。一部の実施形態では、データベースストレージシステム 100 は論理ボリュームマネージャを利用できる。ファイル共有マネージャ 370 を用いてストレージシステムデータストア 390 に記憶されたファイルの共有により、仮想データベースシステム 130 などのリモートコンピュータが、共有ファイル内のデータにアクセスできる。リモートシステムはストレージシステムデータストア 390 が共有するファイルの読み出し、かつそれへの書き込みができる。一実施形態では、ファイルは、NTFS と呼ばれる WINDOWS (登録商標) オペレーティングシステムのファイルシステムまたは UNIX (登録商標) ファイルシステム (UFS) などの所定のファイルシステムディスクレイアウトを模倣するフォーマットにおいて組織される。

20

30

【0057】

仮想データベースマネージャ 375 は、仮想データベースシステム 130 用の仮想データベースを作成するためのリクエストを受信する。仮想データベースを作成するリクエストは、管理システム 140 を利用するデータベース管理者により送信され得、生産データベースシステム 11 と仮想データベースシステム 130 とを特定し、作成されるべき仮想データベースに対応する過去の時点を含む。仮想データベースマネージャ 375 は、作成される仮想データベースに対応する必要なファイルを作成し、そのファイルを仮想データベースシステム 130 と共有する。仮想データベースシステム 130 のデータベース管理者は、生産データベースシステム 110 のデータベース管理者とは別人でもよい。

40

【0058】

インターフェースマネージャ 330 は、管理システム 140 を用いて必要な情報を表示する。データベース管理ユーザは、ストレージシステムデータストア 390 に利用可能な情報を確認し、さらにはデータベースストレージシステムが実行可能な機能を実行できる。例えば、データベース管理者は、種々の生産データベースシステム 110 から取得した、ストレージシステムデータストア 390 が記憶した種々の生産データベースを確認できる。別の例として、データベース管理者は、特定の時点において生産データベースシステム 110 が記憶したデータベースの PIT コピーを生成するリクエストをデータベースス

50

トレージシステム 100 に送信できる。一実施形態では、インターフェースマネージャ 330 は、外部アプリケーションがデータベースストレージシステム 100 内の情報にアクセスできるようにする。例えば、データベースストレージシステムは、第三者ベンダーがデータベースストレージシステム 100 に基づいてアプリケーションを書き込み可能なアプリケーションプログラミングインターフェース (API) を提供できる。一実施形態では、インターフェースマネージャ 330 は、データベースストレージシステム 100 に利用可能な情報にウェブアプリケーションがアクセス可能なウェブサービスを提供する。例えば、データベースストレージシステムは、クラウドコンピュータ環境の一部でもよい。第三者ベンダーはウェブサービスを利用して、本明細書に記載する様々なワークフローシナリオなどの、VDB に基づく様々なワークフローシナリオを実施できる。これにより、VDB に基づくワークフローシナリオの自動化が可能になる。

10

【0059】

システム構成マネージャ 315 は、管理システム 140 を用いるデータベース管理者が、データベースストレージシステム 100 の構成を設定または変更できるようにする。例えば、データベースストレージシステムが初期またはその後に設定される場合に、システム構成マネージャ 315 は、データベース管理ユーザまたは代理人が、生産データベースシステム 110 およびそれと接続する仮想データベースシステム 130 を指定できるようにする。システム構成マネージャ 315 はさらに、生産データベースシステム 110 内のデータベースの PIT コピーを時点コピーマネージャ 310 が取得するスケジュールに加えて、生産データベースシステム 110 からのオンライン取引ログの更新を取引ログマネージャ 320 が取得する頻度および時間を指定するポリシーを適切な役割および権限を有するユーザが設定できるようにする。一実施形態では、スケジュールは、PIT およびログ取得機能を実行する 1 日のうちの頻度および時間を指定するか、同じ機能を実行する暦日を指定する定期スケジュールでもよい。

20

【0060】

一実施形態では、データベース管理者がポリシーを規定し得る。このポリシーは、生産データベースシステム 110 からの時点コピーのローディング、生産データベースシステム 110 からの取引ログのローディング、データベースの時点コピーおよび取引ログ情報を含む、データベースストレージシステム 100 からの情報の消去および仮想データベースシステムのプロビジョニングに関連する種々の操作のためにシステム構成マネージャ 315 に記憶される。ポリシーは特定の操作を実行するための規則を特定する。例えば、ポリシーは、既定のスケジュールに基づいて実行される操作を指定できる。ポリシーは、生産データベースに蓄積した PIT コピーの数に基づいて、データベースストレージシステム 100 が記憶した PIT コピーを消去するタイミングを決定できる。ポリシーは利用可能なストレージ量を測定して、情報を消去するタイミングを決定できる。例えば、利用可能なストレージ量が閾値未満である場合には、選択されたデータベースの古い PIT コピーを消去できる。ポリシーはさらに、情報を消去する前に、用いる生産データベースの優先度を指定でき、例えば、優先度の高いデータベースの情報を消去する前に、優先度の低いデータベースの情報を消去する。特定のワークフローシナリオでは、ポリシーは、生産データベースから新規の情報を取得するタイミング、VDB 情報を自動的に更新するタイミング、および新規の情報に基づいて更新された VDB をプロビジョニングするタイミングを決定できる。

30

40

【0061】

仮想データベースシステム 130 は、データベースサーバ 360 および VDB システムライブラリ 380 を含む。データベースサーバ 360 の機能はデータベースサーバ 345 と類似し、データベースサービスおよびアプリケーションプログラミングインターフェース (API) を提供するコンピュータプログラムであり、データストア 350 が記憶したデータを管理する。データベースサーバ 360 が管理するデータは、ファイル共有システム 120 を利用してデータベースストレージシステム 100 と共有される、ストレージシステムデータストア 390 に記憶され得る。VDB システムライブラリ 380 は、データ

50

ベースストレージシステム 100 が送信するリクエストを処理するためのプログラムコードを含む。代替的な構成では、仮想データベースシステム 130 は異なるおよび / または追加のモジュールを含んでもよい。

【0062】

図 4 は生産データベースシステム 110 内のデータベースが記憶したデータの時点コピーを実現するための、データベースストレージシステム 100 と生産データベースシステム 110 とのやりとりを示す図である。時点コピーマネージャ 310 は、生産データベースシステム 110 のデータベースに関連するデータを取得するためのリクエスト 405 を、生産データベースシステム 110 のベンダーインターフェースモジュール 335 に送信する。一実施形態では、リクエスト 405 は、2つのネットワーク装置間でデータを交換可能なセキュアシェルまたは SSH ネットワークプロトコルを利用して送信される。リクエスト 405 は管理システム 140 からのリクエストに回答して送信されてもよいし、定期スケジュール機能により送信されるように構成されてもよい。例えば、データベースストレージシステム 100 は、毎日所定の時間に生産データベースシステム 110 にリクエスト 405 を送信するように構成されてもよい。図 4 に示すシステム環境は、生産データベースシステム 480 において常に実行される、データベースストレージシステム 100 専用の処理を要求しない。このデータベースストレージシステム 100 に情報を送信するための専用の処理は、生産システムの重要な資源の所望しない消費を招く場合があるため、生産データベースシステム 480 にとって有益である。それ故、データベースストレージシステムは生産データベースシステム 480 から情報が必要なときは常に、リクエスト 405 および 450 を送信する。

【0063】

生産データベースシステム 480 は、リクエストされたデータを時点コピーマネージャ 310 に送信する。リクエスト 405 が、生産データベースシステム 110 が記憶したデータベースに関連するデータに関する最初のリクエストである場合には、生産データベースシステム 480 は、返答としてデータベース全体のデータを送信する。生産データベースシステム 480 はその後のリクエスト 405 に回答して、以前のリクエスト 405 に回答した最後の応答 430 の送信後に変化したデータベースブロックのデータのみを送信する。

【0064】

一実施形態では、ベンダーインターフェースモジュール 335 は、応答 430 に要求される情報を収集するためのリクエスト 410 をデータベースサーバ 345 に送信する。ベンダーインターフェースモジュール 335 はさらに、生産システムライブラリ 385 における利用可能なプログラムコードをロードし得る。データベースサーバは、必要なデータを要求するリクエスト 415 をデータストア 350 に送信し、その応答 420 であるリクエストしたデータを受け取る。データベースサーバ 345 は、リクエスト 410 に回答して、リクエストされたデータ 425 をベンダーインターフェースモジュール 335 に送信する。ベンダーインターフェースモジュール 335 は、データベースサーバから受け取ったデータ 425 を、時点コピーマネージャ 310 に処理され得るフォーマットにパッケージするため、生産システムライブラリ 385 に呼び出し 470 を送信する。生産システムライブラリ 385 は、時点コピーマネージャ 310 用に適切にフォーマットした、リクエストされたデータストリーム 430 を送信する。生産システムライブラリ 385 は、データベースサーバが送信した情報 425 を、時点コピーマネージャ 310 に送信 430 する。生産システムライブラリ 385 のプログラムコードにより動作するベンダーインターフェースモジュール 335 は、データベースストレージシステム 100 に処理されるデータストリームを構築する。

【0065】

他の実施形態では、生産システムライブラリ 385 と関係するベンダーインターフェースモジュール 335 が、要求されるデータをデータストア 350 から直接取得し、時点コピーマネージャ 310 にデータ 430 を送信してもよい。通常、必要な情報を抽出するた

10

20

30

40

50

めの適切なAPIをデータベースサーバ345がサポートする場合に、これらの実施形態は有益である。これらの実施形態では、生産システムライブラリ385は、データストア350が記憶したデータベースのファイル構造を解析するコード、さらにはデータストア350が記憶したデータベースブロックに関連するメタデータを処理するコードを含み、以前の時点から変化したデータベースブロックを検出する。

【0066】

応答430は、データストア350内の複数のファイルに記憶され得るデータベースブロックを含むデータストリームである。応答430に相当するデータストリームは、異なるデータベースブロックに関連する情報をインタリーブし得、例えば、異なるファイル間で取得されたデータベースブロックをインタリーブし得る。それ故、時点コピーマネージャ310のプログラムコードは、データストリーム内で受け取ったデータベースブロックの任意の特定の順序を仮定することなく、データストリームを処理する。これらのデータベースブロックは、異なるデータベースにも属し得る。

10

【0067】

図5は時点コピーマネージャ310における、生産データベースシステム110から受け取ったデータストリームの処理を説明するプロセスフローチャートを示す。時点コピーマネージャ310は、最後のPITコピー後に変化したブロックを含むデータストリームを受け取る510。時点コピーマネージャ310はデータストリームを処理し、データストリーム内のデータベースブロックを特定する515。各データベースブロックは、そのブロックが属するデータベースオブジェクト、データベースブロックのサイズ、データベースブロックが取得されたファイル、データベースブロックが記憶されたファイルにおけるオフセット、およびデータベースブロックが生産データベースシステム110のデータベース内で更新される順序を特定するログシーケンス番号などの、データベースブロックに関連する情報を含むメタデータを含む。

20

【0068】

時点コピーマネージャ310は、各データベースブロックに関するメタデータを解析して、ストレージシステムデータストア390にデータベースブロックを記憶する必要があるか、またはそれを排除すべきかを決定する520。例えば、データベースブロックのメタデータにおけるログシーケンス番号は、生産システムライブラリ385がデータストリームと共にデータベースブロック430を送信する場合でさえも、生産システムライブラリ385から受け取った最後の応答430からデータベースブロックが全く更新されていないことを示す。それ故、ブロックをストレージシステムデータストア390に記憶する必要はなく、それをスキップしてもよい。記憶する必要のないデータベースブロックの他の例は、一時データベースブロック、セッション特定データベースブロック、およびそこにデータが書き込まれていない空データベースブロックを含む。記憶する必要のないデータベースブロックの別の例は、意味を有さないか、データベースソフトウェアにアクセスできないデータベースブロックを含む。別の例は、データベースソフトウェアにより、削除、空または無効であるとマークされたデータベースブロックを含む。

30

【0069】

前述の実施形態では、生産データベースシステム480が送信した情報430は、データストリームがデータベースストレージシステム100により受け取られた後、排除された不要なブロックに含まれた。他の実施形態では、生産システムライブラリ385がデータストリームを構築する間、不要なブロックの一部または全てが排除されてもよい。この実施形態では、生産データベースシステム480がデータベースストレージシステム100に送信するデータストリーム430は、2つのシステム間の通信を効率的にするためにサイズが縮小される。

40

【0070】

記憶する必要のないデータベースブロックのスキップに加えて、記憶したデータベースブロックを圧縮することにより、データベースストレージシステムは、同一のデータベースに対応するデータに関する生産データベースシステムと比較して、データベースファイ

50

ルに要求されるストレージにおいて有意な節約を実現できる。例えば、ストレージシステムデータストア 390 内の生産データベースに対応するデータが占有するストレージスペースは、生産 DB データストア 350 内の生産データベースが占有するスペースの 4 分の 1 であり得る。生産データベースシステムに対応する全情報が、最初の P I T コピーにより取得されることに留意されたい。その後の P I T コピーは、生産 DB 内の变化した情報のみを取得し、最初の P I T コピーに含まれる情報よりもはるかに少なくなり得る。

【 0 0 7 1 】

時点コピーマネージャ 310 が、データストリーム内のデータベースブロックをスキップできることを決定した場合 525 には、時点コピーマネージャ 310 は、処理する次のデータベースブロックの特定 515 に進む。実施形態では、時点コピーマネージャ 310 は、ストリームメタデータに利用可能なサイズのデータベースブロックを用いて、データストリーム内のデータベースブロックの境界を特定する。各ブロックはその後、この境界に従い処理される。

10

【 0 0 7 2 】

時点コピーマネージャ 310 が、データストリーム内のデータベースブロックをデータストレージシステムデータストア 390 に記憶する必要があると決定した場合には、時点コピーマネージャ 310 はデータベースブロックメタデータを解析して、データベースブロックをデータベースファイル、およびファイル内の適切な位置にマッピングする 530。時点コピーマネージャ 310 は、ストレージ割当マネージャ 365 にリクエスト 435 を送信して、データベースブロックを保存する 535。ストレージ割当マネージャ 365 は、データベースブロックに関連する適切なファイルをストレージシステムデータストア 390 に記憶 440 する。時点コピーマネージャ 310 は、データストリームが完全に処理されたか否かを確認する 540。データストリームに未処理データが残っている場合には、時点コピーマネージャ 310 は、処理のための次のデータブロックの特定に進む。

20

【 0 0 7 3 】

ストレージ割当マネージャ 365 は、それが異なる時点において更新された場合には、データベースブロックのデータに対応する、ストレージシステムデータストア 390 内のデータベースブロックのいくつかの異なるバージョンを維持できる。データベースブロックを保存したファイルは、ファイルに関連するメタデータおよびデータベースブロックのシーケンスを含むファイルヘッダを有する。各ベンダー固有データベースサーバ 345 は、それ自体が処理可能なファイルセットとしてデータベース情報を組織化する。ファイルセットを用いた情報のデータベースのための組織化はベンダー固有でもよく、データベースストレージシステムは、ファイルのベンダー固有組織内のデータベース情報を組織化するためのプログラム論理を組み込む。時点コピーマネージャ 310 は、データストア 350 内のデータベースのファイルセットに類似し得るファイル構造セットを作成する。ただし、ストレージシステムデータストア 390 内の情報は、異なる時点における更新情報にその各々が対応する、データベースブロックの複数のバージョンを含み得る。一実施形態では、ストレージ割当マネージャ 365 は、データベースブロックが時点間において更新された場合にのみ、データベースブロックをコピーするような効率的な方法により、ファイルに関連するデータベースブロックを記憶する。例えば、ブロック B1 が時間 T1 で更新されるが、時間 T2 では更新されず、一方、ブロック B2 が時間 T1 および T2 の両方で更新される場合には、ストレージシステムデータストア 390 のデータ構造は、時間 T2 に対するデータベースブロック B1 のコピーを保持しないが、時間 T2 に対するデータベースブロック B2 のバージョンを保持する。

30

40

【 0 0 7 4 】

図 4 はさらに、取引ログマネージャ 320 と生産システムライブラリ 385 とのやりとりを示す。取引ログマネージャ 320 は、生産データベースシステム 110 のデータベース内の取引ログにおける、以前の時点からの増分変化を取得する。一実施形態では、リクエスト 445 はセキュアシェルまたは SSH ネットワークプロトコルを利用して送信される。リクエスト 445 は情報が要求されるデータベースを特定し、取引ログ情報を以前に

50

受け取った時点に対応する時間値を与え得る。生産システムライブラリ 385 は、リクエスト 445 に応答して、リクエストされた情報 450 を取引ログマネージャ 320 に送信する。ベンダーインターフェースモジュール 335 は、前述のようにデータベースサーバ 345 API を呼び出すか、データストア 350 と直接通信することにより、リクエストされた情報を取得できる。生産データベースシステム 110 から取得したデータベースログの増分変化はマネージャ 320 が保存し、これは、ストレージシステムデータストア 390 に情報を記憶 440 するストレージ割当マネージャ 365 にリクエスト 460 を送信することによって行われる。

【0075】

図 6 は生産データベースシステム 110 からデータベースストレージシステム 100 に取引ログファイルをコピーする処理を示す。取引ログマネージャ 320 は、前回の更新の受け取り後に更新された取引ログを取得するためのリクエストを生産データベースシステム 110 に送信する 600。取引ログマネージャ 320 は、生産データベースシステム 110 からの応答をデータストリームとして受け取る 610。取引ログマネージャ 320 は受け取ったデータストリームを解析して、取引ログデータを書き込むログファイルを決定する 620。データストリーム内の受け取ったデータを複数のログファイルに書き込む必要がある可能性がある。取引ログマネージャ 320 は、データストリームからのオンライン取引ログデータを適切なログファイルに書き込む 630。

【0076】

一実施形態では、取引ログマネージャ 320 は一回のログファイルを更新と次の更新間、所定時間待機し 640、取引ログの更新に関する新規の更新があるか否かを確認する次のリクエストを、生産データベースシステム 110 に送信する 650。この時間間隔中に生産データベースが更新されなかった場合には、生産データベースシステム 110 は、その旨を取引ログマネージャ 320 に通知する。この時間間隔中に利用可能な取引ログが新規に更新されなかった場合には、取引ログマネージャ 320 はもう一度その時間間隔待機する 640。取引ログの更新を示す生産データベースシステム 110 からの応答が得られた場合には、取引ログマネージャ 320 は、取引ログの次の更新を取得するための次のリクエストを生産データベースシステム 110 に送信する 600。

【0077】

取引ログの増分変化は、時点コピーマネージャ 310 が実行する時点コピーよりもはるかに頻繁に取引ログマネージャ 320 が取得できる。例えば、時点コピーマネージャは、生産データベースシステム 110 が記憶した時点データベースを 1 日に一度コピーし得、一方、取引ログマネージャ 320 は、取引ログの増分変化を 5 分ごとに取得し得る。取引ログの増分変化を高い頻度で取得することにより、時点コピーマネージャ 310 が時点コピーを実行する時間である時点間で、生産データベースシステム 110 からデータベースのコピーを再作成する能力が得られる。

【0078】

生産データベースシステム 110 は、取引ログファイルを循環方式で再利用し得、これにより、以前のログファイルを上書きできる。ただし、データベースストレージシステム 100 は異なるログファイルへの書き込みを開始するために、データが現在書き込まれているログファイルを閉じることを決定する度に新規のログファイルを作成する。図 7 は生産データベースシステム 110 のログファイルと、データベースストレージシステム 100 のログファイルとを比較する図である。生産データベースシステムのログファイル 710 は、オンライン取引ログファイルに相当する。通常、オンライン取引ログの記憶用に限られた数のファイルが割り当てられる。例えば、図 7 には、オンライン取引ログの記憶用に、生産データベースシステム 110 には、3 つのファイル 710 (a)、710 (b) および 710 (c) が割り当てられる。

【0079】

図 7 に示す矢印 730 は、所定の時間 T_i (時間 T_1 、 T_2 および T_3 は単調に増加すると仮定する) において、生産データベースシステム 110 に取引ログが書き込まれた取

10

20

30

40

50

引ログファイルの変化を示す。例えば、時間 T 1 において、生産データベースシステム 110 はファイル 710 (a) への取引ログの書き込みを停止し、ファイル 710 (b) への取引ログの書き込みを開始した。同様に、時間 T 2 において、生産データベースシステム 110 はファイル 710 (b) への取引ログの書き込みを停止し、ファイル 710 (c) への取引ログの書き込みを開始した。時間 T 3 において、生産データベースシステム 110 はファイル 710 (c) への取引ログの書き込みを停止し、取引ログファイル 710 (a) を再利用することを決定した。取引ログファイルを再利用する前に、生産データベースシステム 110 は、その取引ログファイルにある取引ログが確実に適切なデータベースに適用されるようにする。時間 T 4、T 5 および T 6 におけるログファイルの変化は、前述の変化に類似する。それ故、生産データベースシステムは、通常、循環方式において取引ログファイルを再利用し、ストレージを再利用することができる。

10

【0080】

データベースストレージシステムは、取引ログに基づいてログ保持ポリシーが決定した非常に長い時間履歴情報を保持するため、ログファイルデータに関しては循環再利用方式を利用しない。取引ログに基づく履歴情報を保持することにより、過去の時点における VDB を作成する能力が得られる。過去の時点に対応するデータベーススナップショットの復旧に必要な取引ログが利用可能である限り、過去の時点における VDB を作成できる。取引ログファイルの循環再利用に基づく方式は、時間的に前の取引ログを上書きする。それ故、ログファイルに循環再利用方式を利用するデータベースシステムは、取引ログが上書きされていない最近の時点に関する取引ログに基づいてデータベーススナップショットのみを復旧できる。

20

【0081】

データベースストレージシステム 100 が記憶したログファイル 720 は、保持されたログファイルである。矢印 740 は、生産データベースシステム 110 の取引ログファイル 710 から、データベースストレージシステム 100 の保持されたログファイル 720 への情報の転送を示す。各矢印 740 は、取引ログマネージャ 320 から生産データベースシステム 110 に送信されたリクエスト 445 のいくつか、および取引ログマネージャ 320 が処理し、記憶した、生産データベースシステム 110 から送信された応答 450 のいくつかに対応し得る。

【0082】

30

例えば、矢印 740 (a) は、時間間隔 T 1 から T 2 間のログファイル 710 (a) から 720 (a) への情報のコピーを示す。時間 T 2 において、生産データベースシステムは、ファイル 710 (b) への取引ログの書き込みを開始した。データベースストレージシステムは新規のログファイル 720 (b) を作成し、矢印 740 (b) は、ファイル 710 (b) からログファイル 720 (b) への取引ログ情報の転送を示す。前述の処理が継続するが、時間 T 3 において、生産データベースシステムがログファイル 710 (a) の再利用を開始するが、データベースストレージシステムは新規のログファイル 720 (d) を作成する。矢印 740 (d) はログファイル 720 (d) への取引ログ情報のコピーを示す。それ故、生産データベースシステム 110 の同一の取引ログファイルからの取引ログ情報は、異なる時間においてデータベースストレージシステム 100 内の複数のログファイルにコピーされ得る。例えば、取引ログファイル 710 (a) 内の情報は、T 0 と T 1 との間においてログファイル 720 (a) にコピーされ、T 3 と T 4 との間においてログファイル 720 (d) にコピーされ、時間 T 6 と T 7 との間においてログファイル 720 (g) にコピーされる。データベースストレージシステム 100 は、ログ保持ポリシーが決定するような可能な限り長い時間取引ログ情報を保持するために、ログファイルの再利用を回避する。これによりユーザは、取引ログ情報が利用可能な以前の時点においてデータベースのスナップショットを再作成できる。

40

【0083】

図 8 はストレージシステムデータストア 390 が記憶した、異なる時点における種々の生産データベースシステム 110 から取得した情報を示す。図 8 は生産データベースシス

50

テム 110 から取得した、2つのデータベース、すなわち DB1 および DB2 に関連する情報を示す。情報 850 はデータベース DB1 が取得したデータに相当し、情報 860 はデータベース DB2 が取得したデータに相当する。情報 850 または 860 はデータベースブロックセットおよび取引ログセットを含む。情報 850 (a) は、生産データベースシステム 110 から取得した、データベース DB1 の最初の PIT コピーに相当する。情報 850 (b) は、最初の PIT コピー後にデータベース DB1 の最初の取引ログ更新に相当し、情報 850 (c) は、最初の PIT コピー後のデータベース DB1 の第 2 の取引ログ更新に相当する。情報 850 (d) はデータベース DB1 の第 2 の PIT コピーに相当する。情報 850 (d) は、最初の PIT のコピーが生成された後、データベース DB1 内で変化したデータベースブロックのみを記憶する。情報 850 (e) は、第 2 の PIT コピー後のデータベース DB1 の第 1 の取引ログ更新に相当する。同様に、情報 860 はデータベース DB2 に相当する。情報 850 に付随して示す時間 T_i は、情報がその構成内でコピーされた時間を示す。PIT コピーマネージャ 310 が実行する PIT コピー（ログが更新されていない 850 (a) または 850 (d) など）においては、時間 T_i は、PIT のコピー前にデータベースブロックを最後に更新した時間を示す。850 (b)、850 (c) または 850 (e) などのログ更新に対応する情報においては、時間 T_i は、記憶された対応する取引ログセットにおける最後の取引ログの時間を示す。

【0084】

図 8 に示す矢印 810 は、仮想データベースマネージャ 375 が実行する、情報 850 に基づくデータベースの読み出し / 書き込みコピーに相当するファイルを作成するステップを示す。矢印 830 は、ファイル共有システム 120 を通じて、仮想データベースシステム 130 に対してファイル 870 を利用可能にするステップを示す。図 9 は仮想データベースを作成する処理のフローチャートである。仮想データベースマネージャ 375 は、仮想データベースシステム 130 に用いる仮想データベースを作成するリクエストを受信する 905。VDB を作成するリクエストは、管理システム 140 が受信し得る。VDB を作成するリクエストは、生産データベースシステム 110 および VDB として利用可能とされる必要のある対応するデータベース、VDB を作成する必要のある仮想データベースシステム 130、並びにデータベーススナップショットが VDB として作成される過去の時点 T_n の詳細を含み得る。

【0085】

仮想データベースマネージャ 375 は、 T_n よりも以前の T_j に関する最近の PIT コピーを特定する 910。仮想データベースマネージャ 375 は、 T_j から T_n の期間に更新されたログファイルの一部をさらに特定する 915。特定した PIT コピー内のデータベースブロックおよびログファイルの適切な一部をストレージ効率よくコピーすることにより、読み出し / 書き込みファイル構造 870 を作成する 920。PIT コピーに基づいて作成された VDB に適切な取引ログを適用することにより、PIT コピー後の時点間のソースデータベースのスナップショットを作成できる。したがって、PIT コピーは定期的に、例えば毎日作成され得るが、取引ログを以前の PIT コピーに適切に適用することにより、PIT コピー間の任意の時点における VDB を作成できる。例えば、特定の日付の真夜中に生産データベースにおいて PIT コピーが生成された場合がある。しかしながら、VDB は、午前 10 時 25 分などの 1 日のうちの特定のそれ以降の時間に、その特定の時間に PIT がコピーされていないが、生産データベースの状態に基づいて作成され得る。真夜中からその特定の時間までの生産データベースの変化は、取引ログから取得される。

【0086】

ファイル構造のストレージ効率のよいコピーを実現する機構を本明細書にさらに記載する。仮想データベースマネージャ 375 は、読み出し / 書き込みファイル構造の操作情報を、関連する仮想データベースシステム 130 に送信する 935 (図 8 の矢印 830 に示す)。一部の実施形態では、仮想データベースマネージャ 375 は、リクエストをファイル共有マネージャ 370 に送信することにより、ファイル構造を仮想データベースシステ

ム 1 3 0 に対して利用可にする。ファイル共有マネージャ 3 7 0 はこれにตอบสนองして、ファイル共有システム 1 2 0 を利用して仮想データベースシステム 1 3 0 と適切なファイルを共有する。仮想データベースマネージャ 3 7 5 はまた、適切な保持したログをデータベースブロックに適用することにより、新規の仮想データベースの復旧を実行するリクエストを仮想データベースシステム 1 3 0 に送信する 9 3 0。一部の実施形態では、データベースサーバが仮想データベースシステム 1 3 0 を起動する際に、データベースはその復旧を自動的に実行する。

【 0 0 8 7 】

図 1 0 は V D B に対応する読み出し / 書き込みファイル構造を作成するための、ストレージ効率のよいコピーの作成方法を示す。図 1 0 に示す構造 1 0 1 0 は、生産データベースシステム 1 1 0 のデータベースに対応するファイルを示す。構造 F i および G i は、それぞれ、ファイル 1 0 1 0 に記憶されるデータベースブロックを示す (F i は F 1、F 2、F 3 などを意味し、同様に、G i は G 1、G 2、G 3 などを意味する)。矢印 1 0 1 5 は種々の時点 T i における P I T コピー作成の処理を示す。時間 T 0 において作成される最初の P I T コピー 1 0 3 0 には、データベースの全てのデータベースブロックのコピーが必要となる。例えば、F 1 i はブロック F i のコピーを示し、G 1 i はブロック G i のコピーを示す。時間 T 1 において作成される P I T コピー 1 0 3 5 は、前回の P I T コピー後に変化したブロックのみをコピーし、最初の P I T コピーよりもはるかに少ないデータのコピーで済む。同様に、時間 T 2 において別の P I T コピー 1 0 4 0 が、以前の P I T コピー 1 0 3 5 後に変化したデータベースブロックのみをコピーすることにより作成される。

【 0 0 8 8 】

P I T コピー 1 0 4 0 が図 1 0 に示す構成における最後の P I T コピーであると仮定すると、時点 T 2 において V D B ファイル構造 1 0 5 0 が作成される。構造 1 0 5 0 が作成されると、データを記憶する実際のデータベースブロックを示すポインタとしてブロック V 1 1 ~ V 2 5 が実装され得る。例えば、V 1 1 はブロック F 1 の情報を示し、ブロック F 1 が時間 T 1 および T 2 におけるコピー中に更新されなかったため、F 1 1 を指す。V 1 2 はブロック F 2 の情報を示し、時間 T 1 において F 2 が更新されたため、ブロック F 2 2 を示す。同様に、V 1 3 は時間 T 2 において更新されたブロック F 3 に対応し、ブロック F 3 3 を指す。

【 0 0 8 9 】

図 1 1 は時点 T 1 において作成されたファイル構造 1 1 5 0 を示す。ブロック F 3 に対応する U 1 3 は、時点 T 1 においてブロック F 3 が更新されなかったため、F 1 3 を指すことに留意されたい。また、U 1 4 は時間 T 1 においてコピーされたブロック F 4 に対応するブロック F 2 4 を指す。P I T コピー 1 0 4 0 が時点 T 1 後に作成されたので、構造 1 1 5 0 のいずれも P I T コピー 1 0 4 0 を指さない。

【 0 0 9 0 】

図 1 2 は、ストレージシステムデータストア 3 9 0 が記憶したデータベースブロックが、異なる V D B において作成されたファイル構造と共有される方法を示す。図 1 2 は図 1 0 および図 1 1 に示すような、V D B 用に作成された生産データベースシステム 1 1 0 のファイル 1 0 0 5 に対応するファイル構造を示す。図 1 2 に示すように、ファイル構造 C 5 0 のブロック V 1 3 および V 1 4 は、時間 T 1 において V D B ファイル 1 1 5 0 と共有されないブロック F 3 3 および F 3 4 である最新コピーを指す。しかしながら、T 2 における V D B ファイル 1 0 5 0 のブロック V 1 1 は、T 1 における V D B ファイル 1 1 5 0 のブロック U 1 1 とブロック F 1 1 を共有する。同様に、1 0 5 0 のブロック V 1 2 は、1 1 5 0 のブロック U 1 2 とデータベースブロック F 2 2 を共有する。複数の V D B に渡るブロックの共有利用により、ストレージシステムデータストア 3 9 0 が記憶したデータを効率的に利用できる。V D B の 1 つが共有のデータベースブロックへの書き込みを試みる場合には、書き込みを試みる V D B に共有のデータベースブロックのコピーが作成される。データベースブロックを共有していた残りの V D B は、元のデータベースブロックを

共有し続ける。それ故、コピーされたデータベースブロックにおけるいかなる変化も、データベースブロックに書き込んでいるVDBに固有であるため、残りのVDBには認識されない。

【0091】

VDBはソースとして別のVDBの時点コピーを用いても作成され得る。例えば、仮のVDB1が作成され、仮想データベースシステム130にプロビジョニングされると仮定する。仮想データベースシステム130が最初にデータベースブロックを書き込むときには、そのVDBに関連するデータベースブロックがコピーされる。VDB1の時点コピーも所定のスケジュールに基づいて作成される。これによりユーザは、VDB1の時点コピーに基づいて第2の仮想データベースVDB2を作成できる。VDB1の取引ログも記憶することにより、ユーザは、VDB1の時点コピー間であり得るVDB1の任意の以前の状態に基づいて第2の仮想データベースVDB2を作成できる。

10

【0092】

図13は更に、T2以前の時点T1+t2に対するデータベーススナップショットに対応する、VDBファイル構造1350内へのログファイルの組み込みを示す。図13に示すように、ログファイルデータL1は時間T1+t1において取引ログマネージャ320によりコピーされ、ログファイルデータL2は時間T1+t2においてコピーされる。生産データベースシステム110に書き込まれる追加のログデータL3はコピーされたように示されておらず、T1+t2後の時間に、データベースストレージシステムにコピーされ得る。VDB用に作成したファイル構造1350は、L1およびL2として示される、時間T1とT1+t2との間にコピーされたログ情報に相当する適切なログファイルデータを指す構造VL11を含む。仮想データベースシステム130においてデータベースサーバが起動すると、構造V11が指すログが、データベース復旧処理を利用してデータベースブロック1035に適用され得る。

20

【0093】

図10に示す構造1050、図11に示す構造1150、または図13に示す構造1350が読み出し/書き込み構造であるため、仮想データベースシステム130はこれらの構造からの読み出し、そしてそれへの書き込みが可能になる。仮想データベースシステム130がブロックVijに書き込むと、データベースブロック用のスペースが割り当てられ、割り当てられたスペースに対応するデータベースブロックのデータがコピーされる。例えば、仮想データベースシステム130がブロックV11に書き込む場合には、スペースが割り当てられ、割り当てられたブロックにブロックF11がコピーされる。それ故、ブロックF11の元のコピーが読み出し専用コピーとして維持され、仮想データベースシステム130は、それ自体のために特定の作成された適切なデータベースブロックのコピーを書き込みことができる。これは、対応する仮想データベースシステム130がデータベースブロックに書き込む場合のみ、データベースブロックのコピーを作成する怠惰なメカニズムとみなされ得る。仮想データベースシステム130が書き込むブロックの数、VDBのブロック総数の小さな一部分であり得るので、前述の構造はストレージ効率の高い方法でVDBに関連するデータを記憶する。仮想データベースシステム130によって書き込まれないデータベースブロックは、特定の仮想データベースシステム130のためにコピーをせず、いくつかの仮想データベースシステム間で共有され得る。

30

40

【0094】

VDB操作

図14はデータベースまたはデータベースに依存するアプリケーションを変化させる、ワークフロー内のデータベースのライフサイクルの一例を示す図である。図14に示すように、生産データベース1405のコピーは、開発、調整、検証、品質保証、認証、訓練、および段階分けを含むいくつかの目的のために作成される。従来の方法による多量のデータベースのコピーを作成することは、遅い過程であり得る。さらに、種々の機械におけるデータベースの種々のコピーの稼働は、ハードウェアの非効率的な使用に繋がる。データベースの物理コピーを作成する代わりに仮想データベースを作成することにより、デー

50

データベースに関連する様々なワークフローシナリオは簡易化され、高効率となり得る。複数の仮想データベースがデータベースストレージシステム 100 に記憶され得、システムの資源の利用が効率的になる。

【0095】

VDBに基づくワークフローシナリオにおいて実行されるステップは、従来のシステムを用いた同一のワークフローシナリオにおいて実行される操作とは顕著に異なり得る。これらのステップは、データベースストレージシステム 100 のデータベース管理者が実行してもよいし、スクリプトを利用して自動的に実行されてもよい。仮想データベースに関連する種々の操作を下に説明する。

【0096】

リンク操作は、生産データベースシステム 110 内のデータベースへのアクセスに必要な情報を、データベースストレージシステム 100 のシステム構成マネージャ 315 に提供する操作である。データベースへのアクセスに必要な情報を用いることにより、データベースストレージシステム 100 は、生産データベースシステム 110 からデータを取得できる。この情報は、データベース名、データベースをホストする生産データベースシステム 110 のネットワークアドレス、およびアクセス制御情報を含み得る。リンク操作の一部として、データベースストレージシステムは生産データベースシステム 110 と通信して、データベース情報を認証できる。データベースストレージシステム 100 は、生産データベースシステム 110 内のリンクしたデータベースからデータベースブロックを取得し、それをストレージシステムデータストア 390 に記憶できる。ストレージシステムデータストア 390 に記憶されたデータベースブロックは、仮想データベースの作成に用いることができる。一部の実施形態では、リンク操作は、ソースデータベース全体ではなく、ソースデータベースの一部のみがコピーされる必要があることを指定できる。例えば、リレーショナルデータベースでは、ソースデータベースの一部はテーブルスペース、1 つ以上のテーブルセット、テーブルのサブセット、またはテーブルサブセットのセットをであり得る。一実施形態では、ユーザはデータベースの一部をコンピュータ処理するためのスクリプトを指定できる。

【0097】

ロード操作は、データベースストレージシステム 100 内のストレージに用いられるデータを、生産データベースシステム 110 のデータベースから取得する操作である。データベースは、ロードできるようにするには、データベースストレージシステム 100 にリンクされる必要がある。ロード操作がデータベースのデータを最初に取得するときは、データベースに利用可能なデータ全体を取得する。この結果、最初のロード操作は遅くなり得、データベースのサイズおよび当業界のハードウェア状態に基づくネットワーク回線容量次第では、数時間または数日間かかる可能性もある。その後のロード操作は、データベース内における以前のロード操作からの変化のみを取得するため、はるかに短時間で済む場合もある。ロード操作は定期的に実行され、継続してデータベースの変化を取得する。ロード操作は、データベースのデータベースブロックおよび/または以前の時点後に更新されたデータベースを示す取引ログを取得できる。ロード操作に要求される入力、前もってリンクされているデータベースを特定する情報を含む。ソースデータベースの一部のみがリンク操作により特定されると、その部分のみがロードされる。

【0098】

ロード操作はまた、VDBに利用可能な情報を増加的に更新できる。データベースストレージシステム 100 が生産データベースシステム 110 から取得した情報は、定期的に更新され得る。データベースストレージシステムに利用可能な、生産データベースシステム 110 から取得した情報が更新されると、仮想データベースシステム 130 にプロビジョニングされた情報も更新され得る。仮想データベースシステム 130 がVDB内のデータを更新してもよい。この場合には、増大したロードは仮想データベースシステム 130 による更新を特定し、それと生産データベースシステム 110 から取得した変化とを比較する。2つの更新セットに不一致が無い場合には、生産データベースシステム 110 の変

10

20

30

40

50

化をVDBに適用することにより、ロード操作は成功する。不一致がある場合には、不一致のレポートがデータベース管理者に示され、不一致を解決するために、データベース管理者に入力が必要される。一実施形態では、2つのソース間の更新の不一致は、2つの更新による影響を受けたデータベースブロックを特定することにより検出される。データベースブロックの2つの更新セット間にオーバーラップが無い場合には、データベースストレージシステム100は不一致が無いと判断する。ソースデータベースの一部のみがリンク操作において特定された場合には、その一部の変化のみがロードされる。

【0099】

プロビジョニング操作は、データベースストレージシステム100内の仮想データベースを作成し、それを仮想データベースシステム130に対して利用可能にする。仮想データベースは、ソースデータベースの時点コピー、または別の仮想データベースの時点コピーに基づいて作成され得る。VDB用の1つ以上の読み出し/書き込みファイルが作成され、ファイル共有システム120を利用して仮想データベースシステム130と共有され得る。読み出し/書き込みファイルは、ストレージシステムデータストア390が記憶したデータベースブロックを示す構造を含む。プロビジョニング操作に要求される入力は、以前にリンクされかつロードされたデータベースまたは既存のVDBを特定する情報、データベースの所望の状態に対応する以前の時点、仮想データベースがプロビジョニングされる仮想データベースシステム130を特定する情報を含む。一実施形態では、VDBの一部がプロビジョニングされてもよい。同様に、種々のVDBの一部が共にプロビジョニングされ、新規のVDBを形成してもよい。他の実施形態では、アプリケーション特有の連携方式を用いて、いくつかのVDBが一群として共にプロビジョニングされてもよい。これらの一群に適したプロビジョニングは、アプリケーション論理もしくはプロビジョニングの設定、またはその連携を含み得る。

【0100】

ブックマーク操作は、1つ以上の仮想データベースにおける、アプリケーションとして重要な時点をマークする。結果生じた「ブックマーク」は、プロビジョニング操作を導くために用いられ得る。この操作は、通常、ユーザまたは管理システム140を介して外部プログラムによりトリガされる。データベースストレージシステム100は、結果生じた「ブックマーク」をデータベースストレージシステム100が記憶すると、トークンを返す。その後、ユーザまたは外部プログラムはVDBまたはVDBの群を、返されたトークンを利用して同一のアプリケーションにとって重要な時点にプロビジョニングできる。例えば、外部プログラムは、大量のバッチ処理動作の直後などの特定状態における生産データベースの捕捉を所望する場合がある。ユーザは管理システム140を通じてブックマーク操作を呼び出して、返送されたトークンを保存できる。その後、ユーザは保存したトークンを供給することにより、VDBを同一の状態にプロビジョニングできる。一実施形態では、トークンは文字列の形態でもよい。

【0101】

リフレッシュ操作は、ソースデータベースシステム110からの最新情報に基づいてVDBを定期的に更新するデータベースストレージシステム100に対応する操作である。例えば、VDBはユーザが確認するレポートを生成するレポートティングシステムにおいて用いられ得る。リフレッシュ操作は、1日1回などの定期的に、生産データベースシステム110から最新情報を自動的にロードする。リフレッシュ中のVDBは停止される。VDBは生産データベースシステム110の最新の時点コピーと共に更新され、再起動される。それ故、対応する仮想データベースシステム130のユーザは、生産データベースシステム110内のデータの最新の時点コピーに基づいた最新レポートを確認できる。一実施形態では、生産データベースシステム110から取得した時点コピー間において取得した取引ログに基づいてVDBをリフレッシュしてもよい。リフレッシュ操作に要求される入力は、リフレッシュされるべきVDB、およびデータをリフレッシュするスケジュールを特定する情報を含む。

【0102】

10

20

30

40

50

スクリプト前操作は、別のデータベースストレージシステム 100 の操作の実行前に、特定タスクを実行する専用命令の実行に相当する操作である。例えば、スクリプト前操作は、VDB をプロビジョニングするか、生産データベースサーバ 110 からデータベースをロードする前に実行され得る。データベースは、データベース外部に記憶された特定用途向けデータを要求するアプリケーションと共に用いられ得る。データベースがリフレッシュまたはロードされると、スクリプト前操作が実行され、特定用途向けデータがデータベースストレージシステム 100 にロードされ得る。スクリプト前操作に関する入力、実行される操作、およびスクリプト前操作がその前に実行される、データベースストレージシステム 100 の操作の詳細を指定する実行可能スクリプトを含み得る。

【0103】

スクリプト後操作は、データベースストレージシステム 100 の操作の実行後に、特定タスクを実行する専用命令の実行に対応する操作である。例えば、スクリプト後操作は、VDB を仮想データベースシステム 130 にプロビジョニングした後に実行され得る。生産データベースシステム 110 内のデータベースを用いたアプリケーションの検証および開発は、検証または開発仮想データベースシステム 130 を利用した類似のアプリケーションを実行することにより行われ得る。このシナリオでは、スクリプト前操作により生産データベースサーバ 110 からコピーされた特定用途向けデータはさらに、対応するアプリケーションを実行する仮想データベースシステム 130 にもコピーされる。データベースストレージシステム 100 から仮想データベースシステム 130 に特定用途向けデータをコピーする命令は、プロビジョニング操作後のスクリプト後操作として実行される。スクリプト後操作における入力は、実行される操作、およびスクリプト後操作がその後に実行されるデータベースストレージシステム 100 の操作の詳細を指定する実行可能スクリプトを含む。

【0104】

プレスクリプトおよびスクリプト後操作は、種々の VDB 操作に関連し得る。例えば、スクリプト前操作はリフレッシュ操作の前に実行されてもよいし、対応するスクリプト後操作はリフレッシュ操作後に実行されてもよく、リフレッシュ操作前/後における特有情報のコピー/インストールが可能になる。同様に、スクリプト前/スクリプト後操作は、とりわけリンク、ロード、プロビジョン、およびエクスポートを含む他の VDB 操作に関連し得る。例えば、ソースデータベースからのデータのリンク操作またはローディング中に、スクリプト前/スクリプト後操作により、データベーステーブルの列または行を含むデータを圧縮、マスキングまたは削除することによりデータをそぎ落とすことができる。前スクリプトおよび後スクリプトは、ソースデータベースおよび/または VDB を利用したアプリケーションに関連するアプリケーションデータを処理できる。前スクリプトおよび後スクリプトは、VDB のプロビジョニングに関連するシステム環境の課題を管理でき、VDB のプロビジョニング前/後の始動/停止機能を実行できる。

【0105】

共有操作は、別のユーザが VDB にアクセスするための許可に対応する操作である。一実施形態では、共有操作は、新規の VDB を作成し、それを新規のユーザまたはユーザセットと共有するためにプロビジョニングするステップを含み得る。例えば、検証および開発環境では、VDB を利用した開発が特定の段階に達した後に、VDB はテストユーザ内で共有され得る。共有操作に要求される入力は、共有する VDB の情報、VDB を共有するユーザを特定する情報、およびユーザに与えられた許可レベルを特定するアクセス制御情報を含み得る。

【0106】

エクスポート操作は、データベースに利用可能な情報を、あるコンピュータから別のコンピュータにコピーする操作である。通常、情報はデータベースとして組み立てるためにターゲットコンピュータにコピーされる。段階分け操作は、データベース情報を段階分けサーバにコピーするエクスポート操作に対応する操作である。段階分けサーバは、通常、生産環境におけるデータベースまたはデータベースアプリケーションへの変更を利用する

10

20

30

40

50

前に、データベースのシステムレベルを検証するために用いられる。エクスポート操作における入力、エクスポートされるVDBを特定する情報、およびVDBからデータがエクスポートされるターゲットマシンを特定する情報を含む。

【0107】

マスク操作は、データベース内の情報をコピーするときに、データベースの特定の情報を変更またはスキップする操作である。例えば、データベースをコピーするときに、ソース内の秘密情報はターゲットにはコピーされ得ない。別の例では、データベースをプロビジョニングするときに、データは暗号化される。秘密情報の例はクレジットカード情報または社会保障番号を含む。データベース情報が隠されたシナリオの例は、テスト目的の生産データベースのコピーを含む。VDBを用いて検証するデータベースのユーザは、生産データベースシステム110に記憶されている秘密情報を必要としない場合がある。ソースデータベースからコピーするデータを変換可能な他の操作は、圧縮および暗号化を含む。圧縮操作は、元の情報を保存するが、記憶時により小さいスペースを占有するようにデータのフォーマットを変換する操作である。暗号化操作は、暗号化情報をデコードする論理を有さないアプリケーションが読み出せないフォーマットにデータを変換する操作である。マスク、圧縮、または暗号化操作に対する入力、ソースVDBおよびターゲットデータベースを特定する情報を含む。ターゲットデータベースはそれ自体をVDBとしてもよいし、データを従来のシステムにエクスポートしてもよい。

10

【0108】

消去操作は、VDBから不要な情報を削除する。通常、膨大なスペースを占有し、もはや必要としない情報を消去する。例えば、データベースは、長期間システム内で発生しているイベントに関連するイベントデータを記憶し得る。もはや必要としない古いデータ、またはアーカイブしたデータはデータベースから消去し得る。データベース情報がコピー操作から削除されるべき情報をスキップすることによりコピーされるときに、消去操作が実行される。消去操作における入力、ソースVDBおよびターゲットデータベースを特定する情報を含み得る。ターゲットデータベースはそれ自体をVDBとしてもよいし、従来のデータベースでもよい。

20

【0109】

抽出、変換、およびロード(ETL)操作は、データ保管プロジェクトにおいて実行される標準的な操作を指す。抽出ステップはソースからデータを取得し、変換ステップは操作上の特定の要求に基づいてデータを修正し、ロード操作はデータをターゲットシステムにロードする。ETL操作に要求される入力、ソースデータベースを特定する情報、ターゲットデータベースを特定する情報、およびデータ変換に実行される操作情報を含む。ETL操作における入力、ソースVDBおよびターゲットデータベースを特定する情報を含んでもよい。ターゲットデータベースはそれ自体をVDBとしてもよいし、従来のデータベースでもよい。

30

【0110】

複製操作は、ソースストレージシステムが記憶したデータの変化をターゲットストレージシステムに伝える。複製されるデータはVDB、または1つ以上の生産データベースシステム110から取得したデータベースブロックに対応する、ストレージシステムデータストア390が記憶したデータでもよい。ソースおよびターゲットストレージシステムは、複製操作用に適切に設定されなければならない。ソースストレージシステムにおける複製用のプログラムコードは、ソースストレージシステムが記憶するデータの変化を定期的に特定し、その変化をターゲットストレージシステムに送る。同様に、ターゲットストレージシステムにおけるプログラムコードは、ソースストレージシステムからの変化を受け取り、その変化を適切に処理し組み込むことができる。複製は、例えば、ソースストレージシステムからのデータをターゲットストレージシステムにミラーリングすることにより、可用性を高くするために用いることができる。何らかの理由でソースストレージシステムを使用できない場合でも、ターゲットストレージシステムは利用可能である。複製操作における入力、ソースシステムおよびターゲットシステムを特定する情報を含み得る。

40

50

【 0 1 1 1 】

バックアップ操作は、ストレージシステムに利用可能なデータをコピーして、元のデータが消失した場合に、元のストレージシステムの情報を復旧するためにストレージシステムのバックアップコピーが用いられるようにする操作である。復旧操作はバックアップコピーに利用可能な情報を再生して、情報を復旧する。バックアップの作成以降の元のストレージシステムにおけるいかなる変更も、更新情報が何らかの形式で保存されない限り、消失する可能性があることに留意されたい。一部の実施形態では、バックアップ情報は、可能性として遅い取得速度で、例えばテープバックアップシステムなどの大量のストレージシステムに記憶される。

【 0 1 1 2 】

本明細書に定義した概念に基づく他のVDB操作がデータセンターのワークフローの自動化のために定義され用いられ得る。VDB操作は、既存のVDB操作を組み合わせることでも作成され得る。VDBまたはデータベースストレージシステム100に基づく前述の操作を利用する種々のワークフローシナリオを下に説明する。ワークフローシナリオごとに、従来のシステムに基づくシナリオの簡単な記述が、仮想データベースに基づくシナリオと比較される。

【 0 1 1 3 】

検証および開発ワークフロー

図15は従来のデータベースを用いた生産環境に基づく検証および開発用のシナリオを示す。図15に示すように、生産データベースシステム1505は生産環境に用いられるデータベース1500を含む。従来のシステムにおける生産環境に用いられるソフトウェアの検証および開発は、データベースに記憶されるデータ1500の複数のコピーを要求する場合がある。図15に示すように、データベース1500は開発システム1510のデータストア1515にコピー1550される。一定期間中に開発システム1510において開発動作が実行され得る。データストア1515内のデータベースは定期的に、ソフトウェアおよび/またはデータベースを検証するために検証システム1520内のデータストア1525にさらにコピーされる。検証システム1520において生じた問題は、さらなる開発動作を要求し得るものとしてフィードバック1575され得る。開発および検証処理は複数回繰り返され得る。特定段階において、データベースは検証システム1520から、性能検証、システムインテグレーション、認証、およびユーザ承認を含み得る品質保証を実行する品質保証(QA)システム1530のデータストア1535にコピーされ得る。QAシステム1530に基づくフィードバック1570は、開発システム1510を用いたさらなる開発を要求し得る。開発、検証、およびQAの全体処理は複数回繰り返され得る。十分なQA検証が実行されると、データベースは段階分けシステム1540のデータストア1545にさらにコピーされ得る。ソフトウェアまたはデータベース内の最後の変化は、アップグレード手段などにより、生産データベースシステム1505に伝達1560される。

【 0 1 1 4 】

図16は仮想データベースに基づく検証および開発ワークフローのためのシナリオを示す。図15に記述したワークフローにおけるデータベースのコピーを要求するいくつかのステップは、仮想データベースを用いることにより省略され得る。生産データベースシステム1505のデータベース1500は、データベースストレージシステム100にリンクおよびロード1665される。データベース1500に対応する仮想データベースは、開発システム1610にプロビジョニング1640される。開発システム1610用に作成された仮想データベースは、スケジュールに基づいて複数回リフレッシュ1670され得る。VDBにおける開発動作が特定段階に至ると、VDBが検証システム1615と共有されて、それによりユーザは検証システム1615に適切にアクセスできる。開発VDBと検証VDBとの共有は、開発VDBの時点コピーに基づく検証VDBの作成を含み得る。検証システム1615からのフィードバック1575は、プロビジョニング1640、リフレッシュ1670、および共有1645操作の反復を要求し得る。開発および検証が特

10

20

30

40

50

定段階に至ると、VDBはさらにQAシステム1630とも共有1650され、データストア1635に記憶される。検証または開発VDBとQAシステムとの共有は、対応する検証/開発VDBの時点コピーに基づくQA VDBの作成を要求し得る。代替的に、開発VDBはQAシステムにエクスポートされる。VDBはまた、段階分けシステム1640のデータストア1645に直接段階分け1655されてもよい。

【0115】

いくつかの組織では、ワークフローに伴う種々の動作が、異なる物理的位置で実行され得る。例えば、生産サーバが組織のある場所に位置し、一方、開発および検証が組織の別の場所で行われる。開発および検証を実行する他の場所は離れた場所であり得、これにより、2つの場所間のネットワーク通信が遅くなる。このシナリオでは、図15に示す開発システム1510および検証システム1520はある場所で利用可能であり、生産システム1500、QAシステム1530、および段階分けシステム1540を含むシステムの残りの要素は異なる場所で利用可能となる。

【0116】

図17はこのシナリオにおける種々のシステム間のやりとりを示す図である。図17に示すように、場所は、第1の場所1765および第2の場所1760と名付けられる。データベースストレージシステム1715は第1の場所1765で利用可能であり、第2のデータベースストレージシステム1705は第2の場所1760で利用可能である。生産データベースシステム1505に記憶されるデータベースは、第1の場所1765のデータベースストレージシステム1715にリンクおよびロード1775される。データベースに対応するデータは、データベースストレージシステム1715からデータベースストレージシステム1705に複製1725される。複製操作1725はまた、マスキング、消去、圧縮、および暗号化を含む他の操作と組み合わせられ得る。開発/検証が離れた場所で行われ、第2の場所1760のユーザが生産データベースに利用可能な特有情報にアクセスできない場合があるので、情報は隠され、消去される必要が生じ得る。情報はまた、ネットワーク上をデータが移動する時間を短縮するために圧縮され得、データの流用を防ぐために暗号化され得る。データベースは開発システム1610にプロビジョニング1740およびリフレッシュ1770され、必要に応じて、検証システム1615と共有1745される。ストレージシステムデータストア1710が記憶したデータベースにおける検証および開発による変更は、データベースストレージシステム1715に伝達され、ストレージシステムデータストア1720に記憶され得る。これらの変更の伝達は、圧縮および暗号化と組み合わせられ得る複製1730操作により実行され得る。データベースストレージシステム1715内の更新されたデータベースは、QAシステム1630にエクスポート1750され、かつ/または段階分けシステム1640にエクスポート1755される。

【0117】

バックアップおよび復旧

図18aはデータベースのバックアップおよび復旧のためのシナリオを示す。企業内の複数のデータベースシステム1810は、バックアップシステム1815のデータストア1820にコピー1825される。バックアップシステム1815は、大容量ディスク記憶装置などの持続性メモリ内のバックアップデータを記憶し得、かつ/またはテープバックアップ装置を用いることができる。従来のシステムでは、操作コピー1825は、データベース1810内のデータベースブロックのコピー、またはデータベース1810内のデータの1つ以上のファイルへのエクスポート、データストア1820内に記憶されるようにそれらのファイルをバックアップシステム1815にコピーすることに対応する。いくつかのデータベースシステム1810は、システムにおけるデータベースのスナップショットを記憶し得、これもバックアップが要求される。データベースシステム1810は、別のデータベースシステムを利用してデータベースをミラーリングし、ミラーリングしたデータベース内の変化を元のデータベース1810と同期させる。ミラーリングしたデータベースは、バックアップシステム1815にバックアップされる必要があり得る。いく

つかのシステムでは、故障および災害からデータを保護するために、データベース 1810 と共に追加の予備データベースが用いられ得る。予備データベースもまたバックアップシステム 1815 を用いてバックアップされ得る。データベースのバックアップを支援するベンダー固有のユーティリティの例は、ORACLE データベースと共に用いられる RMAN がある。

【0118】

図 18b は従来のバックアップおよび復旧に必要とされる構成に取って代わる、データベースストレージシステム 1890 を用いたデータベースの復旧のシナリオを示す。この実施形態では、データベースストレージシステム 1890 は、それ自体がデータベースシステム 1865 内のデータベース 1860 のコピーのためのストレージとして機能する。コピー操作 1825 が、リンクおよびロード操作 1830 に変更されている。データベースストレージシステム 1890 がサポートするリンクおよびロード操作を利用する利点は、完全および増分バックアップと比較して、データベース 1860 からデータベースストレージシステム 1890 に転送するデータ量はるかに少ないことである。さらに、リンクおよびロード操作 1830 を利用して実行されるデータベース 1860 のその後の更新は、完全なロードを繰り返す必要はなく、継続してデータベース 1860 内の変化のみが転送される。この結果、データベース 1860 からストレージシステムデータストア 1840 に移動するデータ量は、バックアップソリューションと比較してはるかに少ない。それ故、ストレージシステムデータストア 1840 内のデータが占有するストレージスペースは大幅に減少され、データベース 1860 からストレージシステムデータストア 1840 へのデータの転送は大幅により短時間で完了する。

【0119】

別の実施形態では、ストレージシステムデータストア 1840 に利用可能なデータは、バックアップシステム 1845 を用いてバックアップ 1855 される。バックアップ操作 1855 は、最初は、ストレージシステムデータストア 1840 に利用可能なデータ全体をコピーし、その後、ストレージシステムデータストア 1840 に記憶されているデータの増分変化のみをコピー 1855 できる。ストレージシステムデータストア 1840 に記憶されているデータ量は、データベース 1860 の変化のみを記憶しているため、従来のバックアップシステム 1815 のデータストア 1820 に記憶されているデータ量よりもはるかに少なくなり得る。それ故、ストレージシステムデータストア 1840 のデータのバックアップ 1855 に費やされる時間と、データベース 1860 内のデータのストレージシステムデータストア 1840 へのリンク/ロードに要求される時間との和は、大企業において、特に、ロードおよびソースデータベースが要求する時間に関して、バックアップ操作 1825 に費やされる時間よりもはるかに短くなり得る。

【0120】

データベース複製の保持

いくつかのワークフローシナリオでは、ソースデータベース内の情報を、ターゲットデータベースに定期的にコピーする。例えば、災害によるソースデータベースの破壊に備えて（災害復旧として知られる処理）、ソースデータベースから情報の復旧に用いられるターゲットデータベースに情報をコピーし得る。情報は 1 つ以上のデータベースにもコピーされ得、ユーザによるデータの有用性が向上する。例えば、メンテナンスまたは他の理由によりソースデータベースを停止する場合には、ターゲットデータベースをユーザに対して使用可能にする。一部の使用シナリオでは、ソースデータベースからレポート目的に用いられるターゲットデータベースに情報をコピーする。生産データベースシステムにおけるレポートの実行は、データベースに相当量の負荷がもたらされる。取引処理に生産データベースシステムが用いられるため、レポート生成のために、生産データベースシステムのデータベースと同期する異なるサーバが用いられることが好ましい。ターゲットデータベースは頻繁に更新され、レポートインフラストラクチャを利用して最新レポートを提供する。ソースデータベースからターゲットデータベースへの情報のコピーを要求する別のシナリオとして、ある機械から別の機械へのデータベースの移動がある。データベースの

移動は、企業がソフトウェアをより新しいバージョンにアップグレードする場合に要求され得る。例えば、より新しいバージョンのオペレーティングシステム、より新しいバージョンのデータベース管理システム、より新しいバージョンのアプリケーションへのアップグレード、または新規のハードウェアへのアップグレードをする場合である。データベースはまた、例えば、ある会社が別の会社を買収された場合に、ある物理的位置から別の位置への移動が要求され得る。

【0121】

図19は1つ以上のソースデータベースシステム1905からターゲットデータベースシステム1905に情報をコピーするシステム環境を示す。図19はソースデータベースシステム1905のソースデータストア1935から、ターゲットデータベースシステム1910のターゲットデータストア1940への情報のコピーまたは転送1950を示す。他の実施形態では、1つのソースデータストア1935の情報が、2つ以上のターゲットデータストア1940に転送されてもよい。代替的に、2つ以上のソースデータストア1935の情報が、1つのターゲットデータストア1940に転送1950されてもよい。

10

【0122】

転送速度、転送頻度、転送される情報の種類などの、コピー1950操作に関連する種々のパラメータは、特定シナリオに依存し得る。ソースデータベースシステム1905およびターゲットデータベース1910は、第1の場所1955および第2の場所1960に示すなどの地理的に別々の位置である物理的位置に配置され得る。通常、異なる物理的位置に配置された機械は、同じ物理的位置に配置された機械と比較してネットワークの通信速度が遅い。本明細書に記載した実施形態は、同じ物理的位置に配置されたソースおよびターゲットデータベースシステムにも、互いに異なる位置に配置されたそれらにも適用される。

20

【0123】

図20は図19に示すような従来実施されるワークフローシナリオを実行するための、データベースストレージシステム100に記憶された仮想データベースに基づくシステム環境を示す。図20に示すように、ソースデータストア1935が記憶したデータベース内のデータは、ソースデータベースストレージシステム2005のストレージシステムデータストア2025にリンクおよびロード2020される。操作2020は、ソースデータベースシステム1905の更新に基づいて、ストレージシステムデータストア2025のデータを更新するために実行されるその後のロード操作を含み得る。ソースデータベースストレージシステム2005のストレージシステムデータストア2025内のデータは、ターゲットデータベースストレージシステム2010のストレージシステムデータストア2030に送信2015される。操作2015は、ストレージシステムデータストア内の全情報をコピーするコピー操作、バックアップ操作、またはストレージシステムデータストア2025の更新を、ストレージシステムデータストア2030に増加的にコピーする複製操作であり得る。

30

【0124】

データベースの移動のシナリオでは、操作2015は、ストレージシステムデータストア2025のデータ全体をコピーし得る。複製のシナリオでは、ストレージシステムデータストア2025の変更が、ストレージシステムデータストア2030に定期的にコピーされ得る。ストレージシステムデータストア2030の変更は、リフレッシュ操作を利用してターゲットデータベースシステム1910にプロビジョニングされたVDBに適用され得る。ターゲットデータベースシステム1910による任意の変更がVDBになされると、その変更はストレージシステムデータストア2025に戻るように伝達され得る。

40

【0125】

操作2030により、ストレージシステムデータストア2030に記憶されているデータベースを、ターゲットデータベースシステム1910が利用できるようになる。高可用性システムのシナリオでは、操作2030は、ストレージシステムデータストア2030

50

からターゲットデータベースシステム 1910 への VDB のプロビジョニングに対応し得る。災害復旧のシナリオでは、操作 2030 は、ターゲットデータベースシステム 1910 へのデータベースのエクスポートに対応し得る。図 20 に示すように、ソースデータベースストレージシステム 2005 が、VDB システム 2040 に VDB をプロビジョニング 2035 し得る。ターゲットデータベースストレージシステム 2010 内のデータを使用して同等の VDB が作成され、VDB システム 2050 にプロビジョニング 2045 され得る。ソースデータベースストレージシステム 2005 内の VDB に対するいかなる変更も、ストレージシステムデータストア 2025 に自動的に保存され、転送操作 2015 によりターゲットデータベースストレージシステム 2010 に伝達される。

【0126】

一実施形態では、ターゲットデータベースストレージシステム 2010 は、操作 2015 前の、図 3 に示す全てのモジュールを有してもよい。別の実施形態では、図 3 に示すデータベースストレージシステムのモジュールを有さない機械が、ターゲットデータベースストレージシステム 2010 として用いるために提供され得る。例えば、ユーザは、必要なソフトウェアの全てはインストールされていない新しい機械を、データベースストレージシステム 100 として機能するように提供することができる。この実施形態では、操作 2015 は、データベースストレージシステムのモジュールを実現するプログラムコードを、ストレージシステムデータストア 2025 に記憶されているデータと共にターゲットマシンにコピーする。ターゲットマシンにコピーされたプログラムコードがインストールされ、実行用に用意される。それ故、ターゲットデータベースストレージシステム 2010 として用いられるように提供された機械が、データベースストレージシステム 100 のモジュールを実行するために準備される。ストレージシステムデータストア 2025 に記憶されているデータベースに関連するデータがストレージシステムデータストア 2030 にコピーされた後、ターゲットデータベースストレージシステム 2010 は VDB の関連操作、例えば、仮想データベースの作成、または VDB システム 2050 への仮想データベースのプロビジョニング 2045 を実行できる。

【0127】

図 21 は図 19 に示すような従来実施されるワークフローシナリオを実行するための、データベースストレージシステム 100 に基づくシステム環境の別の実施形態を示す。ソースデータベースシステム 1905 は、データベースストレージシステム 2105 に直接リンクおよびロード 2110 される。図 21 に示すように、データベースストレージシステム 2105 は、ソースデータベースの記憶場所 1955 とは異なる場所 1960 または物理的位置において利用可能であるか、2つのシステムが同一の場所に配置されてもよい。ソースデータベースシステム 1905 のソースデータストア 1935 にたいする変更は、定期的にデータベースストレージシステム 2105 にロード 2110 される。データベースストレージシステム 2105 は、災害復旧に利用され得る、ソースデータストア 1935 のデータベースのコピーとして機能する。データベースストレージシステム 2105 に仮想データベースが作成され得、VDB システム 2150 に利用可能なようにプロビジョニングされ得る。

【0128】

一実施形態では、データベースストレージシステム 2105 はまた、それがソースデータベースシステム 1905 がダウンした場合に用いられ得る予備システムとして機能する、高可用性シナリオに用いられ得る。データベースストレージシステム 2105 は、VDB を作成して、作成した VDB を VDB システム 2150 にプロビジョニング 2115 することにより、予備データベースとして機能する。対応するソースデータベースシステム 1905 が停止した場合には、VDB システム 2150 は予備データベースとして機能し得る。ソースデータベースシステム 1905 が処理したデータベースリクエストは、ソースデータベースシステム 1905 が停止している間に、VDB システム 2150 により処理することができる。ソースデータベースシステム 1905 がリクエストを直ぐに処理できる場合には、VDB システム 2150 によってなされた VDB への変更は、ソーススト

10

20

30

40

50

レンジシステムにエクスポートされる。VDBシステム2150からソースデータベースシステム1935に変更が適用された後、データベースリクエストはソースデータベースシステム1905に転用されてもよい。

【0129】

図22は図19に示すような従来実施されるワークフローシナリオを実行するための、データストレージシステムに基づくシステム環境の別の実施形態を示す。いくつかの企業では、ソースデータベースシステム1905からターゲットデータベースシステム1910にデータを複製する既存のシステムを備えている。それ故、図21に示すようにソースデータベースシステム1905から直接データベースストレージシステム2200にデータをリンクおよびロードする必要がない場合がある。図22に示すリンクおよびロード2265操作は、ソースデータベースシステム1905から情報がコピーされているターゲットデータベースシステム1910において利用可能な情報を用いて実行され得る。データベースストレージシステムからのデータのリンク操作およびローディングは、ソースデータベースシステム1905への負荷を伴うことがあるが、ターゲットデータベースシステム1910などのミラーリングされたシステムからの適切な情報を取得することにより回避され得る。これにより、データベースストレージシステム2200に必要な情報を提供している間、ソースストレージシステム1905には影響を及ぼさない。

【0130】

データウェアハウスを管理するためのワークフロー

図23はデータベースに利用可能なデータを用いてデータウェアハウスおよびデータマートを作成するシステム環境を示す。生産データベースシステム2305は、データストア2330が記憶した1つ以上のデータベース内の取引に基づく最新情報を含む。1つ以上の生産データベースシステム2305からの情報は、解析目的のために、操作可能データストア2310のデータストア2340に同化2380される。操作可能データストア2310内のデータは、抽出変換およびロード(ETL)システム2355によりさらに処理される2385。ETLシステム2355によって処理されたデータは、データウェアハウスシステム2315に送られる2375。ETLシステム2355は、処理のためにデータを一時的に記憶できる。ETLシステム2355が実行する処理により、データはデータウェアハウスシステム2315特有のレポートおよび解析操作に有用な特定の形式で、データウェアハウスシステム2315のデータストア2360に記憶される。データストア2360に記憶されるデータサブセットは、特定の目的用のためのデータサブセットの解析を意図して、データマートシステム2320のデータストア2365のストレージのためにコンピュータ処理2370され得る。データは、前述のいくつかのシステムのデータストアに記憶されているので、バックアップシステム2325を用いてバックアップ2350され得、バックアップデータストア2335に記憶され得る。前述の処理は、データが変化しない場合でも、異なるシステム間で同一データの複数のコピーを保持できる。そうでないと、データの記憶にはいくつかの異なるコンピュータシステムが用いられ、それにより資源の利用が非効率的となる。

【0131】

図24は図23に示すような従来実施されるワークフローシナリオを実行するための、データベースストレージシステム100に基づくシステム環境の実施形態を示す。生産データベースシステム2305のデータストア2330内のデータベースは、データベースストレージシステム2400にリンクおよびロード2450される。最初のロード操作2450後のロード2450は、データストア2330内の対応するデータベースにおいて変更されたデータのみを転送する。仮想データベースを、操作可能データストア2310として用いるために、作成し、プロビジョニング2455することができる。ETLシステム2355は、操作可能データストア2310に関連するVDBから取得したデータを処理2385し、処理したデータをデータウェアハウスシステム2315に送る2375する。データウェアハウス2315のデータストア2360に記憶されているデータは、データベースストレージシステム2400にリンクおよびロード2460される。データ

ベースストレージシステム 2400 は VDB を、データマートシステム 2320 に用いられるように作成し、プロビジョン 2470 できる。操作可能データストア 2310、ETL システム 2355、およびデータマートシステム 2320 を含むシステムは、対応するデータベースを局所的に記憶する必要はなく、データベースを記憶するためにストレージシステムデータストア 2490 を利用できる。さらに、前述のワークフロー内の種々のデータベースのバックアップ処理は、バックアップシステム 2325 のデータストア 2335 にストレージシステムデータストア 2490 をバックアップ 2465 することにより達成できる。図 18 に示すバックアップのワークフローシナリオにおいて記述したように、図 24 に示すようなデータベースストレージシステム 2400 を用いて実行されるバックアップは、図 23 に示すような種々のシステムが実行する個々のバックアップよりも効率的となり得る。ストレージシステムデータストア 2490 のバックアップは効率的である。なぜならば、ストレージシステムデータストア 2490 がデータのコピーを効率的に記憶するため、バックアップするデータ量が大幅に減少し、さらには、単一のシステムからのデータの転送は、複数のシステムからのデータ転送よりも効率的であるからである。

【0132】

コンピューティングマシン構築

図 25 は機械可読媒体からの命令を読み出し得、プロセッサ（またはコントローラ）によりその命令を実行し得る機械の実施形態の構成要素を示すブロック図である。特に、図 25 は本明細書に記述する任意の 1 つ以上の手法を機械に実行させるための命令 2524（例えば、ソフトウェア）を実行し得るコンピュータシステム 2500 の例示の形態における機械の図表を示す。代替的な実施形態では、機械はスタンドアロンデバイスとして動作するか、あるいは他の機械に接続（例えば、ネットワーク接続）してもよい。ネットワーク配置では、機械はサーバ-クライアントネットワーク環境におけるサーバ機械またはクライアント機械として動作し得るか、ピアツーピア（または分散）ネットワーク環境におけるピア機械として動作し得る。

【0133】

機械は、サーバコンピュータ、クライアントコンピュータ、パーソナルコンピュータ（PC）、タブレット PC、セットトップボックス（STB）、携帯情報端末（PDA）、携帯電話、スマートフォン、ウェブアプライアンス、ネットワークルータ、スイッチもしくはブリッジ、またはその機械が取得し得る機能を指定する命令 2524（順次又はそれ以外で）を実行可能な任意の機械であり得る。さらに、単一機械のみを示したが、「機械」という用語は、本明細書に記述した任意の 1 つ以上の手法を用いて命令 2524 を個別にまたは共同で実行する機械の任意の集合も含み得る。

【0134】

例示のコンピュータシステム 2500 は、バス 2508 を通じて互いに通信するように構成されたプロセッサ 2502（例えば、中央処理装置（CPU）、グラフィック処理装置（GPU）、デジタルシグナルプロセッサ（DSP）、1 つ以上の特定用途向け集積回路（ASIC）、1 つ以上の無線周波数集積回路（RFIC）、またはこれらの任意の組み合わせ）、メインメモリ 2504、およびスタティックメモリ 2506 を含む。コンピュータシステム 2500 は、画像表示装置 2510（例えば、プラズマ表示パネル（PDP）、液晶表示装置（LCD）、プロジェクタ、またはブラウン管（CRT））をさらに含んでもよい。コンピュータシステム 2500 は、これもまたバス 2508 を通じて互いに通信するように構成された英数字入力デバイス 2512（例えば、キーボード）、カーソル制御デバイス 2514（例えば、マウス、トラックボール、ジョイスティック、運動センサ、または他のポインティング機器）、ストレージユニット 2516、信号発生デバイス 2518（例えば、スピーカ）、およびネットワークインターフェースデバイス 2520 をさらに含んでもよい。

【0135】

ストレージユニット 2516 は、本明細書に記載された任意の 1 つ以上の手法または機能を具現化する命令 2524（例えば、ソフトウェア）を記憶する機械可読媒体 2522

10

20

30

40

50

を含む。命令 2 5 2 4 (例えば、ソフトウェア)は、コンピュータシステム 2 5 0 0 によるその実行中、メインメモリ 2 5 0 4 またはプロセッサ 2 5 0 2 (例えば、プロセッサのキャッシュメモリ)に完全または少なくとも部分的に存在してもよく、メインメモリ 2 5 0 4 およびプロセッサ 2 5 0 2 が機械可読媒体を構成してもよい。命令 2 5 2 4 (例えば、ソフトウェア)は、ネットワークインターフェースデバイス 2 5 2 0 を通じて、ネットワーク 2 5 2 6 上で送信または受信され得る。

【 0 1 3 6 】

例示の実施形態には単一媒体である機械可読媒体 2 5 2 2 を示したが、「機械可読媒体」という用語が命令 (例えば、命令 2 5 2 4) を記憶可能な単一媒体または複数の媒体 (例えば、集中データベース、分散データベース、または関連するキャッシュおよびサーバ) を含むことは言うまでもない。「機械可読媒体」という用語もまた、機械が実行する命令 (例えば、命令 2 5 2 4) を記憶でき、本明細書に開示する任意の 1 つ以上の手法を機械に実行させる任意の媒体を含むものとして理解される。「機械可読媒体」という用語は、限定されないが固体メモリ、光媒体、および磁気媒体形態のデータ保存場所を含む。

【 0 1 3 7 】

さらなる設定の考慮

本明細書全体を通して、複数の事例が、単一の事例として記述した構成要素、操作、または構造を実現してもよい。1 つ以上の方法の個々の操作を別々の操作として示し、記述したが、1 つ以上の個々の操作を同時に実行してもよく、操作は記述した順序に実行される必要はない。例示の構成における個々の構成要素として示した構造および機能は、組み合わせた構造または構成要素として実現することができる。同様に、単一の構成要素として示した構造および機能性を、個々の構成要素として実現してもよい。これらおよび他の変形、変更、追加および改善は本明細書の主題の範囲内である。

【 0 1 3 8 】

特定の実施形態を論理もしくは複数の構成要素、モジュール、または機構を含むものとして本明細書に記載した。モジュールはソフトウェアモジュール (例えば、機械可読媒体または伝送信号を具現化するコード) として構成してもよいし、ハードウェアモジュールとして構成してもよい。ハードウェアモジュールは特定の操作を実行可能な有形装置であり、特定の方法により構成または配置され得る。例示的な実施形態では、1 つ以上のコンピュータシステム (例えば、スタンドアロン、クライアント、またはサーバコンピュータシステム)、またはコンピュータシステムの 1 つ以上のハードウェアモジュール (例えば、プロセッサまたはプロセッサの一群) は、本明細書に記載した特定の操作を実行するように動作するハードウェアモジュールとして、ソフトウェア (例えば、アプリケーションまたはアプリケーション部) により構成され得る。

【 0 1 3 9 】

種々の実施形態では、ハードウェアモジュールは、機械的または電子的に実装されてもよい。例えば、ハードウェアモジュールは、特定の操作を実行するための、永久的に構成された専用回路または論理 (例えば、フィールドプログラマブルゲートアレイ (FPGA) または特定用途向け集積回路 (ASIC) などの専用プロセッサ) を含み得る。ハードウェアモジュールはまた、特定の操作を実行するための、ソフトウェアにより一時的に構成された (例えば、汎用プロセッサまたは他のプログラマブルプロセッサに含有される) プログラマブル論理または回路を含み得る。専用および永久的に構成された回路、または一時的に構成された (例えば、ソフトウェアにより構成された) 回路において、ハードウェアモジュールを機械的に実施するか否かは、費用および時間を考慮して決定されることが理解される。

【 0 1 4 0 】

それ故、「ハードウェアモジュール」という用語が、本明細書に記載した特定の操作を実行する特定の方法により動作する、物理的にかつ永久的に構成された (例えば、配線に接続された)、または一時的に構成された (例えば、プログラムされた) 実在物である有体物を含有することが当然ながら理解される。本明細書に用いられる「ハードウェア実装

モジュール」は、ハードウェアモジュールを意味する。ハードウェアモジュールが一時的に構成された（例えば、プログラムされた）実施形態を考慮すると、ハードウェアモジュール各々は、任意の一事例において後に構成または例示化される必要はない。例えば、ハードウェアモジュールがソフトウェアを用いて構成される汎用プロセッサを備えている場合には、汎用プロセッサは、様々な時間においてそれぞれが異なるハードウェアモジュールとして構成され得る。それ故、ソフトウェアはプロセッサを構成し、例えば、時間の一事例において特定のハードウェアモジュールを構成し、種々の時間の事例における種々のハードウェアモジュールを構成する。

【 0 1 4 1 】

ハードウェアモジュールは他のハードウェアモジュールに情報を送り、かつそれからの情報を受け取り得る。それ故、記述したハードウェアモジュールは通信可能に接続され得る。このようなハードウェアモジュールが複数同時に存在する場合には、ハードウェアモジュールを（例えば、適切な回路およびバスを通じて）接続する信号伝送を通じて通信が達成され得る。複数のハードウェアモジュールが様々な時間において構成、またはインスタンス化される実施形態では、このようなハードウェアモジュール間の通信は、例えば、ストレージを通じて、および複数のハードウェアモジュールがアクセスするメモリ構造内に情報を取得することによって達成される。例えば、1つのハードウェアモジュールが、操作を実行し、それが通信可能に接続するメモリデバイス内の操作の出力を記憶できる。さらなるハードウェアモジュールが、その後、メモリデバイスにアクセスして、記憶した出力を取得および処理できる。ハードウェアモジュールはさらに、入力または出力デバイスとの通信を起動し得、情報資源に対する動作を実行する（例えば、情報の収集）。

【 0 1 4 2 】

本明細書に記載した例示の方法における種々の操作は、少なくとも部分的に、（例えば、ソフトウェアにより）一時的に構成されたか、永久的に構成された1つ以上のプロセッサを利用して、関連操作を実行できる。一時的に構成されたか、永久的に構成されたかに関わらず、このようなプロセッサは、1つ以上の操作または機能を実行するように動作するプロセッサ実装モジュールを構成し得る。本明細書に参照するモジュールは、一部の例示的な実施形態では、プロセッサ実装モジュールを備えている。

【 0 1 4 3 】

同様に、本明細書に記載した方法は、少なくとも部分的にプロセッサにより実施される。例えば、方法の操作の少なくとも一部は、プロセッサまたはプロセッサ実装ハードウェアモジュールのいずれかにより実行され得る。操作の特定の動作は、1つ以上のプロセッサ間で分配され得、単一機械に存在するだけでなく、複数の機械にわたっても分配され得る。いくつかの例示的な実施形態では、1つ以上のプロセッサは（例えば、家庭環境もしくははオフィス環境内の位置、またはサーバファームとして）単一位置に配置されてもよいが、他の実施形態では、プロセッサは複数位置に配置されてもよい。

【 0 1 4 4 】

1つ以上のプロセッサはまた、「クラウドコンピュータ」環境内の関連操作の性能をサポートするように、または「サービス型ソフトウェア」（SaaS）として動作し得る。例えば、少なくとも一部の操作は、（プロセッサを含む機械の例として）コンピュータ群により実行され得る。これらの操作はネットワーク（例えば、インターネット）および1つ以上の適切なインターフェース（例えば、アプリケーションプログラムインターフェース）を通じてアクセス可能である。

【 0 1 4 5 】

操作の特定の動作は、1つ以上のプロセッサ間で分配され得、単一機械に存在するだけでなく、複数の機械にも分配される。いくつかの例示的な実施形態では、1つ以上のプロセッサまたはプロセッサ実装モジュールは、（例えば、家庭環境もしくははオフィス環境内の位置、またはサーバファームとして）単一の地理的位置に配置されてもよい。他の例示的な実施形態では、1つ以上のプロセッサまたはプロセッサ実装モジュールは、複数の地理的位置にわたって割り当てられてもよい。

【 0 1 4 6 】

本明細書の一部は、機械メモリ（例えば、コンピュータメモリ）内のビットまたは2値デジタル信号として記憶されたデータの操作のアルゴリズムまたは象徴の観点から示される。これらのアルゴリズムまたは記号的表現は、それらの研究における本質を当業界の他の技術者に与えるために、データ処理分野の当業者が用いる技術の例として示される。本明細書に用いる「アルゴリズム」は、所望の結果を導く操作または類似の処理の首尾一貫したシーケンスである。この文脈では、アルゴリズムおよび操作は、物理量の物理的処置を伴う。通常、このような物理量は、必ずしもこれらに限らないが、機械による記憶、アクセス、転送、結合、比較、または他の処理を実行可能な電気、磁気、または光信号という形式であり得る。主に共通使用の理由から、「データ」、「コンテンツ」、「ビット」、

10

「値」、「要素」、「記号」、「文字」、「用語」、「数字」、「数詞」などの言葉を用いてこれらの信号を示すことが場合によっては好都合である。ただし、これらの言葉は、単に好都合な名称であり、適切な物理量に関連するものである。

【 0 1 4 7 】

他に明確に記載しない限り、「処理」、「計算」、「算出」、「決定」、「提示」、「表示」などの言葉を用いた本明細書の記述は、メモリ（例えば、揮発性メモリ、非揮発性メモリ、またはそれらの組み合わせ）、レジスタ、または情報を受け取り、記憶、送り、または表示する他の機械部品の1つ以上の物理量を示すデータ（例えば、電子、磁気、または光学）を操作または変換する機械（例えば、コンピュータ）の機能または処理を意味し得る。

20

【 0 1 4 8 】

本明細書に用いる「一実施形態」または「1つの実施形態」といった言い回しは、その実施形態との関連で記載された特定の要素、特徴、構造、または特性が、少なくとも一部の実施形態に含まれることを意味する。本明細書の様々な位置に記載する句「一実施形態では」は、全て同一の実施形態を必ずしも意味する訳ではない。

【 0 1 4 9 】

一部の実施形態は、「連結された」および「接続された」という表現、さらにはこれらの派生語を用いて記述され得る。これらが互いに同義語として意図されていないことが理解されることは言うまでもない。例えば、いくつかの実施形態で用いられる「接続された」という用語は、2以上の要素が互いに直接に物理または電気接触していることを意味し得る。別の例では、いくつかの実施形態は、2以上の要素が直接に物理または電気接触していることを示す「連結された」という用語を用いて記述され得る。しかし、「連結された」という用語は、2以上の要素が、互いに直接には接触していないが互いに共同動作または通信することも意味することがある。実施形態はこれらの文脈により限定されない。

30

【 0 1 5 0 】

本明細書に用いる「備える」、「備えている」、「含む」、「含んでいる」、「有する」、「有している」という用語またはこれらのあらゆる変形語は、包括的な含有を含むように意図される。例えば、要素の列挙を含む処理、方法、部品、または装置は、必ずしもこれらの要素のみに限定されず、明示的に列挙されていない、またはこのような処理、方法、部品、または装置に元々備えられている他の要素を含み得る。さらに、明確に逆に規定しない限り、「または」は含有を意味し、排他的な「または」は意味しない。例えば、条件「AまたはB」は、下記のいずれかの条件を満たす。すなわち、「Aが真であり（または、存在している）、かつBが偽である（または、存在していない）」、「Aが偽であり（または、存在していない）、かつBが真である（または、存在している）」、または「AおよびBが共に真である（または、存在している）」である。

40

【 0 1 5 1 】

また、本明細書における要素および構成要素の記述に、不定冠詞「a」または「an」を用いている。これは単に便宜上のためであり、本発明の範囲の一般的な意味を与えるためである。この記述は、「1つの」または「少なくとも1つの」を包含するように読まれるべきであり、また、単数は、それが複数を含意しないことが明らかでない限り、複数も

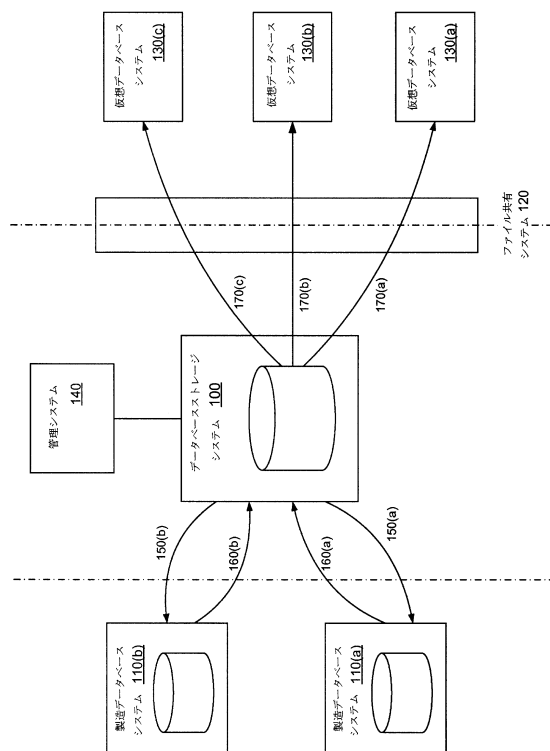
50

包含する。

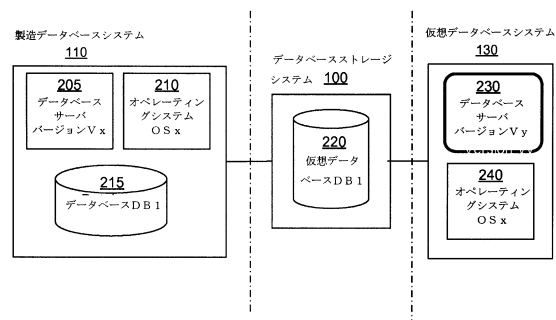
【 0 1 5 2 】

当業者は、本明細書読了後、ストレージマネージャに記憶される生産データベースの時点コピーから仮想データベースを作成するシステムおよび処理のための追加の代替構造および機能の設計を理解する。それ故、特定の実施形態およびアプリケーションを示し、記述したが、開示する実施形態が、本明細書に開示する正確な構造および構成要素に限定されないことが理解される。本明細書に開示する方法および装置の配置、操作、および詳細に関する種々の変更、変化、および変形が、添付の請求項に定義された精神および範囲から逸脱することなく、達成され得ることは当業者に明らかである。

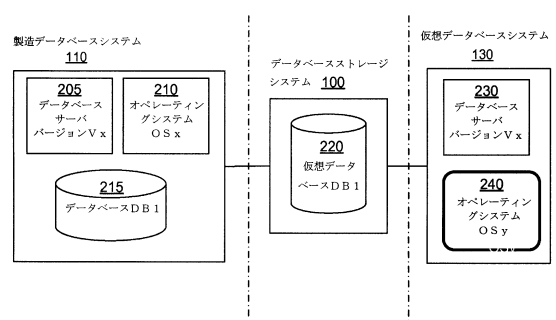
【 図 1 】



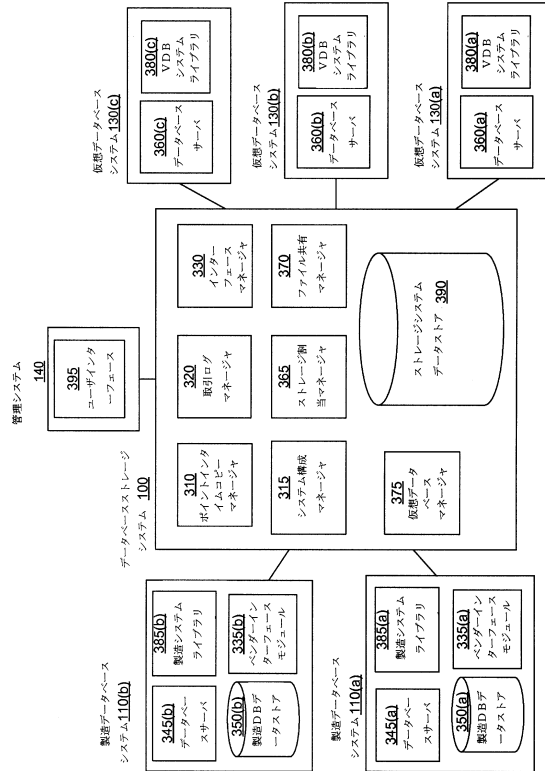
【 図 2 a 】



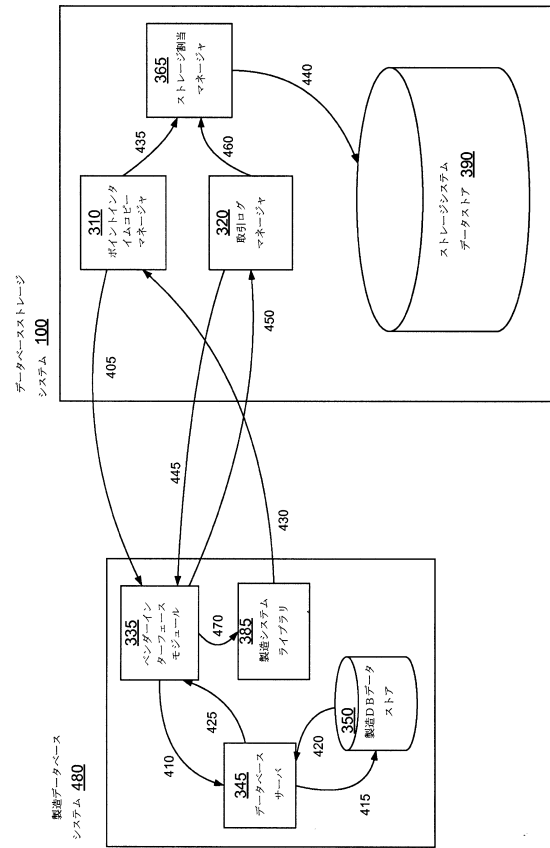
【 図 2 b 】



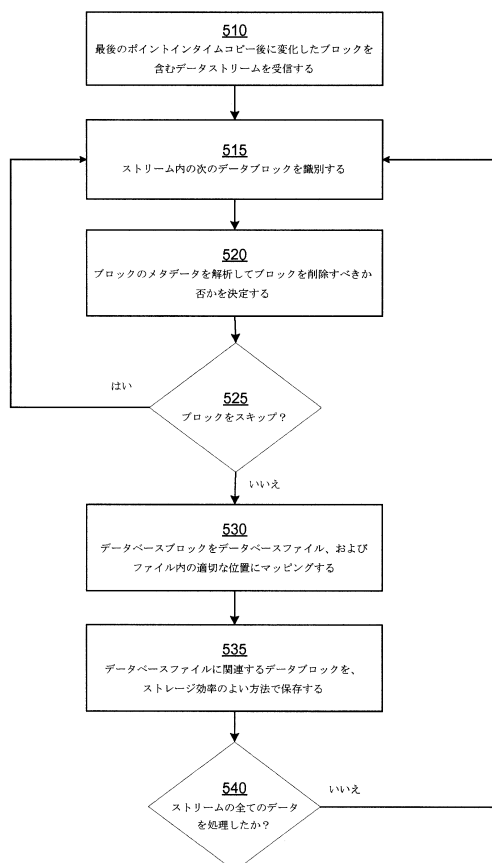
【 図 3 】



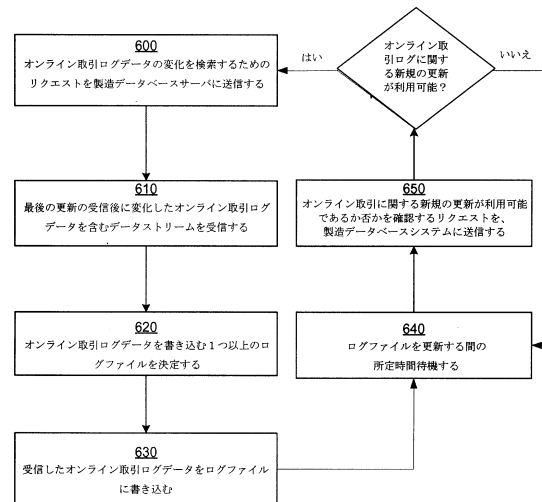
【 図 4 】



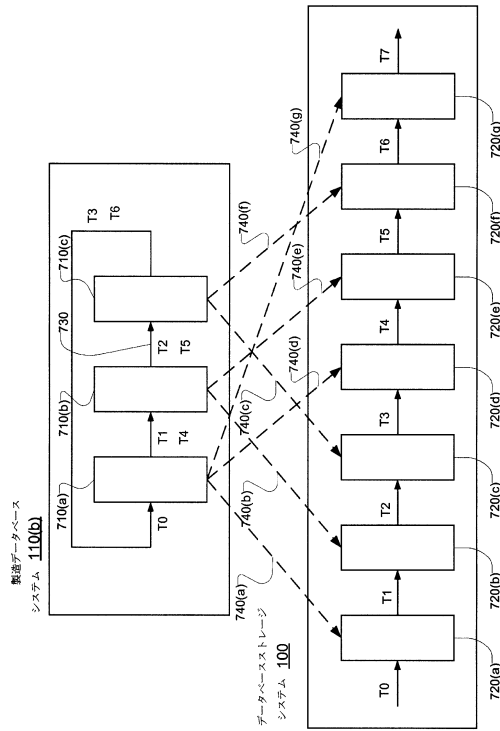
【 図 5 】



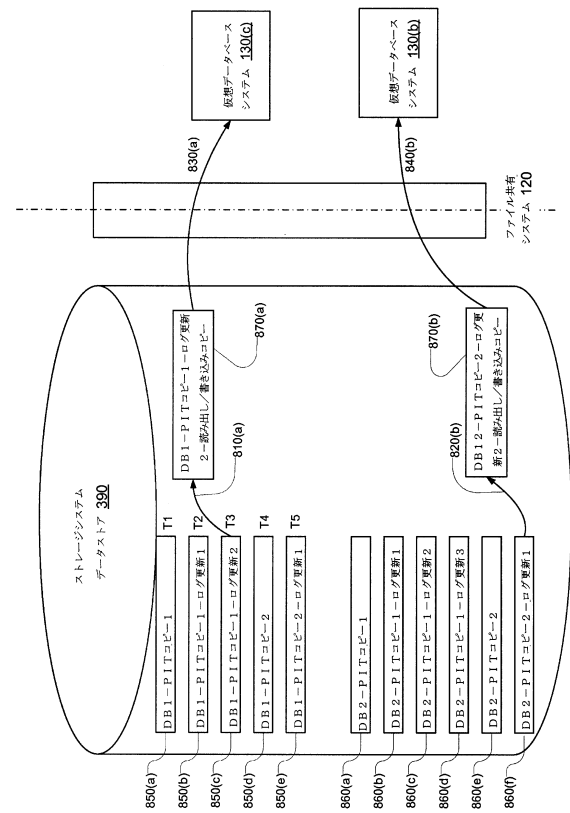
【 図 6 】



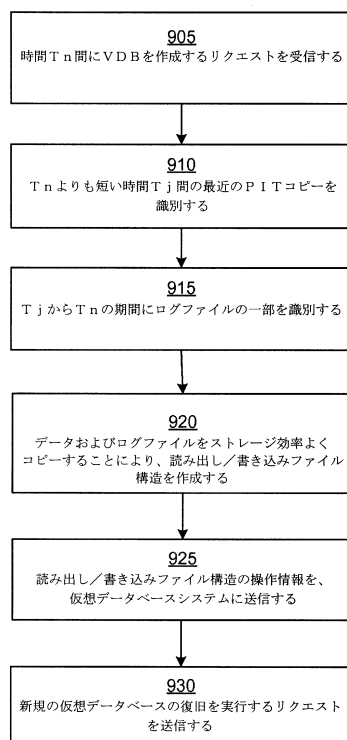
【 図 7 】



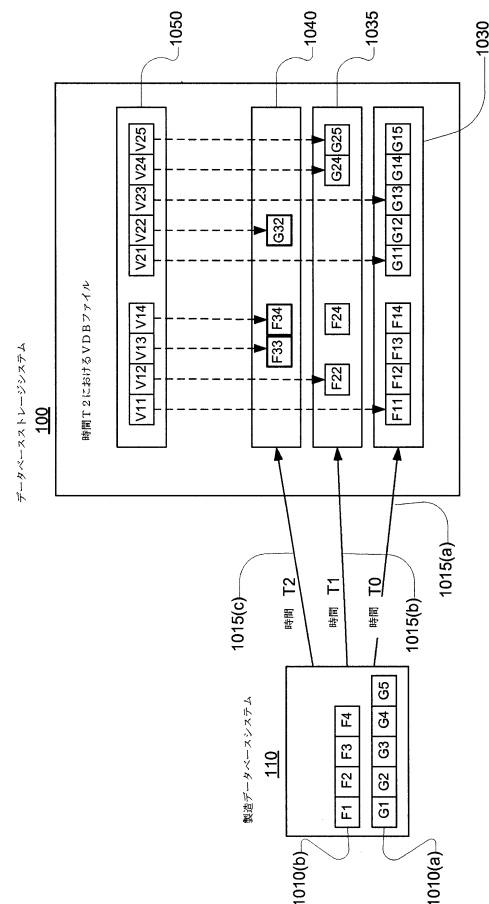
【 図 8 】



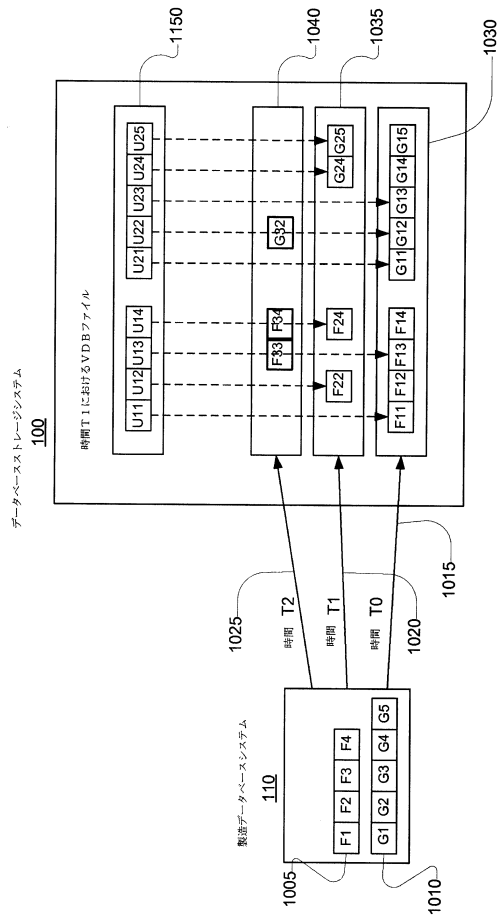
【 図 9 】



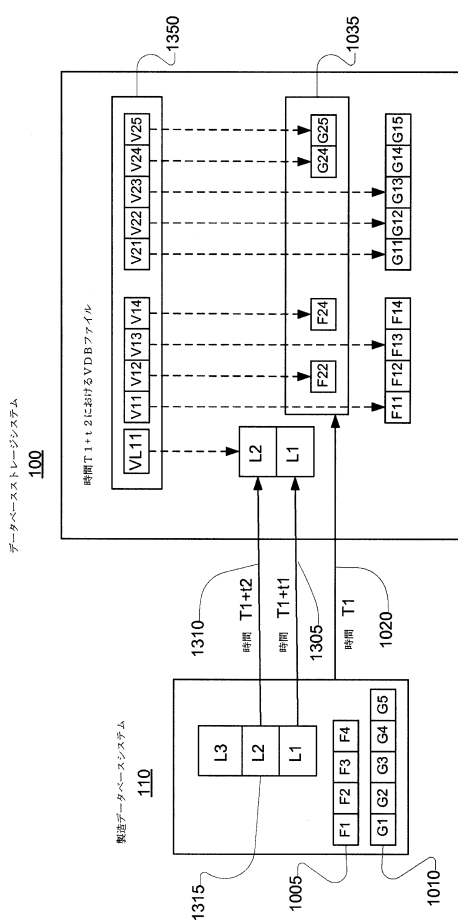
【 図 1 0 】



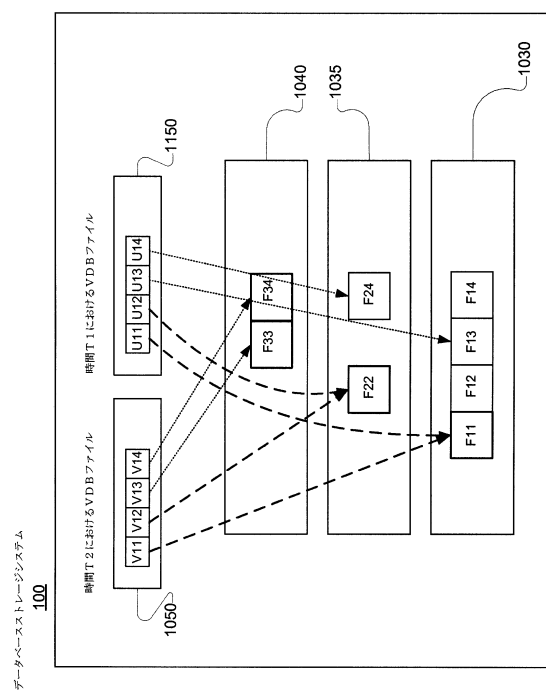
【図 1 1】



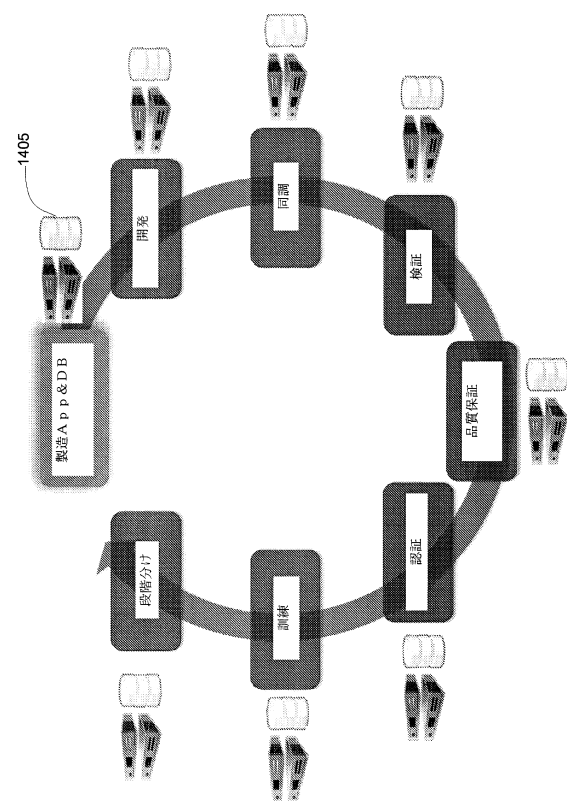
【図 1 3】



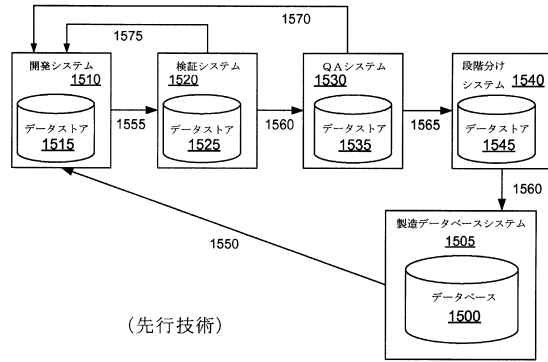
【図 1 2】



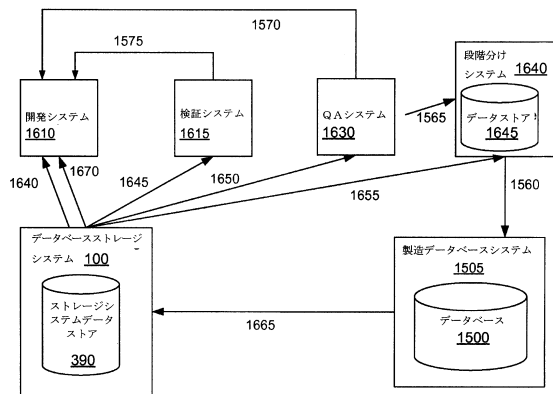
【図 1 4】



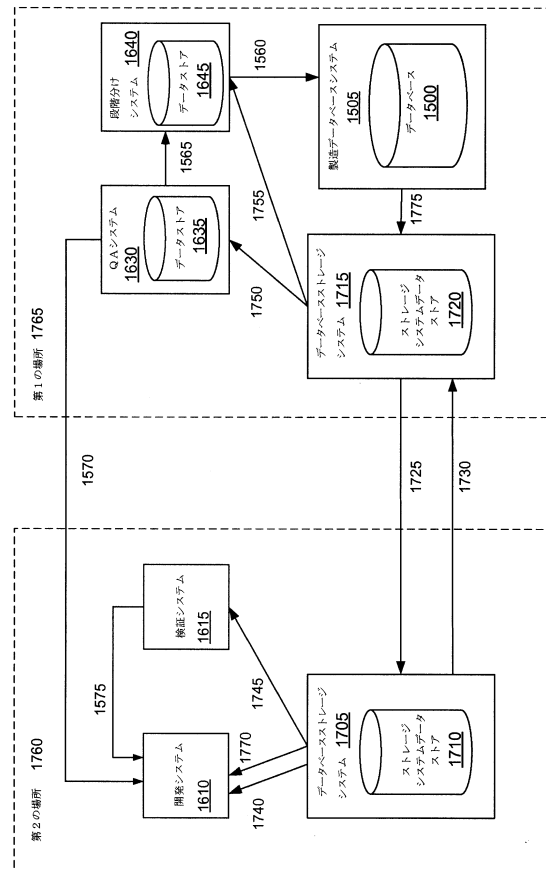
【図 15】



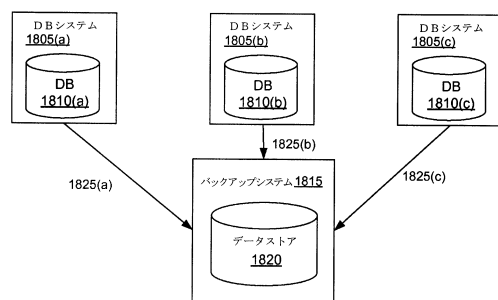
【図 16】



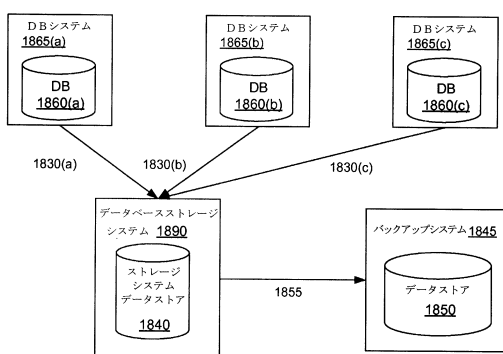
【図 17】



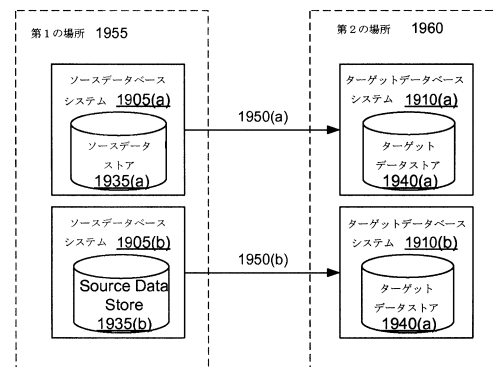
【図 18 a】



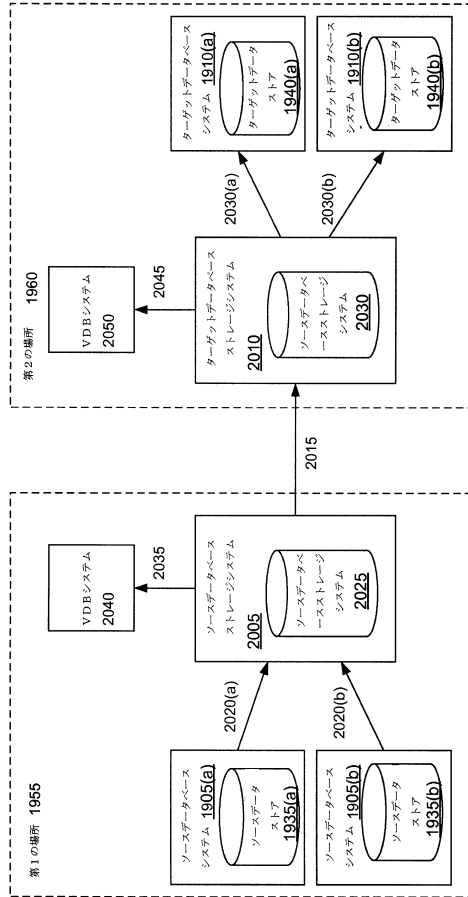
【図 18 b】



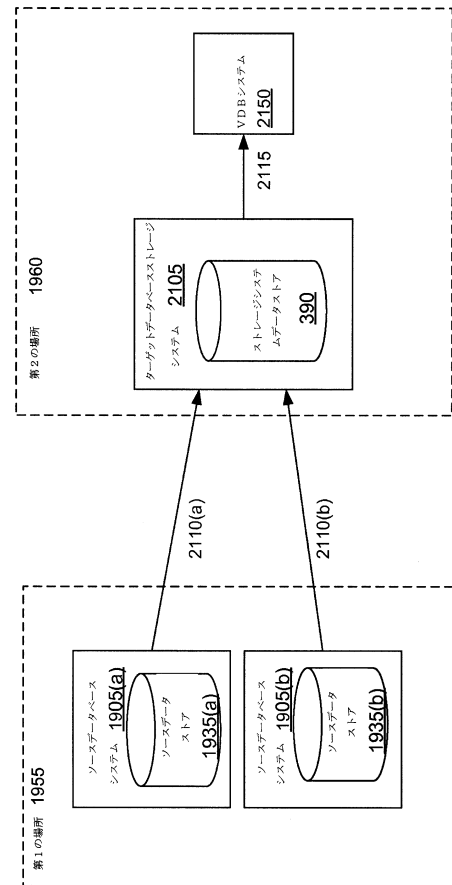
【図 19】



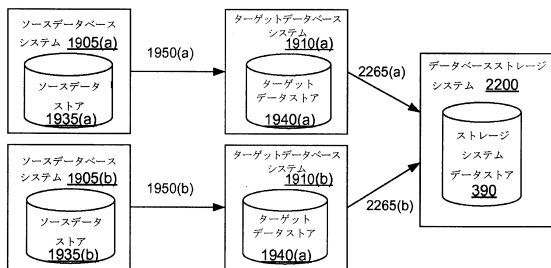
【図 20】



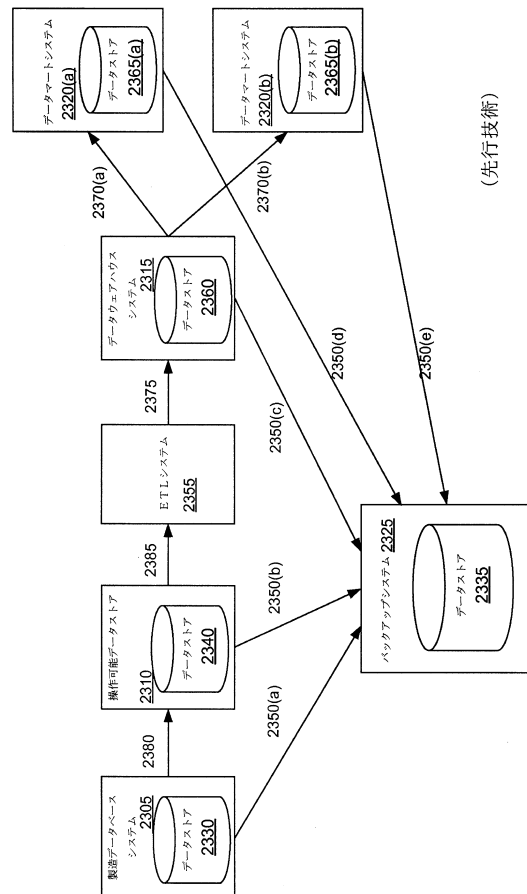
【図 21】



【図 22】

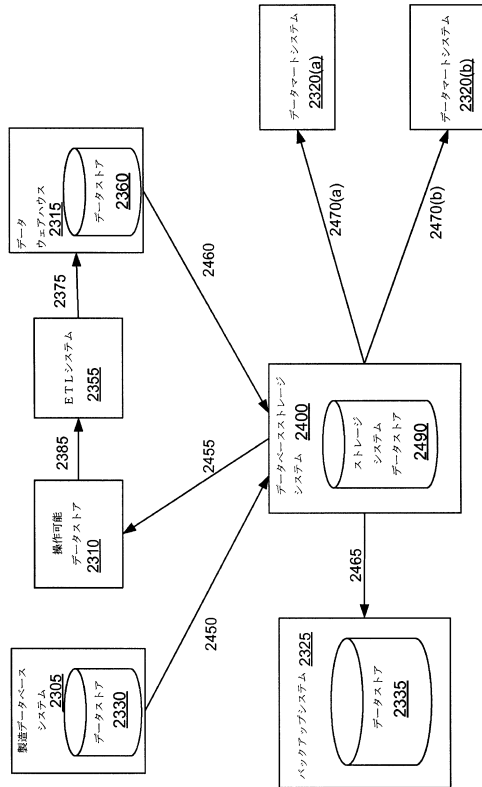


【図 23】

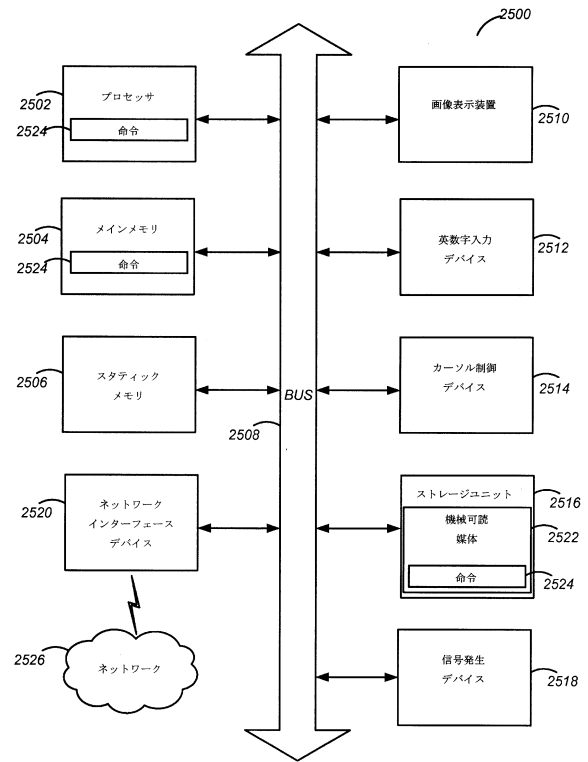


(先行技術)

【図 24】



【図 25】



フロントページの続き

- (72)発明者 ジャ, チャーリー, リ
アメリカ合衆国 カリフォルニア 94303, パロ アルト, セカンド フロア, サン アント
ニオ 960, デルフィクス コーポレーション内
- (72)発明者 ユエ, ジェディディア
アメリカ合衆国 カリフォルニア 94303, パロ アルト, セカンド フロア, サン アント
ニオ 960, デルフィクス コーポレーション内

審査官 池田 聡史

- (56)参考文献 特開2004-110218(JP, A)
特表2009-530756(JP, A)
ハーバウ ローガン, 異種混在型の仮想化環境を的確に管理せよ PART2 仮想化環境向け
「高可用性+ディザスタ・リカバリ」注目5製品徹底レビュー, COMPUTERWORLD,
日本, (株)IDGジャパン, 2009年 1月 1日, 第6巻 第1号, pp.42~51

- (58)調査した分野(Int.Cl., DB名)
G06F 12/00
G06F 17/30
JSTPlus(JDreamIII)