

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5577350号
(P5577350)

(45) 発行日 平成26年8月20日(2014.8.20)

(24) 登録日 平成26年7月11日(2014.7.11)

(51) Int.Cl. F I
G O 6 F 12/00 (2006.01) G O 6 F 12/00 5 3 3 J

請求項の数 14 (全 24 頁)

(21) 出願番号	特願2011-540691 (P2011-540691)	(73) 特許権者	510330264
(86) (22) 出願日	平成21年12月9日 (2009.12.9)		アリババ・グループ・ホールディング・リミテッド
(65) 公表番号	特表2012-511773 (P2012-511773A)		ALIBABA GROUP HOLDING LIMITED
(43) 公表日	平成24年5月24日 (2012.5.24)		英国領、ケイマン諸島、グランド・ケイマン、ジョージ・タウン、ワン・キャピタル・プレイス、フォース・フロア、ピー・オー・ボックス 847
(86) 国際出願番号	PCT/US2009/006457	(74) 代理人	110000028
(87) 国際公開番号	W02010/068260		特許業務法人明成国際特許事務所
(87) 国際公開日	平成22年6月17日 (2010.6.17)	(72) 発明者	チャオ・リン
審査請求日	平成24年11月13日 (2012.11.13)		中華人民共和国 ハンチョウ、ワーナー・ロード、ウエスト・レイク・インターナショナル・プラザ、10階、ナンバー391
(31) 優先権主張番号	12/653,091		最終頁に続く
(32) 優先日	平成21年12月8日 (2009.12.8)		
(33) 優先権主張国	米国 (US)		
(31) 優先権主張番号	200810179394.6		
(32) 優先日	平成20年12月10日 (2008.12.10)		
(33) 優先権主張国	中国 (CN)		

(54) 【発明の名称】 効率的なデータ同期化のための方法及びシステム

(57) 【特許請求の範囲】

【請求項1】

データ同期化方法であって、

同期化予定のデータを保存されているソースデータテーブルに基づいて複数のターゲットデータテーブルを確立することと、

前記複数のターゲットデータテーブルのなかから現ターゲットデータテーブルを決定することと、

前記ソースデータテーブルと前記現ターゲットデータテーブルとを同期化することであって、

前記ソースデータテーブル内において同期化予定データによって占有されている幾つかのストレージ範囲を決定すること、

前記それぞれのストレージ範囲内における前記同期化予定データのストレージアドレス情報を決定すること、

前記ストレージアドレス情報を使用して、前記それぞれのストレージ範囲内の前記同期化予定データを前記現ターゲットデータテーブルへ同期化すること、

を含む、前記ソースデータテーブルと前記現ターゲットデータテーブルとを同期化することと、

同期化が成功した場合に、前記現ターゲットデータテーブルにアクセスするようにアプリケーションサーバに指示することと、

を備える方法。

10

20

【請求項 2】

請求項 1 に記載の方法であって、

前記アプリケーションサーバは、前記現ターゲットデータテーブルを指し示すように事前に更新された上位ビューを通じて前記現ターゲットデータテーブルにアクセスするように指示される、方法。

【請求項 3】

請求項 1 に記載の方法であって、

前記複数のターゲットデータテーブルのなかから前記現ターゲットデータテーブルを決定することは、

前記複数のターゲットデータテーブル内のそれぞれのデータレコードの数を決定することと、

前記ターゲットデータテーブルのうち前記データレコードの数がゼロである 1 つのターゲットデータテーブルを前記現ターゲットデータテーブルとして選択することと、

を含む、方法。

10

【請求項 4】

請求項 1 に記載の方法であって、

前記ストレージアドレス情報を使用して、前記それぞれのストレージ範囲内の前記同期化予定データを前記現ターゲットデータテーブルへ同期化することは、前記ストレージアドレス情報にしたがって、前記それぞれのストレージ範囲内の前記データを 1 つのデータ同期化プロセスにおいて順次、前記現ターゲットデータテーブルへ同期化することを含む、方法。

20

【請求項 5】

請求項 1 に記載の方法であって、

前記ストレージアドレス情報を使用して、前記それぞれのストレージ範囲内の前記同期化予定データを前記現ターゲットデータテーブルへ同期化することは、

前記ストレージ範囲を N 個のグループに分割することと、

対応する N 個のデータ同期化プロセスを開始させることと、

前記ストレージアドレス情報にしたがって、前記グループのうちの対応する 1 つのグループの前記それぞれのストレージ範囲内の前記データを、前記データ同期化プロセスのうちの各プロセスにおいて順次、前記現ターゲットデータテーブルへ同期化することと、

を含む、方法。

30

【請求項 6】

請求項 1 に記載の方法であって、更に、

前記同期化が失敗した場合に、前記それぞれのストレージ範囲内の前記データを前記現ターゲットデータテーブルへ同期化し直すことと、

やり直しの回数がプリセット閾値を超えたときに、前記ストレージ範囲内の前記データを同期化することを終了させることと、

を備える方法。

【請求項 7】

請求項 1 に記載の方法であって、更に、

前記ソースデータテーブルと前記現ターゲットデータテーブルとの同期化が成功した場合、前記現ターゲットデータテーブル以外のターゲットデータテーブルの内容を消去することを備える方法。

40

【請求項 8】

データ同期化システムであって、

1 つ又は 2 つ以上のプロセッサであって、

同期化予定のデータを保存されているソースデータテーブルに基づいて複数のターゲットデータテーブルを確立し、

前記複数のターゲットデータテーブルのなかから現ターゲットデータテーブルを決定し、

50

前記ソースデータテーブルと前記現ターゲットデータテーブルとを同期化し、前記ソースデータテーブルと前記現ターゲットデータテーブルとを同期化することは、

前記ソースデータテーブル内において同期化予定データによって占有されている幾つかのストレージ範囲を決定し、

前記それぞれのストレージ範囲内における前記同期化予定データのストレージアドレス情報を決定し、

前記ストレージアドレス情報を使用して、前記それぞれのストレージ範囲内の前記同期化予定データを前記現ターゲットデータテーブルへ同期化すること、

を含み、及び

同期化が成功した場合に、前記現ターゲットデータテーブルにアクセスするようにアプリケーションサーバに指示するように、

構成されたプロセッサと、

前記1つ又は2つ以上のプロセッサに結合され、前記1つ又は複数のプロセッサに命令を提供するように構成されたメモリと、

を備えるシステム。

【請求項9】

請求項8に記載のシステムであって、

前記アプリケーションサーバは、前記現ターゲットデータテーブルを指し示すように事前に更新された上位ビューを通じて前記現ターゲットデータテーブルにアクセスするように指示される、システム。

【請求項10】

請求項8に記載のシステムであって、

前記複数のターゲットデータテーブルのなかから前記現ターゲットデータテーブルを決定することは、

前記複数のターゲットデータテーブル内のそれぞれのデータレコードの数を決定することと、

前記ターゲットデータテーブルのうち前記データレコードの数がゼロである1つのターゲットデータテーブルを前記現ターゲットデータテーブルとして選択することと、

を含む、システム。

【請求項11】

請求項8に記載のシステムであって、

前記ストレージアドレス情報を使用して、前記それぞれのストレージ範囲内の前記同期化予定データを前記現ターゲットデータテーブルへ同期化することは、前記ストレージアドレス情報にしたがって、前記それぞれのストレージ範囲内の前記データを1つのデータ同期化プロセスにおいて順次、前記現ターゲットデータテーブルへ同期化することを含む、システム。

【請求項12】

請求項8に記載のシステムであって、

前記ストレージアドレス情報を使用して、前記それぞれのストレージ範囲内の前記同期化予定データを前記現ターゲットデータテーブルへ同期化することは、

前記ストレージ範囲をN個のグループに分割することと、

対応するN個のデータ同期化プロセスを開始させることと、

前記ストレージアドレス情報にしたがって、前記グループのうちの対応する1つのグループの前記それぞれのストレージ範囲内の前記データを、前記データ同期化プロセスのうちの各プロセスにおいて順次、前記現ターゲットデータテーブルへ同期化することと、

を含む、システム。

【請求項13】

請求項8に記載のシステムであって、

前記1つ又は2つ以上のプロセッサは、更に、

前記同期化が失敗したときに、前記それぞれのストレージ範囲内の前記データを前記

10

20

30

40

50

現ターゲットデータテーブルへ同期化し直すように、及び

やり直しの回数がプリセット閾値を超えたときに、前記ストレージ範囲内の前記データを同期化することを終了させるように、

構成される、システム。

【請求項 14】

請求項 8 に記載のシステムであって、

前記 1 つ又は 2 つ以上のプロセッサは、更に、前記ソースデータテーブルと前記現ターゲットデータテーブルとの同期化が成功した場合に、前記現ターゲットデータテーブル以外のターゲットデータテーブルの内容を消去するように構成される、システム。

【発明の詳細な説明】

10

【技術分野】

【0001】

[関連出願への相互参照]

本出願は、2008年12月10日付けで出願された、発明の名称を「DATA SYNCHRONIZATION METHOD AND DEVICE (データ同期化の方法及びデバイス)」とする中国特許出願第200810179394.6号の優先権を主張する。当該出願は、あらゆる目的のために参照によって本明細書に組み込まれる。

【0002】

本発明は、コンピュータの分野に関し、特に、データ同期化の方法及びシステムに関する。

20

【背景技術】

【0003】

大規模な電子商取引ウェブサイトでは、フロントエンドアプリケーションサーバによるクエリを処理するために、バックエンドデータウェアハウスと、オンライントランザクション処理 (OLTP) 用のORACLE (商標) データベースなどのフロントエンドデータベースとを、同期化させることが、しばしば必要とされる。

【0004】

ORACLEデータベースは、アンドゥ/ロールバックテーブルスペースを含む。あるセッションがデータベースに対してデータ操作言語 (DML) で更新及び削除の動作を実施するとき、アンドゥ/ロールバックテーブルスペースに、データ行に作用するローデータが保存される。問題のセッションがコミットされる前に別のセッションがこれらのデータ行にアクセスするときは、このセッションは、データが変更されたがまだコミットされていない旨の表示を受信し、保存されたアンドゥ/ロールバックアドレス情報にしたがってアンドゥ/ロールバックテーブルスペースからローデータを読み出す。占有されたロールバックセグメントは、問題のセッションがコミットされるまで又はロールバックが終了されるまで解放されない。

30

【0005】

図 1 は、代表的なデータ同期化技術を示した図である。この例では、テーブル t 1 は、フロントエンドOLTP ORACLEデータベースにあるテーブルであり、データウェアハウス (DW) は、バックエンドデータウェアハウスであり、アプリケーションサーバは、テーブル t 1 からデータを読み出す。代表的なデータ同期化技術は、以下の通りである (以下のコマンドは、全て、フロントエンドデータベースで実行される)。すなわち、第 1 のステップでは、ユーザテーブル t 1 内のデータが、delete from schema1.tablenameなどのコマンドを使用して削除され、削除されたデータは、アンドゥ/ロールバックテーブルスペースに一時的に保存され、第 2 のステップでは、バックエンドデータウェアハウスのユーザテーブル t 2 からデータが読み出され、insert into schema 1.tablename select * from schema2.tablename@lnk_dwなどのコマンドを使用してユーザテーブル t 1 に挿入され、同期化されたがコミットされていないデータもまた、アンドゥ/ロールバックテーブルスペースに一時的に保存され、第 3 のステップでは、データはコミットされ、第 4 のステップでは、もし異常が発生した場合にデータがロールバックされる。ORACLEデータベースは

40

50

、フロントエンド及びデータウェアハウスの両方に採用されるので、データは、データベースリンクを介して直接的にインポート/エクスポートすることができる。例えば、上記ステップの第2のステップで使用される `lnk_dw` は、データベースリンクを表わす。

【0006】

上記のデータ同期化技術では、同じデータサービスの全ての動作が同じテーブル上で実施される、すなわち、バックエンドデータウェアハウスのテーブルはフロントエンドデータベースのテーブルに対応している。このように、動作中は、アンドゥ/ロールバックテーブルスペースに大量のデータを保存する必要がある。既存の技術には、以下の短所がある。

【0007】

(1) 上記のステップでは、アンドゥ/ロールバックテーブルスペースのかなりの部分が第1及び第2のステップで使用され、問題を発生させやすい。例えば、もしアンドゥ/ロールバックテーブルスペースが使い尽くされると、サービスの通常の過程で呼び出される「挿入」、「更新」、及び「削除」などの他の通常DML動作が影響を受ける恐れがある。ピーク期間中にアンドゥ/ロールバックテーブルスペースが満杯になると、アプリケーションサーバからデータベースに送信された各DML宣言は待機しなければならず、サーバにおける接続プールが満杯になる。この結果、アプリケーションサーバに到着するますます多くの要求が接続待ちしなければならず、アプリケーションサーバ上のメモリなどのリソースが使い尽くされてアプリケーションサーバを正常に機能させなくするという悪循環が生じる。

【0008】

(2) 第1及び第2のステップの実行には、長時間要する可能性がある。この時間中に異常が発生すると、第4のロールバックステップが実行される。すると、システムは、誤差耐性に乏しくなる。ロールバックされるデータが大量にあると、ロールバック動作の実施が長期に及び、生産データベースのパフォーマンスを低下させる可能性がある。

【0009】

(3) データの同期化は、主として第1のステップの削除動作ゆえに、非効率的である。

【0010】

(4) 断絶された移送を復旧させるのは、困難である。

【0011】

(5) データの同期化の進行は、監視が容易でない。

【0012】

同期化されるデータの量が増すほど、フロントエンドORACLEデータベースのアンドゥ/ロールバックテーブルスペースは満杯になりやすく、これは、システムのパフォーマンスに悪影響を及ぼす。

【図面の簡単な説明】

【0013】

以下の詳細な説明及び添付の図面において、本発明の各種の実施形態が開示される。

【0014】

【図1】代表的なデータ同期化技術を示した図である。

【0015】

【図2】データ同期化システムの一実施形態を示した図である。

【0016】

【図3】ソースデータテーブルとターゲットデータテーブルとの間でデータを同期化するためのデータ同期化プロセスの一実施形態を示したフローチャートである。

【0017】

【図4】アプリケーションサーバがテーブル `t1` にアクセスし、データがバックエンドデータウェアハウスからヌルテーブル `t2` へ同期化される一例を示す図である。

【0018】

10

20

30

40

50

【図5】アプリケーションサーバがテーブルt2にアクセスし、データがバックエンドデータウェアハウスからヌルテーブルt1へ同期化される一例を示す図である。

【0019】

【図6】ソースデータテーブル（例：バックエンドデータウェアハウス内のデータテーブル）とターゲットデータテーブル（例：フロントエンドORACLEデータベース内のデータテーブル）との間でデータを同期化するためのデータ同期化プロセスの実施形態を示したフローチャートである。

【図7】ソースデータテーブル（例：バックエンドデータウェアハウス内のデータテーブル）とターゲットデータテーブル（例：フロントエンドORACLEデータベース内のデータテーブル）との間でデータを同期化するためのデータ同期化プロセスの実施形態を示したフローチャートである。

10

【0020】

【図8】ソースデータテーブルと現ターゲットデータテーブルとを同期化するためのプロセスの一実施形態を示したフローチャートである。

【0021】

【図9】ストレージフォーマットの一実施形態を示した図である。

【0022】

【図10】ソースデータテーブルから現ターゲットデータテーブルへセグメント式にデータを同期化するためのプロセスの一実施形態を示したフローチャートである。

【0023】

20

【図11】データ同期化システムを示したブロック図である。

【0024】

【図12】マルチプロセスデータ同期化システムを示したブロック図である。

【発明を実施するための形態】

【0025】

本発明は、プロセス、装置、システム、合成物、コンピュータ可読記憶媒体に盛り込まれたコンピュータプログラム製品、並びに / 又は結合先のメモリに格納された命令及び / 若しくは結合先のメモリによって提供される命令を実行するように構成されたプロセッサを含む、数々の形態で実現することができる。本明細書では、これらの実現の形態、又は本発明が取り得るその他の任意の形態は、技術と称することができる。総じて、開示されたプロセスのステップの順序は、本発明の範囲内で変更され得る。別途明記されない限り、タスクを実施するように構成されるものとして説明されるプロセッサ又はメモリなどのコンポーネントは、所定時にそのタスクを実施するように一時的に構成される汎用コンポーネント、又はそのタスクを実施するように製造された特殊コンポーネントとして導入され得る。本明細書で使用される「プロセッサ」という用語は、コンピュータプログラム命令などのデータを処理するように構成された1つ又は複数のデバイス、回路、及び / 又は処理コアを意味するものとする。

30

【0026】

発明の原理を例示した添付の図面を伴って、以下で、発明の1つ又は複数の実施形態に関する詳細な説明が提供される。発明は、このような実施形態に関連して説明されるが、どの実施形態にも限定されない。発明の範囲は、特許請求の範囲によってのみ限定され、数々の代替、変更、及び等価な形態を内包する。以下の説明では、発明の完全な理解を可能にするために、数々の詳細が特定される。これらの詳細は、例示目的で提供されるものであり、発明は、これらの詳細の一部又は全部を伴わなくても特許請求の範囲にしたがって実施され得る。明瞭さを期するため、発明に関連した技術分野で知られている技術物は、発明が不必要に不明瞭にされないように、詳細な説明を省略されている。

40

【0027】

図2は、データ同期化システムの一実施形態を示した図である。同じデータサービスについて、2つ又は3つ以上のターゲットデータテーブル（図2には、ターゲットデータテーブル1及びターゲットデータテーブル2が示されている）が予め確立される。ターゲッ

50

トデータテーブルは、同期化予定のデータが格納されたソースデータテーブルに対応している。この例では、各ソースデータテーブルは、交互に使用される2つのターゲットデータテーブルに対応している。ソースデータテーブル及びターゲットデータテーブルは、バックエンドデータウェアハウス内に、又はORACLEなどのフロントエンドデータベース内に存在することができる。或いは、ソースデータテーブルがバックエンドデータウェアハウス内に、ターゲットデータテーブルがフロントエンドデータベース内にそれぞれ存在することができる、又はその逆であっても良い。

【0028】

この例では、アプリケーションサーバによるターゲットデータテーブルアクセスを促すために、上位ビューが予め確立される。上位ビューは、データベースサーバ上、アプリケーションサーバ上、又はその他の適切なデバイス上に確立され得る。一部の実施形態では、上位ビューは、ターゲットデータテーブルへのポインタ、ハンドル、又はその他の適切な参照として実装される。一部の実装形態では、上位ビューは、ターゲットデータテーブルへポインタ又は参照を返す関数として実装される。アプリケーションサーバは、上位レベルによって指し示される又は参照されるターゲットデータテーブルにアクセスする。図2に示されるように、例えば、上位ビューがターゲットデータテーブル1を指し示すときは、アプリケーションサーバは、上位ビューを介してターゲットデータテーブル1内のデータにアクセスすることができる。その間、ターゲットデータテーブル2は、アクセスを受けず、同期化を所望されるデータがソースデータテーブル内にある場合には、ターゲットデータテーブル2と同期化することができる。同様に、上位ビューがターゲットデータテーブル2を指し示すときは、アプリケーションサーバは、上位ビューを通してターゲットデータテーブル2内のデータにアクセスすることができる。ターゲットデータテーブル1は、このときはアクセスされないため、同期化を必要とするソースデータテーブル内のデータは、ターゲットデータテーブル1と同期化することができる。このように、データ同期化プロセス及びデータアクセスプロセスは、別々のテーブル上で実施され、衝突が回避される。

【0029】

図3は、ソースデータテーブルとターゲットデータテーブルとの間でデータを同期化するためのデータ同期化プロセスの一実施形態を示したフローチャートである。プロセスは、ウェブサーバやデータベースサーバなどのサーバ上で、又はデータベース及びアプリケーションサーバの両方とやり取りするその他の任意の適切なデバイス上で実施され得る。

【0030】

ステップ302では、図2に関連して上述された技術に従い、ソースデータテーブルに基づいて2つ又は3つ以上のターゲットデータテーブルが確立される。ステップ304では、先に確立されたターゲットデータテーブルから現ターゲットデータテーブルが決定される。一部の実施形態では、現ターゲットデータテーブルを決定するために、各ターゲットデータテーブル内のデータレコードの数がカウントされる。レコードを持たないターゲットデータテーブルが、現テーブルであると見なされる。2つ以上のターゲットデータテーブルのレコード数がゼロの場合には、任意のテーブルが選ばれる。しかしながら、レコード数がゼロのターゲットデータテーブルが存在しない場合には、異常が発生しており、データ同期化のフローは終了される。

【0031】

ステップ306では、ソースデータテーブルと、決定された現ターゲットデータテーブルとが同期化される。一部の実施形態では、ソースデータテーブルから、決定された現ターゲットデータテーブルへデータが同期化されるときに、データは、決定された現ターゲットデータテーブル(すなわち、元はヌルのターゲットデータテーブル)へ直接、コピー又は更新することができる。一部の実施形態では、元は空/無効(null)のターゲットデータテーブルが削除され、現ターゲットデータテーブルとして新しいターゲットデータテーブルが作成され、データは、その新しく作成された現ターゲットデータテーブルへコピーされる。

10

20

30

40

50

【 0 0 3 2 】

一部の実施形態では、ターゲットデータテーブルを作成するために、以下のコマンドが使用される。

```
create table t2 as select* from schema2.tablename@lnk_dw
```

【 0 0 3 3 】

コマンドの動作は、複合動作である。1つの動作は、(テーブルt2を作成するための)DDLであり、もう1つの動作は、データ同期化である。ORACLEは、それを、アンドゥ/ロールバックテーブルスペースを実質的に占有しないDDL動作として扱う。また、並列プロンプトも採用可能であり、特定のコマンドは、以下の通りである。

```
create table t2 as select/*+parallel (dw_t2, 4)*/ from schema2.tablename@lnk_dw  
dw_t2
```

ここで、上記コマンドの/*+parallel (dw_t2, 4)*/は、4つのプロセスによってtable dw_t2をスキャンすることを意味する。

【 0 0 3 4 】

このようなスキームの使用は、アンドゥ/ロールバックテーブルスペースが一杯になる可能性を更に下げるとともに、データ同期化の効率をある程度向上させることができる。

【 0 0 3 5 】

ステップ308では、データテーブルの同期化が成功した場合(正常に完了した場合)、アプリケーションサーバは、現ターゲットデータテーブルにアクセスするように指示される。一部の実施形態では、上位ビューは、データ同期化が成功すると随時更新される。更新された上位ビューは、ターゲットデータテーブルへ同期化されたデータにアプリケーションサーバがスムーズにアクセスできることを保証するために、データの新しいインポート先となったターゲットデータテーブルを指し示す。

【 0 0 3 6 】

一部の実施形態では、非現ターゲットデータテーブル(すなわち、データ同期化を受けないターゲットデータテーブル)の情報を消去する動作も、データ同期化が成功すると実行される。

【 0 0 3 7 】

バックエンドデータウェアハウス内のソースデータテーブルとフロントエンドORACLEデータベース内のターゲットデータテーブルとの間のデータ同期化を説明するために、詳細な一例が以下に提示される。同じデータサービス用に、2つのデータテーブルがフロントエンドデータサービス内に確立される。すなわち、バックエンドデータウェアハウス内のソースデータテーブルは、フロントエンドデータベース内の、交互に使用される2つのターゲットデータテーブル(テーブルt1及びテーブルt2)に対応している。

【 0 0 3 8 】

同期化スキームは、図4及び図5に示されている。図4は、アプリケーションサーバがテーブルt1にアクセスし、データがバックエンドデータウェアハウスからヌルテーブルt2へ同期化される一例を示している。図5は、アプリケーションサーバがテーブルt2にアクセスし、データがバックエンドデータウェアハウスからヌルテーブルt1へ同期化される一例を示している。

【 0 0 3 9 】

図4及び図5では、データウェアハウス(DW)は、バックエンドデータウェアハウスに対応しており、テーブルt1及びテーブルt2は、フロントエンドデータベース内の2つのデータテーブルに対応している。これらのテーブルは、同じデータサービスに対応している。バックエンドデータウェアハウス内のデータテーブルは、フロントエンドデータベース内の2つのデータテーブル(t1及びt2)に対応している。したがって、データは、バックエンドデータウェアハウス内のテーブルからテーブルt1又はテーブルt2へ同期化することができる。

【 0 0 4 0 】

アプリケーションサーバは、上位ビューがt1又はt2のいずれを指し示すかに基づい

10

20

30

40

50

て、テーブル t 1 又はテーブル t 2 のいずれのデータにアクセスするかを決定する。したがって、アプリケーションサーバは、もし上位ビューがテーブル t 1 を指し示すならば、上位ビューを通してテーブル t 1 にアクセスする、又はもし上位ビューがテーブル t 2 を指し示すならば、アプリケーションサーバは、上位ビューを通してテーブル t 2 にアクセスする。

【 0 0 4 1 】

一部の実施形態では、データを定期的に同期化するために使用される時間間隔（プリセットデータ同期化期間とも言われる）がある。例えば、一部のシステムは、データウェアハウスからフロントエンドORACLEデータベースへ毎日データ同期化を実施するように構成される。データ同期化のために、テーブル t 1 及びテーブル t 2 は、同期化されたデータを取り込む（すなわち、コピーする）ヌルテーブルを見つけるために調査される。

10

【 0 0 4 2 】

図 4 では、アプリケーションサーバはテーブル t 1 にアクセスし、データはヌルテーブル t 2 中に同期化され、上位ビューは、データの同期化が完了するとテーブル t 2 を指し示すように再定義される。テーブル t 1 内のデータを消去するために、以下のコマンドが実行される。

```
truncate table t1
```

【 0 0 4 3 】

Truncate（切り捨て）の動作は、Delete（削除）の動作とは原則的に異なり、前者の使用の方が、より速いと思われる。

20

【 0 0 4 4 】

データがテーブル t 2 中に同期化され、上位ビューが再定義された後、アプリケーションサーバは、テーブル t 2 内のデータにアクセスし、次に同期化されるときデータは、ヌルテーブル t 1 へ同期化される。図 5 は、アプリケーションサーバがテーブル t 2 にアクセスし、データがヌルテーブル t 1 へ同期化されるケースを示している。データ同期化が完了すると、上位ビューは、テーブル t 1 を指し示すように再定義され、次いで、以下のコマンドが実行される。

```
truncate table t2
```

【 0 0 4 5 】

テーブル t 2 のデータは、消去される。

30

【 0 0 4 6 】

図 6 及び図 7 は、ソースデータテーブル（例：バックエンドデータウェアハウス内のデータテーブル）とターゲットデータテーブル（例：フロントエンドORACLEデータベース内のデータテーブル）との間でデータを同期化するためのデータ同期化プロセスの実施形態を示したフローチャートである。これらの例は、ターゲットデータテーブル t 1 及びターゲットデータテーブル t 2 が事前に確立されていることを前提とする。以下の実施形態で説明されるプロセスは、ウェブサーバ上、データベースサーバ上、又はその他の適切なデバイス上で実行され得る。

【 0 0 4 7 】

図 6 では、2つのターゲットデータテーブル内のレコードの数が順次決定される。一方のターゲットデータテーブル内のレコード数がゼロに決定されるときは、もう一方のターゲットデータテーブル内のレコード数を決定する必要はない。ステップ 601 では、テーブル t 1 内のレコード数がカウントされ、その結果が N 1 として保存される。一部の実施形態では、以下のコマンドを使用して、テーブル t 1 内のレコード数がカウントされ変数 N 1 として保存される。

40

```
Select count(*) into N1 from t1 where rownum<2
```

【 0 0 4 8 】

ここで、rownumは、データレコードを意味し、rownum<2は、データテーブル内に少なくとも1つのレコードが検出される限り、これ以上データレコードを更に検出しなくてもデータの存在を決定できることを示している。

50

【 0 0 4 9 】

ステップ 6 0 2 では、テーブル t 1 内にデータが存在するか否かが決定される。この例では、テーブル t 1 内にデータが存在するか否かは、カウントされたレコード数 N 1 がゼロに等しいか否かを決定することによって決定される。等しい場合に、プロセスは、同期化のために 6 0 3 に進み、等しくない場合には、プロセスは、テーブル t 2 がデータを有するか否かを決定するためにステップ 1 1 1 に進む。

【 0 0 5 0 】

ステップ 6 0 3 では、データが同期化され、バックエンドデータウェアハウスからテーブル t 1 へ取り込まれる。各種の実施形態では、データは、ORACLEEXTENTによってサポートされるセグメント式コミットメント法などの各種の同期化方法を使用して同期化される。

10

【 0 0 5 1 】

ステップ 6 0 4 では、同期化が成功したか否かが決定される。成功した場合には、ステップ 6 0 5 が実行され、成功しなかった場合には、異常を処理するためにステップ 6 0 8 が実行される。異常が発生すると、データ同期化プロセスは終了される。警告メッセージ及び/又はログメッセージが生成されても良い。

【 0 0 5 2 】

ステップ 6 0 5 では、新しい上位ビューが作成される。新しい上位ビューは、テーブル t 1 を指し示すように作成されるので、アプリケーションサーバは、上位ビューを通してテーブル t 1 にアクセスすることができる。

20

【 0 0 5 3 】

ステップ 6 0 6 では、テーブル t 2 に対して消去動作が実施される。一部の実施形態では、古いデータを速やかに削除してテーブル t 2 の内容を消去するために、テーブル t 2 に対して消去動作が実施されるときに、上記のテーブル切り捨てコマンドが使用される。

【 0 0 5 4 】

ステップ 6 0 7 では、データ同期化プロセスは終了する。

【 0 0 5 5 】

ステップ 6 1 1 では、テーブル t 2 内のレコード数がカウントされ、その値が N 2 として保存される。

【 0 0 5 6 】

以下のコマンドを使用して、テーブル t 2 内のレコードの数がカウントされ変数 N 2 として保存される。

```
Select count(*) into N2 from t2 where rownum<2
```

【 0 0 5 7 】

ステップ 6 1 2 では、テーブル t 2 内にデータが存在するか否かが決定される。すなわち、カウントされたレコード数 N 2 がゼロに等しいか否かが決定される。等しい場合には、プロセスは、データ同期化を実施するためにステップ 6 1 3 に進み、等しくない場合には、両テーブル内におけるデータの存在は、異常を示しており、プロセスは、異常を処理するためにステップ 6 0 8 に進む。

【 0 0 5 8 】

ステップ 6 1 3 では、データが同期化され、バックエンドデータウェアハウスからテーブル t 2 に取り込まれる。各種の実施形態では、データは、ORACLEEXTENTによってサポートされるセグメント式コミットメント法などの各種の同期化方法を使用して同期化される。

40

【 0 0 5 9 】

ステップ 6 1 4 では、同期化が成功したか否かが決定される。成功した場合には、プロセスは、ステップ 6 1 5 に進み、成功しなかった場合には、異常があり、プロセスは、異常を処理するためにステップ 6 0 8 に進む。

【 0 0 6 0 】

ステップ 6 1 5 では、新しい上位ビューが作成される。データ同期化が成功すると、同

50

期化されたテーブル、すなわちテーブル t 2 を指し示す上位ビューが作成しなおされるので、アプリケーションサーバは、上位ビューを通してテーブル t 2 にアクセスすることができる。

【 0 0 6 1 】

ステップ 6 1 6 では、テーブル t 1 に対して消去動作が実施される。一部の実施形態では、古いデータを速やかに削除してテーブル t 1 の内容を消去するために、上記のテーブル切り捨てコマンドが実行される。

【 0 0 6 2 】

ステップ 6 0 7 では、プロセスは終了する。

【 0 0 6 3 】

図 7 では、2 つのターゲットデータテーブル内のレコード数が同時にカウントされ、それらのうち、カウント数がゼロであるターゲットデータテーブルが選択される。

【 0 0 6 4 】

ステップ 7 0 1 では、テーブル t 1 内のレコード数及びテーブル t 2 内のレコード数が、それぞれ N 1 及び N 2 としてカウントされる。

【 0 0 6 5 】

テーブル t 1 内のレコード数及びテーブル t 2 内のレコード数は、処理中に、やはり上記の rownum<2 のやり方でそれぞれ N 1 及び N 2 としてカウントすることができ、具体的なコマンド行は、以下の通りである。

```
Select count(*) into N1 from t1 where rownum<2
```

```
Select count(*) into N2 from t2 where rownum<2
```

【 0 0 6 6 】

ステップ 7 0 2 では、テーブル t 1 及びテーブル t 2 にデータが存在するか否かが決定される。一部の実施形態では、テーブル t 1 内のレコード数 N 1 及びテーブル t 2 内のレコード数 N 2 がゼロに等しいか否かが決定される。

【 0 0 6 7 】

N1=0 且つ N2 = 0 は、テーブル t 1 にはデータがなく、テーブル t 2 にはデータがあることを示し、したがって、プロセスは、ステップ 7 0 3 に進む。

【 0 0 6 8 】

ステップ 7 0 3 では、バックエンドデータウェアハウスからテーブル t 1 中にデータが同期化される。

【 0 0 6 9 】

ステップ 7 0 4 では、同期化が成功したか否かが決定される。成功した場合には、プロセスは、ステップ 7 0 5 に進み、成功しなかった場合には、プロセスは、異常状態を処理するためにステップ 7 3 3 に進む。

【 0 0 7 0 】

ステップ 7 0 5 では、新しい上位ビューが作成される。新しい上位ビューは、テーブル t 1 を指し示すように作成されるので、アプリケーションサーバは、上位ビューを通してテーブル t 1 にアクセスすることができる。

【 0 0 7 1 】

ステップ 7 0 6 では、テーブル t 2 に対して消去動作が実施される。一部の実施形態では、消去動作を実施するために、上述の切り捨てコマンドが使用される。

【 0 0 7 2 】

プロセスは、ステップ 7 0 7 で終了する。

【 0 0 7 3 】

N1 = 0 且つ N2=0 は、テーブル t 1 にはデータがあり、テーブル t 2 にはデータがないことを示し、したがって、プロセスは、ステップ 7 1 3 に進む。

【 0 0 7 4 】

ステップ 7 1 3 では、バックエンドデータウェアハウスからテーブル t 2 中にデータが同期化される。

10

20

30

40

50

【 0 0 7 5 】

ステップ 7 1 4 では、同期化が成功したか否かが決定される。成功した場合には、プロセスは、新しい上位ビューを作成するためにステップ 7 1 5 に進み、成功しなかった場合には、異常が発生しており、プロセスは、異常を処理するためにステップ 7 3 3 に進む。

【 0 0 7 6 】

データ同期化が成功すると、ステップ 7 1 5 では、同期化されたデータテーブル t 2 を指し示すように上位ビューが作成しなおされるので、アプリケーションサーバは、上位ビューを通してテーブル t 2 にアクセスすることができる。

【 0 0 7 7 】

ステップ 7 1 6 では、テーブル t 1 に対して消去動作が実施される。

10

【 0 0 7 8 】

プロセスは、ステップ 7 0 7 で終了する。

【 0 0 7 9 】

$N1=0$ 且つ $N2=0$ は、テーブル t 1 にもテーブル t 2 にもデータが存在しないことを示し、プロセスは、ステップ 7 2 3 に進む。

【 0 0 8 0 】

ステップ 7 2 3 では、テーブル t 1 及びテーブル t 2 のなかからデータテーブルが選択される。この選択は、任意の選択であってよい。テーブル t 1 が選択される場合には、7 0 3 ~ 7 0 7 が実施される。テーブル t 2 が選択される場合には、7 1 3 ~ 7 1 6 及び 7 0 7 が実施される。

20

【 0 0 8 1 】

$N1 = 0$ 且つ $N2 = 0$ は、テーブル t 1 にもテーブル t 2 にもデータが存在することを示している。これは、異常状態だと見なされる。プロセスは、異常を処理するためにステップ 7 3 3 に進み、終了する。

【 0 0 8 2 】

図 8 は、ソースデータテーブルと現ターゲットデータテーブルとを同期化するためのプロセスの一実施形態を示したフローチャートである。プロセスは、図 6 のステップ 6 0 3 及び 6 1 3、並びに図 7 のステップ 7 0 3 及び 7 1 3 を実施するために使用され得る。

【 0 0 8 3 】

ステップ 8 0 1 では、ソースデータテーブル内において同期化予定データによって占有されているストレージ範囲が決定される。一部の実施形態では、ソースデータテーブル内において同期化予定データによって占有されているストレージ範囲を得るために、ORACLE データベースのデータディクショナリビューに対してクエリが行われる。

30

【 0 0 8 4 】

ステップ 8 0 2 では、上記ストレージ範囲内における同期化予定データのストレージアドレス情報が決定される。このストレージアドレス情報は、少なくとも、それぞれのストレージ範囲内における同期化予定データのアドレス域を含む。それぞれのストレージ範囲内において同期化予定データが保存されている最小アドレス (e_min_rowid) 及び最大アドレス (e_max_rowid) が、ORACLE データベースのデータディクショナリビューから導出される、ソースデータテーブル内において同期化予定データによって占有されているストレージ範囲を基にして計算される。一部の実施形態では、導出されたアドレス域は、例えばテーブル tmp_rowid などの中間テーブルに保存される。中間テーブルのテーブル構造の一例は、以下の通りである。

40

```
create table tmp_rowid
(
  id number;
  ROWID_MIN VARCHAR2 (100),
  ROWID_MAX VARCHAR2 (100)
)
```

【 0 0 8 5 】

50

ソースデータテーブル内のそれぞれのストレージ範囲内における同期化予定データのアドレス情報は、必要に応じて中間テーブルに対してクエリを行うことによって読み出される。

【 0 0 8 6 】

ステップ 8 0 3 では、それぞれのストレージ範囲内の同期化予定データが、ストレージアドレス情報にしたがって現ターゲットデータテーブルへ同期化される。あるストレージ範囲内のデータの同期化が失敗した場合には、そのストレージ範囲内のデータを同期化するプロセスは、やり直しされ、やり直しの回数がプリセット閾値を超えると終了する。

【 0 0 8 7 】

同期化は幾つかの方法で実施することができる。例えば、第 1 の実施形態では、それぞれのストレージ範囲内のデータは、ストレージアドレス情報にしたがって、1 つのデータ同期化プロセスにおいて順次、現ターゲットデータテーブルへ同期化される。第 2 の実施形態では、決定されたストレージ範囲は、N 個のグループに分割され、対応する N 個のデータ同期化プロセスが開始され、各データ同期化プロセスでは、対応する一グループのそれぞれのストレージ範囲内のデータが、ストレージアドレス情報にしたがって、順次、現ターゲットデータテーブルへ同期化される。第 2 の実施形態は、通常、ORACLE データベースを必要とする。第 2 の実施形態は、アプリケーションサーバを特定のデータから隔てるために、上位ビューを使用する。データウェアハウスからデータが同期化される、フロントエンド ORACLE データベース内のデータテーブルは、例えば、アプリケーションサーバによって現時点で呼び出されているデータが存在する任意のテーブルなどの、システムによって現時点で使用されているどのテーブルとも同じではない。データウェアハウスからフロントエンド ORACLE データベース内のデータテーブルへのデータ同期化、及びアプリケーションサーバによってデータを呼び出す動作は、同じデータテーブル上では実施されないため、これらの動作に起因するアクセスの対立は取り除かれる。

【 0 0 8 8 】

更に、ORACLE データベースへのデータの保存は、以下の特徴を有する。

【 0 0 8 9 】

ORACLE データベース内のデータテーブルは、数々のストレージ範囲を含み、各範囲は、データの中に保存される固定サイズのブロックの連なりを含む。一範囲は、データテーブルスペースの最小単位であり、1 つの範囲と別の範囲は、それらの物理的位置の観点からみて連続的でないことがある。しかしながら、各範囲の内部には、連続した物理アドレスを有する一連のブロックがある。そのストレージフォーマットは、図 9 に示されるようなフォーマットである。

【 0 0 9 0 】

ORACLE データベースへのデータの保存の上記特徴に基づき、データ同期化プロセスがアプリケーションサーバのアクセスプロセスと対立しないとされるときは、同期化予定のデータは、セグメント式にコミットすることができる、すなわち、データ同期化手続きは、サービスシステムによってアクセスされるデータの正確さに何ら影響することなく尚且つ向上されたデータ同期化速度で、第 2 の実施形態において可能とされ得る。

【 0 0 9 1 】

図 1 0 は、ソースデータテーブル（例：バックエンドデータウェアハウス）から現ターゲットデータテーブル（例：フロントエンドテーブル t 1 又は t 2）へセグメント式にデータを同期化するためのプロセスの一実施形態を示したフローチャートである。

【 0 0 9 2 】

ステップ 1 0 0 1 では、同期化予定データによって占有されているストレージ範囲に関してクエリが行われる。一部の実施形態では、ソースデータテーブル内において同期化予定データによって占有されているストレージ範囲を導出するために、ソースデータテーブルがあるバックエンドデータウェアハウスのデータディクショナリビューに対してクエリが行われる。

【 0 0 9 3 】

10

20

30

40

50

ステップ1002では、ソースデータテーブル内において各ストレージ範囲内の同期化予定データが保存されているアドレス域が決定される。一部の実施形態では、アドレス域は、例えばテーブルtmp_rowidなどの保存された中間テーブルに対してクエリを行うこと
によって決定される。

【0094】

ステップ1003では、同期化予定データを保存されているストレージ範囲がN個のグループに分割される。

【0095】

テーブルtmp_rowid内のデータは、例えば識別子(ID)モジュールによってN個のグループに分割するなどの数々のやり方でグループ分けすることができる。データは、次いで、グループごとに処理される。

10

【0096】

ステップ1004では、N個のグループに分割されたそれぞれのストレージ範囲内のデータを現ターゲットデータテーブルへ同期化するために、N個のデータ同期化プロセスが開始される。

【0097】

データは、各ストレージ範囲グループにそれぞれ対応するN個のデータ同期化プロセスで並列に処理される。同期化プロセスは、対立することなく同時に実行される。それぞれのストレージ範囲内のデータの同期化に並列処理を使用することによって、データ同期化の効率が向上される。

20

【0098】

ステップ1005aでは、現グループ(例:第1のグループ)のそれぞれのストレージ範囲内の同期化予定データが、順次、処理される。一部の実施形態では、現グループのストレージ範囲内の同期化予定データが処理され、同期化の成功を確認するメッセージの受信を受けて、現グループの次のストレージ範囲内のデータを同期化するプロセスが開始される。

【0099】

ステップ1006aでは、同期化が成功したか否かが決定される。一部の実施形態では、現ストレージ範囲内の同期化予定データ同期化が成功したか否かが決定される。成功した場合には、次いで、ステップ1008aが実行され、成功しなかった場合には、そのストレージ範囲内のデータを同期化する手続きを再び実行するためにロールバックが必要であり、ステップ1007aが実行される。

30

【0100】

ステップ1007aでは、ロールバックの回数がプリセット閾値を超えるか否かが決定される。一部の実施形態では、ロールバック(すなわち、再試行)の回数の閾値がプリセットされ、再試行の回数がカウントされる。再試行の回数がプリセット閾値を上回る場合には、これ以上再試行はなされず、そのストレージ範囲内のデータの同期化の失敗を示す表示が生成されるとともにループが終了する。再試行の方針は、データ同期化の誤差耐性を向上させるためにデータ同期化手順において採用される。再試行の回数がプリセット閾値未満である場合には、フローは、そのストレージ範囲内のデータをターゲットデータテーブルへ再び同期化するステップ1005aに戻って当該ステップを再び実行し、そうでない場合には、1009aが実行され、必要に応じて失敗の表示が生成される。

40

【0101】

現ストレージ範囲内のデータの同期化が成功した場合には、ステップ1008aにおいて、同期化されたデータがターゲットデータテーブルに対してコミットされる。

【0102】

ステップ1009aでは、処理が終わったか否かが決定される、すなわち、現グループ内の全てのストレージ範囲が処理されたか否かが決定される。処理が終わっている場合には、グループ内の全てのストレージ範囲内の対応データが同期化されたことが示され、プロセスは終了する。必要に応じて、失敗の表示が生成される。処理が終わっていない場合

50

、換言すると、グループ内の全てのストレージ範囲内の対応データが同期化されていない場合には、フローは、グループ内の次のストレージ範囲内のデータを同期化するステップ 1005a に戻って当該ステップを実行する。

【0103】

グループのなかでデータ同期化を終えたストレージ範囲の数を相応に示すために、各グループについて、並列処理中にカウンタ変数 v_done_num を設定することができる。この結果、各プロセスによるデータ同期化の速度及びデータ同期化の全体速度の計算を促すことができる。データ同期化の効率は、上記の統計値に基づいて便利に監視することもできる。

【0104】

ステップ 1005b ~ 1009b は、第 2 のグループのそれぞれのストレージ範囲内のデータを順次、同期化する手続きである。ステップ 1005x ~ 1009x は、第 N グループのそれぞれのストレージ範囲内のデータを順次、同期化する手続きである。フローチャートには、中間グループ用の更なるステップが存在し得る。

【0105】

ステップ 1005b ~ 1009b、ステップ 1005x ~ 1009x、及びあらゆる中間ステップの処理フローは、ステップ 1005a ~ 1009a の処理フローに対応して同じであり、ここでは、説明の繰り返しは省略される。

【0106】

ステップ 1010 では、同期化プロセスは終了する。

【0107】

ソースデータテーブルからターゲットデータテーブルへの上記のデータ同期化手続きにおいて、複数のデータ範囲内のデータが一度に処理されるように複数のプロセスの並列処理が採用される場合には、これらのデータ範囲内のデータは、コミット動作の数を減らすために、そして更に、同期化予定データをコミットする速度及び効率を高めるために、1 つのコミット動作でターゲットデータテーブルに対してコミットすることができる。同時に処理されるストレージ範囲の数は、実際の実装形態に応じて調整することができる。

【0108】

図 8 ~ 10 に示された、ソースデータテーブルから現ターゲットデータテーブルへデータを同期化する上記の手続きは、ソースデータテーブルからターゲットデータテーブルへのデータの同期化が必要とされるその他の手続きに個別に適用可能であるが、本発明の実施形態で提示されたデータ同期化手続きに提示された応用に限定はされない。

【0109】

発明の実施形態における上記のデータ同期化方法にしたがって、図 11 に示されるようなデータ同期化デバイスを構築することができ、該デバイスは、データテーブル作成モジュール 10、決定モジュール 20、データ同期化モジュール 30、及びアクセス&更新モジュール 40 を含む。これらのモジュールは、1 つ若しくは 2 つ以上の汎用プロセッサ上で実行されるソフトウェアコンポーネントとして、又は特定の機能を実施するように設計されたプログラマブルロジックデバイス及び/若しくは特殊用途向け集積回路などのハードウェアとして、又はそれらの組み合わせとして実装することができる。一部の実施形態では、これらのモジュールは、不揮発性ストレージメディア（光ディスク、フラッシュストレージデバイス、モバイルハードディスクなど）に格納可能で尚且つ本発明の実施形態で説明された方法をコンピュータデバイス（パソコン、サーバ、ネットワーク機器など）に実施させるための幾つかの命令を含むソフトウェア製品の形態で実施することができる。これらのモジュールは、1 つのデバイス上に実装されてよい、又は複数のデバイスに分散されてよい。これらのモジュールの機能は、互いに融合されて 1 つにされてよい、又は更に複数のサブ機能に分けられてよい。

【0110】

この例では、データテーブル作成モジュール 10 は、同期化予定のデータを保存されているソースデータテーブルに対応する 2 つのターゲットデータテーブルを予め作成するよ

10

20

30

40

50

うに構成される。

【0111】

決定モジュール20は、2つのターゲットデータテーブルから現ターゲットデータテーブルを決定するように構成される。

【0112】

好ましくは、決定モジュール20は、カウントユニット201及び決定ユニット202を更に含むことができる。

【0113】

カウントユニット201は、2つのターゲットデータテーブル内のそれぞれのデータレコードをカウントするように構成される。

10

【0114】

決定ユニット202は、ターゲットデータテーブルのうちカウントユニット201によってカウントされたデータレコード数がゼロである1つのターゲットデータテーブルを現ターゲットデータテーブルとして決定するように構成される。

【0115】

決定ユニット202は、更に、カウントユニット201によってカウントされた2つのターゲットデータテーブル内のデータレコード数がいずれもゼロでないときに異常を示してデータ同期化フローを終了させるように構成される。

【0116】

データ同期化モジュール30は、ソースデータテーブルから決定モジュール20によって決定された現ターゲットデータテーブルへデータを同期化するように構成される。

20

【0117】

好ましくは、データ同期化モジュールは、更に、ストレージ範囲決定ユニット301、アドレス情報決定ユニット302、及びデータ同期化ユニット303を含むことができる。

【0118】

ストレージ範囲決定ユニット301は、ソースデータテーブル内において同期化予定データによって占有されている幾つかのストレージ範囲を決定するように構成される。

【0119】

アドレス情報決定ユニット302は、それぞれのストレージ範囲内における同期化予定データのストレージアドレス情報を決定するように構成される。

30

【0120】

データ同期化ユニット303は、アドレス情報決定ユニット302によって決定されたストレージアドレス情報にしたがって、それぞれのストレージ範囲内の同期化予定データを現ターゲットデータテーブルへ同期化するように構成される。

【0121】

好ましくは、データ同期化ユニット303は、更に、第1のデータ同期化サブユニット3031及び第2のデータ同期化サブユニット3032を含むことができる。

【0122】

第1のデータ同期化サブユニット3031は、アドレス情報決定ユニット302によって決定されたストレージアドレス情報にしたがって、それぞれのストレージ範囲内の同期化予定データを1つのデータ同期化プロセスにおいて順次、現ターゲットデータテーブルへ同期化するように構成される。

40

【0123】

第2のデータ同期化サブユニット3032は、ストレージ範囲をN個のグループに分割し、対応するN個のデータ同期化プロセスを開始させ、アドレス情報決定ユニット302によって決定されたストレージアドレス情報にしたがって、対応するグループのそれぞれのストレージ範囲内のデータを各データ同期化プロセスにおいて順次、現ターゲットデータテーブルへ同期化するように構成される。

【0124】

50

データ同期化ユニット 303 は、更に、同期化が失敗したときにそのストレージ範囲内のデータを同期化する手続きをやり直すように、及びやり直しの回数がプリセット閾値を超えたときにそのストレージ範囲内のデータを同期化する手続きを終了させるように適応されたやり直しサブユニット 3033 を含む。

【0125】

アクセス&更新モジュール 40 は、データ同期化モジュール 30 がデータ同期化に成功したときに、決定モジュール 20 によって決定された現ターゲットデータテーブルにアプリケーションサーバがアクセスすることを可能にするように構成される。

【0126】

アクセス&更新モジュール 40 は、ターゲットデータテーブルのうち上位ビューが指し示す 1 つのターゲットデータテーブルにアプリケーションサーバがその上位ビューを通してアクセスすることができるように予め上位ビューを作成するように、及びデータ同期化の完了を受けて上位ビューが現ターゲットデータテーブルを指し示すことを可能にするように上位ビューを更新するように、特に構成される。

10

【0127】

上記のデータ同期化デバイスは、更に、データ同期化モジュール 30 がデータ同期化に成功したときに、2 つのターゲットデータテーブルのうちデータ同期化を受けていない方のターゲットデータテーブルの内容を消去するように適応されたデータテーブル消去モジュール 50 を含む。

【0128】

20

上記のデータ同期化デバイスのデータ同期化モジュール 30 は、データ同期化のための個別の手段として用意することができ、任意のデータ転送手続きにおいて個別に使用することができる。

【0129】

図 8 及び図 10 に示された、ソースデータテーブルからターゲットデータテーブルへデータを同期化する上記のフローにしたがって、図 12 に示されるような、グループ分けユニット 121 及びデータ同期化ユニット 122 を含む複数プロセスデータ同期化デバイスを構築することができる。これらのユニットは、1 つ若しくは 2 つ以上の汎用プロセッサ上で実行されるソフトウェアコンポーネントとして、又は特定の機能を実施するように設計されたプログラマブルロジックデバイス及び/若しくは特殊用途向け集積回路などのハードウェアとして、又はそれらの組み合わせとして実装することができる。一部の実施形態では、これらのユニットは、不揮発性ストレージメディア（光ディスク、フラッシュストレージデバイス、モバイルハードディスクなど）に格納可能で尚且つ本発明の実施形態で説明された方法をコンピュータデバイス（パソコン、サーバ、ネットワーク機器など）に実施させるための幾つかの命令を含むソフトウェア製品の形態で実施することができる。これらのユニットは、この例では、1 つのデバイス内にあるものとして示されているが、その他の実施形態では、複数のデバイスに分散されてよい。これらのユニットの機能は、互いに融合されて 1 つにされてよい、又は更に複数のサブ機能に分けられてよい。

30

【0130】

グループ分けユニット 121 は、ソースデータテーブル内で同期化予定データによって占有されているストレージ範囲を幾つかのグループに分割するように構成される。

40

【0131】

データ同期化ユニット 122 は、対応するデータ同期化プロセスを並列に開始させるように、及び各データ同期化プロセスにおいて、対応する一グループのそれぞれのストレージ範囲内の同期化予定データを現ターゲットデータテーブルへ同期化するように構成される。

【0132】

好ましくは、データ同期化ユニット 122 は、制御実行サブユニット 1221 及び決定サブユニット 1222 を更に含むことができる。

【0133】

50

制御実行サブユニット1221は、対応するグループのストレージ範囲内の同期化予定データを処理するように、及び対応するグループの全てのストレージ範囲が処理されたのではないと決定サブユニットによって決定されたときに、対応するグループの次のストレージ範囲内の同期化予定データを処理する手続きを開始させるように、各データ同期化プロセスを制御するように構成される。

【0134】

決定サブユニット1222は、ストレージ範囲内の同期化予定データが同期化されて正常にコミットされた後に、対応するグループの全てのストレージ範囲が処理されたかどうかを決定するように構成される。

【0135】

データ同期化ユニット1222は、同期化が失敗したときにそのストレージ範囲内のデータを同期化する手続きをやり直すように、及びやり直しの回数がプリセット閾値を超えたときにそのストレージ範囲内のデータを同期化する手続きを終了させるように適応されたやり直しサブユニット1223を更に含む。

【0136】

上記のデータ同期化デバイスは、更に以下の構成を含む。

【0137】

ソースデータテーブル内において同期化予定データによって占有されているストレージ範囲を決定するように適応されたストレージ範囲決定ユニット123と、それぞれのストレージ範囲内における同期化予定データのストレージアドレス情報を決定するように適応されたアドレス情報決定ユニット124。

【0138】

本発明の実施形態にしたがったデータ同期化の方法及びデバイスでは、同期化予定データを保存されているソースデータテーブルに対応するターゲットデータテーブルが予め作成され、アプリケーションサーバは、予め作成されている上位ビューを通して、ターゲットデータテーブルのうちその上位ビューが指し示す1つのターゲットデータテーブルにアクセスするので、アプリケーションレイヤは、データが特にどのテーブルから発しているかに注意を払うことはなく、したがって、アプリケーション実装の複雑性が抑えられる。

【0139】

ソースデータテーブルからターゲットデータテーブルへデータが同期化されるときは、まず、2つのターゲットデータテーブルのうち1つのターゲットデータテーブルが現ターゲットデータテーブルとして決定され、次いで、データが、ソースデータテーブルから現ターゲットデータテーブルへ同期化され、そして、データの同期化が成功すると、上位ビューが、現ターゲットデータテーブルを指し示すように更新される。上記の方法では、2つのターゲットデータテーブルは、交互に使用され、アプリケーションサーバによってアクセスされるデータテーブルと、データ同期化プログラムによってアクセスされるデータテーブルは、対立しないので、大量のデータがアンドゥ/ロールバックテーブルスペースへ転送される事態が回避され、それによって、アンドゥ/ロールバックテーブルスペースが満杯にされた後に生じる行列待ち及び混雑の現象が効果的に回避され、また、データ同期化の速度及び効率が向上される。

【0140】

データ同期化の手続きでは、それぞれのストレージ範囲内のデータをターゲットデータテーブルへ同期化するために、それぞれのストレージ範囲に対してデータ移動を実施するために、並びにストレージ範囲ごとにデータを並列に同期化及びコミットするために、ORACLEテーブルのスペース割り当て及びストレージ原理と併せてORACLEデータテーブルの固有なストレージメカニズムを用いることができるので、データ同期化の手続きでは、フロントエンドORACLEデータベースにおけるアンドゥ/ロールバックテーブルスペースの長期に及ぶ占有を抑えること、アンドゥ/ロールバックテーブルスペースの全体の利用率を大幅に下げること、アンドゥ/ロールバックテーブルスペースが満杯にされる可能性を大幅に減らすこと、及び生産データベースのセキュリティを保証することが可能である。デー

10

20

30

40

50

タ同期化のために、ストレージ範囲は、データを並列に移動可能にしてデータ移動の効率を更に高めるようにグループ分けすることもできる。

【0141】

一ストレージ範囲内のデータの同期化が失敗したときは、ロールバックメカニズムの採用が、システムの誤差耐性を大幅に向上させることができ、ロールバック回数制御メカニズムが、例えば無限ロールバックなどのエンドレスループの発生を阻止することができる。動作は、アプリケーションサーバによってアクセスされるテーブルスペースと何ら対立することなくそれぞれのストレージ範囲に対して実施され、初期状態において、使用される現データテーブルはヌルであり対立データを含まないため、データ同期化の中断時もデータのうち同期化されていない部分の同期化を継続させることができ、これは、断絶されたデータ移送の復旧を可能にする。

10

【0142】

更に、データ同期化を終えたストレージ範囲をカウントし、それによってデータ同期化の進行、速度、及び効率を監視するために、カウンタ変数を用意することができる。

【0143】

以上の説明は、発明の好ましい実施形態を例示したものにすぎず、発明の特許請求の範囲は、それらに限定されず、発明の技術的開示に照らして当業者が容易に思い付く任意の変形、代替、又はその他の類似デバイスへの応用が、発明の特許請求の範囲に含まれるものとする。したがって、発明の特許請求の範囲は、添付の特許請求の範囲として定められるものとする。

20

【0144】

以上の実施形態は、理解を明瞭にする目的で幾らか詳細に説明されてきたが、発明は、提供された詳細に限定されず、発明を実現するには、多くの代替的方法がある。開示された実施形態は、例示的なものであり、限定的なものではない。

適用例 1：データ同期化方法は、同期化予定のデータを保存されているソースデータテーブルに基づいて複数のターゲットデータテーブルを確立することと、前記複数のターゲットデータテーブルのなかから現ターゲットデータテーブルを決定することと、前記ソースデータテーブルと前記現ターゲットデータテーブルとを同期化することと、同期化が成功した場合に、前記現ターゲットデータテーブルにアクセスするようにアプリケーションサーバに指示することとを備える。

30

適用例 2：適用例 1 に記載の方法であって、前記アプリケーションサーバは、前記現ターゲットデータテーブルを指し示すように事前に更新された上位ビューを通じて前記現ターゲットデータテーブルにアクセスするように指示される、方法。

適用例 3：請求項 1 または 2 に記載の方法であって、前記複数のターゲットデータテーブルのなかから前記現ターゲットデータテーブルを決定することは、前記複数のターゲットデータテーブル内のそれぞれのデータレコードの数を決定することと、前記ターゲットデータテーブルのうち前記データレコードの数がゼロである 1 つのターゲットデータテーブルを前記現ターゲットデータテーブルとして選択することを含む、方法。

適用例 4：適用例 1 ~ 3 のいずれかに記載の方法であって、前記ソースデータテーブルと前記現ターゲットデータテーブルとを同期化することは、前記ソースデータテーブルから前記現ターゲットデータテーブルへデータを直接コピーすることを含む、方法。

40

適用例 5：請求項 1 に記載の方法であって、前記ソースデータテーブルと前記現ターゲットデータテーブルとを同期化することは、旧ターゲットデータテーブルを削除すること、現ターゲットデータテーブルとして新しいターゲットデータテーブルを作成すること、及び前記データを前記新しく作成された現ターゲットデータテーブルへ同期化することを含む、方法。

適用例 6：適用例 1 ~ 5 のいずれかに記載の方法であって、前記ソースデータテーブルと前記現ターゲットデータテーブルとを同期化することは、前記ソースデータテーブル内において同期化予定データによって占有されている幾つかのストレージ範囲を決定することと、前記それぞれのストレージ範囲内における前記同期化予定データのストレージアドレ

50

ス情報を決定することと、前記ストレージアドレス情報を使用して、前記それぞれのストレージ範囲内の前記同期化予定データを前記現ターゲットデータテーブルへ同期化することと、を含む、方法。

適用例 7：適用例 6 に記載の方法であって、前記ストレージアドレス情報を使用して、前記それぞれのストレージ範囲内の前記同期化予定データを前記現ターゲットデータテーブルへ同期化することは、前記ストレージアドレス情報にしたがって、前記それぞれのストレージ範囲内の前記データを 1 つのデータ同期化プロセスにおいて順次、前記現ターゲットデータテーブルへ同期化することを含む、方法。

適用例 8：適用例 6 に記載の方法であって、前記ストレージアドレス情報を使用して、前記それぞれのストレージ範囲内の前記同期化予定データを前記現ターゲットデータテーブルへ同期化することは、前記ストレージ範囲を N 個のグループに分割することと、

対応する N 個のデータ同期化プロセスを開始させることと、前記ストレージアドレス情報にしたがって、前記グループのうちの対応する 1 つのグループの前記それぞれのストレージ範囲内の前記データを、前記データ同期化プロセスのうちの各プロセスにおいて順次、前記現ターゲットデータテーブルへ同期化することと、を含む、方法。

適用例 9：適用例 6 に記載の方法であって、更に、前記同期化が失敗した場合に、前記それぞれのストレージ範囲内の前記データを前記現ターゲットデータテーブルへ同期化し直すことと、やり直しの回数がプリセット閾値を超えたときに、前記ストレージ範囲内の前記データを同期化することを終了させることと、を備える方法。

適用例 10：適用例 1～9 に記載の方法であって、更に、前記ソースデータテーブルと前記現ターゲットデータテーブルとの同期化が成功した場合、前記現ターゲットデータテーブル以外のターゲットデータテーブルの内容を消去することを備える方法。

適用例 11：データ同期化システムであって、1 つ又は 2 つ以上のプロセッサであって、同期化予定のデータを保存されているソースデータテーブルに基づいて複数のターゲットデータテーブルを確立するように、前記複数のターゲットデータテーブルのなかから現ターゲットデータテーブルを決定するように、前記ソースデータテーブルと前記現ターゲットデータテーブルとを同期化するように、及び同期化が成功した場合に、前記現ターゲットデータテーブルにアクセスするようにアプリケーションサーバに指示するように、構成されたプロセッサと、

前記 1 つ又は 2 つ以上のプロセッサに結合され、前記 1 つ又は複数のプロセッサに命令を提供するように構成されたメモリと、を備えるシステム。

適用例 12：適用例 11 に記載のシステムであって、前記アプリケーションサーバは、前記現ターゲットデータテーブルを指し示すように事前に更新された上位ビューを通じて前記現ターゲットデータテーブルにアクセスするように指示される、システム。

適用例 13：適用例 11 または 12 に記載のシステムであって、前記複数のターゲットデータテーブルのなかから前記現ターゲットデータテーブルを決定することは、前記複数のターゲットデータテーブル内のそれぞれのデータレコードの数を決定することと、前記ターゲットデータテーブルのうち前記データレコードの数がゼロである 1 つのターゲットデータテーブルを前記現ターゲットデータテーブルとして選択することと、を含む、システム。

適用例 14：適用例 11～13 のいずれかに記載のシステムであって、前記ソースデータテーブルと前記現ターゲットデータテーブルとを同期化することは、前記ソースデータテーブルから前記現ターゲットデータテーブルへデータを直接コピーすることを含む、システム。

適用例 15：適用例 11～14 のいずれかに記載のシステムであって、前記ソースデータテーブルと前記現ターゲットデータテーブルとを同期化することは、旧ターゲットデータテーブルを削除すること、現ターゲットデータテーブルとして新しいターゲットデータテーブルを作成すること、及び前記データを前記新しく作成された現ターゲットデータテーブルへ同期化することを含む、システム。

適用例 16：適用例 11～15 のいずれかに記載のシステムであって、前記ソースデー

10

20

30

40

50

テーブルと前記現ターゲットデータテーブルとを同期化することは、前記ソースデータテーブル内において同期化予定データによって占有されている幾つかのストレージ範囲を決定することと、前記それぞれのストレージ範囲内における前記同期化予定データのストレージアドレス情報を決定することと、前記ストレージアドレス情報を使用して、前記それぞれのストレージ範囲内の前記同期化予定データを前記現ターゲットデータテーブルへ同期化することと、を含む、システム。

適用例 17：適用例 16 に記載のシステムであって、前記ストレージアドレス情報を使用して、前記それぞれのストレージ範囲内の前記同期化予定データを前記現ターゲットデータテーブルへ同期化することは、前記ストレージアドレス情報にしたがって、前記それぞれのストレージ範囲内の前記データを 1 つのデータ同期化プロセスにおいて順次、前記現ターゲットデータテーブルへ同期化することを含む、システム。

10

適用例 18：適用例 16 に記載のシステムであって、前記ストレージアドレス情報を使用して、前記それぞれのストレージ範囲内の前記同期化予定データを前記現ターゲットデータテーブルへ同期化することは、前記ストレージ範囲を N 個のグループに分割することと、対応する N 個のデータ同期化プロセスを開始させることと、前記ストレージアドレス情報にしたがって、前記グループのうちの対応する 1 つのグループの前記それぞれのストレージ範囲内の前記データを、前記データ同期化プロセスのうちの各プロセスにおいて順次、前記現ターゲットデータテーブルへ同期化することと、を含む、システム。

適用例 19：適用例 16 に記載のシステムであって、前記 1 つ又は 2 つ以上のプロセッサは、更に、前記同期化が失敗したときに、前記それぞれのストレージ範囲内の前記データを前記現ターゲットデータテーブルへ同期化し直すように、及びやり直しの回数がプリセット閾値を超えたときに、前記ストレージ範囲内の前記データを同期化することを終了させるように、構成される、システム。

20

適用例 20：適用例 11 ~ 19 のいずれかに記載のシステムであって、前記 1 つ又は 2 つ以上のプロセッサは、更に、前記ソースデータテーブルと前記現ターゲットデータテーブルとの同期化が成功した場合に、前記現ターゲットデータテーブル以外のターゲットデータテーブルの内容を消去するように構成される、システム。

【図1】

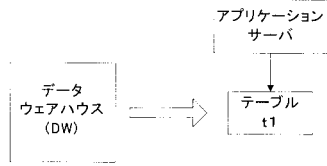


FIG. 1

【図2】

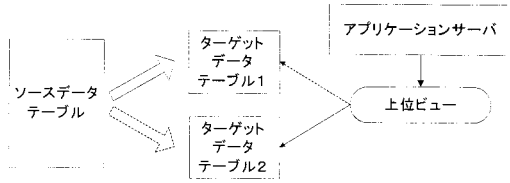


FIG. 2

【図3】

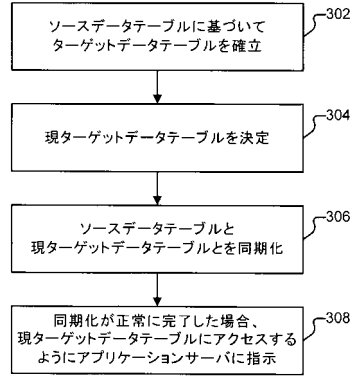


FIG. 3

【図4】

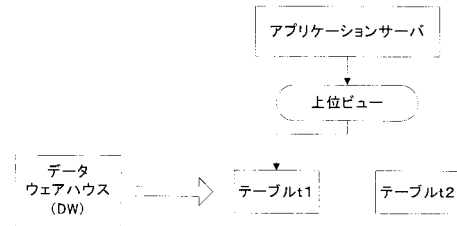


FIG. 4

【図5】

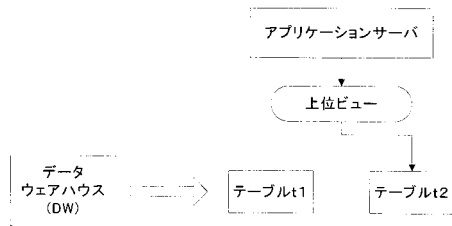


FIG. 5

【図7】

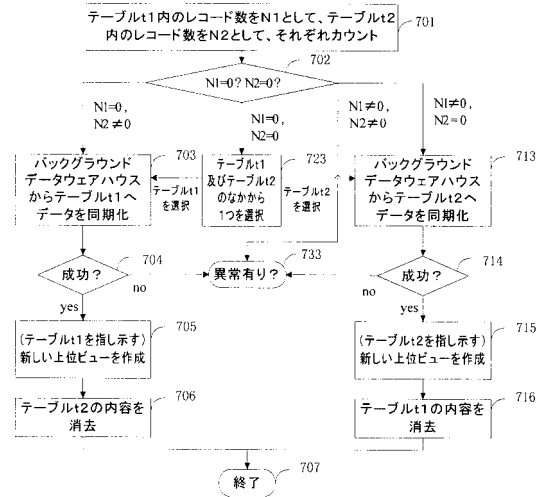


FIG. 7

【図6】

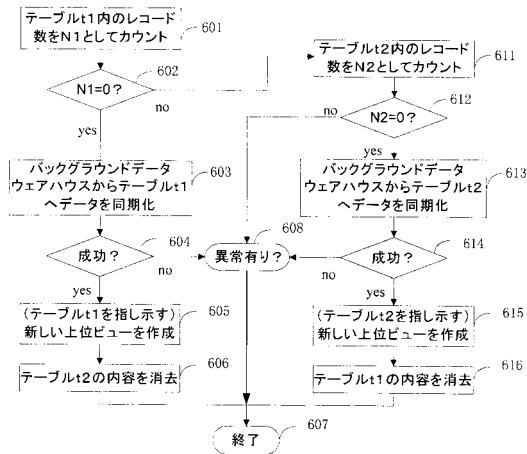


FIG. 6

フロントページの続き

- (72)発明者 ハン・ハイ
中華人民共和国 ハンチョウ, ワーナー・ロード, ウエスト・レイク・インターナショナル・プラ
ザ, 10階, ナンバー391
- (72)発明者 チャン・チン
中華人民共和国 ハンチョウ, ワーナー・ロード, ウエスト・レイク・インターナショナル・プラ
ザ, 10階, ナンバー391
- (72)発明者 チャン・イナ
中華人民共和国 ハンチョウ, ワーナー・ロード, ウエスト・レイク・インターナショナル・プラ
ザ, 10階, ナンバー391

審査官 田川 泰宏

- (56)参考文献 特開2006-301820(JP, A)
特開2007-041888(JP, A)
特開2001-166976(JP, A)
特表2007-502464(JP, A)
特表2002-525755(JP, A)
特表2008-538016(JP, A)
特表2002-523839(JP, A)
特表2005-505829(JP, A)
特開昭62-257535(JP, A)
特開2004-355203(JP, A)
特開2002-202906(JP, A)
特開2006-099748(JP, A)

- (58)調査した分野(Int.Cl., DB名)
G06F 12/00