(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2005-25750
(P2005-25750A)
(43)公開日　平成17年1月27日(2005.1.27)

(51)Int.Cl.[7]	F I		テーマコード（参考）
G06F 12/00	G06F	12/00　518A	5B082

審査請求 未請求　請求項の数 44　ＯＬ　（全 22 頁）

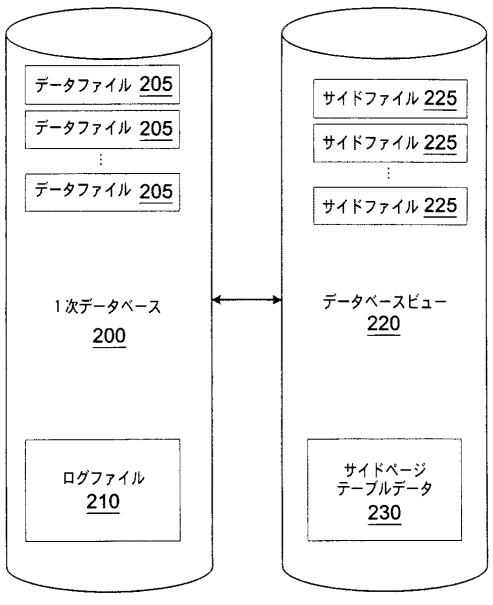
(21)出願番号　特願2004-190397(P2004-190397)
(22)出願日　平成16年6月28日(2004.6.28)
(31)優先権主張番号　10/611,774
(32)優先日　平成15年6月30日(2003.6.30)
(33)優先権主張国　米国(US)

(71)出願人　500046438
マイクロソフト　コーポレーション
アメリカ合衆国　ワシントン州　９８０５２－６３９９　レッドモンド　ワン　マイクロソフト　ウェイ
(74)代理人　100077481
弁理士　谷　義一
(74)代理人　100088915
弁理士　阿部　和夫
(72)発明者　マイケル　ジェイ．ツウィリング
アメリカ合衆国　９８０５２　ワシントン州　レッドモンド　ノースイースト　６８　ストリート　１５２１５

最終頁に続く

(54)【発明の名称】トランザクションの整合性を保つ書き込み時コピーのデータベース

(57)【要約】
【課題】　前のある時点の既存のデータベースのトランザクションの整合性が保たれたビューを提供するデータベースのデータベースビューを作成すること。
【解決手段】　各データベースビューは、１次データベースとともに、前のある時点の１次データベースの内容を決定するのに必要なすべての情報を含む。データベースビューは、１次データベース内の各データファイルに対応するサイドファイルから成る。サイドファイルは、データベースビューが作成されてから変更された対応するデータファイルからのすべてのデータのコピーを含む。スペースを保つために、スパースファイルをサイドファイルのために使用することができる。１次データベースからのページが修正され、古いバージョンがデータベースビューのサイドファイルに格納されているかどうかについて迅速に決定できるようにするために、ページテーブルデータが保持される。
【選択図】　図３



【特許請求の範囲】
【請求項１】
特定の時点のデータベースの内容を反映するトランザクションの整合性を保つデータを含むデータベースビューを提供する方法であって、前記データベースは、データ要素を含み、トランザクションログに関連付けられ、前記トランザクションログは、アクティブトランザクションおよび非アクティブトランザクションを含み、前記データベースビューは、前記データベースからの前記データ要素のうちの少なくとも１つの前のバージョンを格納する差分ストレージを含み、前記方法は、
前記時点に対応する前記トランザクションログにおける分割点を決定するステップと、
前記データベースに対する修正を行う前記分割点より前の各トランザクションを前記トランザクションログ上で見つけるステップと、
前記差分ストレージ内に前記修正のそれぞれを格納するステップと、
前記分割点より前の各アクティブトランザクションを前記トランザクションログ上で見つけるステップと、
前記差分ストレージ内の対応する任意の修正を元に戻すステップと
を備えたことを特徴とする方法。
【請求項２】
前記差分ストレージを初期化するステップをさらに備えたことを特徴とする請求項１に記載の方法。
【請求項３】
前記時点に対応する前記トランザクションログにおける分割点を決定する前記ステップは、ログの切り捨てを使用不可にするステップをさらに含むことを特徴とする請求項１に記載の方法。
【請求項４】
前記データベースビューは、ページテーブルをさらに含み、前記方法は、前記ページテーブルを初期化するステップをさらに備えたことを特徴とする請求項１に記載の方法。
【請求項５】
前記差分ストレージ内の対応する任意の修正を元に戻す前記ステップは、前記対応する修正を削除するステップを含むことを特徴とする請求項１に記載の方法。
【請求項６】
前記差分ストレージ内の対応する任意の修正を元に戻す前記ステップは、
前記データベース内の対応する修正されていないデータを読み取るステップと、
前記対応する修正されていないデータを前記差分ストレージに書き込むステップと
を含むことを特徴とする請求項１に記載の方法。
【請求項７】
前記データ要素のそれぞれは、１ページのデータを含むことを特徴とする請求項１に記載の方法。
【請求項８】
前記差分ストレージは、少なくとも１つのスパースファイルを含むことを特徴とする請求項７に記載の方法。
【請求項９】
前記差分ストレージ内に前記修正のそれぞれを格納する前記ステップは、前記スパースファイルのうちの１つにおいてメモリの領域を割り振るステップを含むことを特徴とする請求項８に記載の方法。
【請求項１０】
前記データ要素のそれぞれは、１ページのデータを含み、前記データベースビューは、ページテーブルをさらに含み、前記ページテーブルは、ページごとに、
前記ページが前記差分ストレージに格納されているかどうかを示す第１の格納データと、
前記差分ストレージにおいて前記領域が割り振られているかどうかを示す第２の格納デ

ータと
を含むことを特徴とする請求項９に記載の方法。
【請求項１１】
前記方法は、
前記ページテーブルが無効であることを検出するステップと、
前記スパースファイル内の領域ごとに、前記領域が割り振られているかどうかを決定するステップと、
前記スパースファイル内の領域ごとに、前記領域が割り振られているかどうかに基づいて前記第２の格納データを設定するステップと
をさらに備えたことを特徴とする請求項１０に記載の方法。
【請求項１２】
前記差分ストレージ内の特定のページにデータが格納されているかどうかの決定は、
前記第１の格納データをチェックし、前記特定のページが前記差分ストレージに格納されていることを、前記第１の格納データが示す場合、前記データは、前記差分ストレージ内の前記特定のページに格納されていることを決定するステップと、
前記第２の格納データをチェックし、前記領域が前記差分ストレージにおいて割り振られていないことを、前記第２の格納データが示す場合、前記データは、前記差分ストレージ内の前記特定のページに格納されていないことを決定するステップと、
前記ページが前記差分ストレージ内に格納されていることを、前記第１の格納データが示さず、前記領域が前記差分ストレージにおいて割り振られてないことを、前記第２の格納データが示さない場合、前記特定のページについての前記差分ストレージの対応するエリアからページデータを読み取り、前記対応するエリアからの前記ページデータが有効であるかどうかを決定するステップと
を含むことを特徴とする請求項１０に記載の方法。
【請求項１３】
前記データベースビューは、ページテーブルをさらに含み、前記ページテーブルは、ページごとに、
前記ページが前記差分ストレージ内に格納されているかどうかを示す第１の格納データを含むことを特徴とする請求項７に記載の方法。
【請求項１４】
データが前記差分ストレージ内の特定のページ内に格納されているかどうかの決定は、
前記第１の格納データをチェックし、前記特定のページが前記差分ストレージに格納されていることを、前記第１の格納データが示す場合、前記データは、前記差分ストレージ内の前記特定のページに格納されていることを決定するステップと、
前記特定のページが前記差分ストレージ内に格納されていることを、前記第１の格納データが示さない場合、前記特定のページについての前記差分ストレージの対応するエリアからページデータを読み取り、前記対応するエリアからの前記ページデータが有効であるかどうかを決定するステップと
を含むことを特徴とする請求項１３に記載の方法。
【請求項１５】
前記データベースビュー内の特定のデータ要素の要求を受け付けるステップと、
データが前記差分ストレージ内の前記特定のデータ要素に対応する場所に格納されているかどうかを決定するステップと、
データが前記差分ストレージ内の前記特定のデータ要素に対応する場所に格納されている場合、前記差分ストレージを読み取ることによって前記要求に応答するステップと、
データが前記差分ストレージ内の前記特定のデータ要素に対応する場所に格納されている場合、前記データベースを読み取ることによって前記要求に応答するステップと
をさらに含むことを特徴とする請求項１に記載の方法。
【請求項１６】
データが前記差分ストレージ内の前記特定のデータ要素に対応する場所に格納されてい

るかどうかを決定する前記ステップは、前記差分ストレージが前記場所に有効なデータを含んでいるかどうかを決定するステップを含むことを特徴とする請求項１５に記載の方法。

【請求項１７】

データが前記差分ストレージ内の前記特定のデータ要素に対応する場所に格納されているかどうかを決定する前記ステップは、ページテーブルを調べるステップを含むことを特徴とする請求項１５に記載の方法。

【請求項１８】

前記方法は、

第２の特定のデータ要素の代わりに前記データベース内のある場所に第１の特定の値を格納する前記データベースに加えられた修正を検出するステップと、

前記データベースビュー内の対応する場所が有効なデータを含んでいるかどうかを決定するステップと、

前記データベースビュー内の前記対応する場所が有効なデータを含んでいない場合、前記第２の特定のデータ要素を前記対応する場所に書き込むステップと

をさらに含むことを特徴とする請求項１に記載の方法。

【請求項１９】

請求項１に記載の方法を実行することを特徴とする、オペレーティングシステム、複数のコンピュータ実行可能命令を格納するコンピュータ読取り可能媒体、コプロセッシング装置、コンピューティング装置、およびコンピュータ実行可能命令を含む変調されたデータ信号のうちの少なくとも１つ。

【請求項２０】

特定の時点のデータベースの内容を反映するトランザクションの整合性を保つデータを含むデータベースビューを提供するシステムであって、前記データベースは、データ要素を含み、トランザクションログに関連付けられ、前記トランザクションログは、アクティブトランザクションおよび非アクティブトランザクションを含み、前記システムは、

前記時点に対応する前記トランザクションログにおける分割点を決定する分割点決定器と、

前記データベースに対する修正を行う前記分割点より前の各トランザクションを前記トランザクションログ上で見つける第１のトランザクションログアナライザと、

前記差分ストレージ内に前記修正のそれぞれを格納する差分ストレージと、

前記分割点より前の各アクティブトランザクションを前記トランザクションログ上で見つける第２のトランザクションログアナライザと、

前記差分ストレージ内の対応する任意の修正を元に戻す差分ストレージ変更器と

を備えたことを特徴とするシステム。

【請求項２１】

前記分割点決定器は、ログの切り捨てを使用不可にするログ切り捨て使用不可器をさらに含むことを特徴とする請求項２０に記載のシステム。

【請求項２２】

前記システムは、特定のデータ要素が前記差分ストレージに格納されているかどうかを示すデータを含むページテーブルをさらに含むことを特徴とする請求項２０に記載のシステム。

【請求項２３】

前記ページテーブルは、前記特定のデータ要素が前記差分ストレージに格納されているかどうかを示す第１の格納データを含むことを特徴とする請求項２２に記載のシステム。

【請求項２４】

前記差分ストレージは、スパースファイルを含み、

前記ページテーブルは、前記差分ストレージおいて前記特定のデータ要素に対応する領域が割り振られているかどうかを示す第２の格納データをさらに含むことを特徴とする請求項２３に記載のシステム。

【請求項２５】
前記システムは、
前記スパースファイル内の領域ごとに、前記領域が割り振られているかどうかを決定する領域割り振り決定器と、
前記スパースファイル内の領域ごとに、前記領域が割り振られているかどうかに基づいて前記第２の格納データを設定する第２の格納データ設定機器と
をさらに備えたことを特徴とする請求項２４に記載のシステム。
【請求項２６】
前記データベースビュー内の特定のデータ要素の要求を受け付け、データが前記差分ストレージ内の前記特定のデータ要素に対応する場所に格納されているかどうかを決定し、データが前記差分ストレージ内の前記特定のデータ要素に対応する場所に格納されている場合、前記差分ストレージを読み取ることによって前記要求に応答し、データが前記差分ストレージ内の前記特定のデータ要素に対応する場所に格納されている場合、前記データベースを読み取ることによって前記要求に応答する要求応答器をさらに含むことを特徴とする請求項２０に記載のシステム。
【請求項２７】
特定の時点のデータベースの内容を反映するトランザクションの整合性を保つデータを含むデータベースビューを提供するコンピュータ読取り可能媒体であって、前記データベースは、データ要素を含み、トランザクションログに関連付けられ、前記トランザクションログは、アクティブトランザクションおよび非アクティブトランザクションを含み、前記データベースビューは、前記データベースからの前記データベース要素のうちの少なくとも１つの前のバージョンを格納する差分ストレージを含み、前記コンピュータ読取り可能媒体は、
前記時点に対応する前記トランザクションログにおける分割点を決定するステップと、
前記データベースに対する修正を行う前記分割点より前の各トランザクションを前記トランザクションログ上で見つけるステップと、
前記差分ストレージ内に前記修正のそれぞれを格納するステップと、
前記分割点より前の各アクティブトランザクションを前記トランザクションログ上で見つけるステップと、
前記差分ストレージ内の対応する任意の修正を元に戻すステップと
を含む動作を実行する命令を備えたことを特徴とするコンピュータ読取り可能媒体。
【請求項２８】
前記動作は、前記差分ストレージを初期化するステップをさらに含むことを特徴とする請求項２７に記載のコンピュータ読取り可能媒体。
【請求項２９】
前記時点に対応する前記トランザクションログにおける分割点を前記決定するステップは、ログの切り捨てを使用不可にするステップをさらに含むことを特徴とする請求項２７に記載のコンピュータ読取り可能媒体。
【請求項３０】
前記データベースビューは、ページテーブルをさらに含み、前記動作は、前記ページテーブルを初期化するステップをさらに含むことを特徴とする請求項２７に記載のコンピュータ読取り可能媒体。
【請求項３１】
前記差分ストレージ内の対応する任意の修正を元に戻す前記ステップは、前記対応する修正を削除するステップを含むことを特徴とする請求項２７に記載のコンピュータ読取り可能媒体。
【請求項３２】
前記差分ストレージ内の対応する任意の修正を元に戻す前記ステップは、
前記データベース内の対応する修正されていないデータを読み取るステップと、
前記対応する修正されていないデータを前記差分ストレージに書き込むステップと

を含むことを特徴とする請求項２７に記載のコンピュータ読取り可能媒体。
【請求項３３】
前記データ要素のそれぞれは、１ページのデータを含むことを特徴とする請求項２７に記載のコンピュータ読取り可能媒体。
【請求項３４】
前記差分ストレージは、少なくとも１つのスパースファイルを含むことを特徴とする請求項３３に記載のコンピュータ読取り可能媒体。
【請求項３５】
前記差分ストレージ内に前記修正のそれぞれを格納する前記動作は、前記スパースファイルのうちの１つにおいてメモリの領域を割り振るステップを含むことを特徴とする請求項３４に記載のコンピュータ読取り可能媒体。
【請求項３６】
前記データ要素のそれぞれは、１ページのデータを含み、前記データベースビューは、ページテーブルをさらに含み、前記ページテーブルは、ページごとに、
前記ページが前記差分ストレージに格納されているかどうかを示す第１の格納データと、
前記差分ストレージにおいて前記領域が割り振られているかどうかを示す第２の格納データと
を含むことを特徴とする請求項３５に記載のコンピュータ読取り可能媒体。
【請求項３７】
前記動作は、
前記ページテーブルが無効であることを検出するステップと、
前記スパースファイル内の領域ごとに、前記領域が割り振られているかどうかを決定するステップと、
前記スパースファイル内の領域ごとに、前記領域が割り振られているかどうかに基づいて前記第２の格納データを設定するステップと
をさらに備えたことを特徴とする請求項３６に記載のコンピュータ読取り可能媒体。
【請求項３８】
前記差分ストレージ内の特定のページにデータが格納されているかどうかの決定は、
前記第１の格納データをチェックし、前記特定のページが前記差分ストレージに格納されていることを、前記第１の格納データが示す場合、前記データは、前記差分ストレージ内の前記特定のページに格納されていることを決定するステップと、
前記第２の格納データをチェックし、前記領域が前記差分ストレージにおいて割り振られていないことを、前記第２の格納データが示す場合、前記データは、前記差分ストレージ内の前記特定のページに格納されていないことを決定するステップと、
前記ページが前記差分ストレージ内に格納されていることを、前記第１の格納データが示さず、前記領域が前記差分ストレージにおいて割り振られてないことを、前記第２の格納データが示さない場合、前記特定のページについての前記差分ストレージの対応するエリアからページデータを読み取り、前記対応するエリアからの前記ページデータが有効であるかどうかを決定するステップと
を含むことを特徴とする請求項３６に記載のコンピュータ読取り可能媒体。
【請求項３９】
前記データベースビューは、ページテーブルをさらに含み、前記ページテーブルは、ページごとに、前記ページが前記差分ストレージ内に格納されているかどうかを示す第１の格納データを含むことを特徴とする請求項３３に記載のコンピュータ読取り可能媒体。
【請求項４０】
データが前記差分ストレージ内の特定のページ内に格納されているかどうかの決定は、
前記第１の格納データをチェックし、前記特定のページが前記差分ストレージに格納されていることを、前記第１の格納データが示す場合、前記データは、前記差分ストレージ内の前記特定のページに格納されることを決定するステップと、

前記ページが前記差分ストレージ内に格納されていることを、前記第１の格納データが示さない場合、前記特定のページについての前記差分ストレージの対応するエリアからページデータを読み取り、前記対応するエリアからの前記ページデータが有効であるかどうかを決定するステップと
を含むことを特徴とする請求項３９に記載のコンピュータ読取り可能媒体。
【請求項４１】
前記動作は、
前記データベースビュー内の特定のデータ要素の要求を受け付けるステップと、
データが前記差分ストレージ内の前記特定のデータ要素に対応する場所に格納されているかどうかを決定するステップと、
データが前記差分ストレージ内の前記特定のデータ要素に対応する場所に格納されている場合、前記差分ストレージを読み取ることによって前記要求に応答するステップと、
データが前記差分ストレージ内の前記特定のデータ要素に対応する場所に格納されている場合、前記データベースを読み取ることによって前記要求に応答するステップと
をさらに含むことを特徴とする請求項２７に記載のコンピュータ読取り可能媒体。
【請求項４２】
データが前記差分ストレージ内の前記特定のデータ要素に対応する場所に格納されているかどうかを決定する前記ステップは、前記差分ストレージが前記場所に有効なデータを含んでいるかどうかを決定するステップを含むことを特徴とする請求項４１に記載のコンピュータ読取り可能媒体。
【請求項４３】
データが前記差分ストレージ内の前記特定のデータ要素に対応する場所に格納されているかどうかを決定する前記動作は、ページテーブルを調べるステップを含むことを特徴とする請求項４１に記載のコンピュータ読取り可能媒体。
【請求項４４】
前記コンピュータ読取り可能媒体は、
第２の特定のデータ要素の代わりに前記データベース内のある場所に第１の特定の値を格納する前記データベースに加えられた修正を検出するステップと、
前記データベースビュー内の対応する場所が有効なデータを含んでいるかどうかを決定するステップと、
前記データベースビュー内の前記対応する場所が有効なデータを含んでいない場合、前記第２の特定のデータ要素を前記対応する場所に書き込むステップと
をさらに含むことを特徴とする請求項２７に記載のコンピュータ読取り可能媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、一般的に、データベースシステムの分野に関する。より詳細には、本発明はトランザクションの整合性を保つデータベースのコピーの作成に関する。
【背景技術】
【０００２】
データベースおよびデータベース製品は、今日よく使用されている。データベースは、レコード、テーブル、インデックスなどのオブジェクトに情報を格納する。データベースに格納されている現在のバージョンの情報に加えて、前バージョンの情報もユーザにとって有用となり得る。
【０００３】
前バージョンのデータに関する情報を提供する１つの方法は、ユーザが関心を持つある時点のデータベースの完全なコピーを作成することである。これが行なわれるときは、データベースに関連するすべてのファイルが格納される。しかし、ある量の記憶スペースに格納されているデータベースでは、各コピーもその量の記憶スペースを必要とするため、この技術ではスペースを使いすぎる。またこの手法では、コピー自体が大量のデータの移

動に関与するため、時間もかかりすぎる。

【０００４】

変更されたデータのコピーのみを格納する書き込み時コピー（copy-on-write）機構を使用して、ディスクボリュームスナップショット（disk volume shapshot）を提供することができる。元のデータベースに変更が加えられるたびに、前に格納されていたデータが変更されたデータのコピーに書き込まれる。しかし、このボリュームスナップショットでは、トランザクションの整合性が保たれない。つまり、格納された変更は、進行中のトランザクションの一部である可能性があり、したがってボリュームスナップショットは、あるトランザクションに関する中途半端な情報を含んでいる可能性がある。こうしたトランザクションの不整合の可能性があるため、アプリケーションがボリュームスナップショットを使用するには特殊なフックが必要である。このように作成されたボリュームスナップショットは、格納されているインデックスおよびテーブルの構造上の修正が不完全であるため、構造的にも整合性のないものとなり得る。さらに、ボリュームスナップショットでは、コピーの粒度はボリュームレベルである。非データベースデータもコピーすることがあるが、これは不要であり、時間およびリソースの無駄につながる。

【０００５】

また、特定のトランザクションによってアクセスされたデータを使用可能にすることによってデータベースのバージョニング（versioning）が提供されている。所与のトランザクションについてこのデータのみが格納され、したがって所与のトランザクションによってアクセスされたものに関するデータが必要な場合、（トランザクションによって任意の変更が加えられる前に）アクセスされたデータをそのトランザクションを開始したユーザに提供することはできる。しかしこのデータは、複数のトランザクションまたはユーザからはアクセスできない。複数のトランザクションまたはユーザが同じデータにアクセスを試みた場合、これらの異なるアクセスごとにデータが異なり得る。さらに、こうしたバージョニングは、データベースサーバの再起動後は永続的ではない。

【発明の開示】

【発明が解決しようとする課題】

【０００６】

したがって、迅速に作成することができ、トランザクションの整合性を保ち、整合性のある情報を複数のトランザクションまたはユーザに提供し、データベースサーバの再起動後も永続性のあるデータベースを表示する方法が必要となる。

【課題を解決するための手段】

【０００７】

データベースの完全なコピーを作成することなく、前のある時点（データベースビューが作成されたとき）の既存のデータベースのトランザクションの整合性が保たれたビューを提供するデータベースのデータベースビューを作成する。１次データベースとは使用中のデータベースであり、１または複数のデータベースビューが作成される。

【０００８】

各データベースビューは、１次データベースとともに、前のある時点の１次データベースの内容を決定するのに必要なすべての情報を含む。データベースビューは、１次データベース内の各データファイルに対応するサイドファイルから成る。サイドファイルは、データベースビューが作成されてから変更された対応するデータファイルからのすべてのデータのコピーを含む。

【０００９】

一実装形態では、データベースのデータベースビューが実装される。データベースは、データ要素を含み、トランザクションログに関連付けられる。トランザクションログは、アクティブトランザクションおよび非アクティブトランザクションの両方を含む。データベースビューは、現在のバージョンのデータベースと前のバージョンのものとの差のストレージ（「差分ストレージ（difference storage）」）を含む。これは、サイドファイルとして知られているファイルに格納される。

【００１０】
新しいデータがデータベースに書き込まれた場合は必ず、前バージョンのデータのコピーが差分ストレージに格納される。別の変更が加えられた場合、差分ストレージはすでに前バージョンのデータのコピーを含んでいるため、新しいデータは格納されない。
【００１１】
一実施形態で、トランザクションログが分析され、前記トランザクションログにおける分割点（split point）が識別される。修正を行う、ログ内の分割点より前の各トランザクションが見つけられ、修正の結果が差分ストレージに書き込まれる。次いでアクティブトランザクションによってもたらされるトランザクションログに対する修正は、データベースビューを更新してトランザクションを元に戻すことによって、データベースビュー内で元に戻される。このように、分割点の時点でのデータベースのトランザクションの整合性を保つビューが提供される。
【００１２】
本発明の他の特徴については後述する。
上記の概略および以下の好ましい実施形態の詳細な説明は、添付の図面と併せ読めば、よりよく理解できる。本発明を説明するために本発明の構造例を図面に示しているが、本発明は、開示されている特定の方法および手段に限定されるものではない。
【発明を実施するための最良の形態】
【００１３】
（コンピューティング環境の例）
図１は、本発明の態様を実施できるコンピューティング環境の例を示している。コンピューティングシステム環境１００は、適したコンピューティング環境の一例にすぎず、本発明の使用または機能の範囲に関する限定を示唆するものではない。また、コンピューティング環境１００を、動作環境１００の例に示した構成要素のいずれか１つ、またはその組合せに関連する依存性または必要条件を有しているものと解釈すべきではない。
【００１４】
本発明は、他の多くの汎用または専用コンピューティングシステム環境または構成で動作可能である。本発明との使用に適したよく知られているコンピューティングシステム、環境、および／または構成の例には、それだけには限定されないが、パーソナルコンピュータ、サーバコンピュータ、ハンドヘルドまたはラップトップ装置、マルチプロセッサシステム、マイクロプロセッサベースのシステム、セットトップボックス、プログラム可能家庭用電化製品、ネットワークＰＣ、ミニコンピュータ、メインフレームコンピュータ、組み込みシステム、上記の任意のシステムまたは装置を含む分散コンピューティング環境などがある。
【００１５】
本発明は、コンピュータによって実行されるプログラムモジュールなどのコンピュータ実行可能命令の一般的な文脈で説明することができる。一般にプログラムモジュールは、特定のタスクを実行する、または特定の抽象データ型を実装するルーチン、プログラム、オブジェクト、構成要素、データ構造などを含む。また、本発明は、タスクが通信ネットワークまたは他のデータ通信媒体によってリンクされているリモート処理装置によって実行される分散コンピューティング環境でも実施することができる。分散コンピューティング環境では、プログラムモジュールおよび他のデータを、メモリ記憶装置を含むローカルおよびリモートのコンピュータ記憶媒体に置くことができる。
【００１６】
図１を参照すると、本発明を実施するシステムの例は、汎用コンピューティング装置をコンピュータ１１０の形で含んでいる。コンピュータ１１０の構成要素は、それだけには限定されないが、処理ユニット１２０、システムメモリ１３０、およびシステムメモリを含む様々なシステム構成要素を処理ユニット１２０に結合するシステムバス１２１を含む。処理ユニット１２０は、マルチスレッドプロセッサ上でサポートされるものなど、複数の論理処理ユニットを表し得る。システムバス１２１は、様々なバスアーキテクチャのう

ちの任意のものを使用するメモリバスまたはメモリコントローラ、周辺バス、およびローカルバスを含むいくつかのタイプのバス構造のうちどんなものでもよい。こうしたアーキテクチャには、それだけには限定されないが一例として、ＩＳＡ（Industry Standard Architecture）バス、ＭＣＡ（Micro Channel Architecture）バス、ＥＩＳＡ（Enhanced ISA）バス、ＶＥＳＡ（Video Electronics Standards Association）ローカルバス、および（メザニンバスとしても知られている）ＰＣＩ（Peripheral Component Interconnect）バスなどがある。システムバス１２１は、通信装置間のポイントツーポイント接続、スイッチングファブリックなどとして実装することもできる。

【００１７】

コンピュータ１１０は、一般に様々なコンピュータ読取り可能媒体を含む。コンピュータ読取り可能媒体は、コンピュータ１１０からアクセスできる使用可能な任意の媒体とすることができ、揮発性および不揮発性媒体、リムーバブルおよび非リムーバブル媒体を含む。コンピュータ読取り可能媒体は、それだけには限定されないが一例として、コンピュータ記憶媒体および通信媒体を含み得る。コンピュータ記憶媒体には、コンピュータ可読命令、データ構造、プログラムモジュール、他のデータなど、情報を記憶するための任意の方法または技術で実施される揮発性および不揮発性のリムーバブルおよび非リムーバブル媒体がある。コンピュータ記憶媒体には、それだけには限定されないが、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、フラッシュメモリまたは他のメモリ技術、ＣＤ－ＲＯＭ、デジタル多用途ディスク（ＤＶＤ）または他の光ディスク記憶装置、磁気カセット、磁気テープ、磁気ディスク記憶装置または他の磁気記憶装置、または所望の情報の格納に使用でき、コンピュータ１１０からアクセスできる他の任意の媒体などがある。通信媒体は一般に、コンピュータ可読命令、データ構造、プログラムモジュール、または他のデータを搬送波または他の移送機構などの変調されたデータ信号に組み込む。これには任意の情報配送媒体がある。「変調されたデータ信号」という用語は、信号内の情報を符号化するように設定または変更された１または複数のその特徴を有する信号を意味する。通信媒体には、それだけには限定されないが一例として、有線ネットワーク、直接配線された接続などの有線媒体、および音響、ＲＦ、赤外線、その他の無線媒体などの無線媒体がある。また、上記のどんな組合せでもコンピュータ読取り可能媒体の範囲内に含まれるものとする。

【００１８】

システムメモリ１３０は、読取り専用メモリ（ＲＯＭ）１３１やランダムアクセスメモリ（ＲＡＭ）１３２など、揮発性および／または不揮発性メモリの形のコンピュータ記憶媒体を含む。基本入出力システム１３３（ＢＩＯＳ）は、例えば起動中など、コンピュータ１１０内の要素間での情報の転送を助ける基本ルーチンを含み、一般にＲＯＭ１３１に格納されている。ＲＡＭ１３２は一般に、処理ユニット１２０から直接アクセス可能な、かつ／または処理ユニット１２０が現在処理しているデータおよび／またはプログラムモジュールを含む。図１は、それだけには限定されないが一例として、オペレーティングシステム１３４、アプリケーションプログラム１３５、他のプログラムモジュール１３６、およびプログラムデータ１３７を示している。

【００１９】

コンピュータ１１０は、他のリムーバブル／非リムーバブル、揮発性／不揮発性コンピュータ記憶媒体を含むこともできる。一例にすぎないが、図１は、非リムーバブル不揮発性磁気媒体から読み取り、あるいはそこに書き込むハードディスクドライブ１４０、リムーバブル不揮発性磁気ディスク１５２から読み取り、あるいはそこに書き込む磁気ディスクドライブ１５１、およびＣＤ－ＲＯＭや他の光媒体など、リムーバブル不揮発性光ディスク１５６から読み取り、あるいはそこに書き込む光ディスクドライブ１５５を示している。動作環境の例で使用できる他のリムーバブル／非リムーバブル、揮発性／不揮発性コンピュータ記憶媒体には、それだけには限定されないが、磁気テープカセット、フラッシュメモリカード、デジタル多用途ディスク、デジタルビデオテープ、半導体ＲＡＭ、半導体ＲＯＭなどがある。ハードディスクドライブ１４１は一般に、インタフェース１４０などの非リムーバブルメモリインタフェースを介してシステムバス１２１に接続され、磁気

ディスクドライブ１５１および光ディスクドライブ１５５は一般に、インタフェース１５０などのリムーバブルメモリインタフェースによってシステムバス１２１に接続される。

【００２０】

上述し、図１に示したドライブおよびその関連のコンピュータ記憶媒体は、コンピュータ可読命令、データ構造、プログラムモジュール、およびコンピュータ１１０の他のデータの記憶を提供する。図１では例えば、ハードディスクドライブ１４１は、オペレーティングシステム１４４、アプリケーションプログラム１４５、他のプログラムモジュール１４６、およびプログラムデータ１４７を格納するものとして示されている。これらの構成要素は、オペレーティングシステム１３４、アプリケーションプログラム１３５、他のプログラムモジュール１３６、およびプログラムデータ１３７と同じであっても、異なっていてもよいことに注意されたい。オペレーティングシステム１４４、アプリケーションプログラム１４５、他のプログラムモジュール１４６、およびプログラムデータ１４７は少なくとも異なるコピーであることを示すために、ここではそれらに異なる番号を付している。ユーザは、キーボード１６２、および一般にマウス、トラックボール、またはタッチパッドと呼ばれるポインティング装置１６１などの入力装置を介してコマンドおよび情報をコンピュータ１１０に入力することができる。他の入力装置（図示せず）には、マイクロフォン、ジョイスティック、ゲームパッド、衛星パラボラアンテナ、スキャナなどがある。これらおよび他の入力装置は、しばしばシステムバスに結合されているユーザ入力インタフェース１６０を介して処理ユニット１２０に接続されるが、パラレルポート、ゲームポート、ユニバーサルシリアルバス（ＵＳＢ）など他のインタフェースおよびバス構造で接続してもよい。モニタ１９１または他のタイプの表示装置もまた、ビデオインタフェース１９０などのインタフェースを介してシステムバス１２１に接続される。モニタに加えて、コンピュータは、出力周辺インタフェース１９５を介して接続できるスピーカー１９７、プリンタ１９６などの他の周辺出力装置を含むこともできる。

【００２１】

コンピュータ１１０は、リモートコンピュータ１８０など１または複数のリモートコンピュータへの論理接続を使用してネットワーク環境で動作することができる。リモートコンピュータ１８０は、パーソナルコンピュータ、サーバ、ルータ、ネットワークＰＣ、ピア装置、または他の一般のネットワークノードでよく、一般にコンピュータ１１０に関連して上述した多くまたはすべての要素を含むが、図１にはメモリ記憶装置１８１のみを示している。図１に示した論理接続は、ローカルエリアネットワーク（ＬＡＮ）１７１および広域エリアネットワーク（ＷＡＮ）１７３を含むが、他のネットワークを含んでいてもよい。こうしたネットワーキング環境は、オフィス、全社規模のコンピュータネットワーク、イントラネット、およびインターネットではごく一般的である。

【００２２】

ＬＡＮネットワーキング環境で使用する場合、コンピュータ１１０は、ネットワークインタフェースまたはアダプタ１７０を介してＬＡＮ１７１に接続される。ＷＡＮネットワーキング環境で使用する場合、コンピュータ１１０は一般に、モデム１７２、またはインターネットなどＷＡＮ１７３を介して通信を確立する他の手段を含む。モデム１７２は、内蔵のものでも外付けのものでもよく、ユーザ入力インタフェース１６０または他の適切な機構を介してシステムバス１２１に接続することができる。ネットワーク環境では、コンピュータ１１０に関連して示したプログラムモジュール、またはその一部をリモートメモリ記憶装置に格納することができる。図１は、それだけには限定されないが一例として、リモートアプリケーションプログラム１８５をメモリ装置１８１上に存在するものとして示している。図示したネットワーク接続は例であり、コンピュータ間の通信リンクを確立する他の手段を使用してもよいことは理解されよう。

【００２３】

（データベースおよびデータベースビュー）

一般にデータベースは、データベースファイルおよびログファイルの２つの種類のファイルから成る。ログファイルは、ある期間にわたってデータベースファイルに追加された

変更を記載する一連のログレコードを含む。ログレコードは、ログシーケンス番号（ＬＳＮ）で識別することができる。図２に示すように、一実施形態では、１次データベース２００は、１組のデータベースファイル２０５およびログファイル２１０から成る。データファイルは、ページと呼ばれる複数のストレージの塊に分割される。

【００２４】

データベースの完全なコピーを作成することなく、前のある時点の既存のデータベースのトランザクションの整合性が保たれたビューを提供するデータベースのデータベースビューが作成される。データベースビューは、データベースとともに、前のある時点のデータベースのコピーを生成するのに必要なすべての情報を含む。しかしデータベースビューは、それ自体情報のすべてを含んでいるわけではなく、したがって全コピーより小さいサイズになり得る。さらに、修正がデータベースに加えられると、その場でビューが作成され、それによってコスト（時間および処理）を時間にわたって分散することができる。前の時点でデータベースビューからコピーを作成した場合、時間および処理のコストは、一時に集中することになる。さらに、データベースに対する更新アクティビティが続行している間にデータベースビューを作成することができる。１次データベースとは使用中のデータベースであり、１または複数のデータベースビューが作成される。

【００２５】

上述したように、データベースビューは、１次データベースとともに、前のある時点の１次データベースの内容を決定するのに必要なすべての情報を含む。データベースビューは、１次データベース内の各データファイルに対応するサイドファイルから成る。サイドファイルは、データベースビューが作成されてから変更された対応するデータファイルからのすべてのデータのコピーを含む。一実施形態では、サイドファイル内のページから１次ファイル内のページにテーブルをマッピングする必要を回避するために、サイドファイルがスパースファイル（sparse file）に格納される。スパースファイル内の実際に書き込まれるファイルの部分のみが記憶スペースを必要とする。ファイルの他のすべての領域は割り振りされていない。他の実施形態では、サイドファイルのストレージは、スパースファイル内にはない。

【００２６】

一実施形態では、スパースファイル機構は、標準領域サイズで動作する。１つの領域内のデータがスパースファイルに書き込まれると、たとえデータが領域全体を埋めなくても、領域全体のためのスペースが割り振られる。このスペースが割り振られ、そこからの読み取りが可能であるため、有効な値で埋められた領域のエリアと、その領域内の任意のストレージが必要な場合にスパースファイルの粒度によってあるサイズの領域が割り振られることが求められるために存在する領域のエリアとの間に区別をつける必要がある。

【００２７】

データベースビューは、データベースビューが作成されてから１次データベースで変更されたすべてのデータの元の値を含んでいるため、データベースビューの作成時点のデータベースのデータは、データベースビューから読み取ることができる。データベースビューからのデータの要求に応答するため、サイドファイルが要求のデータを含んでいる場合、データは、データベースビューのサイドファイルから読み取られる。読み取るべきデータであってサイドファイルに存在しないものは、データベースビューが作成されてから変更されておらず、１次データベースから読み取られる。

【００２８】

一実施形態で、サイドファイルは、１次データベースからのデータのページを含む。１次データベースの任意のページの任意のデータが変更されると、そのデータのページがサイドファイルに格納される。本発明では、１次データベース内のデータの単位としてページについて説明しているが、１次データベースのデータの他の単位を使用することもできることを企図している。

【００２９】

どのデータがサイドファイルに書き込まれたか、またどのデータを１次データベースか

ら読み取るべきかを決定するために、サイドファイル内の有効なデータの存在を確認する必要がある。一実施形態では、有効なデータが存在するかどうかを確認するためにサイドファイルを直接読み取る。別の実施形態では、サイドページテーブルが作成され、所与のページが存在するかどうか、また有効かどうかに関するデータが格納される。

【００３０】

一実施形態では、１次データベース内のページごとに、サイドページテーブルは、ページは変更されておらず、それを１次データベースから読み取るべきか、またはページが変更されているのでそれをサイドファイルから読み取るべきかに関する情報を格納する。サイドページテーブルによって、所与のページがサイドファイルに存在するかどうかの迅速な決定が可能になる。

【００３１】

（１ビットおよび２ビットのページテーブル）

一実施形態では、サイドファイルおよびスパースファイル機構は、いずれも同じページ／領域サイズを使用する。つまり、サイドファイルが１次データベースから格納したページは、任意のメモリがスパースファイルに書き込まれたときにスパースファイルが格納した領域と同じサイズである。例えば、スパースファイル領域が８ＫＢであり、１次データベースから格納されたページも８ＫＢである場合、ページサイズおよび領域サイズは等しい。この場合、埋められる任意の領域は、１次データベースから読み取られたページによって完全に埋められ、無効なデータがその領域に格納される可能性はない。

【００３２】

別の実施形態では、いくつかのサイドファイル領域が各ページに正確に対応する。スパースファイル領域は８ＫＢ（キロバイト）であり、１次データベースから格納されたページは１６ＫＢである場合、サイドファイルに格納される各ページによって２つの領域が埋められる。この場合も、埋められる任意の領域は、１次データベースから読み取られたページからの内容によって完全に埋められる。この場合もまた、無効なデータがその領域に格納される可能性はない。

【００３３】

これらの実施形態の場合、サイドページテーブルは、サイドファイル内のページごとに１ビットの情報を保持するメモリ内ビットマップを含む。サイドファイル内のページごとに、対応するビットは、ページがサイドファイル内にあるかどうかを示す。

【００３４】

別の実施形態では、サイドファイル領域の粒度は、１次データベースから格納されたページの粒度より大きい。例えば、サイドファイルの各領域は６４ＫＢであり、ページのサイズは８ＫＢである場合、サイドファイル内の領域の存在は、必ずしもその領域内のすべての情報が１次データベースからの有効なデータであることを示すとは限らない。１つのページのみがサイドファイルにコピーされる場合、この例では、割り振られた領域内の６４ＫＢのうちの８ＫＢのみが有効なデータを含む。別の実施形態では、一部のサイドファイルページが領域にわたって分散される。

【００３５】

これらの実施形態では、サイドページテーブルは、サイドファイル内のページごとに２ビットの情報を保持する２つのメモリ内ビットマップを含む。これらをビット１およびビット２と呼ぶ。サイドファイル内のページごとに、対応するビットは、（ビット１）ページが確かにサイドファイル内にあるかどうか、（ビット２）ページがサイドファイル内にある可能性があるかどうかを示す。ビット２は、ページがサイドファイル内に格納される領域が割り振られていることを示すものとも考えられる。しかし、後述するように、一実施形態では、このビット２は、サイドページテーブルが再構築されたときにのみ設定される。

【００３６】

ビットマップは、メモリ内で維持され、したがって永続的ではない可能性がある。ビットマップは、消去されると、スパースファイル情報から再構築される。スパースファイル

を調べ、ページごとに、サイドファイルでページが配置される領域にメモリが割り振られている場合、ビット２は、そのページがサイドファイル内にある可能性があることを示すように設定される。ページごとに、ビット１は、ページがサイドファイル内にあることを確定しないことを示すように設定される。

【００３７】

サイドページテーブルが永続的となるように維持されると、領域およびページの粒度を無視することができ、１ビットのサイドページテーブルを使用することができる。しかし一実施形態では、データベースサーバの再起動後、永続的なデータベースビューをサポートするために、２ビットのページテーブルを使用する。

【００３８】

一実施形態では、サイドファイルのためのページテーブルは作成されない。この場合、コピーがデータベースビュー内のページから作成されているかどうかを決定することが必要なときはいつでも、データベースビューを調べる。本発明では、１ビットまたは２ビットのページテーブルが存在する実施形態に関して後述するが、ページテーブルがなく、データベースビューを検査して、１次データベースからコピーされたページが含まれているかどうかを決定する必要がある実施形態についても検討する。

【００３９】

図３に示すように、１次データベース２００のデータベースビュー２２０は、サイドファイル２２５から成る。１次データベース２００内のデータファイル２０５のそれぞれは、データベースビュー２２０内に対応するサイドファイル２２５を有する。さらに、サイドページテーブルデータ２３０は、データベースビュー２２０のメモリ内に格納される。一実施形態では、サイドページテーブルデータ２３０は、サイドファイル２２５のすべてをカバーする１つのサイドページテーブルである。別の実施形態では、個別のサイドページテーブルがサイドファイル２２５ごとに存在する。

【００４０】

（トランザクションログ）

データベースでは、トランザクションログとは、トランザクションログを最後にバックアップしてからデータベースに対して実行されたすべてのトランザクションのシリアルレコードである。トランザクションログを使用して、障害時にデータベースを回復する。一実施形態では、トランザクションログは循環キューとしてモデル化される。トランザクションログは、ログの非アクティブな部分を削除することによって切り捨てることができる。この非アクティブ部分は、回復する必要のない完了したトランザクションを含む。逆に、トランザクションログのアクティブ部分は、完了したトランザクション、および依然として稼動しており、まだ完了していないトランザクション（アクティブトランザクション）を含む。切り捨ては、トランザクションログが拡大し続け、より多くのスペースを使用できるようにする代わりに、トランザクションログ内の非アクティブなスペースを最小限に抑えるために行われる。

【００４１】

アクティブトランザクションは、トランザクションの不整合を引き起こす可能性がある。アクティブトランザクションでは、データファイルの一部の修正がバッファキャッシュからデータファイルに書き込まれなかった可能性があり、データファイル内に未完了のトランザクションからの一部の修正がある可能性がある。ログファイルを使用して、データベースの回復によって確実にトランザクションの整合性が保たれるようにする。これは、ＡＲＩＥＳ（Algorithms for Recovery and Isolation Exploiting Semantics）型の回復を使用して行われる。データファイルに書き込まれていない可能性のある、ログに記録されているすべての修正は、データベースへの修正を行うことによってロールフォーワードされる。データベースの保全性を確保するために、トランザクションログ内にある未完了のすべてのトランザクションは、データベースに対する修正を元に戻すことによってロールバックされる。

【００４２】

（データベースビューの作成）
データベースビューを作成するために、データベースビューの物理的構造（サイドファイルおよびページテーブル）を初期化する必要がある。まず、サイドファイル２２５が１次データベース２００内のデータファイル２０５ごとに作成される。上述したように、サイドファイルはスパースファイルでよい。あるいは別の実施形態では、データファイル２０５と同じサイズの非スパースファイルとすることができる。サイドファイル２２５は、１次データベース２００内のデータファイル２０５に関連付けられる。

【００４３】
トランザクションは継続的に行われており、データベースビューではトランザクションの整合性が保たれるため、データベースビューの作成中、トランザクションログを使用する必要がある。データベースビューに使用すべきトランザクションに関する情報が確実に破棄されないようにするために、１次データベース２００において、（存在する場合は）ログの切り捨てを使用不可にする。

【００４４】
一実施形態では、サイドページテーブル２３０は、データベースビューのために初期化される。最初に、サイドページテーブルは、サイドファイル２２５内にページが存在しないことを示すように設定され、２ビットのサイドページテーブルの場合、サイドファイル２２５にページが潜在的に、または確実に存在しないことを示すように設定される。

【００４５】
初期化が完了すると、データベースビューは、「オンライン」になる用意ができている。データベースビューは、この時点で１次データベース２００と平行して稼動しており、修正が行われると、修正されたページの元の値のコピー（すなわち更新が行われる前のページの内容）がデータベースビューに格納される。図５は、本発明の一実施形態によるデータベースのトランザクションの整合性を保つビューを実装する方法のフロー図を示している。図５のステップ５００に示すように、トランザクションログにおいて分割点が決定される。この分割点は、データベースビューが表す時点に対応する。データベースビューが作成されると、１次データベース２００上のログの端部のＬＳＮが取得される。このＬＳＮが、１次データベース２００およびデータベースビュー２２０が分岐を開始する「分割点」である。次いで１次データベース２００は、データベースビュー処理が要求されるようにマーク付けされる。後述するように、１次データベース２００内のデータベースビューサポートが開始する。

【００４６】
データベースビューが整合性を保つようにするために、分割点より前の１次データベース２００のログを分析して、分割点の時点でどのトランザクションがアクティブであったかを決定する必要がある。ログ内で（分割点の時点で）最も古いアクティブトランザクションが識別される。その最も古いアクティブトランザクションの前にログの切り捨てが使用可能となる。

【００４７】
ＡＲＩＥＳ型回復と同じように、分割点より前の最も古いアクティブトランザクションからの１次データベース２００のログ内のすべての操作がデータベースビューに対して行われる。図４は、本発明の好ましい実施形態によるトランザクションログの例、ログファイル２１０を示すブロック図である。ログファイル２１０内のログエントリは、ログエントリ４００、４１０、４２０、４３０、４４０、４５０、４６０、４８０、４９０、および４９９を含む。分割点４７５が決定される。トランザクションは、引き続きログに書き込まれ、しかし切り捨ては使用不可である。ログファイル２１０を検査し、最も古いアクティブトランザクションから分割点まで（図４の例ではログエントリｎ４００からログエントリｎ＋７まで）のトランザクションの結果、データベースに対する任意の修正がサイドファイル２２５に対して行われる。これらの各トランザクションにおける修正の結果がサイドファイル２２５に格納される。次いでこれらのトランザクションを検査する。ログ内の任意のアクティブトランザクションによってログファイルに書き込まれた修正、例え

ばログエントリｎ４００、ログエントリｎ＋２　４２０、ログエントリｎ＋６がサイドファイル２２５において元に戻される。

【００４８】

図５のステップ５００からわかるように、トランザクションログにおいて分割点を選択する。次にステップ５１０で、データベースに対して修正を行う各トランザクションを見つける。一実施形態では、トランザクションを分析し、トランザクションの結果としてある値がデータベースのある場所に書き込まれる場合、ステップ５２０からわかるように、以下で詳述するデータベースビューを修正する方法を使用して、その値がサイドファイル内の対応する場所に保存される。このように、データベースに書き込む必要のあるすべての変更（「ダーティページ」の変更など）がデータベースビューのために格納される。

【００４９】

しかし、一部のトランザクションは、依然としてコミットされていない。したがって、ログ内の分割点までのこれらのアクティブトランザクションを探し出し（ステップ５３０）、元に戻す（ステップ５４０）必要がある。一実施形態で、未完了のトランザクションがデータベース内のある場所の値を変更する場合、上記でサイドファイルに追加された変更は、サイドファイルから削除される。代替実施形態では、後述するように、データベースビューを修正し、サイドファイル内のデータを分割点の時点でのデータベース内のデータと一致するように設定することによって、トランザクションが元に戻される。

【００５０】

このように、ログからの完了したトランザクションのみがデータベースビューに反映される。ログ上のトランザクションがデータベースビューにおいて反映されると、元に戻された、分割点が生じたときにアクティブであったトランザクションを除いて、１次データベース２００上でログの切り捨てが使用可能になる。データベースビュー処理は使用可能であったため、１次データベース２００に変更が加えられると、データベースビューが更新されるので、データベースビューを使用して分割点の時点での１次データベース２００の内容を決定することができる。

【００５１】

（データベースビューの回復）

データベースサーバが（正常または異常に）シャットダウンした後で再起動すると、データベースビューを再度初期化する必要がある。そのために、メモリ内に格納されているサイドページテーブルを再度初期化する必要がある。

【００５２】

サイドページテーブルを再初期化するには、２ビットのサイドページテーブルの実装では、割り振られたサイドページテーブル内の領域ごとに、割り振られた領域内のページごとのサイドページテーブル内のデータ（ビット２）は、ページがサイドファイル２２５に書き込まれた可能性があることを示すように設定される。他のすべてのページのサイドページテーブル内のデータは、ページがサイドファイル２２５に書き込まれた可能性がないことを示すように設定される。ただし、ページがサイドファイル２２５に書き込まれたことを確定するものではなく、したがって、ビット１は、最初は設定されない。

【００５３】

あるいは、２ビットのサイドページテーブルの実装、または１ビットのサイドページテーブルの実装では、上述したように、サイドファイル２２５を検査して、ページごとに、サイドファイル２２５内のページが有効であるかどうかを決定することができる。ページテーブルは、存在しているページごとに、ページが実際にサイドファイル２２５に存在することを示すように設定される。すべての他のページは、ページがサイドファイル２２５に存在しないことを示すように設定される。

【００５４】

（１次データベース内のデータベースビューサポート）

データが上書きされる前にデータベースビューが１次データベース２００から情報を格納するようにするために、１次データベース２００は、データベースビューの作成をサポ

ートする必要がある。１次データベース２００が修正するページごとに、ページがデータベースビュー内にあるかどうかに関して決定を行う必要がある。データベースビュー内にページが存在する場合、それは正しいバージョンのページである。例えばこれは、前の修正が１次データベース２００内のそのページに加えられたときであり得る。ページが１次２００内で再度変更されると、データベースビューのバージョンを変更する必要がある。

【００５５】

ページが変更されているという情報を１次データベース２００から受信したとき、ページがサイドファイル２２５にある場合は何もする必要がない。ページがサイドファイル２２５内にない場合、ページをサイドファイル２２５に書き込む必要があり、サイドページテーブルで適切なビットを設定する必要がある。２ビットのページテーブルがある場合、次の表１で示すように、そのページでのビット１およびビット２の場合で３つの可能性がある。

【００５６】

【表１】

表１　２ビットのページテーブルの様々なケース

	ビット1はページがサイドファイル内に確実にあることを示す	ビット1はページがサイドファイル内に確実にあることを示さない
ビット2はページがサイドファイル内にある可能性があることを示す	ケース1: ページはサイドファイル内にある	ケース2: ページはサイドファイル内にある可能性がある
ビット2はページがサイドファイル内に絶対にないことを示す	ケース1: ページはサイドファイル内にある［あるいはケース4: 無効］	ケース3: ページは絶対にサイドファイル内にない

【００５７】

一実施形態では、ページがサイドファイル２２５内に確実にあることを、ビット１が示すとき、ビット２は無視される。したがって表１に示すように、ページがサイドファイル２２５内に確実にあることを、ビット１が示す場合には、ビット２がどのように示していても、ページはサイドファイル２２５内にあるものとみなされる。代替実施形態では、ページがサイドファイル２２５に確実に存在することを、ビット１が示すように設定されると、ビット２は、ページがサイドファイル２２５内にある可能性があることを示すように設定される。またこの代替実施形態では、ページはサイドファイル２２５に確実にあることを、ビット１が示しているが、ビット２はサイドファイル２２５に絶対にないことを示しているとき、このケースは無効であり、エラーが生じる。

【００５８】

１次データベース２００は、２ビットのページテーブルの場合にページが変更されていることを示しているとき、上述の場合にとるべき動作は次の通りである。

ケース１：何も行わない

ケース２：ページがサイドファイル２２５内にあるかどうかを決定し、ない場合はページをサイドファイル２２５に書き込む

ケース３：ページをサイドファイル２２５に書き込む。

【００５９】

ページがサイドファイル２２５に書き込まれるとき、ケース１またはケース２のいずれかで、１次データベース２００内の古いバージョン（現在１次データベース２００によって修正されているバージョン）のページがサイドファイル２２５に書き込まれる。さらに、ページテーブルは、ページが現在サイドファイル２２５内にあることを示すように設定され、その後のページへのすべての書き込みはケース１に従って処理され、データベースビューに適切なページは、サイドファイル２２５に格納されたままである。

【００６０】
ケース２において、ページがサイドファイル２２５内にあるかどうかを決定するために、ページに対応するデータがサイドファイル２２５から読み取られる。データが有効であり、前のバージョンのページがサイドファイル２２５内にある場合、それを上書きする必要はない。一実施形態では、ページに対応するページテーブルのビット１は、ページがサイドファイル２２５に確実に存在することを示すように設定されるため、ページへの将来の書き込みはケース１に基づいて処理される。
【００６１】
有効なデータはまだその領域に書き込まれていないことを示すために、新しく割り振られた領域に配置されたデータによってデータの無効を示すことができる。例えば、すべてゼロしかないデータベースのページがないことがわかっている場合、新しく割り振られた領域にすべてゼロを書き込むことができる。そうである場合、サイドファイル２２５内のページの存在は、割り振られた領域の一部であり、ゼロ以外の何らかのデータを含むサイドファイル２２５内の対応するページによって示される。
【００６２】
（データベースビューの読み取り）
表１に詳しく示したケースは、データベースビューに格納されているデータの読み取りを行うのにも有用である。ページ内のデータをデータベースビューから読み取るとき、ページがサイドファイル２２５内に存在している場合は、そこから読み取る必要がある。そこにない場合、ページは１次データベース２００から読み取る必要がある。２ビットのページテーブルシステムでは、３つのケースの場合にとるべきアクションは、次の通りである。
ケース１：サイドファイル２２５からページを読み取る
ケース２：ページがサイドファイル２２５内にあるかどうかを決定し、そこにある場合はサイドファイル２２５からページを読み取り、そこにない場合は１次データベース２００からページを読み取る
ケース３：１次データベース２００からページを読み取る。
【００６３】
（データベースビューの修正）
データベースビューは、前のある時点でのデータベースの状態を表す。ユーザは、そのデータベースビューをデータベースとして使用することを選択することができる。例えばユーザは、前の時点のデータベースビューに対してそのアクションが行われたかのように、データベースビューに対してアクションを行い、データベースのデータベースビューを作成することができる。さらに、初期化中、上記で詳述したように、データベースビューに対してトランザクションを実行し、元に戻すこともできる。
【００６４】
データベースビューを修正するには、修正は、データベースビュー内のデータに基づく必要があり、その結果得られたページをデータベースビューに格納する必要がある。ページのデータベースビューにデータが存在しない場合、修正は、１次データベース２００内のデータに基づく必要があり、その結果得られたページをデータベースビューに格納する必要がある。
【００６５】
２ビットのページテーブルシステムでは、３つのケースの場合にとるべきアクションは、以下の通りである。
ケース１：サイドファイル２２５からページを読み取り、修正を行い、ページをサイドファイル２２５に書き込む
ケース２：ページがサイドファイル２２５内に存在するかどうかを決定し、存在する場合はケース１と同様に処理し、存在しない場合はケース３と同様に処理する
ケース３：１次データベース２００からページを読み取り、サイドファイル２２５にページを書き込み、ページがサイドファイル２２５にあることを示すようにページテーブ

ルを設定する。ページへの修正を行い、適切な場合はサイドファイル２２５に修正されたページを書き込む。

【００６６】

上記の例は、単に説明の目的で提供したものであり、決して本発明を限定するものと解釈されるものではないことに注意されたい。本発明を様々な実施形態に関連して説明してきたが、本明細書で使用した単語は、限定するものではなく説明および例示のためのものであることを理解されたい。さらに、本明細書において本発明を特定の手段、材料、および実施形態に関連して説明してきたが、本発明は、本明細書に開示した詳細に限定されるものではなく、添付の特許請求の範囲内に含まれるものなど機能的に等価なすべての構造、方法、用途にまで拡張される。本明細書の教示の恩恵を受ける当分野の技術者は、それに様々な修正を行うことができ、本発明の態様の範囲および意図から逸脱することなく変更を加えることができる。

【図面の簡単な説明】

【００６７】

【図１】本発明の態様を実施できるコンピューティング環境の例を示すブロック図である。

【図２】本発明の一実施形態によるデータベースを示すブロック図である。

【図３】本発明の一実施形態によるデータベースビューおよびデータベースを示すブロック図である。

【図４】本発明の一実施形態によるトランザクションログの例を示すブロック図である。

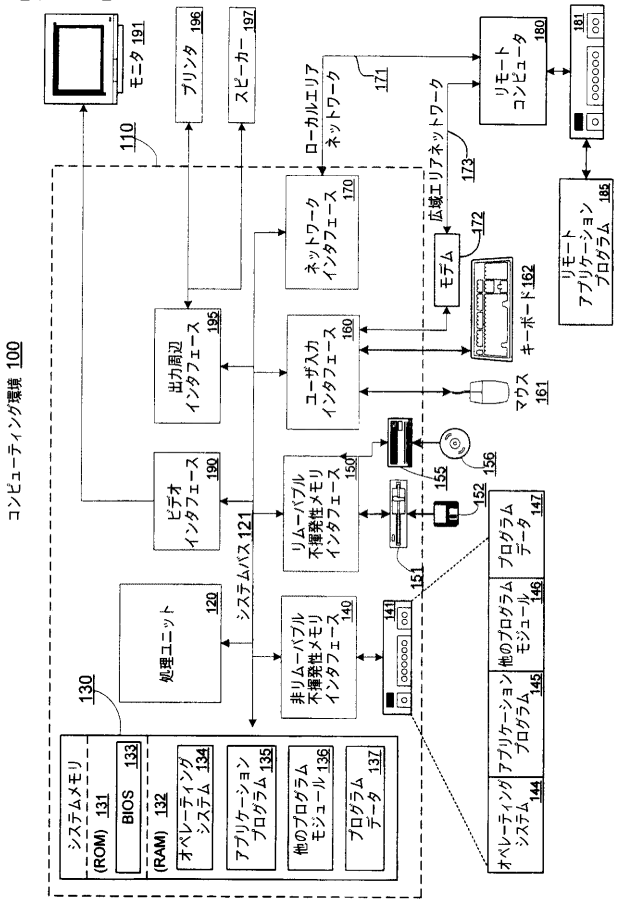
【図５】本発明の一実施形態によるトランザクションの整合性を保つデータベースビューを実装する方法を示すフロー図である。

【符号の説明】

【００６８】

２００　１次データベース
２０５　データファイル
２１０　ログファイル
２２０　データベースビュー
２２５　サイドファイル
２３０　サイドページテーブルデータ

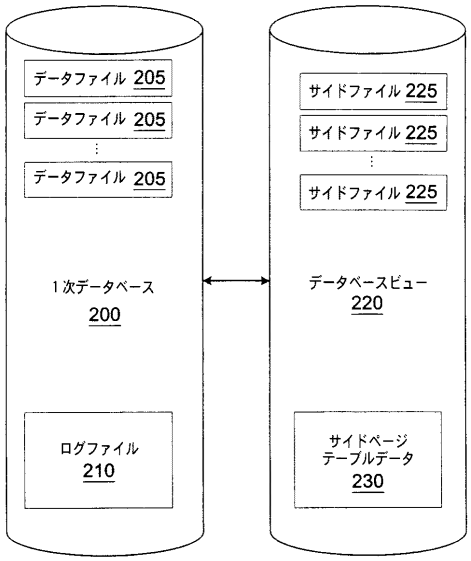
【図１】


コンピューティング環境 100
110
モニタ 191
プリンタ 196
スピーカー 197
ローカルエリア ネットワーク 171
広域エリアネットワーク 173
リモート コンピュータ 180
181
リモート アプリケーション プログラム 185
ネットワーク インタフェース 170
モデム 172
キーボード 162
出力周辺 インタフェース 195
ユーザ入力 インタフェース 160
マウス 161
ビデオ インタフェース 190
リムーバブル 不揮発性メモリ インタフェース 150
155
156
152
151
システムバス 121
処理ユニット 120
非リムーバブル 不揮発性メモリ インタフェース 140
141
130
システムメモリ (ROM) 131
BIOS 133
(RAM) 132
オペレーティング システム 134
アプリケーション プログラム 135
他のプログラム モジュール 136
プログラム データ 137
プログラム データ 147
他のプログラム モジュール 146
アプリケーション プログラム 145
オペレーティング システム 144

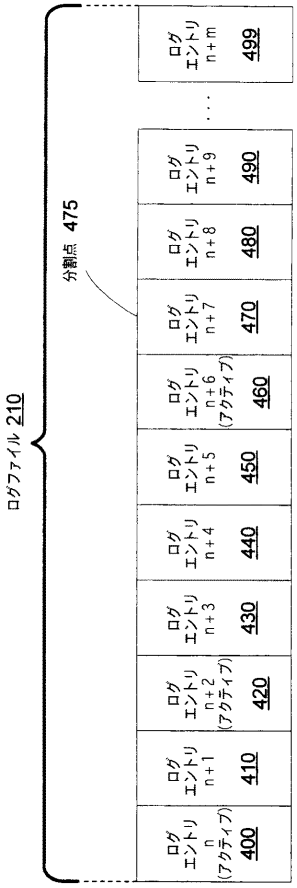
【図２】


データファイル 205
データファイル 205
データファイル 205
1次データベース
200
ログファイル
210

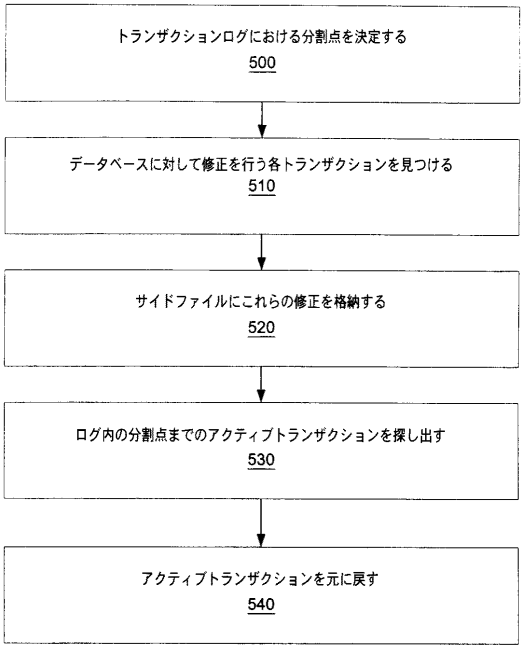
【図３】


データファイル 205
データファイル 205
データファイル 205
1次データベース
200
ログファイル
210
サイドファイル 225
サイドファイル 225
サイドファイル 225
データベースビュー
220
サイドページ テーブルデータ 230

【図４】


ログファイル 210
分割点 475
ログ エントリ n (アクティブ) 400
ログ エントリ n+1 410
ログ エントリ n+2 (アクティブ) 420
ログ エントリ n+3 430
ログ エントリ n+4 440
ログ エントリ n+5 450
ログ エントリ n+6 (アクティブ) 460
ログ エントリ n+7 470
ログ エントリ n+8 480
ログ エントリ n+9 490
ログ エントリ n+m 499

【図５】


トランザクションログにおける分割点を決定する
500
データベースに対して修正を行う各トランザクションを見つける
510
サイドファイルにこれらの修正を格納する
520
ログ内の分割点までのアクティブトランザクションを探し出す
530
アクティブトランザクションを元に戻す
540

フロントページの続き

(72)発明者　ルイス　エス．ブルック
アメリカ合衆国　９８０５３　ワシントン州　レッドモンド　ノースイースト　６０　プレイス　２１００９
(72)発明者　サミート　エイチ．アガーウォル
アメリカ合衆国　９８０５２　ワシントン州　レッドモンド　１４９　プレイス　ノースイースト　８１２７　ユニット　ビー３０９
(72)発明者　ヤン　カンロン
アメリカ合衆国　９８０２９　ワシントン州　イサコア　サウスイースト　４１　ドライブ　２５００１

Ｆターム(参考) 5B082 GB06