(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3593366号
(P3593366)
(45)発行日　平成16年11月24日(2004.11.24)　(24)登録日　平成16年9月3日(2004.9.3)

(51)Int.Cl.[7]　F I
G06F 17/30　G06F 17/30　４１９Ｂ
G06F 12/00　G06F 12/00　５３１Ｒ

請求項の数 3　(全 23 頁)

(21)出願番号　特願平6-222930
(22)出願日　平成6年9月19日(1994.9.19)
(65)公開番号　特開平8-87510
(43)公開日　平成8年4月2日(1996.4.2)
審査請求日　平成13年3月15日(2001.3.15)

(73)特許権者　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(73)特許権者　000233055
日立ソフトウエアエンジニアリング株式会社
神奈川県横浜市鶴見区末広町一丁目１番４３
(74)代理人　100075096
弁理士　作田　康夫
(74)代理人　100068504
弁理士　小川　勝男

最終頁に続く

(54)【発明の名称】データベース管理方法

(57)【特許請求の範囲】
【請求項１】
障害発生時の再開始回復処理を前提としたトランザクション処理を可能とするデータベース管理システムにおけるシンクポイントを取得するデータベース管理方法であって、
前記シンクポイント取得開始時に外部記憶装置上のデータベースに書き込みされていない主記憶装置上にマッピングされたバッファ中のすべての更新されたページを管理するテーブルにシンクポイント取得処理中であるマークをつけ、
前記シンクポイント取得処理中は前記マークのついたバッファ中の更新されたページを前記データベースに書き込み、
前記シンクポイント取得処理中に前記マークのついたバッファ中の更新されたページを更新するトランザクションは前記更新されたページをアクセスする前に前記データベースに書き込みを行い、前記マークをはずした後アクセスを許可し、
前記マークのついた更新ページをすべて前記データベースに書き込みが完了した時点をシンクポイント取得完了とし、
前記シンクポイントを取得する間隔がある一定量に達した時点より、前記シンクポイント取得開始時点まで前記バッファ中の更新ページを非同期に前記データベースに書き込みを行うプレシンクポイント取得処理を行うことにより、
前記シンクポイント処理中であってもトランザクション処理続行可能とするとともに前記シンクポイント取得処理における前記バッファ中の更新ページ数を削減することを特徴とするデータベース管理方法。

【請求項２】
請求項１記載のデータベース管理方法において、
前記プレシンクポイント取得処理において、前記シンクポイント取得開始時点までに前記データベースに書き込めなかった前記バッファ中の更新ページは、前記シンクポイント取得開始時にバッファ中のすべての更新されたページを管理するテーブルにシンクポイント取得処理中であるマークをつけることによりシンクポイントを取得する
ことを特徴とするデータベース管理方法。
【請求項３】
請求項２記載のデータベース管理方法において、
前記プレシンクポイント取得処理において、前記シンクポイント取得開始時にプレシンクポイント取得が完了したか否かを記憶しておき、前記プレシンクポイント取得が完了していない場合は前記データベースに書き込めなかった前記バッファ中の更新ページ数から前記プレシンクポイント取得開始点を早くするように変更し、前記プレシンクポイント取得がすでに完了している場合は、前記シンクポイント取得時に存在した更新ページ数から前記プレシンクポイント取得開始点を遅くするように変更し、前記プレシンクポイント取得開始点を補正することを特徴とするデータベース管理方法。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】
本発明は、複数のトランザクションからアクセスされるデータを記憶するデータベースを管理するデータベース管理システムにおける、データベースのシンクポイント処理を行う方法に関する。
【０００２】
【従来の技術】
多重処理環境におけるデータベース管理システムでは、一定間隔おきにシンクポイントあるいはチェックポイントという確認点を設けている。これは、障害が発生しデータベースの再開始回復処理を実行しなければならない場合、障害が発生した時点の直前のシンクポイントの状態にデータベースを復元し、そこから処理の再開をはかることによってデータベースの再開始回復処理を行う。シンクポイントは、更新を含むトランザクション処理の進行とは非同期に取得することができる。
【０００３】
データベース管理システム、特にリレーショナル・データベース管理システムは、データベース、スキーマ、表、行といったように論理的に階層的な資源で構成される。物理的に行を格納するための不揮発性記憶装置である磁気ディスクなどの場合、行を物理的に固定長のブロックあるいはページという単位に格納する。トランザクション処理において行われるデータベースアクセスは、このページを単位として主記憶装置上に確保されたバッファプールを介して行われる。
【０００４】
あるトランザクションにおいて、必要なページがバッファプールに読み込まれ、バッファプール内で検索、更新処理が行われる。バッファプールに読み込まれたページは、できるだけ長くバッファプール上に存在するように管理されることによって、磁気ディスクなどの外部記憶装置との入出力処理オーバヘッドを削減している。あるトランザクションにおいて行われたデータベースの更新処理については、その更新履歴情報をログとして取得する。ログは外部記憶装置のような不揮発性の記憶装置に永久的に記録される。ログレコードはプロセッサ内の主記憶装置上の揮発性のログバッファにまず書き込まれ、トランザクションが確定されるときに外部記憶装置上のログファイルに移される。この場合、トランザクションにおいて行われたバッファプール上のページの更新は、いずれ外部記憶装置上のデータベースに反映されなければならない。
【０００５】
トランザクション確定時に、当該トランザクションによって更新されたバッファプール上

の全てのページを外部記憶装置上のデータベースに反映する方法がある。この方法では、トランザクション確定時の外部記憶装置への書き込みオーバヘッドがかかる。そこで、もう一つの方法では、トランザクション確定時とは遅延させてバッファプールより更新されたページを外部記憶装置上のデータベースに反映することによって、同一ページへの複数のトランザクションの更新をまとめて外部記憶装置への書き込みを行うことで、トランザクション確定時の外部記憶装置への書き込みオーバヘッドを削減するものである。

【０００６】

トランザクション確定時とよりも遅延させて外部記憶装置への書き込みを行う場合、外部記憶装置に更新されたページが書き込まれる前に障害等によりデータベース管理システムが終了してしまうとそのページの情報は失われる。この場合、システムの再開始が行われ、システムが終了した時点までに行われたトランザクションの回復を保証する必要がある。トランザクションが確定される前に障害が生じた場合は、そのトランザクションで行われたすべての更新を確実に元に戻す。また、トランザクション確定後に障害が生じた場合は、、そのトランザクションにて行われた更新をすべて保証されなければならない。

【０００７】

システムの再開始時には、前回のシンクポイントより行われた全てのトランザクションを外部記憶装置上に保存されたログファイルからログレコードを読み出して回復する。ログレコードによるトランザクション回復時には、確定されたトランザクションであっても実際に更新したページについては、バッファプール上から外部記憶装置へ反映されていないことがある。その場合は、外部記憶装置上のデータベースから回復対象のページを再度バッファプール上に読み込み、ログレコードを基に更新処理を再実行（ＲＥＤＯ）する。

【０００８】

逆に、確定されたトランザクションおよび確定されていないトランザクションで行われた更新がすでにバッファプールから外部記憶装置へ反映されてしまっている場合もある。これに対応する対策としては、ページ中に更新が発生したときのログのログシーケンス番号（ＬＳＮ）を持っておくことによって解決される。ページに対してあるトランザクションによって更新が行われると、そのときのログを取得するがそのログレコードのＬＳＮを合わせてページ内に書き込んでおく。そうすると、バッファプールから更新されたページが外部記憶装置に書き込まれることによって、そのトランザクションで行った更新がどのログに対応する更新まで外部記憶装置に反映されているかがわかる。これにより、システムの再開始時に回復対象の更新ページすべてを回復する必要がなくなる。

【０００９】

更新対象ページをバッファプールに読み込んだ場合、読み込んだページのＬＳＮとログレコードのＬＳＮを比較し、ページのＬＳＮの方がログレコードのＬＳＮより大きいときは、そのログレコードに記録された更新情報はすでに反映されていることになる。ということは、ページのＬＳＮの方がログレコードのＬＳＮより小さいときだけ、ログを使用してＲＥＤＯを行えばよいということになる。データベース管理システムでは、こうした再開始時の処理時間をできる限り短時間で行うことが望ましい。そのためには、再開始処理時のＲＥＤＯ処理を極力削減することが望まれる。それには、通常、バッファプール上で更新されたページを適切な時期にシステムプロセスを使用して、バックグランドで外部記憶装置上のデータベースに書き込む。

【００１０】

ここで、再開始処理において直前のシンクポイントから回復処理を行う場合に前提とされるのは、直前のシンクポイント時点のデータベースの状態からログを用いて回復が行えることを保証することである。そのために、シンクポイントを取得する処理では、まず、ログレコードとしてシンクポイント開始ログを取得することにより開始される。この後、トランザクションの進行を停止させ、ログバッファ上のログレコードを外部記憶装置上のログファイルに書き込む。

【００１１】

ログの書き込みが終了すると、次はバッファプール上の全ての更新されたページを外部記

憶装置上のデータベースに書き込む。これが終わると、シンクポイント取得完了ログをログバッファに書き込んで、停止したトランザクション処理の続行を開始してシンクポイント取得を完了する。これによって、次のシンクポイントが取得されるまでに障害等によりシステムが終了しても、今、取得したシンクポイントのデータベースの状態が保証されたので、取得したシンクポイントを起点にして再開始処理を行うことができる。

【００１２】

このシンクポイント取得処理では、説明したようにシンクポイント取得が開始されるとトランザクションの進行が一時的に妨げられるので、シンクポイント取得処理中はトランザクションのスループットが一時的に沈み込むという問題がある。シンクポイント取得処理中であっても、トランザクション処理を続行できる方法としては、シンクポイント取得処理においてバッファプール中のすべての更新ページ番号と各ページのＬＳＮをログに出力しておくことによって、シンクポイント取得を完了する方法がある。この方法では、システム再開始時に直前のシンクポイント時に取得した更新ページ番号のログから最も古いＬＳＮ探索する。そして、最も古いＬＳＮを持つログレコードを再開始の回復起点とするものである。しかし、シンクポイント取得時にバッファプールから長い間、外部記憶装置上のデータベースに書き込まれていないページがあると、ＲＥＤＯ対象となるログレコード数が増加することになるため、再開始処理時間を最小にするための工夫が必要となる。

【００１３】

これらの従来技術は、Ｃ．Ｍｏｈａｎ，Ｄｏｎ　Ｈａｄｅｒｌｅ等の”ＡＲＩＥＳ：Ａ　Ｔｒａｎｓａｃｔｉｏｎ　Ｒｅｃｏｖｅｒｙ　Ｍｅｔｈｏｄ　Ｓｕｐｐｏｒｔｉｎｇ　Ｆｉｎｅ－Ｇｒａｎｕｌａｌｉｔｙ　Ｌｏｃｋｉｎｇ　ａｎｄ　Ｐａｒｔｉａｌ　Ｒｏｌｌｂａｃｋｓ　Ｕｓｉｎｇ　Ｗｒｉｔｅ－Ａｈｅａｄ　Ｌｏｇｇｉｎｇ”，ＡＣＭ　Ｔｒａｎｓａｃｔｉｏｎｓ　ｏｎ　Ｄａｔａｂａｓｅ　Ｓｙｓｔｅｍｓ、第１７巻、第１号、ｐｐ．　９４－１６２（１９９２年３月）に詳細に説明されている。さらに、特開平５－６２９７号公報に開示されているトランザクション処理方法およびシステムにも詳細に説明されている。

【００１４】

【発明が解決しようとする課題】

本発明の目的は、シンクポイント取得処理が発生してもトラザクション処理のスループットを低下させないようにすることにある。システム再開始時における再開始処理時間の削減するための方法を提供することが関連した目的である。特に、シンクポイント取得処理の開始時点においてバッファプール中に存在した更新ページとそれ以外の非更新ページのアクセスをシンクポイント取得中でも同時にアクセスするための方法を提供することである。

【００１５】

【課題を解決するための手段】

上記目的を達成するために、本発明のデータベース管理方法では、障害発生時の再開始回復処理を前提としたトランザクション処理を可能とするデータベース管理システムにおけるシンクポイントを取得する方法において、シンクポイント取得開始時に外部記憶装置上のデータベースに書き込みされていない主記憶装置上にマッピングされたバッファ中のすべての更新されたページを管理するテーブルにシンクポイント取得処理中であるマークをつける。

【００１６】

シンクポイント取得処理中はマークのついたバッファ中の更新されたページをデータベースに書き込み、シンクポイント取得処理中にマークのついたバッファ中の更新されたページを更新するトランザクションは更新されたページをアクセスする前にデータベースに書き込みを行い、マークをはずした後アクセスを許可し、マークのついた更新ページをすべてデータベースに書き込みが完了した時点をシンクポイント取得完了とすることにより、シンクポイント処理中であってもトランザクション処理続行可能とすることを特徴とするデータベース管理方法によって解決される。

【００１７】
【作用】
次に、第一の目的を達成するため、本発明のデータベース管理方法は、複数のトランザクションが、各々データベースをアクセスしてバッファプール上に読み込んだページに対して更新処理を行っている場合、ある時点でデータベースのシンクポイント取得時点に達すると、バッファプール上の更新されたページに対応するバッファを管理するテーブルにシンクポイント取得中であるマークをつけることにより、バックグランドでデータベースへの書き込みを行うプロセスは、該バッファプール上でシンクポイント取得中であるマークのついたバッファに対応するページを対象に外部記憶装置上のデータベースへ書き込みを行うようにする。これによって、シンクポイント取得時にログファイルへの書き込みとログに対応した更新ページのデータベースへの書き込みを同期することができる。
【００１８】
シンクポイント取得中に、該マークのついたバッファに格納されたページに対するアクセス要求は、一旦バッファプール上から外部記憶装置上のデータベースに対して書き込みを行ってから、アクセスを許可するようにする。これによって、シンクポイント取得中であっても、トランザクションを停止させることなく、システムとしてのスループットを低下させることがない。
【００１９】
【実施例】
以下、本発明の一実施例を図面を用いて具体的に説明する。
図２は、本発明の一実施例にかかるコンピュータシステム１０の構成を示す。コンピュータシステム１０は、ＣＰＵ１２、主記憶装置１４、磁気ディスク等の外部記憶装置１６及び多数の端末１８で構成される。主記憶装置２上には、データベース管理システム２０が置かれ、外部記憶装置１６上にはデータベース管理システム２０が管理するデータベース３６ａ、３６ｂ及びデータベース更新に関する更新履歴情報であるログファイル３７が格納される。
【００２０】
データベース管理システム２０は、ユーザからのデータベース問い合わせ要求であるＳＱＬを受け取り、構文解析意味解析処理を通してデータベースアクセスの最適なアクセス経路を決定する最適化処理を行ない、決定したアクセス経路に基づいてデータベース処理用の内部処理コードを生成する質問解析処理部２１、生成された内部処理コードを基にデータベース・アクセスを行うデータベース演算処理部２２、外部記憶装置１６に格納されたデータベース１６からデータを主記憶装置１４上に確保したバッファプールとの間でデータの入出力を行うバッファプール管理部２３、前記端末１８から投入されたトランザクションを管理するトランザクション管理部２４、バッファプール３４の上で行われたデータの更新をトランザクションとは非同期に外部記憶装置１６上のデータベース３６に書き込みを行う遅延書き込み処理部２５、定期的にデータベースを整合性が取れた状態にすることを保証する処理を行うシンクポイント取得処理部２６及びトランザクションで行われたデータベースへの更新履歴情報であるログ（以下、ログと略す）を管理し、各ログレコードのログシーケンス番号（以下、ＬＳＮと略す）を割り振り、トランザクションの完了時に主記憶装置１４上のログバッファ３５から外部記憶装置１６上のログファイル３７に書き込みを行うログ書き込み処理部２７で構成される。前記バッファプール管理部２３は、主記憶装置１４上に確保したバッファプール３４をデータベースのデータを蓄積する単位である物理適に固定長のページと対応させて管理するためのバッファプール管理テーブル３１、ページ管理テーブル３２及びハッシュ管理テーブル３３を使用する。
【００２１】
図３は、先に述べたデータベースの入出力単位であるページ４０の構造を示す。データベース管理システム、特にリレーショナルデータベース管理システムではデータは複数の行からなるリレーションと呼ばれる表形式を成している。リレーションは、データベース管理システム２０の入出力単位である物理的に固定長の複数のページに分割され、外部記憶

装置１６のデータベース３６に格納されている。ページ４０には行が格納され、ページ内の状態はページ内制御情報４２で管理され、格納された行１～行６は各々ページ内の相対アドレスを持っているスロット領域４４によりポイントされている。

【００２２】

図４は、図３に示したページ内制御情報４２に含まれる管理情報を示す。ページ内制御情報４２には、ページ内に割り当てられたスロット数である割当スロット数４２１、実際に格納されている行に使用されたスロット数である使用スロット数４２２、ページ内の空き領域長を管理する空き領域長４２３、ページ内の未使用領域の先頭を相対アドレスで管理する未使用領域先頭オフセット４２４及びページ内で行われた更新を反映したログのＬＳＮ４２５で構成される。このＬＳＮ４２５は、データベース３６からバッファプール３４に読み込まれた時に設定されているＬＳＮ４２５が実際にそのＬＳＮ４２５までのログがすべて反映されていることを示す。逆に、バッファプール３４上で行われたページ上の行に対する更新が行われると、その更新に対応したログのＬＳＮが書き込まれ、バッファプール３４からデータベース３６に書き込まれた時には、そのＬＳＮが示す更新はデータベースに反映したことが保証される。もちろん、バッファプール３４からデータベース３６に書き込まれる前に、一般的に良く知られるＷＡＬ（Ｗｒｉｔｅ　Ａｈｅａｄ　Ｌｏｇｇｉｎｇ）プロトコルによりログがログファイル３７に書き込まれることは言うまでもない。

【００２３】

図５にバッファプール管理部２３がデータベース３６との入出力を行うためのバッファプール３４を管理する各テーブルの関連を示す。バッファプール管理テーブル３１は、ハッシュ管理テーブル３３及びページ管理テーブル３２を管理している。バッファプール管理テーブル３１は、図６に示すような情報を持つ。バッファプール３４が持つページを格納するバッファ面数３１１、ハッシュ管理テーブル３３へのポインタ３１２、できるだけバッファプール上にページを留めるように管理するための手法であるＬＲＵ（Ｌｅａｓｔ　Ｒｅｃｅｎｔｌｙ　Ｕｓｅｄ）アルゴリズムに基づいたＬＲＵチェインの最も新しいページを管理するページ管理テーブル３２へのポインタ３１３、逆に最も使用されていない古いページを管理するページ管理テーブルへのポインタ３１４、前記ＬＲＵチェインとは別に更新されたページについてのＬＲＵチェインを管理するための最も新しいページを管理するページ管理テーブル３２へのポインタ３１５、逆に、最も使用されていない古いページを管理するページ管理テーブルへのポインタ３１６、ページ管理テーブル３２が管理するページに対応したバッファが無効状態である。すなわち、ＦＲＥＥ状態であるページ管理テーブルのポインタ３１７、バッファプール３４上に存在する更新ページ数３１８及び遅延書き込み状態を示すフラグ３１９から成る。

【００２４】

次に、図６にページ管理テーブル３２の情報について示す。ページ管理テーブル３２には、記憶したページのオブジェクト識別子３２１、バッファ番号３２２、ＬＲＵチェイン又はＦＲＥＥチェインに接続された場合の前方ポインタ３２３と後方ポインタ３２４、ハッシュチェインに接続された場合の前方ポインタ３２５と後方ポインタ３２６、ページ番号３２７、ロックカウンタ３２８、排他モードでページを入力した状態を示すＸＧＥＴフラグ３２９、出力要求フラグ３３０、データベースからの読み込み中である状態を示す入力中フラグ３３１、データベースへの書き込み中である状態を示す出力中フラグ３３２、データベースからの入出力待ち状態である場合の入出力待ちチェイン３３３、ロック待ちチェイン３３４、及び本発明の特長であるシンクポイント取得時の出力対象ページとなったことを示すシンクポイント取得中フラグ３３５で構成される。

【００２５】

図５を用いて、各テーブルの関連について説明する。バッファプール管理テーブル３１からハッシュ管理テーブル３３へは、ハッシュ管理テーブルポインタ３１２でポイントされ、ハッシュ管理テーブルはデータベースを構成するページ番号からハッシングすることによって該当するハッシュエントリにそのページを管理するページ管理テーブル３２が接続

される。例えば、ハッシュエントリ１には同じハッシュ値を持つページ管理テーブル３２ａ及び３２ｂが接続される。バッファプール管理テーブル３１のＬＲＵチェインにはページ管理テーブル３２上の出力要求フラグ３３０が出力要求状態でないものがチェインされ、ページ管理テーブル３２ｃ、及び３２ｆがチェインされている。バッファプール管理テーブル３１の更新ＬＲＵチェインにはページ管理テーブル３２上の出力要求フラグ３３０が出力要求状態のものがチェインされ、ページ管理テーブル３２ｄ、及び３２ｈがチェインされている。また、ＦＲＥＥチェインにはページ管理テーブル３２ｉ、及び３２ｊがチェインされており、バッファプール管理部２３はバッファプール上に存在しないページを入力する場合、このＦＲＥＥチェイン３１７のページ管理テーブル３２を割り当て、当該ページ管理テーブルに対応したバッファにページを入力する。

【００２６】

ここで、本発明に関わるトランザクション処理について図８から図１１を用いて説明する。図２における端末１８からトランザクションが投入されるとトランザクション管理部２４はトランザクション開始処理を行う。トランザクション開始処理の概略処理フローを図８に示す。トランザクションが開始されると、まず、トランザクション管理テーブルを当該トランザクション用に割り当て、初期化する（ステップ５２）。当該テーブルにはトランザクション番号、トランザクション開始ログのＬＳＮ及びカレントな当該トランザクションのＬＳＮなどが格納される。

【００２７】

次に、トランザクション開始ログを作成し、ログ書き込み処理部２７にログの出力要求を発行し、主記憶装置１４上のログバッファ３５に出力し、データベース管理システム内の一貫したＬＳＮを割り振り、ＬＳＮを取得する（ステップ５４）。取得したＬＳＮを当該トランザクションのトランザクション管理テーブルのトランザクション開始ログとして登録する（ステップ５６）。こうして、トランザクション開始処理が完了すると、データベースに対する質問処理が実行される。データベースに対する質問処理のうち、更新処理が行われた場合の概略処理フローを図９に示す。当該データベース管理システムが行レベルの排他制御を行うデータベース管理システムの場合、更新対象となった行に排他モードでロックを取得する（ステップ６１）。更新対象行にロックを確保できたならば、バッファプール管理部２３に対して更新対象行が記憶されているページを排他モードでページ入力要求を発行する（ステップ６２）。

【００２８】

そして、更新対象行への更新操作を行う前に更新操作を回復するためのＵＮＤＯログを作成し、ログ書き込み処理部２７に対し書き込み要求を発行するとともにＵＮＤＯログのＬＳＮを取得する（ステップ６３）。ＵＮＤＯログが取得できたら、行の更新操作を行う（ステップ６４）。そして、更新した行に対する再実行による回復のためのＲＥＤＯログを出力し、ＲＥＤＯログのＬＳＮを取得する（ステップ６５）。取得したＲＥＤＯログのＬＳＮをページ内制御情報４２のＬＳＮ４２５に書き込み、更新する（ステップ６６）。ＲＥＤＯログのＬＳＮを更新した後、当該ページに対する更新をデータベースに反映するため、バッファプール管理部２３に対してページ出力要求を行う（ステップ６８）。

【００２９】

ただし、当該要求は直ちにデータベース１６に書き込まれるのではなく、当該ページを管理するページ管理テーブル３２の出力要求フラグ３３０を出力要求状態に設定するのみとする。出力要求状態に設定することで、当該バッファプールの置換アルゴリズムにしたがって、バッファスチール（他のページを入力するためにデータベースに強制的に書きだす）されるか、遅延書き込み処理部２５によって非同期にデータベース１６に書き込まれるまで、主記憶装置１４上のバッファプール３４上に置かれる。次に、トランザクション内で行ったデータベースへのアクセス処理が終わるとトランザクションを有効なものとするために２フェーズに分けてトランザクションを完結させる。

【００３０】

まず、図１０にトラザンクションのＰＲＥＰＡＲＥ処理フェーズについて説明する。まず

、トランザクションのＰＲＥＰＡＲＥログをログバッファに書き込む（ステップ７２）。次いで、当該ＰＲＥＰＡＲＥログをログファイル３７に強制出力する（ステップ７４）。ＰＲＥＰＡＲＥフェーズが終わるとトランザクションを正常に終了させるためにＣＯＭＭＩＴフェーズに入る。ＣＯＭＭＩＴフェーズの処理概略フローを図１１に示す。トランザクションＣＯＭＭＩＴログをログバッファに出力する（ステップ８２）。ＣＯＭＭＩＴログをログファイル３７に強制出力する。これにより、トランザクションが確約できたので当該トランザクションで取得したすべての排他資源に対するロックを解放する（ステップ８４）。ＣＯＭＭＩＴされたトランザクションは、トランザクション実行時に割り当てられたトランザクション管理テーブルが不要となるため、システム内のアクティブトランザクション表から当該ＣＯＭＭＩＴされたトランザクションを取り除くとともにトランザクション管理テーブルを解放する（ステップ８６）。

【００３１】

次に、本発明に関わるシンクポイント取得処理について、概略処理フローを図１を用いて説明する。シンクポイントの設定の目的は、システムが障害により停止してしまったとしても確実にシステムすなわちデータベースを回復でき、システムを存続できることを確かなものにすることである。また、シンクポイントを設定することによってデータベースの回復時のＲＥＤＯの必要なログの使用を小さくさせることにもある。シンクポイントを取得するタイミングは、システムインプリメンテーションによって様々であるが、定期的に何らかのアクションによって実施される。

【００３２】

例えば、ログファイルへのログの出力回数が一定の回数に達した時点によってシンクポイントを取得する方法がある。この場合、シンクポイント取得のトリガを与えるのはログ書き込み処理部２７である。まず、シンクポイント取得処理部２６にシンクポイント取得処理要求が渡されると、シクポイント取得開始ログを作成し、ログバッファに書き込む（ステップ２５１）。次にシンクポイント取得開始時点でのバッファプールの物理的整合性を一時的に保証するためにロックを確保する（ステップ２５２）。当該ロックは、ラッチと呼ぶことも有り、データベースの論理的な整合性を保証するために用いるロックとは異なり、一般のロックよりも保持時間が短い。バッファプールのロックを確保できたならば、当該ロックが解放されるまでの間はトランザクショによるバッファプールへのアクセスは一時的に待たされる。

【００３３】

次にバッファプール３４中に存在するすべてのページのページ管理テーブルを走査し、出力要求フラグ３３０が出力要求状態になっているページについてシンクポイント取得中フラグ３３５をシンクポイント取得状態に設定する（ステップ２５３）。ステップ２５３と同時にシンクポイント取得においてデータベース１６へ書き込む対象となるページを管理するページ管理テーブルリスト３８を作成する（ステップ２５４）。

【００３４】

図１２に出力対象ページ管テーブルリスト３８の構成を示す。出力対象ページ管理テーブルリスト３８には、当該リスト作成時にリストに登録されたバッファプール３４中の対象ページのページ管理テーブル３２のアドレス３８２ａから３８２ｄとその出力対象ページ数であるシンクポイント出力ページ数カウンタ３８１で構成される。登録されたリストの最後には、次の出力対象ページ管理テーブルがないことを示す情報３２８ｅとして０を設定する。そして、作成された出力対象ページ管理テーブルリスト３８の当該リストが作成された時点でシンクポイント出力ページ数カウンタ３８１が決定される（ステップ２５５）。当該カウンタは、セマフォにより実現されていてもよい。

【００３５】

当該シンクポイントでデータベースへの書きだしを行うページが確約されたので直ちにバッファプールのロックを解放する（ステップ２５６）。当該バッファプールのロック期間中には、一切の外部記憶装置との入出力処理を行わないのでＣＰＵ処理だけの短いロック時間となる。これにより、バッファプールへのロック待ちとなっていたトランザクション

の処理が続行可能となる。そして、作成された出力対象のページ管理テーブルリストに基づいて当該シンクポイントで出力すべきページを遅延書き込み処理部２５に対して要求しデータベースへの出力を開始する（ステップ２５７）。

【００３６】

シンクポイントにおいて出力対象となったすべてのページの出力が完了したか否かは、前記シンクポイント出力ページ数カウンタ３８１が０になったかどうかによって決定される（ステップ２５８）。したがって、シンクポイント出力ページ数カウンタ３８１が０になるまで当該処理は待ち状態となる。これは、シンクポイント出力対象ページの書き込み処理中に他のトランザクションによって書き込み処理が行われることがあるため、他のトランザクションによって書き込み処理が行われ、シンクポイント出力ページ数カウンタ３８１をカウントダウンして０になるまで同期を取る必要があるからである。シンクポイントにおいて出力対象となったすべてのページの出力が完了すると、すなわちシンクポイント出力ページ数カウンタ３８１が０になると待ち状態が解除されるので、シンクポイント取得完了ログを作成し、ログバッファに書き込むと同時にログファイルへの書き込みが強制される（ステップ２５９）。

【００３７】

シンクポイント取得処理が完了した時点で、当該シンクポイントをカレントな有効シンクポイントとするため当該シンクポイント開始ログのＬＳＮをカレント有効シンクポイントとする（ステップ２６０）。この有効シンクポイントのＬＳＮは外部記憶装置１６のような不揮発な記憶装置に格納されることが望ましい。つまり、システムがこの後、障害や故障によりシステムが停止しても、システムの再開始時にこの有効シンクポイントのＬＳＮを読み出して、当該ＬＳＮを起点としてデータベースの回復が行えるようになる。次に、図１で示したステップ２５７のシンクポイント出力対象ページの書き込み処理の処理が概略フローを図１３を用いて説明する。

【００３８】

先ず、出力対象となった出力対象ページ管理テーブルリスト３８の先頭アドレスにカレントポジション（以下、ＣＰと略す）を設定する（ステップ２５６１）。次に、当該ＣＰの指すページ管理テーブルのアドレスが０か否かを判定し、０すなわちすべての出力対象ページのデータベースへの書き込みが完了したか出力対象ページがない場合は当該出力処理を終了する（ステップ２５６２）。以降、出力対象ページが存在する間、以下の処理を繰り返す。まず、ページ管理テーブル３２のロックカウンタ３２８をチェックし、ロックカウンタ３２８が０以上ならば当該出力処理の対象からはずし、当該ページを使用中のトランザクションによって出力を強制するようにする。ロックカウンタ３２８が０の場合、バッファプールにロックを確保する（ステップ２５６４）。そして、出力中フラグ３３２を出力中状態に設定することによってデータベース１６への書き込みが完了するまで他のトランザクションによる参照を待たせる（ステップ２５６５）。

【００３９】

次に、ロックカウンタ３２８が０ということは更新ＬＲＵチェインに接続されているということであるので、更新ＬＲＵチェインから切り離し、参照ＬＲＵチェインの先頭に接続する。さらに、出力要求フラグ３３０をＯＦＦにする（ステップ２５６７）。本処理が完了するとバッファプールのロックを解放する（ステップ２５６８）。そして、データベース１６への書き込み処理を行う（ステップ２５６９）。データベース１６への書き込みが完了した時点で出力中フラグ３３２をＯＦＦにする（ステップ２５７０）。出力が完了するまでの間にトランザクションから参照された場合には入出力待ちチェイン３３３に参照したトランザクションが登録されているので、登録されたすべてのトランザクションをアクティブにするための同期制御処理を行う（ステップ２５７１、２５７２）。こうして、一つのページの処理が終了するとＣＰを次のページ管理テーブルリストのアドレスに更新する。当該出力処理によって、シンクポイント取得開始ログ以前のすべてのログに対応した更新がデータベースに反映されることになる。

【００４０】

シンクポイント取得処理によって選択された出力対象ページが出力処理の対象からはずされたページについては、バッファプール管理部２３がトランザクションの処理によって確実にデータベース１６に書きだされることを保証している。つまり、出力対象となったページを管理するページ管理テーブル３２にはシンクポイント取得中フラグ３３５が設定されたままの状態になっており、トランザクションによって当該シンクポイント取得中フラグ３３５が設定されているページを使用する場合あるいは使用を終了する場合にデータベース１６への出力を強制することによってトランザクション処理を止めることなく確実にシンクポイント取得処理が行われる。バッファプール管理部２３においてどのようにシンクポイント出力対象ページがデータベース１６に書きだされているかを図１４から図１９を用いて説明する。

【００４１】

図１４は、トランザクション内で行われたデータベースに対する更新処理において図９のステップ６７で示した更新対象行格納ページ出力要求処理のバッファプール管理部２３における処理フローを示す。まず、バッファプールにロックを確保する（ステップ４０１）。次にページ管理テーブル３２の出力要求フラグ３３０がＯＮであるかどうかチェックし（ステップ４０２）、ＯＮならばさらにシンクポイント取得中フラグ３３５がＯＮか否かをチェックし（ステップ４０３）、ＯＮということは当該ページがシンクポイント取得中であるので当該トランザクションによってデータベース１６への書き込みを強制される必要がある。

【００４２】

まず、出力要求フラグ３３０をＯＦＦにし（ステップ４０４）、参照ＬＲＵチェインの先頭に接続する（ステップ４０５）。そして、出力中フラグ３３２をＯＮにし（ステップ４０６）、ロックカウンタ３２８を0にし（ステップ４０７）、この時点でバッファプールのロックを解放する（ステップ４０８）。バッファプールのロックを解放したら、直ちにデータベース１６への書き込みを行う（ステップ４０９）。データベース１６への書き込みが完了した時点で出力中フラグ３３２をＯＦＦにする（ステップ４１０）。シンクポイント取得時の対象となったペ－ジをデ－タベ－ス３６に書き込みを行ったのでシンクポイント出力ペ－ジ数カウンタ３８１から1を減ずる（ステップ４１１）。出力が完了するまでの間に他のトランザクションから参照された場合にはロック待ちチェイン３３４に参照したトランザクションが登録されているので、登録された先頭のトランザクションをアクティブにするための同期制御処理を行う（ステップ４１２、４１３）。また、ステップ４０２で出力要求フラグ３３０がＯＦＦの場合は、出力要求フラグ３３０をＯＮにし（ステップ４１４）、更新ＬＲＵチェインの先頭に接続し（ステップ４１５）、ロックカウンタ３２８を0に設定し（ステップ４１６）、バッファプールのロックを解放する（ステップ４１７）。そして、ステップ４１２に進み、ロック待ちチェインがあれば同期制御処理を行う。

【００４３】

図１５は、トランザクション内で行われたデータベースに対するアクセスにおいてページの入力要求処理のバッファプール管理部２３における処理フローを示す。基本的に要求されたページがバッファプール上に存在すればデータベース１６からの読み込み無しで入力処理が完了し、バッファプール上に存在しない場合はデータベース１６からの入力処理を伴う。その際、バッファプール上にページが存在し、ページ入力要求モードが排他モードの場合でかつシンクポイント取得中フラグ３３５がＯＮになっているページについてはあらかじめデータベース１６への書き込みを強制した後、入力要求処理を満たすようにする。

【００４４】

まず、バッファプール３４にロックを確保し（ステップ５０１）、バッファプール３４上に目的のページが存在するか否かをハッシュ管理テーブル３３を使用してサーチする（ステップ５０２）。目的のページがバッファプール３４上に存在したならばページロックモードの種別によって処理を振り分ける（ステップ５０３、５０４）。ページロックモード

が共用モード（Ｓモード）ならば、共用モード入力処理を行い（ステップ５０５）、ページロックモードが排他モード（Ｘモード）ならば排他モード入力処理を行う（ステップ５０６）。目的のページが存在しない場合は空きバッファが有るか否かをバッファプール管理テーブル３１のＦＲＥＥチェインポインタ３１７によりチェックし（ステップ５０７）、ＦＲＥＥチェインポインタ３１７に接続されたバッファが存在する場合は当該ＦＲＥＥチェインに接続されたページ管理テーブル３２を割り当てる（ステップ５０８）。ＦＲＥＥチェインポインタ３１７に該当するページ管理テーブル３２が接続されていない場合は、参照ＬＲＵチェイン及び更新ＬＲＵチェインより最も使用されていないページ管理テーブルをスチールの対象とし、目的ページ入力用のバッファとして割り当てる（ステップ５０９）。

【００４５】

こうして、割り当てらたバッファのページ管理テーブル３２の入力中フラグ３３１を入力中状態に設定するためＯＮにし（ステップ５１０）、ロックカウンタ３２８に１を設定し（ステップ５１１）、バッファプールのロックを解放する（ステップ５１２）。この場合、ページロックモードが排他モードの場合はＸＧＥＴ中フラグ３２９もＯＮに設定する。そして、目的のページをデータベース１６から読み込み（ステップ５１３）、読み込みが完了すると直ちに入力中フラグ３３１をＯＦＦにするとともに入出力待ちチェインあるいはロック待ちチェインに登録されているトランザクションがあれば同期制御処理を行う（ステップ５１４、５１５、５１６）。

【００４６】

次に、ページロックモードが共用モードの場合の入力処理の処理フローを図１６に示す。まず、ロックカウンタ３２８が０か否かをチェックし（ステップ６０１）、０ならば参照ＬＲＵチェイン又は更新ＬＲＵチェインに存在することになる。出力要求フラグ３３０がＯＮか否かをチェックし（ステップ６０２）、ＯＮならば更新ＬＲＵチェインに存在するので更新ＬＲＵチェインから切り離す（ステップ６０３）。出力要求フラグ３３０がＯＦＦならば参照ＬＲＵチェインに存在するので参照ＬＲＵチェインから切り離す（ステップ６０４）。そして、ロックカウンタ３２８に１を設定し（ステップ６０５）、バッファプールのロックを解放する（ステップ６０７）。

【００４７】

また、ステップ６０１によりロックカウンタ３２８が０でない場合、すなわちすでに他のトランザクションが使用中の場合は、ロック待ちチェイン３３４に登録されているトランザクションが存在するか、又は当該ページのロックモードが排他モードで使用中であることを示すＸＧＥＴフラグがＯＮになっているか否かをチェックし（ステップ６０７）、いずれかの条件を満たす場合、ロック待ちチェイン３３４の最後尾に当該入力要求トランザクションを登録し（ステップ６０９）、一旦バッファプールのロックを解放し（ステップ６０９）、当該ロック待ち状態から解放されるのを同期制御処理によって待つ（ステップ６１０）。ステップ６０７の条件をいずれも満たさない場合、すなわち、共用モードで使用中のページの場合は、ロックカウンタ３２８を１増やし（ステップ６１１）、入出力中フラグがＯＮか否かをチェックし（ステップ６１２）、データベース１６からの入力中の場合は、入出力待ちチェイン３３３に当該トランザクションを登録し（ステップ６１３）、一旦バッファプールのロックを解放した後（ステップ６１４）、当該入出力が完了し、当該トランザクションをアクティブ状態にしてくれるまで待つ同期制御処理を行う（ステップ６１５）。ステップ６１２で入出力状態でない場合は、直ちにバファプールのロックを解放することによって処理を終了する（ステップ６１６）。

【００４８】

次に、ページロックモードが排他モードの場合の入力処理の処理フローを図１７に示す。まず、ロックカウンタ３２８が０か否かをチェックし（ステップ７０１）、０ならば参照ＬＲＵチェイン又は更新ＬＲＵチェインに存在することになる。出力要求フラグ３３０がＯＮか否かをチェックし（ステップ７０２）、ＯＮならば更新ＬＲＵチェインに存在するので更新ＬＲＵチェインから切り離す（ステップ７０３）。出力要求フラグ３３０がＯＦ

Ｆならば参照ＬＲＵチェインに存在するので参照ＬＲＵチェインから切り離す（ステップ７０４）。そして、ロックカウンタ３２８に１を設定する（ステップ７０５）。

【００４９】

そして、ロックモードが排他モードであるのでＸＧＥＴフラグ３２９をＯＮにし、入力中フラグ３３１がＯＮか否かをチェックし（ステップ７０７）、データベース１６からの入力中の場合は入出力待ちチェイン３３３に当該トランザクションを登録し（ステップ７０８）、一旦バッファプールのロックを解放した後（ステップ７０９）、当該入出力が完了し、当該トランザクションをアクティブ状態にしてくれるまで待つ同期制御処理を行う（ステップ７１０）。ステップ７０７で入出力状態でない場合は、シンクポイント取得中フラグ３３５がＯＮか否かチェックし（ステップ７１４）、ＯＮの場合は出力中フラグ３３２をＯＮにし（ステップ７１５）、一旦バッファプールのロックを解放する（ステップ７１６）。

【００５０】

そして、目的のページをデータベース１６へ書き込みを強制し（ステップ７１７）、書き込み完了後、出力中フラグ３３２をＯＦＦにし（ステップ７１８）、さらにシンクポイント取得中フラグ３３５及び出力要求フラグ３３０をＯＦＦにする。また、シンクポイント取得の出力対象ページをデータベース３６に出力したのでシンクポイント出力ページ数カウンタ３８１を１減ずる（ステップ７１９）。入出力待ちチェインに登録されているトランザクションがあれば同期制御処理を行う（ステップ７２０、７２１）。ステップ７１４によりシンクポイント取得中フラグ３３５がＯＦＦの場合は直ちにバッファプールのロックを解放する（ステップ７２２）。

【００５１】

図１８は、トランザクションにおいてアクセスしたページの使用が終わった場合にページのロックを解放する処理の処理フローを示す。まず、バッファプールのロックを確保する（ステップ８０１）。次にシンクポイント取得中フラグ３３５がＯＮで有るか否かをチェックし（ステップ８０２）、ＯＮの場合は出力中フラグ３３２をＯＮにし（ステップ８０３）、一旦バッファプールのロックを解放する（ステップ８０４）。そして、当該ページをデータベース１６へ書き込む（ステップ８０５）。データベース１６への書き込みが完了するとページ管理テーブル３２内の出力中フラグ３３２、シンクポイント取得中フラグ３３５、及び出力要求フラグ３３０をＯＦＦにする。また、シンクポイント取得の出力対象ページをデータベース３６に出力したのでシンクポイント出力ページ数カウンタ３８１を１減ずる（ステップ８０６）。

【００５２】

次に、もしこのデータベースへの書き込み最中に当該ページへのアクセスが行われた場合には入出力待ちチェイン３３３に登録されているトランザクションをアクティブにするための同期制御処理を行った後（ステップ８０７、８０８）、バッファプールのロックを再度確保する（ステップ８０９）。次に、ステップ８０２においてシンクポイント取得中フラグ３３５がＯＦＦの場合、あるいはＯＮの場合にデータベース１６への書き込みが終了するとロックカウンタ３２８を１減らし（ステップ８１０）、ロックカウンタ３２８が０でかつＸＧＥＴフラグ３２９がＯＮの場合はＸＧＥＴフラグ３２９をＯＦＦにする（ステップ８１１、８１２、８１３）。

【００５３】

そして、出力要求フラグ３３０がＯＮの場合は更新ＬＲＵチェインへ接続し、出力要求フラグ３３０がＯＦＦの場合は参照ＬＲＵチェインへ接続する（ステップ８１４、８１５、８１６）。ステップ８１１においてロックカウンタ３２８が０でない場合、すなわち他にまだ使用中のトランザクションがいる場合とＬＲＵチェインに接続している最中にロック待ちチェイン３３４に登録されたトランザクションがある場合は、そのトランザクションのロック待ちを解除するための同期制御処理を行い、バッファプールのロックを解放する（ステップ８１７、８１８、８１９）。

【００５４】

図１９は、トランザクションにおいてアクセスしたページが不要になった場合に、当該ページを管理するページ管理テーブルをＦＲＥＥチェインへ登録する処理の処理フローを示す。まず、バッファプールのロックを確保する（ステップ９０１）。次にシンクポイント取得中フラグ３３５がＯＮで有るか否かをチェックし（ステップ９０２）、ＯＮの場合は出力中フラグ３３２をＯＮにし（ステップ９０３）、一旦バッファプールのロックを解放する（ステップ９０４）。そして、当該ページをデータベース１６へ書き込む（ステップ９０５）。データベース１６への書き込みが完了するとページ管理テーブル３２内の出力中フラグ３３２、シンクポイント取得中フラグ３３５、及び出力要求フラグ３３０をＯＦＦにする。また、シンクポイント取得の出力対象ページをデータベース３６に出力したのでシンクポイント出力ページ数カウンタ３８１を１減ずる（ステップ９０６）。

【００５５】

次に、もしこのデータベースへの書き込み最中に当該ページへのアクセスが行われた場合には入出力待ちチェイン３３３に登録されているトランザクションをアクティブにするための同期制御処理を行った後（ステップ９０７、９０８）、バッファプールのロックを再度確保する（ステップ９０９）。次に、ステップ８０２においてシンクポイント取得中フラグ３３５がＯＦＦの場合、あるいはＯＮの場合にデータベース１６への書き込みが終了するとロックカウンタ３２８を０に設定する（ステップ９１０）。

【００５６】

さらに、ＸＧＥＴフラグ３２９及び出力要求フラグ３３０をＯＦＦにする（ステップ９１１、９１２）。ＦＲＥＥチェインに接続する前にロック待ちチェイン３３４に登録されたトランザクションがある場合は参照ＬＲＵチェインに接続するようにし、そのトランザクションのロック待ちを解除するための同期制御処理を行う（ステップ９１３、９１４、９１５）。ステップ９１３においてロック待ちチェイン３３４に登録されているトランザクションがない場合は、バッファプール管理テーブル３１のＦＲＥＥチェインに当該ページ管理テーブル３２を接続し、ハッシュチェインからも切り離す（ステップ９１７、９１８）。これらの処理が終了するとバッファプールのロックを解放する（ステップ９１６）。

【００５７】

以上、本発明を実施例に基づき具体的に説明したが、本発明により構成されたデータベース管理システムがトランザクション処理を始めた場合、システムのスループット、すなわち単位時間当たりのトランザクション処理量を時間の経過で示すと図２０のような効果になる。トランザクションが一様に投入された場合のシステムのスループットを示したものであり、ある一定間隔においてシンクポイントが発生してもスループットの低下は見られない性能効果が得られる。

【００５８】

次に、本発明の効果をさらに引き出すための実施例について説明する。前述した実施例では、シンクポイント取得開始時点のバッファプール上の更新ページをシンクポイント取得時にデータベースの整合性を保証するようにする処理について説明したが、通常シンクポイント間隔では図２に示した遅延書き込み処理部２５がバッファプール上の更新ページ数を基に非同期にデータベースへの書き込みを行っているので、シンクポイント取得時点になるべくバッファプール上の更新ページ数が少なくなるように制御されている。

【００５９】

しかし、図２３の（ａ）を見てわかるようにシンクポイント取得時点の更新ページ数がバッファプール全体に閉める割合が大きい場合には、データベースへ書き出すページ数が増加するためシンクポイント取得に要する処理時間が長くなってしまうことがある。これは、通常シンクポイント間隔で行われる遅延書き込みの頻度を多くする、すなわちバッファプール全体に占める更新ページ数の比率を小さい値に設定することによって、その水準に達した時点で遅延書き込みを起動する方法が考えられる。ここでは、他の方法としてシンクポイント取得開始時点までに、あらかじめ遅延書き込み処理を利用してバッファプール中の更新ページをできるだけ多くデータベースに書き出すためのプレシンクポイントを行う処理の実施例を示す。

【００６０】
図２１は、図２におけるログ書き込み処理部２７において実施されるログファイル３７への書き込み回数をシンクポイント間隔としてプレシンクポイント及びシンクポイントを取得する処理の制御を行う処理の概略フローを示す。トランザクションのコミットやロールバックにおいてログバッファ３５に書き込まれたログをログファイル３７に強制出力する場合、まず、ログファイル３７への書き込みを行う（ステップ２７１０）。ログファイル３７への書き込みが終了すると、シンクポイント間隔を制御するためのログ出力回数を１増やす（ステップ２７１２）。
【００６１】
そして、ログ出力回数がプレシンクポイント取得点に到達したか否かを比較し（ステップ２７１２）、プレシンクポイント取得点に到達した場合は、バッファプールにロックを確保し（ステップ２７１６）、バッファプール管理テーブル３１の遅延書き込み中フラグ３１９をＯＮにする（ステップ２７１８）。遅延書き込み中フラグ３１９をＯＮにすると、バッファプールのロックを解放し（ステップ２７２０）、プレシンクポイント取得処理に制御を渡す（ステップ２７２２）。ステップ２７１４により、プレシンクポイント取得点に達していないか、あるいは通過した場合は、シンクポイント取得点に達したか否かを比較する（ステップ２７２４）。プレシンクポイントに達した場合は、さらに遅延書き込み中フラグ３１９がＯＮか否かをチェックする（ステップ２７２６）。遅延書き込み中フラグ３１９がＯＮの場合は、まだプレシンクポイント取得処理が完了していないので、プレシンクポイント取得処理を直ちに停止させる要求をプレシンクポイント取得処理を行っているプロセスに通達する（ステップ２７２８）。この方法は、シグナルやセマフォなどによって実現される。
【００６２】
プレシンクポイント取得処理が停止するか、あるいはすでに終了していた場合にはシンクポイント取得処理に制御を渡す（ステップ２７３０）。シンクポイント取得処理が終了すると、ログ出力回数を０にクリアする（ステップ２７３２）。次に、プレシンクポイント取得処理の処理フローを図２２を用いて説明する。プレシンクポイント取得処理では基本的にバッファプール管理テーブル３１の更新ＬＲＵチェインに登録されたページを対象にデータベース３６への書き込みを行う。あらかじめ、プレシンクポイント取得点における更新ＬＲＵチェイン中のページのＬＳＮ４２５を参照し、最も最新のＬＳＮをプレシンクポイント取得時の最大ＬＳＮとして記憶しておく。
【００６３】
まず、更新ＬＲＵチェインの先頭アドレスすなわち最も古い更新ページを管理しているページ管理テーブルのアドレスをカレントポジション（以下、ＣＰと略す）に設定する（ステップ２７４０）。ＣＰの指すページ管理テーブル３２が管理するページのＬＳＮ４２５をプレシンクポイント取得時の最大ＬＳＮと比較し（ステップ２７４１）、ＣＰの指すページ管理テーブル３２が管理するページのＬＳＮ４２５の方が小さい間、プレシンクポイント取得処理のデータベース３６への出力対象ページとして処理を行う。
【００６４】
次に、プレシンクポイント取得処理中にシンクポイント取得点に達した場合は、プレシンクポイント取得停止要求が発行されていることがあるため、プレシンクポイント取得停止要求が発行されたか否かをチェックし（ステップ２７４２）、プレシンクポイント取得停止要求が発行されていない場合は、プレシンクポイント取得処理を継続する。まず、バッファプールにロックを確保する（ステップ２７４３）。そして、出力中フラグを出力中状態に設定することによってデータベース２６への書き込みが完了するまで他のトランザクションによる参照を待たせる（ステップ２７４４）。
【００６５】
次に、更新ＬＲＵチェインから切離し、参照ＬＲＵチェインに登録する（ステップ２７４５）。そして、出力要求フラグ３３０をＯＦＦにする（ステップ２７４６）。本処理が完了するとバッファプールのロックを解放する（ステップ２７４７）。そして、データベー

ス１６への書き込みを行う（ステップ２７４８）。データベース１６への書き込みが完了した時点で出力中フラグ３３２をＯＦＦにする（ステップ２７４９）。データベース１６への書き込みが完了するまでの間にトランザクションから参照された場合には入出力待ちチェイン３３３に参照したトランザクションが登録されているので、登録されたすべてのトランザクションをアクティブにするための同期制御処理を行う（ステップ２７５１、２７５２）。ステップ２７４１、２７４２によって、プレシンクポイント取得処理が終了したか、プレシンクポイント取得停止要求を受け付けた場合は、プレシンクポイント取得処理の終了処理として遅延書き込み中フラグ３１９をＯＦＦにする。

【００６６】

プレシンクポイントを設けた場合の性能効果について図２３に示す。プレシンクポイント取得時点は、シンクポイント間隔の設定方法としてログの出力回数を使用する場合にはログ出力回数がある一定数に達した時点とする。シンクポイント間隔の設定方法は、ログの出力回数のみではなく他の別の実施方法であってもよい。例えば、一定時間で設定されるようなシンクポイント間隔ならばその一定時間に達する前のある時間に達した時点をプレシンクポイント取得時点とすることができる。

【００６７】

本実施例では、シンクポイント間隔をログの出力回数によって設定される場合について述べる。この場合、プレシンクポイント取得開始のトリガを与えるのは図２におけるログ書き込み処理部２７である。ログ書き込み処理部２７は、トランザクションからのログの書き込み要求に応じてログバッファ３５にキャッシュし、ログバッファ３５が満杯になるか、トランザクションのコミット、ロールバック時点によってログファイル３７への書き出しが強制される。その場合、ログファイル３７への出力回数を主記憶装置上に記憶しておき、シンクポイント間隔のある一定数に達した時点で遅延書き込み処理部２５に制御を渡す。

【００６８】

例えば、シンクポイント間隔が１０００であった場合、ログファイルへの出力回数が１０００に達した時点がシンクポイント取得時点となり、プレシンクポイントはその８０％の８００に達した時点とするようにする。通常、遅延書き込み処理では、バッファプール中の更新ページ数がある一定数に達するとデータベースへの書き込みを開始するが、すべての更新ページがデータベースへ書き出されるのではなく、一定のページ数に留めることが多い。しかし、プレシンクポイント取得時点では、その時点に存在するバッファプール上のすべての更新ページをデータベースに書き出す。そうすると、図２０の（ｂ）及び（ｃ）のようにシンクポイント取得時点までにバッファプール上のほとんどの更新ページが書き出されるようになる。

【００６９】

理想的には、シンクポイント取得時点にバッファプール上の更新ページ数が０になることが望ましいが、通常（ｂ）のようにプレシンクポイントで更新ページをデータベースに書き出す処理を行ってもシンクポイント取得時点までに全てのページを書き出せないことがある。その場合は、残った更新ページをシンクポイント取得処理にてデータベースへの書き出しを行うようになる。また、逆に（ｃ）のようにプレシンクポイントでバッファプール上の更新ページを全てデータベースに書き出せたが、シンクポイント取得時点に達しない場合もある。その場合は、シンクポイント取得時点に存在するバッファプール上の更新ページがシンクポイント取得時点でのデータベースへの書き出し対象ページとなる。

【００７０】

プレシンクポイントの取得開始点は、図２０の（ｂ）、（ｃ）のようにシンクポイント取得開始時点での更新ページ数を残さないようにするため、シンクポイント取得開始時にプレシンクポイント取得が完了したか否かを記憶しておき、プレシンクポイント取得が完了していない場合はデータベースへ書き込みができなかった更新ページ数からプレシンクポイント取得開始点を早くするようにし、プレシンクポイント取得がすでに完了している場合は、シンクポイント取得時に存在した更新ページ数からプレシンクポイント取得開始点

を遅くするようにプレシンクポイント取得開始点を補正し、次のシンクポイントのプレシンクポイント取得開始点とする処理を行う。

【００７１】

【発明の効果】

以上、説明したように、本発明によれば、データベース管理システムにおいてシンクポイント取得開始時にバッファプール上に存在するすべての更新ページにマークをつけるだけで、そのマークのついた更新ページをデータベースへの書き込み対象とするため、シンクポイント取得中であってもトランザクション処理を全く停止させることなく入出力処理を可能にするので、シンクポイント取得中のトランザクション処理量が０になることがない。また、プレシンクポイント処理を行うことにより、シンクポイント取得時点のバッファプール上に存在する更新ページ数を削減することができるので、シンクポイント取得処理時間を大幅に短縮できる。

【図面の簡単な説明】

【図１】本発明の特長となるシンクポイント取得処理の概略処理フローを示す。

【図２】本発明を実施するためのデータベース管理システムの構成図を示す。

【図３】データベースを構成するページの構造を示す。

【図４】図３におけるページ内の制御情報の構成を示す。

【図５】図２におけるバッファプールを管理するテーブルの構造を示す。

【図６】図２におけるバッファプール管理テーブルの構成情報を示す。

【図７】図２におけるバッファプール中の各ページを管理するページ管理テーブルの構成情報を示す。

【図８】トランザクションの開始処理の処理フローを示す。

【図９】あるトランザクションにおけるデータベースの更新処理の処理フローを示す。

【図１０】トランザクションのＰＲＥＰＡＲＥ処理の処理フローを示す。

【図１１】トランザクションのＣＯＭＭＩＴ処理の処理フローを示す。

【図１２】出力対象ページ管理テーブルリストの構成情報を示す。

【図１３】シンクポイント出力対象ページの書き込み処理の処理フローを示す。

【図１４】図２のバッファプール管理部における出力要求処理の処理フローを示す。

【図１５】図２のバッファプール管理部における入力要求処理の処理フローを示す。

【図１６】図３の共用モードのページ入力処理の処理フローの詳細を示す。

【図１７】図３の排他モードのページ入力処理の処理フローの詳細を示す。

【図１８】図２のバッファプール管理部におけるページロック解放処理の処理フローを示す。

【図１９】図２のバッファプール管理部におけるバッファＦＲＥＥ処理の処理フローを示す。

【図２０】本発明実施時のデータベース管理システムの性能特性のグラフを示す。

【図２１】プレシンクポイント及びシンクポイント取得を制御するログ書き込み処理の処理フローを示す。

【図２２】プレシンクポイント取得処理の処理フローを示す。

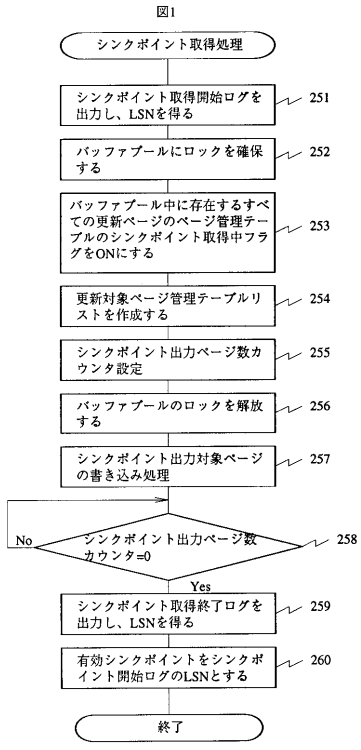
【図２３】本発明実施時に効果を上げるためのプレシンクポイント処理を行った場合のバッファプールにおける更新ページ数の時間経過における比率のグラフを示す。

【符号の説明】

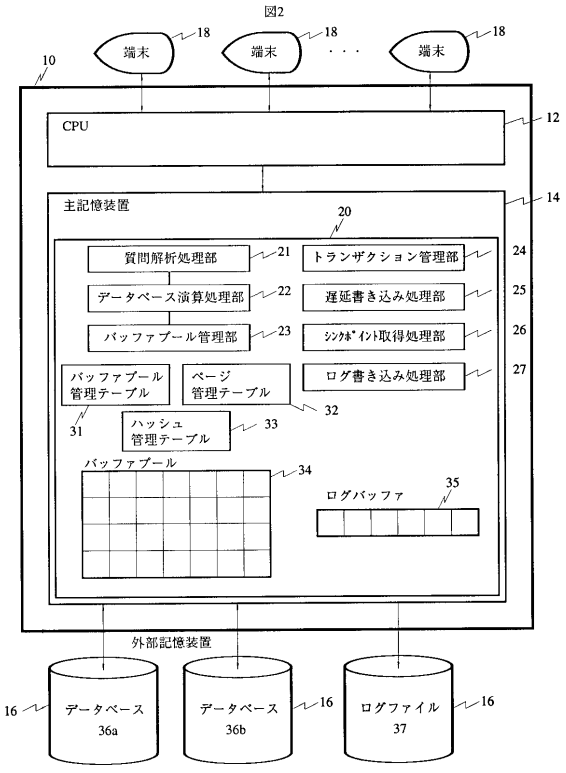
１０：コンピュータシステム、１２：ＣＰＵ、１４：主記憶装置、１６：外部記憶装置、１８：端末、２０：データベース管理システム、２１：質問解析処理部、２２：データベース演算処理部、２３：バッファプール管理部、２４：トランザクション管理部、２５：遅延書き込み処理部、２６：シンクポイント取得処理部、２７：ログ書き込み処理部、３１：バッファプール管理テーブル、３２：ページ管理テーブル、３３：ハッシュ管理テーブル、３４：バッファプール、３５：ログバッファ、３６：データベース、３７：ログファイル、３８：出力対象ページ管理テーブルリスト、４０：ページ、４２：ページ内制御情報、４４：スロット領域、３１９：遅延書き込み中フラグ、３２８：ロックカウンタ、

３２９：ＸＧＥＴフラグ、３３０：出力要求フラグ、３３５：シンクポイント取得中フラグ、３８１：シンクポイント出力ページ数カウンタ、４２１：割当スロット数、４２２：使用スロット数、４２３：空き領域長、４２４：未使用領域先頭オフセット、４２５：ＬＳＮ

【図１】

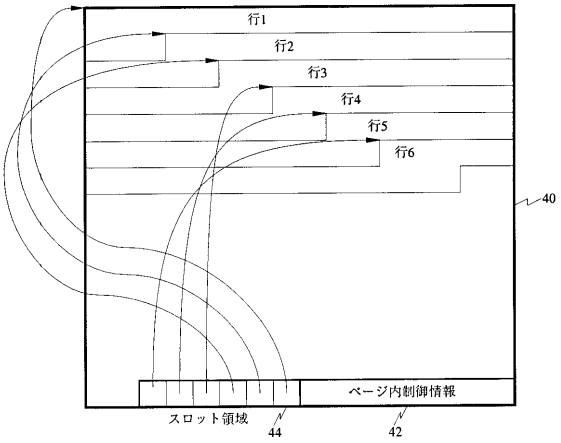


【図２】



【図３】

図3



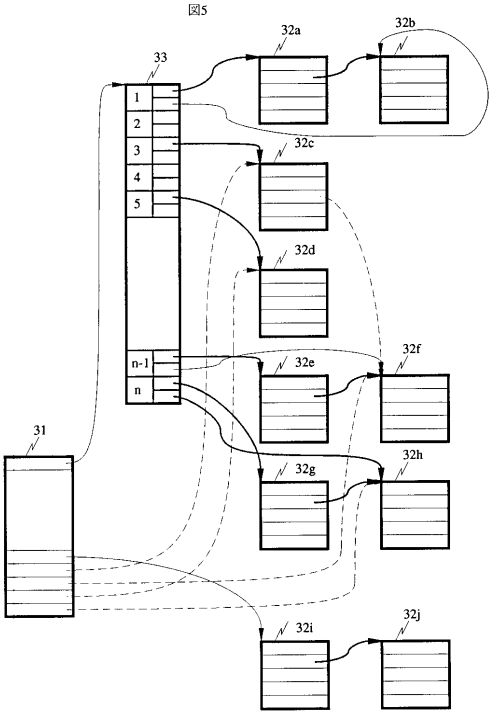
【図４】

図4

割当スロット数	使用スロット数	空き領域長	未使用領域先頭オフセット	LSN
421	422	423	424	425

【図５】

図5



【図６】

図6

バッファプール管理テーブル 31	
バッファ面数	311
ハッシュ管理テーブルポインタ	312
LRUチェイン先頭ポインタ	313
LRUチェイン末尾ポインタ	314
更新LRUチェイン先頭ポインタ	315
更新LRUチェイン末尾ポインタ	316
FREEチェインポインタ	317
更新ページ数	318
遅延書き込み中フラグ	319

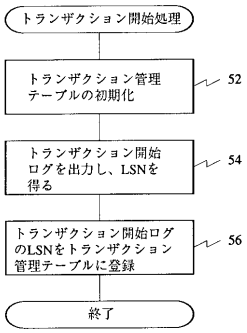
【図７】

図7

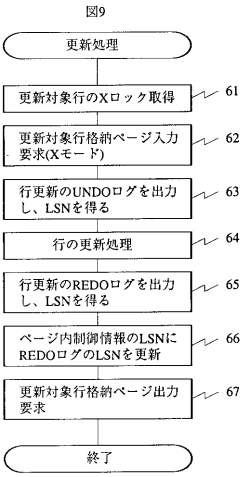
ページ管理テーブル 32	
オブジェクト識別子	321
バッファ番号	322
ページ管理テーブル前方ポインタ	323
ページ管理テーブル後方ポインタ	324
ハッシュチエイン前方ポインタ	325
ハッシュチェイン後方ポインタ	326
ページ番号	327
ロックカウンタ	328
XGETフラグ	329
出力要求フラグ	330
入力中フラグ	331
出力中フラグ	332
入出力待ちチェイン	333
ロック待ちチェイン	334
シンクポイント取得中フラグ	335

【図８】

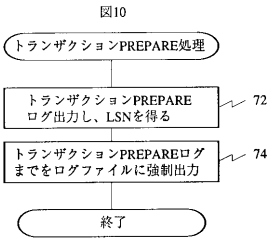
図8



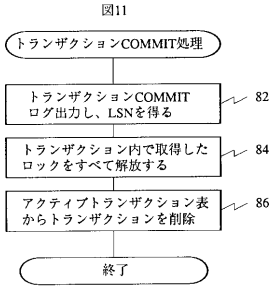
【図９】



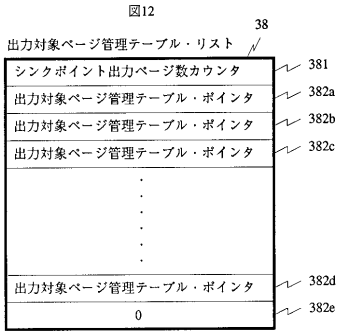
【図１０】



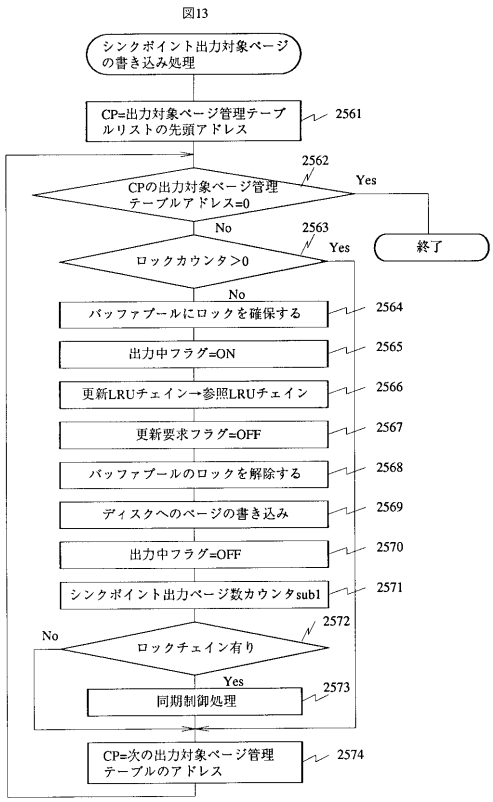
【図１１】



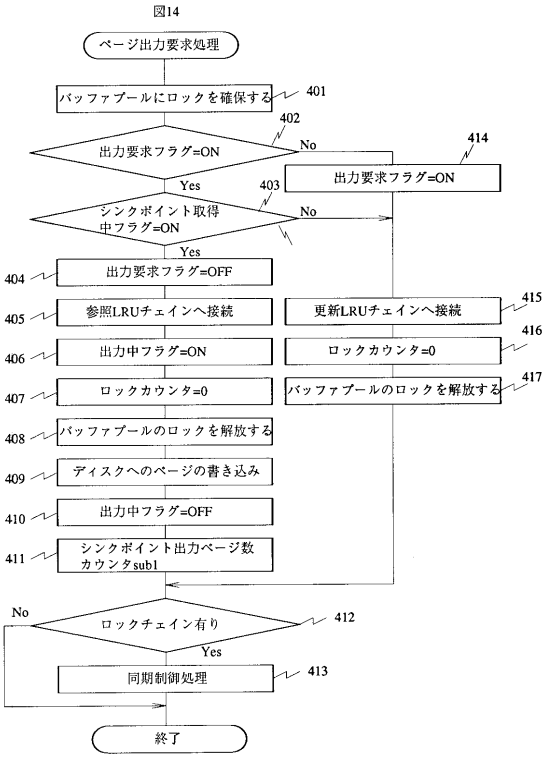
【図１２】



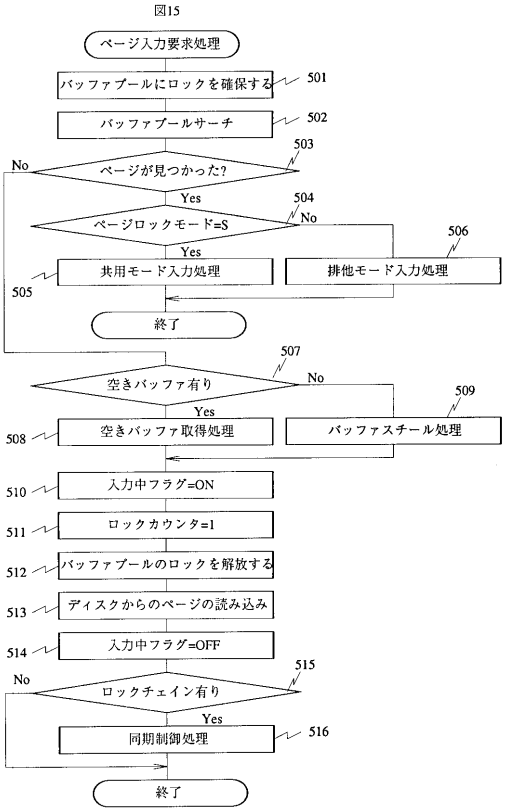
【図１３】



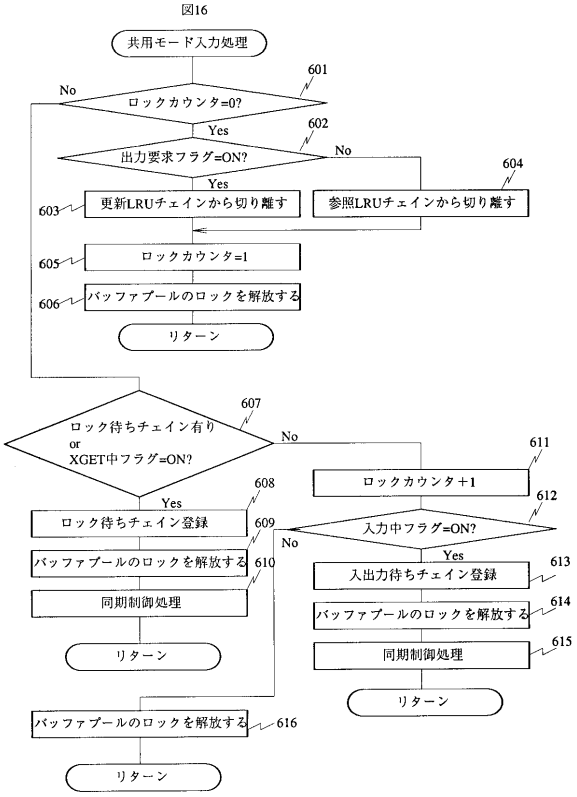
【図１４】



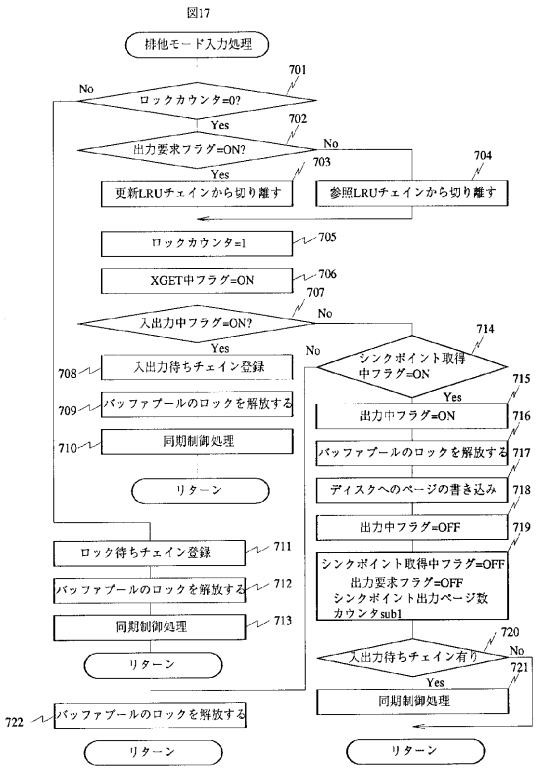
【図１５】



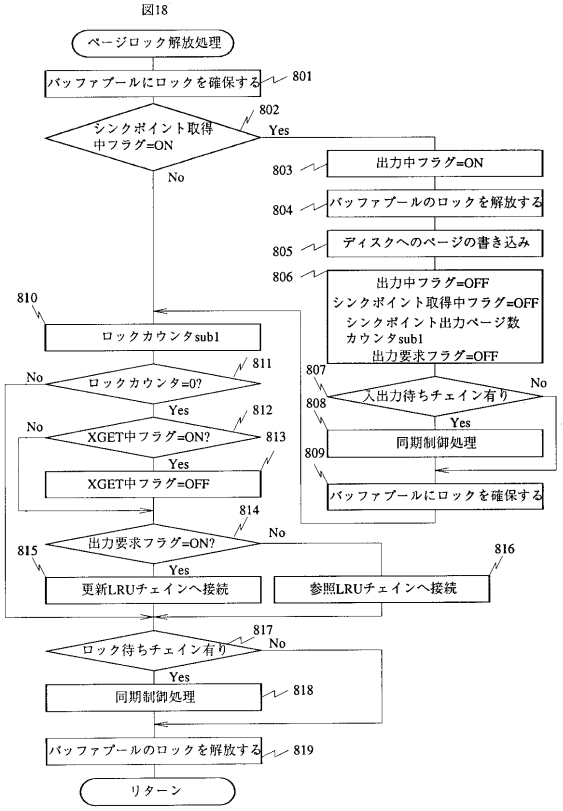
【図１６】



【図１７】

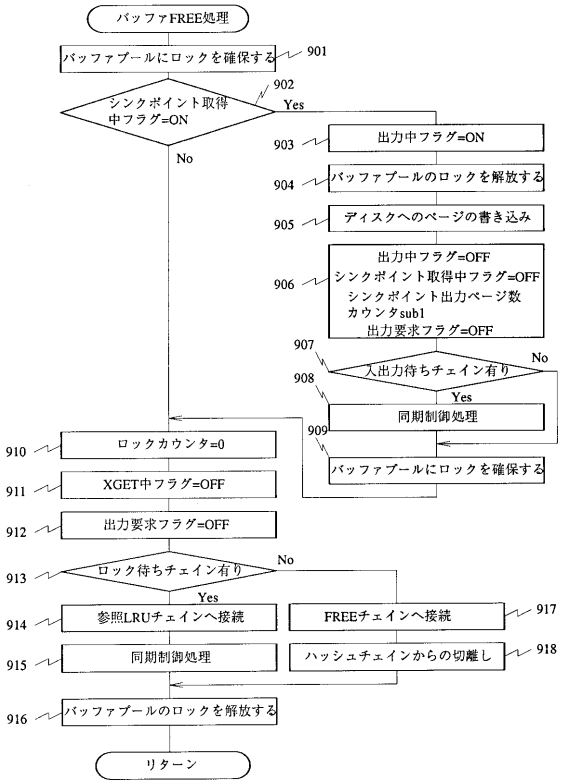


【図１８】



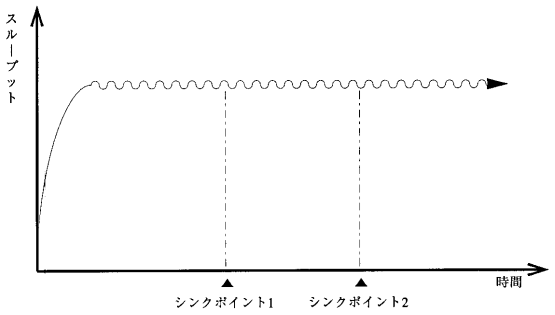
【図１９】

図19



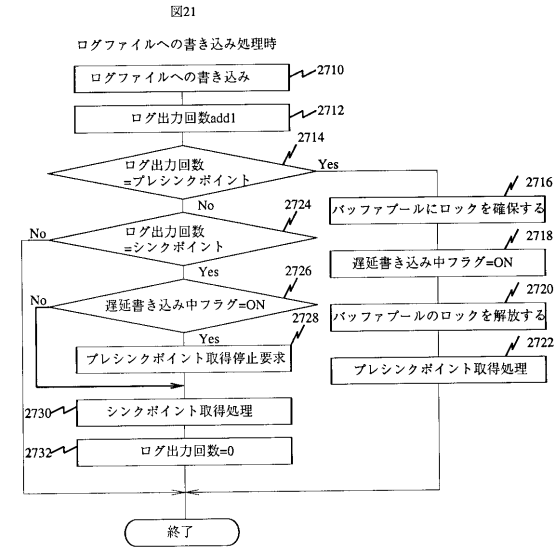
【図２０】

図20



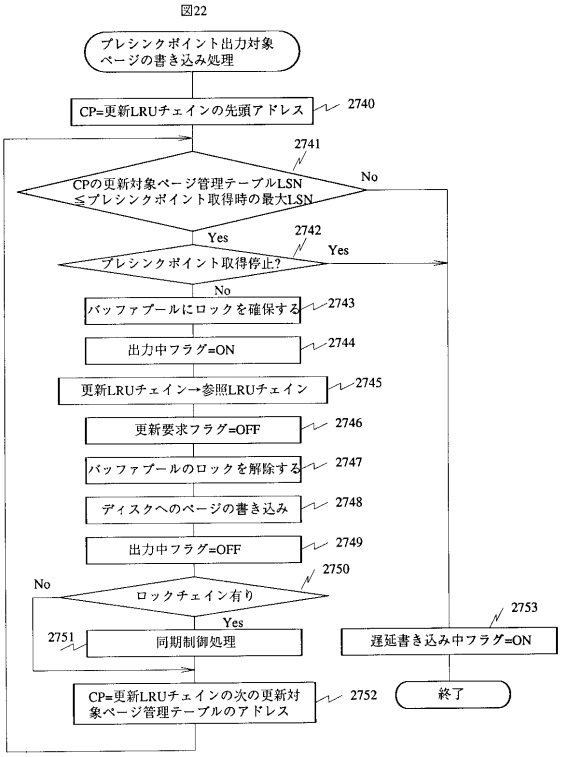
【図２１】

図21



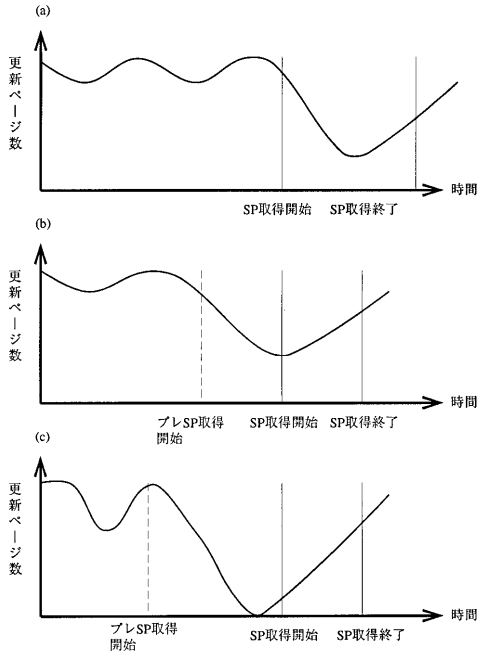
【図２２】

図22



【図２３】

図23


(a)
更新ページ数
時間
SP取得開始
SP取得終了
(b)
更新ページ数
時間
プレSP取得開始
SP取得開始
SP取得終了
(c)
更新ページ数
時間
プレSP取得開始
SP取得開始
SP取得終了

フロントページの続き

(72)発明者　河村　信男
神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内
(72)発明者　正井　一夫
神奈川県横浜市戸塚区戸塚町５０３０番地　株式会社日立製作所　ソフトウエア開発本部内
(72)発明者　山下　信之
神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内
(72)発明者　永井　浩
神奈川県横浜市中区尾上町６丁目８１番地　日立ソフトウエアエンジニアリング株式会社内

審査官　辻本　泰隆

(56)参考文献　特開平０４－２８２７３３（ＪＰ，Ａ）
特開平０６－２１４８５７（ＪＰ，Ａ）
特開平０６－２１４８４８（ＪＰ，Ａ）
特開昭６３－１９５７５５（ＪＰ，Ａ）
特開平０５－１５８７７８（ＪＰ，Ａ）
特開平０３－２６８１４６（ＪＰ，Ａ）
特開昭６３－１３３２４０（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06F 17/30,
G06F 12/00