(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6445049号
(P6445049)
(45)発行日　平成30年12月26日(2018.12.26)　(24)登録日　平成30年12月7日(2018.12.7)

(51)Int.Cl.　　F I
G06F 12/00 (2006.01)　　G06F 12/00 531J
G06F 11/34 (2006.01)　　G06F 12/00 531R
　　G06F 11/34 176

請求項の数 4　(全 18 頁)

(21)出願番号　特願2016-570238 (P2016-570238)
(86)(22)出願日　平成27年1月20日 (2015.1.20)
(86)国際出願番号　PCT/JP2015/051343
(87)国際公開番号　WO2016/117022
(87)国際公開日　平成28年7月28日 (2016.7.28)
審査請求日　平成28年11月28日 (2016.11.28)

(73)特許権者　000005108
株式会社日立製作所
東京都千代田区丸の内一丁目６番６号
(74)代理人　110001678
特許業務法人藤央特許事務所
(72)発明者　友田　敦
東京都千代田区丸の内一丁目６番６号　株式会社日立製作所内
(72)発明者　礒田　有哉
東京都千代田区丸の内一丁目６番６号　株式会社日立製作所内
(72)発明者　牛嶋　一智
東京都千代田区丸の内一丁目６番６号　株式会社日立製作所内

最終頁に続く

(54)【発明の名称】ログの管理方法及び計算機システム

(57)【特許請求の範囲】
【請求項１】
プロセッサとメモリとストレージ装置とを有する計算機システムで、前記プロセッサが所定の処理を実行し、前記ストレージ装置に前記処理の内容を含むログを格納するログの管理方法であって、
前記プロセッサが、前記所定の処理の内容を含むログを生成する第１のステップと、
前記プロセッサが、前記ログを前記ストレージ装置のログファイルに書き込む第２のステップと、
前記プロセッサが、前記ログを格納するログ領域の終端を決定して前記ストレージ装置のログファイルに書き込む第３のステップと、
前記プロセッサが、前記所定の処理の対象となるデータのバックアップデータを生成して、前記ストレージ装置に格納する第４のステップと、
前記プロセッサが、前記バックアップデータと前記ログファイルを読み込んで、前記ログを前記バックアップデータに適用して、前記ログファイルに書き込まれたログ領域の終端までリカバリを実行する第５のステップと、
を含み、
前記第２のステップは、
当該ログが正常であるか否かを示すログヘッダ識別子を当該ログに設定するステップと、
当該ログのサイズを示す情報を当該ログに設定するステップと、を含み、

前記第５のステップは、
前記ログヘッダ識別子を読み込んで当該ログヘッダ識別子が正常であるか否かを判定するステップと、
前記ログヘッダ識別子が不正である場合には、前記サイズを読み込んで次のログを探索するステップと、
前記ログヘッダ識別子が正常なログの場合には、前記ログを前記バックアップデータに適用してリカバリを行うステップと、
を含むことを特徴とするログの管理方法。
【請求項２】
プロセッサとメモリとストレージ装置とを有する計算機システムで、前記プロセッサが所定の処理を実行し、前記ストレージ装置に前記処理の内容を含むログを格納するログの管理方法であって、
前記プロセッサが、前記所定の処理の内容を含む固定長のログを生成する第１のステップと、
前記プロセッサが、前記ログを前記ストレージ装置のログファイルに書き込む第２のステップと、
前記プロセッサが、前記ログを格納するログ領域の終端を決定して前記ストレージ装置のログファイルに書き込む第３のステップと、
前記プロセッサが、前記所定の処理の対象となるデータのバックアップデータを生成して、前記ストレージ装置に格納する第４のステップと、
前記プロセッサが、前記バックアップデータと前記ログファイルを読み込んで、前記ログを前記バックアップデータに適用して、前記ログファイルに書き込まれたログ領域の終端までリカバリを実行する第５のステップと、
を含み、
前記第２のステップは、
当該ログが正常であるか否かを示すログヘッダ識別子を当該ログに設定するステップを含み、
前記第５のステップは、
前記ログヘッダ識別子を読み込んで当該ログヘッダ識別子が正常であるか否かを判定するステップと、
前記ログヘッダ識別子が不正である場合には、予め設定されたサイズを読み込んで次のログを探索するステップと、
前記ログヘッダ識別子が正常な場合には、前記ログを前記バックアップデータに適用してリカバリを行うステップと、
を含むことを特徴とするログの管理方法。
【請求項３】
プロセッサとメモリとストレージ装置とを有する計算機システムであって、
前記計算機システムは、
所定の処理を実行して前記処理の内容を含むログを生成し、前記ストレージ装置に前記ログを格納し、前記所定の処理の対象となるデータのバックアップデータを生成して前記ストレージ装置に格納する実行部と、
前記ログを格納するログ領域の終端を決定して前記ストレージ装置のログファイルに書き込み、リカバリの際には前記バックアップデータと前記ログファイルを読み込んで、前記ログを前記バックアップデータに適用するリカバリを、前記ログファイルに書き込まれたログ領域の終端まで行うログ管理部と、
を有し、
前記実行部は、
当該ログが正常であるか否かを示すログヘッダ識別子と、当該ログのサイズを示す情報を当該ログに設定し、
前記ログ管理部は、

前記ログヘッダ識別子を読み込んで当該ログヘッダ識別子が正常であるか否かを判定し、前記ログヘッダ識別子が不正である場合には、前記サイズを読み込んで次のログを探索し、前記ログヘッダ識別子が正常なログの場合には、前記ログを前記バックアップデータに適用してリカバリを行うことを特徴とする計算機システム。

【請求項４】

プロセッサとメモリとストレージ装置とを有する計算機システムであって、

前記計算機システムは、

所定の処理を実行して前記処理の内容を含む固定長のログを生成し、前記ストレージ装置に前記ログを格納し、前記所定の処理の対象となるデータのバックアップデータを生成して前記ストレージ装置に格納する実行部と、

前記ログを格納するログ領域の終端を決定して前記ストレージ装置のログファイルに書き込み、リカバリの際には前記バックアップデータと前記ログファイルを読み込んで、前記ログを前記バックアップデータに適用するリカバリを、前記ログファイルに書き込まれたログ領域の終端まで行うログ管理部と、

を有し、

前記実行部は、

当該ログが正常であるか否かを示すログヘッダ識別子を当該ログに設定し、

前記ログ管理部は、

前記ログヘッダ識別子を読み込んで当該ログヘッダ識別子が正常であるか否かを判定し、前記ログヘッダ識別子が不正である場合には、予め設定されたサイズを読み込んで次のログを探索し、前記ログヘッダ識別子が正常な場合には、前記ログを前記バックアップデータに適用してリカバリを行うことを特徴とする計算機システム。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、リレーショナルデータベースのマネージメントシステムにおけるトランザクション処理の永続化を目的としたログ出力の同時実行性能の向上に関する。

【背景技術】

【０００２】

デバイスの発達によりサーバに搭載されるＣＰＵに集積されるコア数や、メモリ容量が増大している。これによりリレーショナルデータベースのテーブルやインデックスといった主要なデータをメモリ上に配置するインメモリデータベースが普及してきている。

【０００３】

インメモリデータベースにおいても、システムの障害時にデータベースの状態を復元するためには、ハードディスクやフラッシュメモリなどから構成されるストレージ装置に変更履歴をログとして出力しておく必要がある。

【０００４】

従来のハードディスクドライブ等をログを記録する永続デバイスとして想定した計算機システムにおいては、一回のＩ／Ｏ出力に対するレスポンスがトランザクション処理を行うサーバの処理時間に対して非常に大きかった。このため、トランザクションごとにログのＩ／Ｏ出力を行うのではなく、メモリ上に確保されたログバッファ領域にログを格納した後、ログマネージャスレッドが、ログバッファ領域のログをまとめて永続デバイスに時系列的かつシーケンシャルに出力していた。なお、永続デバイスは、永続的にデータを保持する不揮発性記憶装置である。

【０００５】

複数のトランザクションで１つのログファイルを共有することを効率的に行う方法として、ログファイルを固定サイズのスロットに切り分け、複数のスレッド間でこれらのスロットをログマネージャで順に予約し、ログを書き込んでいく方法が特許文献１に開示されている。

【先行技術文献】

【特許文献】
【０００６】
【特許文献１】米国特許出願公開第２０１４／０２０８０８３号明細書
【発明の概要】
【発明が解決しようとする課題】
【０００７】
昨今、半導体製造技術の進歩によって１つのサーバに搭載可能なコア数が増加し、複数のスレッドで並列的にトランザクションを処理するため、前述のログマネージャを介して行っていたログ出力完了の待ち合わせ処理がボトルネックになってきていた。
【０００８】
一方、複数のスレッド間でのＩ／Ｏ処理に伴うリソースの競合に関しては、ＮＶＭ（Ｎｏｎｅ　Ｖｏｌａｔｉｌｅ　Ｍｅｍｏｒｙ）　Ｅｘｐｒｅｓｓと呼ばれる規格において、スレッドごとにＩ／Ｏキュー等を独立しておき、Ｉ／Ｏ処理がスレッドごとに独立して実行させる技術が確立されている。
【０００９】
これにより、各スレッドは他のスレッドと競合することなくＩ／Ｏ出力を行えるようになった。しかしながら、スレッドごとに割り当てられたＩ／Ｏキューなどのリソースを活用すると、以下の問題が生じる。
【００１０】
第一はログを格納する記憶装置であるログデバイスの歯抜けの発生である。トランザクション処理を行う各スレッドが、それぞれ予約したログデバイスの領域に対して独立して割り当てられたＩ／Ｏキューを利用してＩ／Ｏ出力を行うと、スレッドが独立して発行したＩ／Ｏ処理の完了順序は必ずしも保証されない。
【００１１】
したがって、ある時点においてサーバに障害が発生すると、ログデバイス上に本来は連続して配置されているはずのログのうち、Ｉ／Ｏ処理が完了したスレッドのログだけが永続化され、Ｉ／Ｏ処理が完了しなかったスレッドが書き込むはずであった領域は書き込みされる前の状態（空白）のまま残ってしまう。以下では書き込みが行われなかったブランク領域のことを歯抜け領域（ｔｏｏｔｈｌｅｓｓ　ａｒｅａ）と呼ぶ。
【００１２】
第二は、障害回復時の処理時間の増大である。従来例に示したログマネージャがまとめてログをＩ／Ｏ出力していたときには、ログマネージャが複数のトランザクションのログを連続したアドレスに記録されるように整形してから一度のＩ／Ｏ出力で書き込みを行っていた。このため、上記の歯抜け領域は発生せず、１つのログの後ろの領域にログが記録されていない領域が最初に出現して以降の領域は、ログが書かれていない領域だった。
【００１３】
しかしながら、各スレッドが独立してログを書き出す場合では、上述のように歯抜け領域が発生する場合がある。したがって、ログをもとにデータベースを障害前の状態に復旧する障害回復処理において、ログデバイスを走査する際に、一旦ログが書かれていない領域（歯抜け領域）が出現した場合でも、走査をつづけた先に別のログが記録されている可能性がある。そこで、歯抜け領域が出現しても、ログデバイス全体を走査する必要があり、回復時間を大幅に増大させてしまう、という問題があった。
【００１４】
そこで本発明は、上記問題点に鑑みてなされたもので、複数のログをストレージ装置へ書き込む計算機システムで、リカバリ時にログの読み込みを高速化することを目的とする。
【課題を解決するための手段】
【００１５】
本発明は、プロセッサとメモリとストレージ装置とを有する計算機システムで、前記プロセッサが所定の処理を実行し、前記ストレージ装置に前記処理の内容を含むログを格納

するログの管理方法であって、前記プロセッサが、前記所定の処理の内容を含むログを生成する第１のステップと、前記プロセッサが、前記ログを前記ストレージ装置のログファイルに書き込む第２のステップと、前記プロセッサが、前記ログを格納するログ領域の終端を決定して前記ストレージ装置のログファイルに書き込む第３のステップと、前記プロセッサが、前記所定の処理の対象となるデータのバックアップデータを生成して、前記ストレージ装置に格納する第４のステップと、前記プロセッサが、前記バックアップデータと前記ログファイルを読み込んで、前記ログを前記バックアップデータに適用して、前記ログファイルに書き込まれたログ領域の終端までリカバリを実行する第５のステップと、を含み、前記第２のステップは、当該ログが正常であるか否かを示すログヘッダ識別子を当該ログに設定するステップと、当該ログのサイズを示す情報を当該ログに設定するステップと、を含み、前記第５のステップは、前記ログヘッダ識別子を読み込んで当該ログヘッダ識別子が正常であるか否かを判定するステップと、前記ログヘッダ識別子が不正である場合には、前記サイズを読み込んで次のログを探索するステップと、前記ログヘッダ識別子が正常なログの場合には、前記ログを前記バックアップデータに適用してリカバリを行うステップと、を含む。

【発明の効果】

【００１６】

本発明によれば、ログ領域の終端がログファイルに記載されているので、障害回復等のリカバリ時にストレージ装置を走査する領域の範囲が特定され、ストレージ装置の走査時間が短縮され、結果として計算機システムの障害回復時間を短縮することができる。

【図面の簡単な説明】

【００１７】

【図１】本発明の第１の実施例を示し、計算機システムの一例を示すブロック図である。

【図２】本発明の第１の実施例を示し、データベースサーバの一例を示す機能ブロック図である。

【図３】本発明の第１の実施例を示し、ログ管理部の構成の一例を示すブロック図である。

【図４】本発明の第１の実施例を示し、トランザクション処理の一例を示すフローチャートである。

【図５】本発明の第１の実施例を示し、ログ出力処理のフローチャートである。

【図６】本発明の第１の実施例を示し、ログのデータ構造の一例を示す図である。

【図７】本発明の第１の実施例を示し、リカバリ処理の一例を示すフローチャートである。

【図８】本発明の第１の実施例を示し、可変長のログでブランク領域の次のログを探索する例を示す図である。

【図９】本発明の第２の実施例を示し、固定長のログの構造の一例を示す図である。

【発明を実施するための形態】

【００１８】

以下、本発明の一実施形態について添付図面を用いて説明する。

【実施例１】

【００１９】

以下、図面を参照して、一実施例を説明する。

【００２０】

なお、以下の説明では、同種の要素の参照符号には同一の親番号を含んでいる。同種の要素を区別して説明する場合には、個々の要素を識別する参照符号（例えば、アルファベット）を使用し（例えばスレッド１１０Ａ、１１０Ｂ、…）、同種の要素を区別しないで説明する場合には、要素の参照符号のうちの親番号のみ使用することがある（例えばスレッド１１０）。

【００２１】

また、以下の説明では、「プログラム」を主語として処理を説明する場合があるが、プ

ログラムは、プロセッサによって実行されることで、予め定められた処理を、適宜に記憶資源（例えばメインメモリ４１６）及び／又は通信インターフェイスデバイス等を用いながら行うため、処理の主語がプロセッサとされてもよい。プログラムを主語として説明された処理は、プロセッサ或いはプロセッサを有する装置（例えばデータベースサーバ）が行う処理としてもよい。また、プロセッサは、処理の一部又は全部を行うハードウェア回路を含んでもよい。プログラムは、プログラムソースから各コントローラにインストールされてもよい。プログラムソースは、例えば、プログラム配布計算機又は記憶メディアであってもよい。

【００２２】

図１は、本発明の計算機システムの構成の一例を示すブロック図である。データベースサーバ４０１に外部ストレージ装置４０２が、例えば通信ネットワーク４０３を介して接続されている。

【００２３】

データベースサーバ４０１は、計算機であって、例えば、パーソナルコンピュータや、ワークステーション又はメインフレームであってよく、もしくは、これらの計算機において仮想化プログラムによって構成された仮想的な計算機であってよい。

【００２４】

データベースサーバ４０１は、Ｉ／Ｏアダプタ４１３と、メインメモリ４１６と、記憶デバイス４１５及びそれらに接続されたプロセッサ４１４を有する。プロセッサ４１４は、例えば、マルチコアのマイクロプロセッサ又は、マイクロプロセッサと専用ハードウェア回路を含んだモジュールで構成してもよい。プロセッサ４１４は、メインメモリ４１６にロードされたコンピュータプログラムを実行する。プロセッサ４１４により実行されるコンピュータプログラムは、例えば、オペレーティングシステム（ＯＳ）１１７及びデータベース管理システム（Data Base Management System：以下ＤＢＭＳ）４１２である。

【００２５】

メインメモリ４１６は、例えば、揮発性のＤＲＡＭ（Dynamic Random Access Memory）等であり、プロセッサ４１４によって実行されるプログラムと、プログラムが使用するデータを一時的に記憶する。なお、メインメモリ４１６には、不揮発性の半導体メモリを採用してもよい。

【００２６】

記憶デバイス４１５は、不揮発性の記憶媒体を含み、例えば、ＨＤＤ（Hard Disk Drive）又はＳＳＤ（Solid State Drive）である。記憶デバイス４１５は、プログラム及びプログラムが使用するデータを格納してよい。Ｉ／Ｏアダプタ４１３は、通信ネットワーク４０３とデータベースサーバ４０１を接続する。

【００２７】

外部ストレージ装置４０２は、複数の記憶デバイスを含む記憶デバイス群４４３を有する装置であり、例えば、ディスクアレイ装置である。なお、外部ストレージ装置４０２は、複数の記憶デバイスに代えて、単一の記憶デバイスであってもよい。

【００２８】

外部ストレージ装置４０２は、複数のログを格納したログファイル３０１を記憶する。外部ストレージ装置４０２は、データベースサーバ４０１からログのＩ／Ｏ要求を受け付ける。外部ストレージ装置４０２は、当該Ｉ／Ｏ要求に従いデータ（例えばログ）の読み書きを行い、読み書きの結果をデータベースサーバ４０１に応答する。なお、記憶デバイス群４４３が有する記憶デバイスは、不揮発性の記憶媒体を有するデバイスであって、例えば、ＨＤＤ又はＳＳＤである。記憶デバイス群４４３は、ＲＡＩＤ（Redundant Array of Independent Disks）グループで構成してもよく、所定のＲＡＩＤレベルでデータを記憶してもよい。記憶デバイス群４４３の記憶空間に基づく論理的な記憶デバイス（例えば、論理ユニット、論理ボリューム、ファイルシステムボリューム）が、データベースサーバ４０１に提供され、当該論理的な記憶デバイス上にログファイル３０１が格納されてもよい。なお、本実施例において、ログファイル３０１は、ログを格納するログ格納領

域の一例である。

【００２９】

外部ストレージ装置４０２は、記憶デバイス群４４３に加えて、Ｉ／Ｏアダプタ４４１及びこれらに接続されたストレージコントローラ４４２を有する。Ｉ／Ｏアダプタ４４１は、外部ストレージ装置４０２を通信ネットワーク４０３に接続し、通信ネットワーク４０３を介して、データベースサーバ４０１と接続される。通信ネットワーク４０３を介した通信プロトコルとしては、例えば、ファイバチャネル（ＦＣ）や、ＳＣＳＩ（Small Computer System Interface）、又は、ＴＣＰ／ＩＰ（Transmission Control Protocol／Internet Protocol）が採用されてよい。例えば、ファイバチャネルもしくはＳＣＳＩが採用される場合、Ｉ／Ｏアダプタ４４１（及び４１３）は、ホストバスアダプタと呼ばれることがある。

【００３０】

ストレージコントローラ４４２は、例えば、メモリ及びプロセッサを含み、データベースサーバ４０１からのＩ／Ｏ要求に応じて、ログファイル３０１を格納した記憶デバイス群４４３との間でデータの読出し、もしくは、書込みを行う。

【００３１】

図２は、実施例に係るデータベースサーバ４０１の機能ブロック図である。

【００３２】

本実施例１のＤＢＭＳ４１２は、例えば、インメモリデータベースである。ＤＢＭＳ４１２は、テーブル１１２と、インデックス１１３とを、メインメモリ４１６上に配置する。

【００３３】

さらにＤＢＭＳ４１２はロック機構１１６を含んでよい。ロック機構１１６は２以上のスレッド１１０Ａ～１１０Ｃが競合することを避けるためである。ロック機構１１６は、テーブル１１２やインデックス１１３をロックするモジュールである。ロック機構１１６は、ロックの取得済か否かを表す情報を含むことができる。例えば、ロック機構１１６は、ロック取得済なら値「１」でよくロック未取得なら値「０」でよい。

【００３４】

ＤＢＭＳ４１２は、ログバッファ１１４とログ管理部１１５を有する。ログバッファ１１４は、テーブル１１２やインデックス１１３への更新履歴を含んだログを一時的に格納する。ログ管理部１１５は、ログファイル３０１およびログファイル３０１へのログの書き出しを管理する。なお、ログ管理部１１５は、ログファイル３０１とバックアップデータを読み込んで、ログをバックアップデータに適用してテーブル１１２を復元するリカバリ処理部１２５を含むことができる。

【００３５】

ＤＢＭＳ４１２は、クエリ発行元からクエリを受信し、受信したクエリの実行において、１又は複数のトランザクションを実行する。具体的には、ＤＢＭＳ４１２は、クエリ受付部４２１、クエリ実行プラン生成部４２２及びクエリ実行部４２４を有する。

【００３６】

クエリ受付部４２１は、クエリ発行元が発行するクエリを受け付ける。クエリは、例えば、構造化問合せ言語（ＳＱＬ、Structured Query Language）によって記述される。１つのクエリで複数のトランザクションが記述されていてもよいし、複数のクエリで複数のトランザクションが記述されてもよい。

【００３７】

また、クエリ発行元は、ＤＢＭＳ４１２の内部のコンピュータプログラムであってよいし、ＤＢＭＳ４１２外部のコンピュータプログラムであってよい。例えば、外部のコンピュータプログラムは、データベースサーバ４０１内で実行されるコンピュータプログラム（例えばアプリケーションプログラム）であってもよいし、データベースサーバ４０１に接続されたクライアント計算機等の装置で実行されるコンピュータプログラム（例えば、アプリケーションプログラム）であってもよい。

【００３８】
クエリ実行プラン生成部４２２は、クエリ受付部４２１が受け付けたクエリから、当該クエリを実行するために必要な１以上のデータベースオペレーションを含むクエリ実行プランを生成する。
【００３９】
クエリ実行プランは、例えば、１以上のデータベースオペレーションと、データベースオペレーションの実行順序の関係を含む情報であり、クエリ実行プラン情報４２３として格納される。クエリ実行プラン情報４２３は、データベースオペレーションをノード、データベースオペレーションの実行順序の関係をエッジとする木構造で表されることがある。
【００４０】
クエリ実行部４２４は、クエリ実行プラン生成部４２２が生成したクエリ実行プランに従って、クエリ受付部４２１が受け付けたクエリを実行し、クエリの実行結果をクエリ発行元に応答する。
【００４１】
この際、クエリ実行部４２４は、データベースオペレーションの実行に必要なデータの読出し要求（参照要求）を発行し、当該読出し要求に従いテーブル１１２からデータを読み出す。クエリ実行部４２４は、読み出したデータを使用して、クエリに応じたデータベースオペレーションを実行してデータを算出し、読出し元レコードのデータを算出後のデータに更新する書込み要求を発行する。
【００４２】
クエリ実行部４２４は、データベースオペレーションを、１以上のスレッド１１０Ａ～１１０Ｃを実行することにより行う。なお、ＤＢＭＳ４１２において、複数のスレッド１１０Ａ～１１０Ｃが並行して実行される。このため、プロセッサ４１４は、複数のコアを有する。複数のコアは、１又は複数のＣＰＵに存在する。スレッド１１０は、タスクと呼ばれてもよい。スレッド１１０の実装としては、例えば、ＯＳ１１７が実現するプロセスやカーネルスレッド等のほか、ライブラリ等が実現するユーザスレッドを用いてよい。１つのスレッド１１０により、１以上のデータベースオペレーションに対応した１つのトランザクションが実行されてよい。以下、クエリ実行部４２４がスレッド１１０を実行することにより行われる処理の主語を、スレッド１１０、とすることがある。
【００４３】
クエリ実行部４２４（スレッド１１０）は、トランザクションの実行と、トランザクションの実行結果（または処理内容）を含むログを生成する。そして、クエリ実行部４２４の各スレッド１１０は、外部ストレージ装置４０２内のログファイル３０１にログを書き込むために、外部ストレージ装置４０２に対するＩ／Ｏ要求をＯＳ１１７に発行する。ＯＳ１１７は、当該Ｉ／Ｏ要求を受け付け、外部ストレージ装置４０２へＩ／Ｏ要求を発行する。
【００４４】
Ｉ／Ｏアダプタ４１３には、複数のＩ／Ｏキュー２０１（２０１Ａ～２０１Ｃ）が設定される。トランザクションの処理において、スレッド１１０が、ログの書込みのためのＩ／Ｏ要求を外部ストレージ装置４０２へ発行するが、Ｉ／Ｏキュー２０１には、当該Ｉ／Ｏ要求が格納される。具体的には、Ｉ／Ｏ要求は、ＯＳ１１７によりＩ／Ｏキュー２０１に格納される。
【００４５】
外部ストレージ装置４０２が、ログファイル３０１を記憶する。ログファイル３０１に、Ｉ／Ｏ要求の書込み対象のログが記録される。
【００４６】
本実施例１では、クエリ実行部のスレッド１１０と、Ｉ／Ｏキュー２０１が、１：１対応している。つまり、スレッド１１０Ａ～１１０Ｃ毎に、１つのＩ／Ｏキュー２０１Ａ～２０１Ｃが設定される。具体的には、スレッド１１０ＡにＩ／Ｏキュー２０１Ａが関連付

けられている。例えば、スレッド１１０Ａが、テーブル１１２のレコードを更新したことを表すログのＩ／Ｏ要求を、ログファイル３０１に対して発行するようになっている。発行されたＩ／Ｏ要求は、ログバッファ１１４を経由して、ＯＳ１１７に送られる。ＯＳ１１７が、ログファイル３０１に対するＩ／Ｏ要求を受けて、当該Ｉ／Ｏ要求を、スレッド１１０Ａに対応するＩ／Ｏキュー２０１Ａに格納する。Ｉ／Ｏキュー２０１Ａに格納されたＩ／Ｏ要求は、ＯＳ１１７によりＩ／Ｏキュー２０１から外部ストレージ装置４０２に送られる。外部ストレージ装置４０２は、当該Ｉ／Ｏ要求の書込み対象データであるログを、ログファイル３０１に書き込む。

【００４７】

図２に示すＤＢＭＳ４１２の構成は一例に過ぎない。例えば、或る構成要素は複数の構成要素に分割されていてもよく、複数の構成要素が１つの構成要素に統合されていてもよい。

【００４８】

図３は、ログ管理部１１５の構成の一例を示すブロック図である。ログ管理部１１５は、ロック機構１２１と、ログファイルアドレス１２２と、ログ領域終端アドレス１２３と、ログ領域追加フラグ１２４と、リカバリ処理部１２５とを有する。

【００４９】

ロック機構１２１も、図２に示したロック機構１１６と同様に、ログ管理部１１５のロックの取得済か否かを表すデータでよい。例えば、ロック機構１２１は、ロック取得済なら値「１」でよくロック未取得なら値「０」でよい。ログファイルアドレス１２２、ログ領域終端アドレス１２３、ログ領域追加フラグについて書き込みまたは読み出しを行うときには、ロック機構１２１のロックを取得していなければならない。

【００５０】

ログファイルアドレス１２２は、ログファイル３０１におけるログの書込み先アドレスである。ログファイルアドレス１２２が表すアドレス（値）は、ログファイル３０１にログを書き出す度に出力するログのサイズ分だけ加算される。なお、ログファイルアドレス１２２や後述のログ領域終端アドレス１２３は、外部ストレージ装置４０２上でのログの格納領域の終端を示す値であり、例えばＬＢＡ（Ｌｏｇｉｃａｌ　Ｂｌｏｃｋ　Ａｄｄｒｅｓｓ）等を採用することができる。

【００５１】

ログ領域終端アドレス１２３はログファイルアドレス１２２の上限値であり、当該上限値を超えた場所にログを書くことはできない。また、リカバリ処理においては、ログ領域終端アドレス１２３にリカバリ処理におけるログファイルの走査範囲の上限がセットされる。リカバリ処理において、ログ領域終端アドレス１２３を超えて走査を行わない。

【００５２】

ログ領域追加フラグ１２４は、トランザクションを処理するスレッドが、ログ出力する領域で領域の追加処理を実行中の場合には、本フラグがセットされる。例えば、本フラグは追加中なら「１」、そうでない場合には「０」でよい。

【００５３】

リカバリ処理部１２５は、ログファイル３０１のログを適用してテーブル１１２を復旧するリカバリ処理を実行する。リカバリ処理部１２５は、図示しない管理装置などから所定の指令を受け付けたときに処理を開始する。

【００５４】

図４は、トランザクション処理のフローチャートである。なお、以下の説明では、１つのトランザクションを例に取り説明する。スレッド１１０Ａが、トランザクションＡを開始すると、トランザクションＡに対応した指示（クエリ中の指示）に基づいて、参照及び更新セットの生成を行う（Ｓ３０１）。

【００５５】

参照及び更新セットは、レコードの参照（テーブル１１２の読出し要求）とレコードの更新（テーブル１１２、インデックス１１３の書込み要求）とのセットである。参照及び

更新セットは、テーブル１１２と、インデックス１１３を更新するための要求セットであるが、ステップＳ３０１の時点では、テーブル１１２、インデックス１１３の変更は行われず、トランザクションＡに対応したローカルメモリ領域（メインメモリ４６１上に確保された領域（図示せず））に参照及び更新セットが保持される。

【００５６】

次に、スレッド１１０Ａが、コミット判定を行う（Ｓ３０２）。コミット判定は、例えば、トランザクションＡが参照及び更新セットに基づいて、テーブル１１２、インデックス１１３に対して行う変更が他のトランザクションとの整合性を保てているか否かをデータベースのアイソレーションレベル（またはトランザクション分離レベル）に応じて行われる。

【００５７】

コミット判定がＮＧ（処理失敗など）の場合（Ｓ３０３：Ｎｏ）、スレッド１１０Ａは、アボート処理を行い（Ｓ３０７）、アボート完了通知を出力して、トランザクションを終了する。

【００５８】

コミット判定がＯＫ（処理完了など）の場合（Ｓ３０３：Ｙｅｓ）、スレッド１１０Ａは、ログ出力処理を実行する（Ｓ３０４）。ログ出力処理は、後述するように、所定の処理（トランザクション）が完了する度に処理の内容を含むログをログファイル３０１へ書き込む処理である。

【００５９】

次に、スレッド１１０Ａは、参照及び更新セットに基づいてテーブル１１２、インデックス１１３をそれぞれ更新し（Ｓ３０５）、コミット完了通知を出して（Ｓ３０６）、トランザクションを終了する。

【００６０】

上記処理により、スレッド１１０Ａは、トランザクション処理が完了すればコミット完了通知を出力し、トランザクション処理が失敗すればアボート完了通知を出力する。

【００６１】

図５は、ログ出力処理（図４のＳ３０４）の一例を示すフローチャートである。まず、トランザクションを実行しているスレッド１１０Ａは、ログ管理部１１５のロック機構１２１を取得する（Ｓ５０１）。

【００６２】

次に、スレッド１１０Ａは、ログ領域終端アドレス１２３とログファイルアドレス１２２の差が一定値以下であり、さらにログ領域追加フラグ１２４がセットされていないことを判定する（Ｓ５０２）。ここで、一定値は、ログ領域終端アドレス１２３の更新に要する時間内にログ領域を使い切らない十分な容量であり、予め設定されてＤＢＭＳ１０２で保持しているものとする。

【００６３】

ステップＳ５０２の判定結果がＹｅｓの場合、スレッド１１０Ａは、ログ領域を拡張する準備を行う。つまり、スレッド１１０Ａは、ログ領域追加フラグ１２４をセットし（Ｓ５０３）、ログバッファ１１４に生成した当該トランザクションのログに、ログ領域拡張ログ（またはログ領域拡張情報）を追加する（Ｓ５０４）。

【００６４】

一方、ステップＳ５０２の判定結果がＮｏの場合には、スレッド１１０Ａは、ログファイルアドレス１２２を取得し、ログバッファ１１４に生成したログ（すなわち、書き込み予定のログ）のサイズをログファイルアドレスに加算し（Ｓ５０５）、ログ管理部１１５のロック機構１２１を解放する（Ｓ５０６）。

【００６５】

スレッド１１０Ａは、ログバッファ１１４に用意したログの書き込み要求（ログ管理部１１５から取得したログファイルアドレス１２２を指定した書込み要求）を発行する（Ｓ５０７）。スレッド１１０Ａは、外部ストレージ装置４０２からＩ／Ｏアダプタ４１３を

介して書込み完了通知を受信した場合に書込み処理を完了する（Ｓ５０８）。
【００６６】
上記の処理により、スレッド１１０Ａによってログファイル３０１には新たなログが書き込まれ、ログファイル３０１のログファイルアドレス１２２が更新される。
【００６７】
書き込みが完了すると、スレッド１１０Ａは、当該トランザクションのログにログ領域拡張ログを追加したか否かを判定する（Ｓ５０９）。このログ領域拡張ログは、ステップＳ５０４で追加される情報である。
【００６８】
ステップＳ５０９の判定結果が、Ｙｅｓの場合、スレッド１１０Ａは、ログ領域の拡張を行う。まず、スレッド１１０Ａは、ログ管理部１１５からロック機構１２１を取得する（Ｓ５１０）。
【００６９】
次に、スレッド１１０Ａは、上記ステップＳ５０４でログファイル３０１に記載したログ領域拡張情報に対応する値を加算してログ領域終端アドレス１２３にセットする（Ｓ５１１）。すなわち、スレッド１１０Ａは、ログ領域の終端アドレスに予め設定したサイズを加算してログ領域を拡張する。
【００７０】
そして、スレッド１１０Ａは、ログファイル３０１のログ領域終端アドレス１２３を更新する。さらに、スレッド１１０Ａは、ログ領域追加フラグ１２４をクリアする（Ｓ５１３）。上記ステップＳ５０９～Ｓ５１３の処理により、ログファイル３０１のログ領域が拡張される。
【００７１】
一方、ステップＳ５０９において、判定結果がＮｏの場合には、スレッド１１０Ａは、そのままログ出力処理を終了する。
【００７２】
上記処理により、スレッド１１０Ａは、ログをログファイル３０１へ書き込む際に、ログ領域が不足する恐れがある場合にはログ領域を拡張することができる。
【００７３】
図６は、ログのデータ構造の一例を示す図である。１つのトランザクションに対して、少なくとも１つのログ３０が生成される。
【００７４】
ログファイル３０１に格納されるログ３０は、ログヘッダ識別子３３とログサイズ３４から構成されるログヘッダ３１と、データベースへの変更履歴およびログ領域拡張ログを格納したログ本体３２とから構成される。なお、ログサイズ３４は、ログ３０が可変長の場合にログ３０のサイズを示す値である。
【００７５】
ログヘッダ識別子３３は、ログヘッダ３１の先頭に格納されており、値を所定の方法で検証することにより、当該アドレスからログ３０が正常に開始していることが判定できるものである。例えばもっとも容易な実装方法として、ログヘッダ３１の開始アドレスの値（またはアドレスのハッシュ値）が格納されている場合、当該アドレスから有効なログヘッダ３１が開始していると判定することができる。なお、偶然に記録されていたビット列が、このような判定で正常と誤判定されることがないように、例えば判定に使用するビット長を長く、アドレスとハッシュ値を併記するなどしてもよい。
【００７６】
リカバリ処理部１２５は、ログヘッダ３１からログヘッダ識別子３３を読み込んで、検証することにより当該ログ３０が有効であるか否かを判定することができる。
【００７７】
なお、ログファイル３０１は、図示のログ３０を複数格納するファイルである。ログファイル３０１には、上記ログ３０に加えてログファイルアドレス１２２とログ領域終端ア

ドレス１２３が格納される。また、ログファイルアドレス１２２とログ領域終端アドレス１２３は、ログファイル３０１内の所定の領域（例えば、先頭）に格納される。

【００７８】

図７は、リカバリ処理のフローチャートである。リカバリ処理では、ＤＢＭＳ１０２が外部ストレージ装置４０２へバックアップしたある時点のデータベース（テーブル１１２）と、バックアップを生成した時点からの差分を逐次的に記録したログ３０とから最新のデータベース（テーブル１１２）を復元する。

【００７９】

本実施例１では、ＤＢＭＳ１０２が所定のタイミングとなる度に外部ストレージ装置４０２へデータベースのバックアップを行う。また、ＤＢＭＳ１０２は、テーブル１１２の更新、追加、削除を行う度にログ３０を生成してログファイル３０１へ書き込む。そして、ＤＢＭＳ１０２は、所定の指令を受け付けるとリカバリ処理部１２５でリカバリ処理を開始する。なお、所定のタイミングは、所定時間の経過や、テーブル１１２が更新されたときなどである。

【００８０】

リカバリ処理部１２５は、まず、外部ストレージ装置４０２へバックアップされたデータベース（テーブル１１２）をメインメモリ４１６上にロードする（Ｓ７０１）。次に、リカバリ処理部１２５は、ログファイル３０１に記録されたログ領域終端アドレス１２３を、ログ管理部１１５の該当領域にセットする（Ｓ７０２）。また、リカバリ処理部１２５は、ログファイルアドレス１２２にログファイル３０１の先頭アドレスなど、リカバリを開始するアドレスを設定する。なお、リカバリを開始するアドレスは、データベースサーバ４０１の管理者などが設定しても良く、バックアップデータの生成時刻に対応するログ３０のアドレスなどを指定することができる。

【００８１】

次に、リカバリ処理部１２５は、ログファイル３０１のログファイルアドレス１２２がログ領域終端アドレス１２３以下であるか否かを判定する（Ｓ７０３）。

【００８２】

ステップＳ７０３の判定結果がＮＯの場合、リカバリ処理部１２５は、適用すべきログ３０はすべてデータベースに適用されているため、リカバリ処理を終了する。

【００８３】

一方、ステップＳ７０３の判定結果がＹＥＳの場合、リカバリ処理部１２５は、ログファイルアドレス１２２が指し示すアドレスから、ログヘッダ３１を読み出し（Ｓ７０５）、ログヘッダ識別子３３が正常であるか否かを判定する（Ｓ７０６）。

【００８４】

ログヘッダ識別子３３の正常化否かの判定は、上述のようにログヘッダ３１の開始アドレスと、ログヘッダ識別子３３の値が一致した場合などで、リカバリ処理部１２５は、ログヘッダ３１が正常に記録されていると判定することができる。なお、本実施例１では、ログヘッダ３１が正常であれば、ログ本体３２も正常に記録されていると判定する。

【００８５】

一方、ステップＳ７０６の判定結果が正常でない（ＮＯ）場合、リカバリ処理部１２５は、ログファイルアドレス１２２に予め設定した値を加算して、ステップＳ７０３に戻って上記処理を繰り返す。

【００８６】

ここで、ログファイルアドレス１２２に加算する予め設定した値とは、例えば、ログ３０のサイズを指している。例えばログサイズが常に３２バイトの倍数の固定長である場合には、ログファイルアドレス１２２に３２バイト加算する。これは、正常なログを失った場合に、次の有効なログ３０を探索するため、次のログ３０の開始アドレスの可能性があるアドレスを順に走査するためである。

【００８７】

一方、ログ３０のサイズが可変長の場合、ログファイルアドレス１２２に加算する所定

数は、ログヘッダ識別子３３のサイズとすることができる。ログヘッダ識別子３３は固定長で指定され、例えば、４バイトで記述される。この場合、図８で示すように、ログヘッダ識別子３３のサイズをログファイルアドレス１２２に順次加算することで、リカバリ処理部１２５は、ブランク領域が存在する場合でも、次に出現するログヘッダ識別子３３（ログヘッダ３１）を探索することができる。なお、図８は、ログ本体３２が可変長の場合に、ブランク領域の次のログヘッダ３１を探索する例を示す図である。

【００８８】

ステップＳ７０６の判定結果が正常（ＹＥＳ）である場合、リカバリ処理部１２５は、ログヘッダ３１に格納されたログサイズ３４を読み出し（Ｓ７０７）、ログファイルアドレス１２２に、当該ログサイズ３４を加算する（Ｓ７０８）。さらに、リカバリ処理部１２５は、当該加算されたログファイルアドレス１２２までのログ本体３２を読み込んで、ログ本体３２の内容をバックアップデータに適用してデータベース（テーブル１１２）を復旧する（Ｓ７０９）。その後、リカバリ処理部１２５は、ログファイルアドレス１２２を次のログ３０に設定してからステップＳ７０３に戻ってログ領域終端アドレス１２３まで上記処理を繰り返す。

【００８９】

以上の処理により、ログファイル３０１中のログ３０に歯抜け領域（ブランク領域）が存在する場合でも、ログヘッダ３１単位で探索することで、次のログ３０を取得することができる。

【００９０】

本実施例１によれば、データベースサーバ４０１で稼働するＤＢＭＳ１０２のスレッド１１０Ａ～１１０Ｃが、それぞれのＩ／Ｏキュー２０１Ａ～２０１Ｃへログを並列的に格納する。その後、Ｉ／Ｏアダプタ４１３からネットワーク４０３を介して外部ストレージ装置４０２のログファイル３０１へログを書き込む。スレッド１１０Ａ～１１０Ｃは、ログファイル３０１で自身のログ３０を書き込む領域のアドレスよりも小さいアドレスの領域（直前のログ３０）へのログ書き出しが完了しなくても、自身のログ領域へログ３０を書き出すことができる。

【００９１】

これにより、前記従来例に示したログマネージャスレッドによるトランザクション間のログ書き出し処理の待ち合わせ処理時間が削減される。したがって、トランザクションあたりのプロセッサ処理時間が削減され、計算機システム全体の処理性能が向上する。

【００９２】

そして、ＤＢＭＳ１０２のスレッド１１０は、ログ３０を格納する外部ストレージ装置４０２のログファイル３０１に、障害回復時（リカバリ時）にログファイル３０１を走査する領域の範囲（ログ領域終端アドレス１２３）を記録する。

【００９３】

これにより、リカバリ処理の際には、ログファイル３０１の走査時間を短縮し、障害回復時間を短縮することができる。各スレッド１１０は、ログ領域終端アドレス１２３とログファイルアドレス１２２の差が所定値以下になると、新たなログ領域を確保するためにログ領域終端アドレス１２３を所定量だけ増大させる。すなわち、ログ３０の追加に応じて、徐々にログ領域を拡張することで、リカバリ処理の際の走査範囲を抑制することができるのである。

【００９４】

したがって、ログファイル３０１の領域は、ログ３０の追加に応じて徐々に増大するので、リカバリ処理の際に探索するブランク領域の範囲を低減でき、障害回復までの時間を短縮することができる。

【００９５】

また、ログ３０が可変長の場合には、各ログ３０に固定長のログヘッダ識別子３３を設定する。データベースサーバ４０１の障害によって、１以上のスレッド１１０でログ３０の書き込みが完了しないブランク領域が発生しても、リカバリ処理の際には、ログヘッダ

識別子３３サイズ単位で、ログファイルアドレス１２２を加算することで、リカバリ処理部１２５は次の有効なログヘッダ識別子３３を検出することが可能となる。

【００９６】

なお、上記実施例１ではデータベースサーバ４０１のＩ／Ｏアダプタ４１３にスレッド１１０毎のＩ／Ｏキュー２０１を設けた例を示したが、これに限定されるものではなく、外部ストレージ装置４０２にＩ／Ｏキュー２０１Ａ～２０１Ｃを設けても良い。この場合、例えば、Ｉ／Ｏアダプタ４４１にＩ／Ｏキュー２０１Ａ～２０１Ｃを設けることができる。

【００９７】

また、上記実施例１では、各スレッド１１０は、所定の条件を満たすとログ領域終端アドレス１２３を変更して、ログ領域を拡張する例を示したが、これに限定されるものではない。例えば、スレッド１１０がログ３０を書き込む度に、書き込んだログ３０のサイズをログ領域終端アドレス１２３に加算し、ログファイル３０１を更新しても良い。

【００９８】

すなわち、スレッド１１０が、ログ領域へログを書き込む度にログ領域の終端を決定してログ出力を行っても良いし、ログ領域を拡張する際にログ領域の終端を決定してログ出力を行っても良い。

【実施例２】

【００９９】

図９は、第２の実施例を示し、固定長のログ３０の構造の一例を示す図である。本実施例２では、前記実施例１のログ３０のサイズを固定長としたもので、その他の構成は前記実施例１と同様である。

【０１００】

固定長のログ３０は、前記実施例１と同様のログヘッダ識別子３３と、ログ本体３２から構成される。

【０１０１】

リカバリ処理の際、リカバリ処理部１２５は、ひとつのログ３０の読み込みとログ適用が完了すると、ログファイルアドレス１２２に所定値を加算して、次のログ３０の開始アドレスを算出する。リカバリ処理部１２５は、次のログ領域のログヘッダ識別子３３を読み込んで、有効か否かを判定する。

【０１０２】

ログヘッダ識別子３３が有効でない場合、リカバリ処理部１２５は、ログファイルアドレス１２２に所定値を加算して、次のログ３０の開始アドレスを算出して上記処理をログ領域終端アドレス１２３まで返す。

【０１０３】

ログ３０のサイズを固定長とすることで、リカバリ処理の際に有効なログ３０を探索する処理は可変長の場合に比して高速になる。可変長の場合は、図８で示したように、ブランク領域内でログヘッダ識別子３３を探索する必要がある。これに対して、固定長の場合では、図９で示すように図中ログ４の次のログ領域がブランク領域なので、リカバリ処理部１２５は、現在のログファイルアドレス１２２に所定値を加算することで、次のログ６の開始アドレスからログヘッダ識別子３３を読み込むことができる。

【０１０４】

以上のように、本実施例２では、ログ３０のサイズを固定長とすることで、リカバリ処理の際にはブランク領域であってもログ領域単位で処理を行うことが可能となって、障害回復までの時間を短縮することが可能となるのである。

【０１０５】

＜まとめ＞

なお、本発明は上記した実施例に限定されるものではなく、様々な変形例が含まれる。例えば、上記した実施例は本発明を分かりやすく説明するために詳細に記載したものであり、必ずしも説明した全ての構成を備えるものに限定されるものではない。また、ある実

施例の構成の一部を他の実施例の構成に置き換えることが可能であり、また、ある実施例の構成に他の実施例の構成を加えることも可能である。また、各実施例の構成の一部について、他の構成の追加、削除、又は置換のいずれもが、単独で、又は組み合わせても適用可能である。

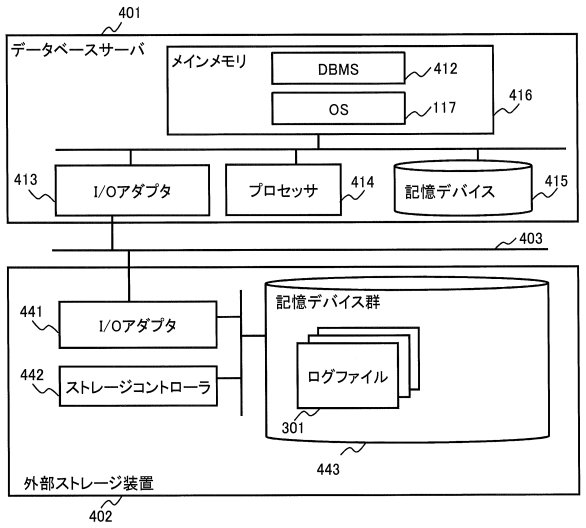
【０１０６】

また、上記の各構成、機能、処理部、及び処理手段等は、それらの一部又は全部を、例えば集積回路で設計する等によりハードウェアで実現してもよい。また、上記の各構成、及び機能等は、プロセッサがそれぞれの機能を実現するプログラムを解釈し、実行することによりソフトウェアで実現してもよい。各機能を実現するプログラム、テーブル、ファイル等の情報は、メモリや、ハードディスク、ＳＳＤ（Ｓｏｌｉｄ Ｓｔａｔｅ Ｄｒｉｖｅ）等の記録装置、または、ＩＣカード、ＳＤカード、ＤＶＤ等の記録媒体に置くことができる。

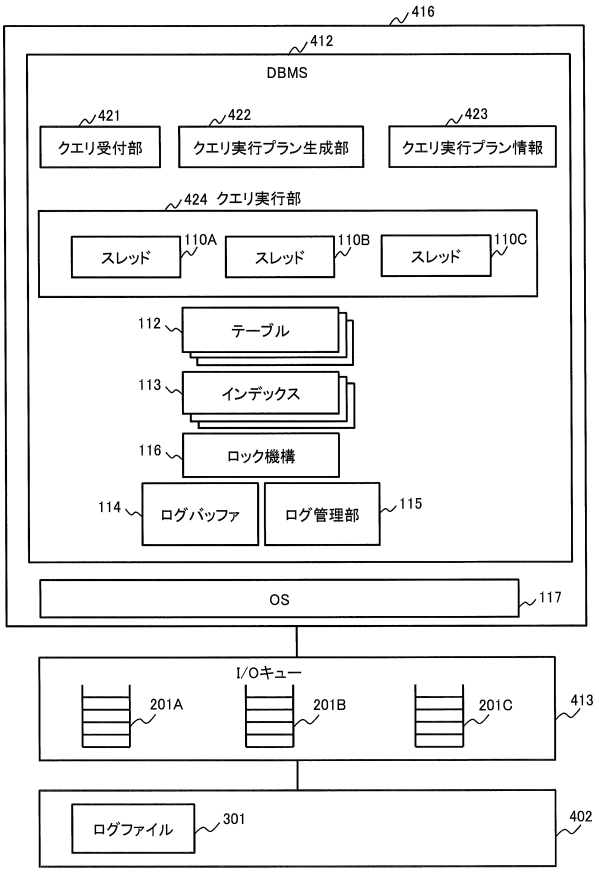
【０１０７】

また、制御線や情報線は説明上必要と考えられるものを示しており、製品上必ずしも全ての制御線や情報線を示しているとは限らない。実際には殆ど全ての構成が相互に接続されていると考えてもよい。

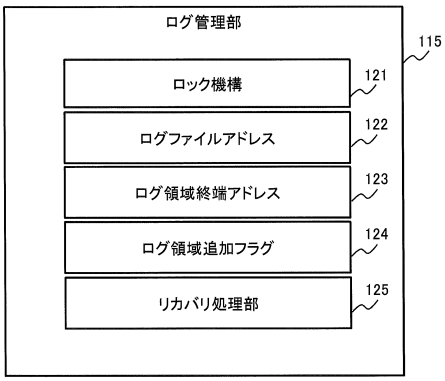
【図１】



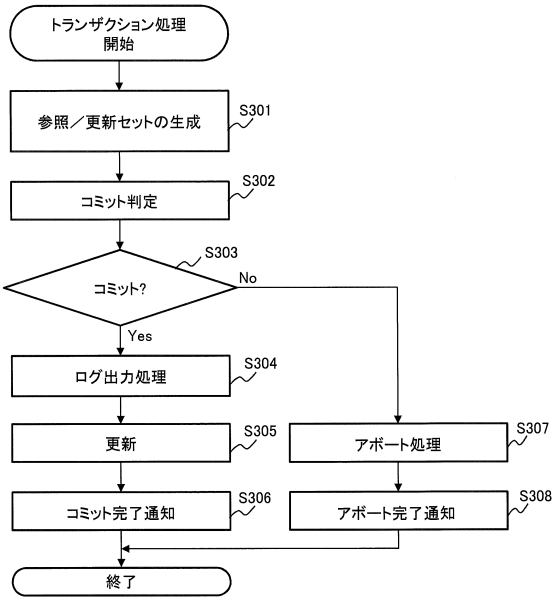
【図２】



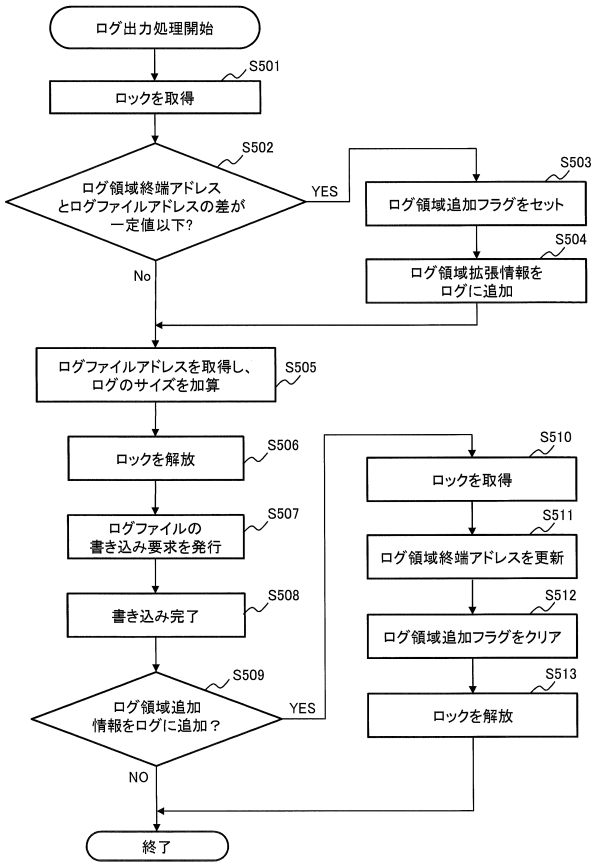
【図３】


ログ管理部
115
ロック機構
121
ログファイルアドレス
122
ログ領域終端アドレス
123
ログ領域追加フラグ
124
リカバリ処理部
125

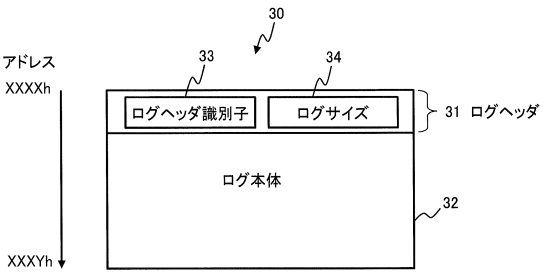
【図４】


トランザクション処理
開始
参照／更新セットの生成
S301
コミット判定
S302
コミット?
S303
No
Yes
ログ出力処理
S304
更新
S305
アボート処理
S307
コミット完了通知
S306
アボート完了通知
S308
終了

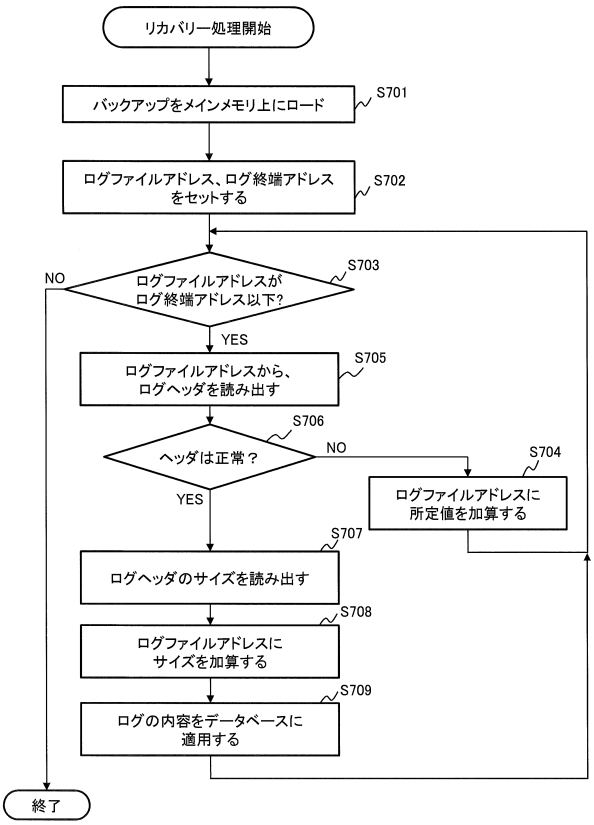
【図５】


ログ出力処理開始
ロックを取得
S501
ログ領域終端アドレス
とログファイルアドレスの差が
一定値以下?
S502
YES
No
ログ領域追加フラグをセット
S503
ログ領域拡張情報を
ログに追加
S504
ログファイルアドレスを取得し、
ログのサイズを加算
S505
ロックを解放
S506
ログファイルの
書き込み要求を発行
S507
書き込み完了
S508
ログ領域追加
情報をログに追加？
S509
YES
NO
ロックを取得
S510
ログ領域終端アドレスを更新
S511
ログ領域追加フラグをクリア
S512
ロックを解放
S513
終了

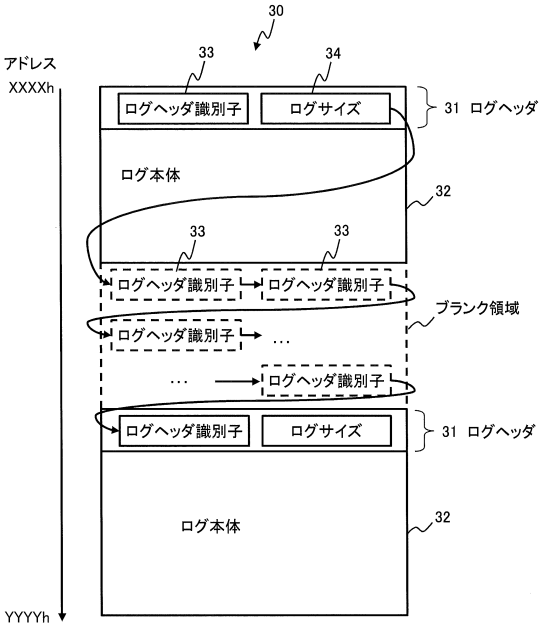
【図６】


30
アドレス
XXXXh
XXXYh
33
34
ログヘッダ識別子
ログサイズ
31 ログヘッダ
ログ本体
32

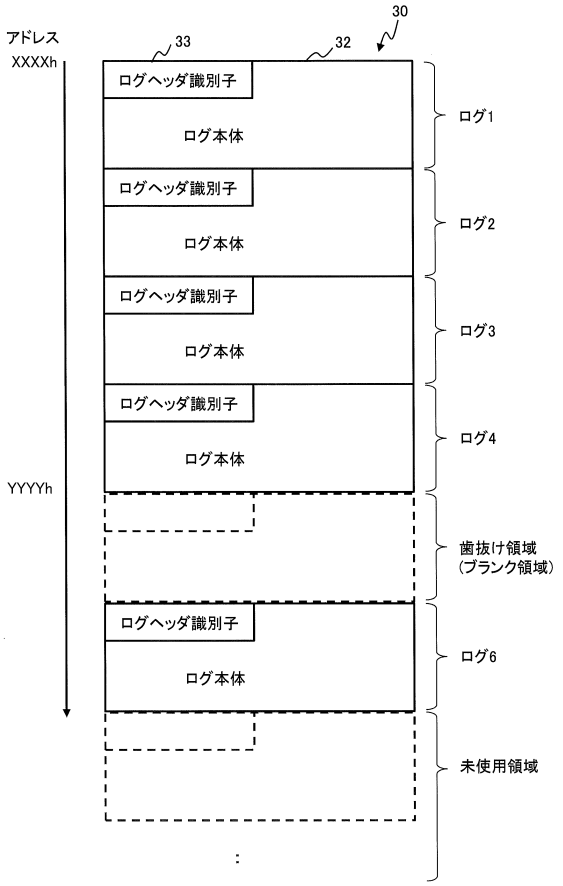
【図７】


リカバリー処理開始
バックアップをメインメモリ上にロード
S701
ログファイルアドレス、ログ終端アドレス
をセットする
S702
ログファイルアドレスが
ログ終端アドレス以下?
S703
NO
YES
ログファイルアドレスから、
ログヘッダを読み出す
S705
ヘッダは正常？
S706
NO
YES
ログファイルアドレスに
所定値を加算する
S704
ログヘッダのサイズを読み出す
S707
ログファイルアドレスに
サイズを加算する
S708
ログの内容をデータベースに
適用する
S709
終了

【図８】


30
アドレス
XXXXh
33
34
ログヘッダ識別子
ログサイズ
31 ログヘッダ
ログ本体
32
33
33
ログヘッダ識別子
ログヘッダ識別子
ブランク領域
ログヘッダ識別子
…
…
ログヘッダ識別子
ログヘッダ識別子
ログサイズ
31 ログヘッダ
ログ本体
32
YYYYh

【図９】


30
アドレス
XXXXh
33
32
ログヘッダ識別子
ログ本体
ログ1
ログヘッダ識別子
ログ本体
ログ2
ログヘッダ識別子
ログ本体
ログ3
ログヘッダ識別子
ログ本体
ログ4
YYYYh
歯抜け領域
(ブランク領域)
ログヘッダ識別子
ログ本体
ログ6
未使用領域
:

フロントページの続き

審査官　原　秀人

(56)参考文献　特開平０７－２４８９４９（ＪＰ，Ａ）
特開２００６－２６８５３１（ＪＰ，Ａ）
特開平１０－２４７１５７（ＪＰ，Ａ）
ジム　グレイ，トランザクション処理　［下］，日本，日経ＢＰ社，２００１年１０月２９日，第1版，pp. 593(3)-616(26)

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１２／００
Ｇ０６Ｆ　１１／３４