

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4139613号  
(P4139613)

(45) 発行日 平成20年8月27日(2008.8.27)

(24) 登録日 平成20年6月13日(2008.6.13)

(51) Int.Cl.

F I

**G 0 6 F 12/00 (2006.01)**

G 0 6 F 12/00 5 3 5 B

G 0 6 F 12/00 5 0 5

G 0 6 F 12/00 5 2 0 A

請求項の数 3 (全 14 頁)

(21) 出願番号 特願2002-73540 (P2002-73540)  
 (22) 出願日 平成14年3月18日(2002.3.18)  
 (65) 公開番号 特開2003-271436 (P2003-271436A)  
 (43) 公開日 平成15年9月26日(2003.9.26)  
 審査請求日 平成16年7月20日(2004.7.20)

(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 110000350  
 ポレール特許業務法人  
 (74) 代理人 100068504  
 弁理士 小川 勝男  
 (74) 代理人 100086656  
 弁理士 田中 恭助  
 (74) 代理人 100094352  
 弁理士 佐々木 孝  
 (72) 発明者 山下 信之  
 神奈川県川崎市幸区鹿島田890番地 株  
 式会社日立製作所 ビジネスソリューション  
 事業部内

最終頁に続く

(54) 【発明の名称】 データ処理方法

(57) 【特許請求の範囲】

【請求項1】

コンピュータシステムのメモリ内にプログラムとして保持され、プロセッサにより実行されるデータ管理システムを演算処理するデータ処理方法において、

前記データ管理システムは、

オブジェクト管理部と、

ロック管理部とを有し、

領域の確保および開放が一つのものとしておこなわれるデータ単位であるオブジェクトと、複数のオブジェクトより構成されるオブジェクトグループと、あるオブジェクトと次のオブジェクトを指し示し、あるオブジェクトから次のオブジェクトをアクセスすることができるようにするポインタとよりなるデータ構造を有し、

オブジェクトの参照、更新、挿入および削除の処理をおこなうときに、前記オブジェクトグループに対して、ロックを設定し、その後、前記オブジェクトグループ内の特定のオブジェクトから、処理対象のオブジェクトまでの前記ポインタによって関連付けられたオブジェクトに対して、前記オブジェクトのロックを設定して、オブジェクトの参照、更新、挿入および削除の処理を複数並行して、前記データ管理システムにより動作させるデータ処理方法であって、

前記ロック管理部のプログラムが前記プロセッサにより演算処理されることによって実行されて、削除処理をおこなうときに、前記オブジェクトグループに第1のロックを設定する第1のステップと、

前記ロック管理部のプログラムが前記プロセッサにより演算処理されることによって実行されて、前記オブジェクトグループ内の特定のオブジェクトから、削除処理をおこなう削除対象オブジェクトまでの前記ポインタによって関連付けられたオブジェクトに対して、第2のロックを設定する第2のステップと、

前記オブジェクト管理部のプログラムが前記プロセッサにより演算処理されることによって実行されて、削除処理をおこなう削除対象オブジェクトの属する前記オブジェクトグループのオブジェクトにロックを取得している前記削除処理以外の全ての処理の情報を記憶する第3のステップと、

前記オブジェクト管理部のプログラムが前記プロセッサにより演算処理されることによって実行されて、削除処理をおこなう削除対象オブジェクトをポインタするオブジェクトから、前記削除対象オブジェクトをポイントしないようにする第4のステップと、

前記オブジェクト管理部のプログラムが前記プロセッサにより演算処理されることによって実行されて、前記オブジェクトグループに対してロックを取得している前記削除処理以外の全ての処理が完了したことを判断する第5のステップと、

前記オブジェクト管理部のプログラムが前記プロセッサにより演算処理されることによって実行されて、前記全ての処理が完了したと判断したときに、前記削除対象オブジェクトの領域を解放する第6のステップとを具備することを特徴とするデータ処理方法。

#### 【請求項2】

前記第5のステップにおいて、前記オブジェクト管理部のプログラムが、前記オブジェクトグループにロックを取得している前記削除処理以外の処理から処理完了通知を受けることを特徴とする請求項1記載のデータ処理方法。

#### 【請求項3】

前記オブジェクト管理部のプログラムが前記プロセッサにより演算処理されることによって実行されて、あるオブジェクトに対して、参照、更新、挿入および削除の処理を実行する場合に、前記オブジェクトグループの基点となる特定のオブジェクトから、前記ポインタを順次たどって、処理を実行するオブジェクトを検索することを特徴とする請求項1記載のデータ処理方法。

#### 【発明の詳細な説明】

#### 【0001】

#### 【発明の属する技術分野】

本発明は、ポインタにより複数のオブジェクトが関連づけられるデータ構造を持ち、オブジェクトの参照、更新、挿入および削除を行なう処理が複数並行して動作するシステムのデータ処理に関する。

#### 【0002】

#### 【従来の技術】

#### （従来技術1）

本発明に関するインデックスにより高速にデータにアクセスする機能を持ったデータ管理システムは基本的なデータへのアクセス方法として次の4つの機能を提供するものとする；

（1）参照機能：指定したキー値に関連するデータの参照

（2）挿入機能：指定したキー値およびそれに関連する指定したデータの挿入

（3）削除機能：指定したキー値およびそれに関連するデータの削除

（4）更新機能：指定したキー値に関連するデータを指定したデータへの更新

これらの機能は、複数並列して処理要求されるものとする。以下、それぞれの機能を実現する処理方法を参照処理、挿入処理、削除処理そして更新処理と呼ぶものとする。

#### 【0003】

これらの機能を実現する従来技術を図1および図2を用いて説明する。

図1、101にデータ構造の一例を示す。3つのオブジェクトA、BおよびCから構成されるオブジェクトグループTを示す。オブジェクトはデータの単位であり、領域の確保および解放はオブジェクト単位で行なわれるものとする。またひとつのデータはひとつのオ

10

20

30

40

50

プロジェクトに格納されるものとする。

【0004】

オブジェクトBにはキー値10とそれに関連するデータが格納されているものとする。同様にオブジェクトCにはキー値20とそれに関連するデータが格納されているものとする。オブジェクトAには、オブジェクトBが保持するキー(すなわちキー値10)とオブジェクトBの格納位置情報(以下、ポインタと呼ぶ)およびオブジェクトCが保持するキー(すなわちキー値20)とオブジェクトCのポインタが格納されている。

【0005】

ここで、前記の4機能の実現方法について説明する。以上の機能を複数の処理が並列実行することを可能とするため、ロックの設定・解除が必要となる。ロックについては例えば、「Gray, J., Reuter, A., TRANSACTION PROCESSING: CONCEPTS AND TECHNIQUES, Morgan Kaufmann Publishers, Inc., 1993, p449-484」に実現方法が述べられている。ここでは前記文献にある、S、X、IS、IX各モードのロックを利用する。SモードのロックはSおよびISモードのロックと並列実行可能であり、Xモードはすべてのモードのロックと並列実行はできず、ISモードのロックはS、ISおよびIXモードのロックと並列実行可能であり、IXモードのロックはISおよびIXモードのロックと並列実行可能である。ロック設定の方針は、参照処理の場合はオブジェクトグループTにISモードのロックを、挿入、削除および更新処理の場合はオブジェクトグループTにIXモードのロックを設定し、その後アクセスするオブジェクトに対し、参照する場合はSモード、更新する(可能性がある)場合はXモードでロックを設定するといったように階層的にロックを設定するものである。即ち、オブジェクトグループに対してロックを設定した後、特定のオブジェクトにロックを設定する。

【0006】

オブジェクトグループに対しロックを設定する場合のリソース名称は、各処理が共通して知っている名称を指定するものとする。また、オブジェクトに対しロックを設定する場合、オブジェクトの格納位置に対応するリソース名称(リソース名称をオブジェクトが格納されている位置情報、例えば、ポインタの値とする)を指定するものとする。つまり、格納位置がわかればオブジェクトにロックを設定可能である。また、オブジェクトグループ内の基点となるオブジェクト(図1ではオブジェクトA)の格納位置はすべての処理が知っており、かつ格納位置が移動したりしないものとする。

【0007】

各処理のタイムチャートを図2に示す。タイムチャートにおける時間は左から右方向に流れるものとする。両端が黒丸で示される線分は、線分の左側に示されるオブジェクトにロックを設定している期間を表し、線分の上にロックのモードを示し、線分の下に処理を示す。両端が白丸で示される線分は、線分の左側に示されるオブジェクトに対し(ロックを設定せずに)アクセスを行なっていることを表し、線分の下に処理を示す。図2(a)は参照処理のフローチャートを示す。ここでは、キー値が10のオブジェクトを参照する例を示している。図2(b)は挿入処理のフローチャートを示す。ここでは、キー値が30のオブジェクトをグループTに挿入する例を示している。図2(c)は削除処理のフローチャートを示す。ここでは、キー値が20のオブジェクトを削除する例を示している。図2(d)は更新処理のフローチャートを示す。ここでは、キー値が10のオブジェクトを更新する例を示している。

【0008】

並列処理での状況を示すために、仮にキー値30の挿入処理と、キー値20の削除処理が、ほぼ同時に動作している状況での場合について説明する。挿入処理202は、まずオブジェクトグループTに対し、IXモードでロックを設定する。そして新規のデータを格納するためのオブジェクト領域を確保(オブジェクトDとする)して、その領域にキー値30および関連するデータを設定する。一方、削除処理203も、同様にオブジェクトグループTに対し、IXモードでロックを設定するが、このロックは挿入処理におけるIXと競合しても並列実行可能であり、いずれかが待たされることはない。次は、挿入処理20

10

20

30

40

50

2、削除処理203とともにオブジェクトAに対してXモードのロックを設定する処理であり、先行してロック設定した処理がロックを解除するまで他方は待たされることになる。

【0009】

まず、削除処理203が挿入処理202に先立ってロックの設定に成功した場合について説明する。挿入処理202は削除処理203がオブジェクトAのロックを解除するまで待たされることになる。削除処理203はオブジェクトAをアクセスしてキー値20に対応するオブジェクトBへのポインタを取得して、オブジェクトCに対しXモードでロックを設定し、オブジェクトA内のキー値20およびオブジェクトCへのポインタを削除（以下、ポインタの切り離しと呼ぶ）して、オブジェクトAに対するロックを解除して（この時点で挿入処理202の待ちは解除される）、オブジェクトCを格納していたオブジェクト領域を解放し、オブジェクトCのロックを解除し、最後にオブジェクトグループTのロックを解除する。この状態を図1の102に示す。

10

【0010】

削除処理203によるオブジェクトAのロックの解除を受けて、挿入処理202はオブジェクトAへのロックの設定に成功する。挿入処理202では、オブジェクトAにアクセスして、オブジェクトAにキー値30とオブジェクトDの格納位置を設定し、オブジェクトAに設定したロックを解除し、オブジェクトグループTに設定したロックを解除する。この状態を図1、104に示す。

【0011】

一方、挿入処理202が削除処理203に先立ってロックの設定に成功した場合は、削除処理203は挿入処理202がオブジェクトAのロックを解除するまで待たされることになる。挿入処理202は前記同様に処理を行い、図1、103の状態となる。その後、削除処理203が実行され、図1、104に示す状態となる。

20

【0012】

参照処理（キー値10）については図2、201に、更新処理（キー値10）については図2、204に示す。参照処理201や更新処理204でオブジェクトAに対するロックを解除する前にオブジェクトBのロックを設定しているのは、並列してオブジェクトBの削除処理が動作していた場合に、誤動作を防止するためである。つまり、仮に、オブジェクトAに対するロックを解除した後にオブジェクトBのロックを設定するようになっていると、その間に削除処理によってオブジェクトBの領域解放が可能であり、さらにこの解放領域が別用途に再割当てされて設定されたデータをオブジェクトBのデータと見てしまい誤動作する可能性がある。

30

【0013】

また、より平易な方法として、参照処理の場合はオブジェクトグループTにSモードのロックを、挿入、削除および更新処理の場合はオブジェクトグループTにXモードのロックを設定するという方法もある。この方法では参照処理同士以外は、並列処理実行はできないため、上記の例より更にディスクやプロセッサといった資源を有効に活用できず、スループットおよびレスポンスタイムが低くなる。

【0014】

このように、従来技術1では、オブジェクトAに対してロックをかけた状態でオブジェクトBにもロックをかけるなど同時に2つのオブジェクトにかける。このことによりオブジェクトの削除処理で削除したオブジェクトの領域を解放することが出来るという利点があるが、並列実行性能がやや低いという特性がある。

40

【0015】

（従来技術2）

DB管理システムにおける、Bツリーインデクスは、従来技術1を拡張したデータ構造になっている。図7、701にその一例として、7つのページ（従来技術1のオブジェクトに相当）からなる3段のBツリーインデクスを示す。図7、701の下方に位置するページP4、P5、P6およびP7はリーフページと呼ばれ、それぞれ1個以上のキー（ページ内の下方に記した値。例えばページP6ではキー値50およびキー値70）とそれに対

50

応するデータの格納位置情報の組、右側のページのポインタ（一番右側のページP7は除く）、ページ内MAXキー値（リーフページ内の右上に記した値。ページ内に格納されているデータが持つ最大のキー値。例えばページP4ではキー値20がページ内Maxキー値である）が格納されている。各リーフページは格納を担当するキーの範囲が設定されており、該当ページおよびその左側に位置するページのページ内Maxキー値によって示される。例えばページP5が格納を担当するキーの範囲は、ページP5の左側のページP4のページ内Maxキー値20より大きく、ページP5のページ内Maxキー値40以下のキーである。同様にページP6、P7が担当するキーの範囲はそれぞれ40より大きく70以下のキー、および70より大きいキーである。左側に位置するページがないページP4は、ページ内Maxキー値以下（20以下）のキーを担当する。

10

**【0016】**

リーフページの上方には上位ページ（図7、701のページP1、ページP2およびページP3、ノードとも呼ばれる）と呼ばれるページが配置される。上位ページには、1個以上の直下のページへのポインタおよびキー値の組、（存在すれば）右側のページへのポインタが格納される。ここでのキー値は組になっているポインタで指されるページのページ内Maxキー値と一致する。特にページP1はルートページと呼ばれる。あるキー値が格納されているリーフページをサーチするためには、ルートページからアクセスして目的のキー値以上の最小のキー値と組になるポインタを辿っていけばよい。領域の確保はページ単位で行なわれる。

**【0017】**

ここで、キー値20～40の範囲のキー値を指定した削除処理について図8、801を用いて説明する。まずBツリーインデクス全体Tに対し、IXモードのロックを設定して、ルートページP1に対し、Sモードでロックを設定して、指定範囲の下限であるキー値20以上であってページP1内の最小のキー値であるキー値40と組になっているページP2へのポインタを取得して、ページP1のロックを解除する。ページP2に対しSモードでロックを設定し、同様にキー値20以上であってページP2内の最小のキー値であるキー値20と組になっているページP4へのポインタを取得して、ページP2のロックを解除する。そしてページP4に対しXモードでロックを設定し、ページP4内の削除対象であるキー値20～40の範囲にあるキー値すべて削除する。

20

**【0018】**

このときページP4には、キー値20しか存在しないため、ページ内にキー値が残らなくなるが、ページP4の領域解放は行なわない。その大きな理由は、領域解放するには、解放対象ページへのポインタを適当なロックを設定しながら、設定し直す必要があり、並列実行性の劣化につながるからである。

30

**【0019】**

またページP4のページ内Maxキー値が20であることから、削除対象であるキー値20～40の範囲のうち、20より大きいキー値は右側のページに格納されていることがわかるので、右ページへのポインタ（ページP5へのポインタ）を取得して、ページP4のロックを解除する。ページP5に対するXモードのロックを設定して、ページP5内の削除対象であるキー値20～40の範囲にあるキー値すべて削除する。このときページP5には、キー値40しか存在しないため、ページ内にキー値が残らなくなるが、ページP5の領域解放は行なわない。ページP5のページ内Maxキー値が40であることよりキー値20～40の範囲のキーすべてを削除できたことになる。ページP5のロックを解除し、最後にBツリーインデクス全体Tのロックを解除し削除処理を完了する。削除処理が完了した状態を図7、702に示す。

40

**【0020】**

また、図7、702の状態から、挿入処理（キー値60）を行なう手順について図8、802を用いて説明する。Bツリーインデクス全体Tに対し、IXモードのロックを設定して、前記記述の手順と同様にページP1、ページP3とページをアクセスしていき、ページP6に対してXモードのロックを設定してページP6にアクセスする。ページP6には

50

既にキー値 50 および 70 のデータが格納されており、キー値 60 のデータを格納するのに十分なスペースがないものとする。この場合スプリットと呼ばれる、ページ分割処理を行なう。まず、新規にページを確保（ページ P8 とする）する。ここでは分割キー値が 50 であったものとする。つまり、キー値 50 のデータをページ P6 に残し、キー値 70 のデータおよび挿入するキー値 60 のデータを新規に確保するページ領域に移動するものとする。ページ P8 に必要な設定を行い、ページ P6 のページ内 Max キー値を 50 に設定し直し、右ページのポインタをページ P8 に設定し直すなどの処理を行なった後、ページ P6 のロックを解除して、ページ P3 に対し X モードでロックを設定して、ポインタを設定し直し、ページ P3 のロックを解除し、最後に B ツリーインデクス全体 T のロックを解除し挿入処理を完了する。挿入処理が完了した状態を図 7、703 に示す。

10

**【0021】**

前述したように、B ツリーインデクスにおいて、並列実行性を高く保つために、データの削除によりページ内にデータが一件もない状態になっても、ページの領域解放を行なわない。そのため、一般にデータの挿入・削除を繰り返すうちに格納効率が低下し、それに伴いアクセス性能が劣化していく。その性能劣化を抑えるために、適当な時期にインデクスのアクセスを禁止して、インデクスに格納されたデータを一時領域に出力して、データを詰め込み直す処理（再編成処理）が一般に行なわれている。

**【0022】**

このように、従来技術 2 では同時に 2 つのページにロックをかけることがない。従って、並行実行可能性は高い。一方、従来技術 1 で述べたような誤ったページの参照が生じる可能性があるので、データを削除しても引き続いてその領域を解放することが出来ない。

20

**【0023】****【発明が解決しようとする課題】**

以上のような従来技術 1、従来技術 2 の説明から分かるように、データの削除処理でのその領域の解放処理が可能であることと、並列実行性が高いことはトレードオフの関係にあるものと考えられていた。

**【0024】**

本発明の目的は、ポインタにより複数のオブジェクトが関連づけられるデータ構造を持ち、オブジェクトの参照、更新、挿入および削除を行なう処理が複数並行して動作するシステムにおいて、参照、更新および挿入の並列実行性が高く、かつ削除処理を参照、更新および挿入処理と並列実行可能とすることである。

30

**【0025】**

さらに、データの挿入および削除の繰り返しにより格納効率・アクセス効率が低下（削除されたがその領域が解放されないオブジェクトが増えることや、目的のオブジェクトにアクセスするために不必要なオブジェクトを経由することになること）した B ツリーインデクスに対し、データ参照、更新および挿入を抑止することなく空きとなっているオブジェクトの領域の回収し、格納効率、アクセス効率を向上させることである。

**【0026】****【課題を解決するための手段】**

参照、更新および挿入処理において、ポインタを辿ってのオブジェクト間の移動に際し、移動前のオブジェクトに対するロックを解除した後に、移動後のオブジェクトに対するロックを設定することにより、参照、更新および挿入の並列実行性を高くする。この際、処理の始めにオブジェクトグループに対し IS モード（参照処理の場合）または IX モード（更新および挿入処理の場合）を設定し、処理完了時にロックを解除する。一方、削除処理において、削除対象へのポインタを切り離し、あるいは付け替えた後、オブジェクトグループに対してロックを取得している全処理が完了した後に、削除対象オブジェクトの領域を解放する。

40

**【0027】****【発明の実施の形態】**

（実施の形態 1）

50

本発明を実現するシステム構成例を図9に示す。ここに示すシステム構成例は実施の形態1～3において共通である。本システムは、プロセッサ2とメモリ3とを有するコンピュータシステム1と、それに接続された外部記憶装置4と、コンピュータシステム1にネットワーク6を介して接続された複数の端末7を備えている。

【0028】

メモリ3内にはプロセッサ2で解釈可能なプログラムの形で表現されたデータ管理システム10が配置され、プロセッサ2によって命令が読み込まれ演算処理がなされる。データ管理システム10は、システム制御部11、オブジェクト管理部12、ロック管理部13から構成される。システム制御部11は、端末7からの参照、挿入、更新および削除要求を受け取り、要求内容を解析してオブジェクト管理部12に制御を渡し、処理結果をネットワーク6を通じて端末7へ応答する。オブジェクト管理部12はシステム制御部11から渡された指示に従って、外部記憶装置4内に格納されたオブジェクト5の管理を行なう。ロック管理部12は、指定されたリソースに対するロック機能をオブジェクト管理部12に提供する。

10

【0029】

従来技術1における説明に使用した図1において、参照、更新および挿入の並列実行性がより高く、かつ削除処理を参照、更新および挿入処理と並列実行可能な方法の例を図3および図4により開示する。

【0030】

図3、301は本実施形態における参照処理を示すタイムチャートである。図2、201と比較して、オブジェクトAに対するロックの解除時期とオブジェクトBに対するSモードのロックの設定時期が入れ替わっている。即ち、従来はオブジェクトAについてロックを設定して、オブジェクトBへのポインタの値を記憶し、その後、オブジェクトBについてロックを設定し、それからオブジェクトAについてのロックを解放していた。これに対し本実施形態ではオブジェクトAについてロックを設定し、オブジェクトBへのポインタの値を記憶したならば、オブジェクトAについてのロックを解放する。それからオブジェクトBについてロックを設定する。図3、302は本実施形態における挿入処理を示すタイムチャートである。これは図2、201と同じものである。

20

【0031】

図3、304は本発明における更新処理を示すタイムチャートである。図2、204と比較して、オブジェクトAに対するロックの解除時期とオブジェクトBに対するXモードのロックの設定時期が入れ替わっている。

30

【0032】

図3、303は本発明における削除処理を示すタイムチャートである。図2、204とは、大きく変わっており、比較しながら説明する。オブジェクトグループTに対するIXモードのロックの設定およびオブジェクトAに対するXモードのロックの設定は同様であるが、その後オブジェクトグループTに対してロックを取得している全処理を記憶する。その後、オブジェクトAから削除対象のオブジェクトCに対するポインタを切り離すが、この実施形態ではここで前記記憶した全処理の完了を待つ。完了の待ち方として、ロックの解除時点でオブジェクトグループTに対してロックを設定している処理側から削除処理へ完了を通知する方法と、削除処理が一定間隔で他の処理の処理完了をチェックする方法とがある。いずれかの方法を用いて前記記憶した全処理の完了を検知できた後、オブジェクトCの領域を解放する。

40

【0033】

解放した領域に他処理がアクセスする可能性はないので、オブジェクトCに対するロックを設定する必要はない。その理由について、図4に例で説明する。図4では、削除処理401、参照処理402、挿入処理403、更新処理404および更新処理405が並列して処理が行なわれていることを示す。406時点、つまり削除処理401においてオブジェクトAに対しXモードでロックを設定した直後の時点でオブジェクトグループTに対しロックを設定していない処理（ここでは更新処理404および更新処理405）は、オブ

50

ジェクトCにアクセスし得ない。なぜならばオブジェクトCにアクセスする前にオブジェクトCへのポインタを設定してある時点でのオブジェクトAにアクセスする必要があるが削除処理401がXモードで押さえているため、更新処理404および更新処理405でオブジェクトAを見られるときにはオブジェクトCへのポインタは切り離されている。また、参照処理402や更新処理403はオブジェクトCを見てデータを取得することができるが、それらの処理はすべてが完了したことを検知できているからである。

【0034】

前記ISモードおよびIXモードは例であって、その代わりに、それらの組と等価な、あるいはより並列実行性の低いロックモードを用いることも出来る。

【0035】

前記処理方法により、従来処理に比べ、参照、挿入および更新処理の並列実行性を高くして、かつ削除処理によって解放された領域を誤って他処理がアクセスすることも回避している。更に詳しくは、上記実施形態では参照、挿入および更新処理において、一つの処理が同時に2つのオブジェクトをロックすることがないため、従来技術と同等の高い並列実行性を実現出来る。また、削除処理において、ポインタを切り離した後、削除処理の対象となっているオブジェクトグループにロックをかけている他の処理の完了を待って削除されたオブジェクトの領域を解放することにより、安全かつ他の処理の実行を阻害することなく領域の解放も実行出来る。

【0036】

(実施の形態2)

図5および図6を用いて第2の実施の形態を開示する。

図5、501はあるハッシュ値に対応するデータをチェーンにより管理する場合などに典型的に用いられるデータ構造である(ここでは、ハッシュ関数がキー値を10で割った余りと定義されていて、ハッシュ値=2のキーおよびデータを管理する部分であると想定している)。この状態のデータ構造に対し、キー値32の削除処理とキー値12の挿入処理が要求された場合の手順について、図6に示す各処理のタイムチャートと合わせ説明する。

【0037】

図6の削除処理601では、オブジェクトBについて削除要求がなされる。まずオブジェクトグループTに対し、IXモードのロックを設定し、オブジェクトAに対しXモードでロックを設定する。オブジェクトAに格納されているオブジェクトBへのポインタを取得して、(オブジェクトAに対するロックを保持したまま)オブジェクトBに対しXモードのロックを設定し、オブジェクトBをアクセスする。オブジェクトB、即ち、キー値が32のオブジェクトがあったことを確認し、この時点でオブジェクトグループTに対しロックを設定しているすべての処理を記憶する(この時点で挿入処理602も並行して処理していてオブジェクトグループTに対しロックを設定しているものとする)。オブジェクトAに格納されているオブジェクトBへのポインタをオブジェクトBに格納されているポインタ(オブジェクトCへのポインタ)に設定し直して(ポインタの付け替え)、オブジェクトAおよびBのロックを解除し(図5、502)、記憶した処理の完了待ちとなる。

【0038】

一方、挿入処理602ではオブジェクトグループTに対し、IXモードのロックを設定し、挿入用のオブジェクト領域確保を行い(オブジェクトDとする)、キー値12および関連するデータの設定を行なう。続いてオブジェクトAに対しXモードでロックを設定(削除処理がロック保持中であれば、解除まで待つことになる)し、オブジェクトAに格納されているポインタをオブジェクトDへのポインタに設定し、オブジェクトAからのポインタを、オブジェクトDを指すように付け替え、オブジェクトAのロックを解除し、最後にオブジェクトグループTのロックを解除し、挿入処理を完了する(図5、503)。

【0039】

挿入処理602のオブジェクトグループTのロックを解除により、削除処理601はオブジェクトBの領域を(ロックを設定することなく)解放し、オブジェクトグループTのロ

10

20

30

40

50

ックを解除して、削除処理を完了する(図5、504)。

#### 【0040】

次に、参照処理(キー値62)603の処理手順について説明する。まず、オブジェクトグループTに対しISモードのロックを設定し、オブジェクトAから順にポインタを辿っていき、キー値が62であるオブジェクトを探す。オブジェクト間を移動する際には、移動元のオブジェクトのロックを解除してから移動先のオブジェクトに対し、Sモードでロックを設定するものとする。なお、前記の削除処理601と並行して動作している場合は、オブジェクトAにアクセスした際の状態が図5、502の状態であれば、オブジェクトBへアクセスすることはなくポインタはオブジェクトCへのものが記憶される。従って、図6、603の点線に囲まれた処理は行なわれない。603の点線で囲まれた処理はオブジェクトBがまだ501の状態にあるときに実行される。このとき、オブジェクトAのアクセスでオブジェクトBへのポインタが記憶され、点線で囲まれた処理でオブジェクトCへのポインタが記憶される。このように参照のタイミングにより処理が変わってくる。

10

#### 【0041】

続いて、更新処理(キー値62)604の処理手順について説明する。まず、オブジェクトグループTに対しIXモードのロックを設定し、オブジェクトAから順にポインタを辿っていき、キー値が62であるオブジェクトを探し、見つけたら関連付けられているデータを更新する。オブジェクト間を移動する際には、移動元のオブジェクトのロックを解除してから移動先のオブジェクトに対し、Xモードでロックを設定するものとする。なお、前記の削除処理601と並行して動作している場合、オブジェクトAにアクセスした際の状態が図5、502の状態であれば、オブジェクトBへアクセスすることはなく、図6、604の点線に囲まれた処理は行なわれない。

20

#### 【0042】

前記処理方法により、削除処理によって解放された領域を誤ってアクセスすることなく、参照、挿入および更新処理の高い並列実行性を実現できる。更に述べれば、ハッシュ関数を用いたデータの管理においても実施の形態1と同様に、参照、挿入、および更新処理において、1つの処理が同時に2つのオブジェクトのロックを設定することがなく高い並列実行性を持つと共に、削除処理において、削除対象となるオブジェクトへのポインタを切り離れた後にそのオブジェクトグループにロックをかけている他の処理の完了を待って領域の解放を実行することにより他の処理の実行を阻害することなく、かつ誤ったデータが使われることなく領域の解放が出来る。

30

#### 【0043】

(実施の形態3)

従来技術2で説明したBツリーインデクスにおいて、データ削除やデータ挿入処理と並列実行可能な空きページ回収処理方法を図7および図8を用いて開示する。空きページ回収処理は、データの挿入・削除を繰り返すうちに格納効率が低下し、それに伴いアクセス性能が劣化している適当な時期に再編成に代わり実行することを想定している。

#### 【0044】

図7、703の状態において、空きページ回収処理が実行された場合の処理手順を図8、803に示す。まず、Bツリーインデクス全体Tに対しIXモードでロックを設定する。以降の説明において、ページに対するロックのモードについては[]内に記すものとする。次にルートページから一番左のリーフページに向かってページを辿る(ページP1[S]、P2[S]およびP4[X])。リーフページP4においてデータ件数が0件であることを確認し、回収予定であることを示すフラグ(図7、704 ページP4内の丸枠に「回」を記した記号)を設定する。続いてページP4の上位ページであるページP2にアクセス[X]して、ページP4へのポインタを切り離す(図7、704)。そして、ページP4の右側のページであるページP5にアクセス[X]して、データ件数が0件であることを確認し、回収予定であることを示すフラグを設定する。次に、上位ページであるページP2にアクセス[X]して、ページP5へのポインタを切り離すとページP2内にはキーが残らないことを確認して、ページ2に回収予定であることを示すフラグを設定する。続いてペー

40

50

ジ P 2 の上位ページであるページ P 1 にアクセス[X]して、ページ P 2 へのポインタを切り離す(図 7、705)。ページ P 5 の右側のページであるページ P 6 にアクセス[X]してデータ件数が 1 件以上であることを確認しページ P 6 の右側のページであるページ P 8 にアクセス[X]してデータ件数が 1 件以上であることを確認し、ページ P 8 の右側のページであるページ P 7 にアクセス[X]してデータ件数が 1 件以上であることを確認し、回収可能なページのフラグ設定およびポインタの切り離しが完了する。この時点で B ツリーインデクス全体 T に対しロックを設定している処理を記憶し、それらの処理の完了を待つ。記憶した処理がすべて完了したら、回収予定としたページ P 2、P 4 および P 5 の領域を解放し、最後に B ツリーインデクス全体 T のロックを解除して空きページ回収処理を完了する(図 7、706)。

10

#### 【0045】

前述した空き回収処理方法で得られた最終的な形では、領域解放を行なったページが格納を担当していたキーは、その右側のページが担当を引き継ぐ形となる。例えば、ページ P 6 はキー値が 50 以下のデータを担当する。それに伴い、参照、挿入および更新処理において、アクセスしたページに回収予定フラグが設定されているのを検知した場合、右側のページへ移動してアクセスするように処理を一部変更する。

#### 【0046】

ここまで示した処理方法により、B ツリーインデクスにおいても参照、挿入および更新処理の並列実行性を大きく損なうことなく、かつデータ参照、更新および挿入を抑止することなく空きとなっているページの領域を回収し、格納効率、アクセス効率を向上させることが可能である。

20

#### 【0047】

以上述べた実施の形態で説明したように、ポインタにより複数のオブジェクトが関連づけられるデータ構造を持ち、オブジェクトの参照、更新、挿入および削除を行なう処理が複数並行して動作するシステムにおいて、参照、更新および挿入の並列実行性が高く、かつ削除処理を参照、更新および挿入処理と並列実行可能な処理が可能である。

#### 【0048】

さらに、データの挿入および削除の繰り返しにより格納効率・アクセス効率が低下した B ツリーインデクスに対し、データ参照、更新および挿入を抑止したり、一時リソースを使用したりすることなく空きとなっているページの領域の回収し、格納効率、アクセス効率を向上させることが可能である。

30

#### 【0049】

##### 【発明の効果】

本発明によれば、参照、挿入および更新処理の並列実行性を持ち、かつ削除処理によって解放された領域を誤って他処理がアクセスすることも回避して削除処理後の解放(回収)も可能としている。

##### 【図面の簡単な説明】

【図 1】従来技術 1 および実施の形態 1 におけるデータ構造を示す図である。

【図 2】従来技術 1 におけるタイムチャートを示す図である。

【図 3】本発明の実施の形態 1 におけるタイムチャートを示す図である。

40

【図 4】本発明の実施の形態 1 における並列処理の様子を示す図である。

【図 5】本発明の実施の形態 2 におけるデータ構造を示す図である。

【図 6】本発明の実施の形態 2 におけるタイムチャートを示す図である。

【図 7】従来技術 2 および本発明の実施の形態 3 におけるデータ構造を示す図である。

【図 8】従来技術 2 および本発明の実施の形態 3 におけるタイムチャートを示す図である。

【図 9】本発明を実現するシステム構成の例を示す図である。

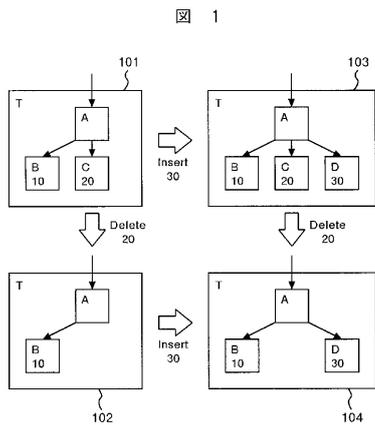
##### 【符号の説明】

1 ... コンピュータシステム、2 ... プロセッサ、3 ... メモリ、4 ... 外部記憶装置、5 ... オブジェクト、6 ... ネットワーク、7 ... 端末、10 ... データ管理システム、11 ... システム制

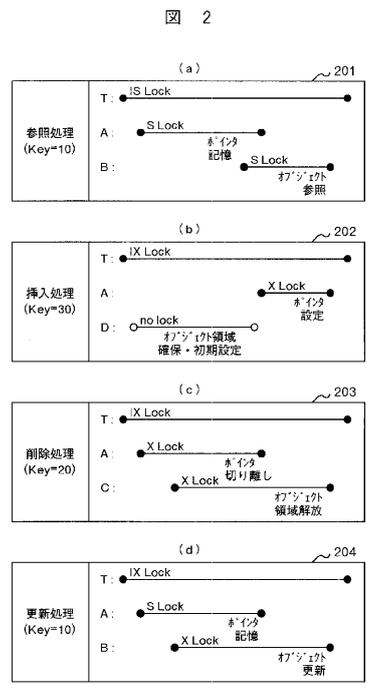
50

御部、 1 2 ... オブジェクト管理部、 1 3 ... ロック管理部

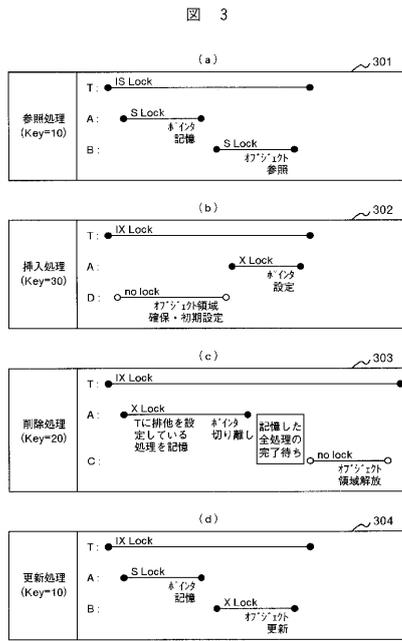
【 図 1 】



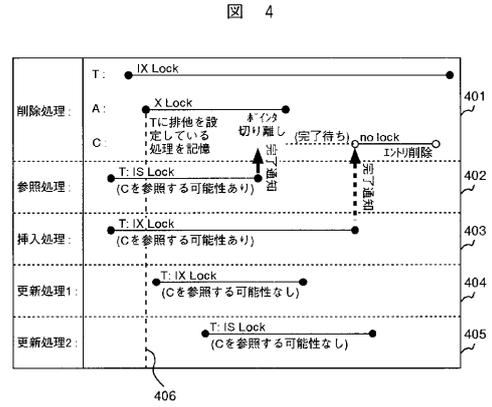
【 図 2 】



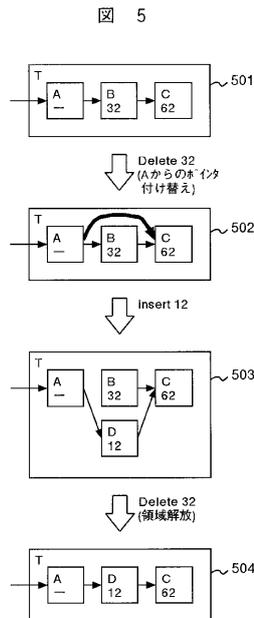
【 図 3 】



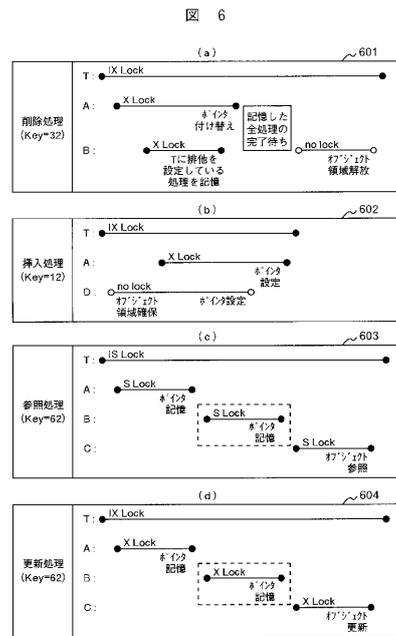
【 図 4 】



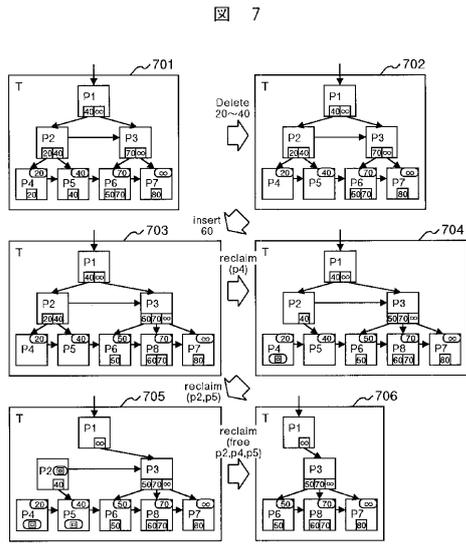
【 図 5 】



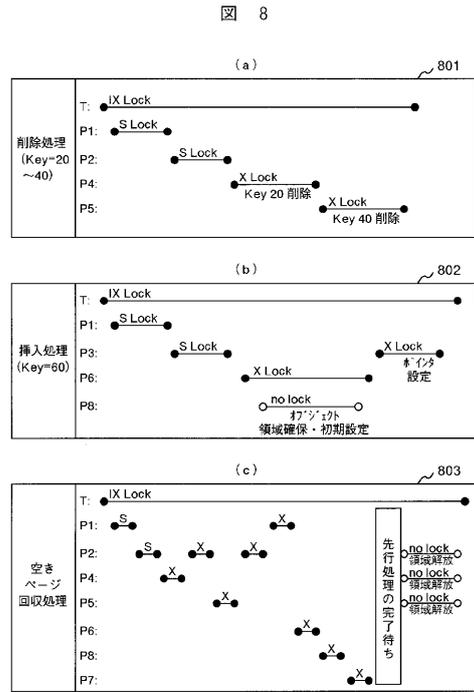
【 図 6 】



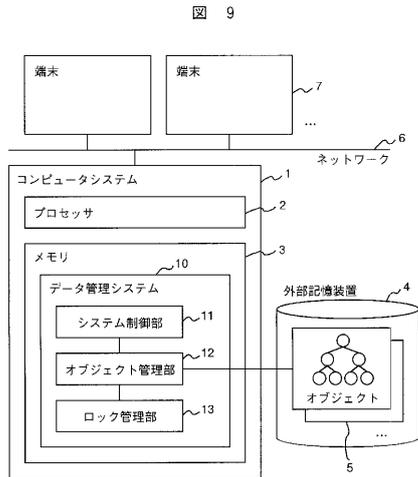
【 図 7 】



【 図 8 】



【 図 9 】



---

フロントページの続き

(72)発明者 中野 幸生

神奈川県川崎市幸区鹿島田890番地 株式会社日立製作所 ビジネスソリューション事業部内

審査官 浜岸 広明

(56)参考文献 米国特許第05832484(US,A)

グレイ ジム, トランザクション処理 [上], 日本, 日経BP社, 2001年10月29日,  
第1版, p. 541 - 581

(58)調査した分野(Int.Cl., DB名)

G06F 12/00

G06F 9/52