

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4340226号
(P4340226)

(45) 発行日 平成21年10月7日 (2009. 10. 7)

(24) 登録日 平成21年7月10日 (2009. 7. 10)

(51) Int. Cl.

F I

G 0 6 F 12/00 (2006. 01)

G 0 6 F 12/00 5 3 5 F

G 0 6 F 12/00 5 1 8 M

請求項の数 18 (全 20 頁)

(21) 出願番号 特願2004-504144 (P2004-504144)
 (86) (22) 出願日 平成15年4月29日 (2003. 4. 29)
 (65) 公表番号 特表2005-532615 (P2005-532615A)
 (43) 公表日 平成17年10月27日 (2005. 10. 27)
 (86) 国際出願番号 PCT/US2003/013326
 (87) 国際公開番号 W02003/096226
 (87) 国際公開日 平成15年11月20日 (2003. 11. 20)
 審査請求日 平成18年4月10日 (2006. 4. 10)
 (31) 優先権主張番号 60/379, 899
 (32) 優先日 平成14年5月10日 (2002. 5. 10)
 (33) 優先権主張国 米国 (US)
 (31) 優先権主張番号 10/263, 493
 (32) 優先日 平成14年10月2日 (2002. 10. 2)
 (33) 優先権主張国 米国 (US)

(73) 特許権者 502303739
 オラクル・インターナショナル・コーポレ
 イション
 アメリカ合衆国、94065 カリフォル
 ニア州、レッドウッド・ショアーズ、オラ
 クル・パークウェイ、500
 (74) 代理人 100064746
 弁理士 深見 久郎
 (74) 代理人 100085132
 弁理士 森田 俊雄
 (74) 代理人 100083703
 弁理士 仲村 義平
 (74) 代理人 100096781
 弁理士 堀井 豊

最終頁に続く

(54) 【発明の名称】 データ項目の使用可能バージョンの提供

(57) 【特許請求の範囲】

【請求項 1】

データ項目をトランザクションに提供するための、コンピュータにより実施される方法であって、

前記コンピュータのプロセッサが、データ項目の第1のバージョンを含むデータブロックの第1のバージョンを、揮発性メモリ内に配置するステップと、

データブロックの第1のバージョンがトランザクションにより使用可能かどうかにかかわらず、データ項目に関連する鮮度時点がトランザクションに関連するスナップショット時点以下であるとデータ項目の第1のバージョンがトランザクションにより使用可能であり、データ項目に関連する鮮度時点がトランザクションに関連するスナップショット時点より大きいとデータ項目の第1のバージョンがトランザクションにより使用可能でないと判断するステップと、

データ項目の第1のバージョンがトランザクションにより使用可能である場合、前記データ項目を、前記トランザクションに提供可能な候補として前記プロセッサが確立するステップと、

データ項目の第1のバージョンがトランザクションにより使用可能ではない場合、トランザクションにより使用可能なデータ項目のバージョンを、前記第1のバージョンとは異なるデータブロックの第2のバージョンから前記プロセッサが取得するステップとを含み、

鮮度時点は、データブロックの第1のバージョン内のデータ項目の第1のバージョンを

最近更新した、コミットされたトランザクションのコミット時点を示す、方法。

【請求項 2】

データ項目をトランザクションに提供するための、コンピュータにより実施される方法であって、

前記コンピュータのプロセッサが、データ項目の第 1 のバージョンを含むデータブロックの第 1 のバージョンを、揮発性メモリ内に配置するステップと、

データブロックの第 1 のバージョンがトランザクションにより使用可能かどうかにかかわらず、データ項目に関連する鮮度時点がトランザクションに関連するCANNOT-SEE時点よりも小さく、かつデータブロックに関連するEXCLUDE時点がトランザクションに関連するMUST-SEE時点よりも大きいとデータ項目の第 1 のバージョンがトランザクションにより使用可能であり、データ項目に関連する鮮度時点がトランザクションに関連するCANNOT-SEE時点よりも大きいとデータ項目の第 1 のバージョンがトランザクションにより使用可能でないと前記プロセッサが判断するステップと、

データ項目の第 1 のバージョンがトランザクションにより使用可能である場合、前記データ項目を、前記トランザクションに提供可能な候補として前記プロセッサが確立するステップと、

データ項目の第 1 のバージョンがトランザクションにより使用可能ではない場合、トランザクションにより使用可能なデータ項目のバージョンを、前記第 1 のバージョンとは異なるデータブロックの第 2 のバージョンから前記プロセッサが取得するステップとを含み、

鮮度時点は、データブロックの第 1 のバージョン内のデータ項目の第 1 のバージョンを最近更新した、コミットされたトランザクションのコミット時点を示す、方法。

【請求項 3】

前記プロセッサが、データブロックの第 1 のバージョンを修正することによって、データブロックの第 2 のバージョンを生成するステップをさらに含む、請求項 1 または 2 に記載の方法。

【請求項 4】

トランザクションにより使用可能なデータ項目のバージョンを、データブロックの第 2 のバージョンから前記プロセッサが取得するステップは、使用可能なデータ項目のバージョンを、データブロックの第 1 のバージョンとは別個に保持されているデータブロックのバージョンから前記プロセッサが取得することによって行なわれる、請求項 1 または 2 に記載の方法。

【請求項 5】

データ項目の第 1 のバージョンがロックされているかどうかを前記プロセッサが判断するステップと、

データ項目の第 1 のバージョンがロックされている場合、データ項目の第 1 のバージョンに対するロックを保持するトランザクションが実際にコミットされたかどうかを前記プロセッサが判断するステップと、

第 1 のデータ項目に対するロックを保持するトランザクションが実際にコミットされていた場合、ロックを保持する前記トランザクションのコミット時点、前記データ項目の鮮度時点として前記プロセッサが使用するステップとを行なうことによって鮮度時点を前記プロセッサが識別するステップをさらに含む、請求項 1 または 2 に記載の方法。

【請求項 6】

データ項目の第 1 のバージョンがロックされているかどうかを前記プロセッサが判断するステップと、

データ項目の第 1 のバージョンがロックされている場合、データ項目の第 1 のバージョンに対するロックを保持するトランザクションが実際にコミットされたかどうかを前記プロセッサが判断するステップと、

第 1 のデータ項目に対するロックを保持するトランザクションが実際にコミットされて

10

20

30

40

50

いなかった場合、第1のデータ項目に対するロックを保持するトランザクションが、データ項目を必要としているトランザクションであるかどうかを前記プロセッサが判断するステップと、

第1のデータ項目に対するロックを保持するトランザクションが、データ項目を必要としているトランザクションではない場合、データ項目の第1のバージョンはトランザクションにより使用可能ではないと前記プロセッサが判断するステップとを行なうことによって鮮度時点を識別するステップをさらに含む、請求項1または2に記載の方法。

【請求項7】

データ項目の第1のバージョンは、トランザクションにより使用可能なデータ項目のバージョンである複数の候補のうちの一候補として確立され、

トランザクションに関連するMUST-SEE時点は、トランザクションに関連するMUST-SEE時点とデータ項目に関連する鮮度時点とのうちの大きい方に変更され、

トランザクションに関連するCANNOT-SEE時点は、トランザクションに関連するCANNOT-SEE時点とデータブロックに関連するEXCLUDE時点とのうちの小さい方に変更され、

前記方法は、複数の候補のうち前記トランザクションに提供する候補に、MUST-SEE時点およびCANNOT-SEE時点により制限される範囲を最も広くする候補を前記プロセッサが選択するステップを含む、請求項2に記載の方法。

【請求項8】

データ項目の第1のバージョンは、トランザクションにより使用可能なデータ項目のバージョンである複数の候補のうちの一候補として確立され、

トランザクションに関連するMUST-SEE時点は、トランザクションに関連するMUST-SEE時点とデータ項目に関連する鮮度時点とのうちの大きい方に変更され、

トランザクションに関連するCANNOT-SEE時点は、トランザクションに関連するCANNOT-SEE時点とデータブロックに関連するEXCLUDE時点とのうちの小さい方に変更され、

前記方法は、複数の候補のうち前記トランザクションに提供する候補に、最大のCANNOT-SEE時点をもたらす候補を前記プロセッサが選択するステップを含む、請求項2に記載の方法。

【請求項9】

トランザクションは、マルチバージョンングデータベースシステム内で実行されるトランザクションであり、

データ項目は、前記データベースシステム内のテーブルの行である、請求項1または2に記載の方法。

【請求項10】

データ項目をトランザクションに提供するための命令を搬送するコンピュータ読取可能媒体であって、命令は、

データ項目の第1のバージョンを含むデータブロックの第1のバージョンを、揮発性メモリ内に配置するステップと、

データブロックの第1のバージョンがトランザクションにより使用可能かどうかにかかわらず、データ項目に関連する鮮度時点がトランザクションに関連するスナップショット時点以下であるとデータ項目の第1のバージョンがトランザクションにより使用可能であり、データ項目に関連する鮮度時点がトランザクションに関連するスナップショット時点より大きいとデータ項目の第1のバージョンがトランザクションにより使用可能でないと前記プロセッサが判断するステップと、

データ項目の第1のバージョンがトランザクションにより使用可能である場合、前記データ項目を、前記トランザクションに提供可能な候補として確立するステップと、

データ項目の第1のバージョンがトランザクションにより使用可能ではない場合、トランザクションにより使用可能なデータ項目のバージョンを、前記第1のバージョンとは異なるデータブロックの第2のバージョンから取得するステップとをコンピュータのプロセッサが行なうための命令を含み、

鮮度時点は、データブロックの第1のバージョン内のデータ項目の第1のバージョンを

10

20

30

40

50

最近更新した、コミットされたトランザクションのコミット時点を示す、コンピュータ読取可能媒体。

【請求項 1 1】

データ項目をトランザクションに提供するための命令を搬送するコンピュータ読取可能媒体であって、命令は、

データ項目の第 1 のバージョンを含むデータブロックの第 1 のバージョンを、揮発性メモリ内に配置するステップと、

データブロックの第 1 のバージョンがトランザクションにより使用可能かどうかにかかわらず、データ項目に関連する鮮度時点がトランザクションに関連するCANNOT-SEE時点よりも小さく、かつデータブロックに関連するEXCLUDE時点がトランザクションに関連するMUST-SEE時点よりも大きいとデータ項目の第 1 のバージョンがトランザクションにより使用可能であり、データ項目に関連する鮮度時点がトランザクションに関連するCANNOT-SEE時点よりも大きいか、もしくはデータブロックに関連するEXCLUDE時点がトランザクションに関連するMUST-SEE時点よりも小さいとデータ項目の第 1 のバージョンがトランザクションにより使用可能でないと判断するステップと、

データ項目の第 1 のバージョンがトランザクションにより使用可能である場合、前記データ項目を、前記トランザクションに提供可能な候補として確立するステップと、

データ項目の第 1 のバージョンがトランザクションにより使用可能ではない場合、トランザクションにより使用可能なデータ項目のバージョンを、前記第 1 のバージョンとは異なるデータブロックの第 2 のバージョンから取得するステップとをコンピュータのプロセッサが行なうための命令を含み、

鮮度時点は、データブロックの第 1 のバージョン内のデータ項目の第 1 のバージョンを最近更新した、コミットされたトランザクションのコミット時点を示す、コンピュータ読取可能媒体。

【請求項 1 2】

データブロックの第 1 のバージョンを修正することによって、データブロックの第 2 のバージョンを生成するステップを前記プロセッサが行なうための命令をさらに含む、請求項 1 0 または 1 1 に記載のコンピュータ読取可能媒体。

【請求項 1 3】

トランザクションにより使用可能なデータ項目のバージョンを、データブロックの第 2 のバージョンから取得するステップは、使用可能なデータ項目のバージョンを、データブロックの第 1 のバージョンとは別個に保持されているデータブロックのバージョンから取得することによって行なわれる、請求項 1 0 または 1 1 に記載のコンピュータ読取可能媒体。

【請求項 1 4】

データ項目の第 1 のバージョンがロックされているかどうかを判断するステップと、

データ項目の第 1 のバージョンがロックされている場合、データ項目の第 1 のバージョンに対するロックを保持するトランザクションが実際にコミットされたかどうかを判断するステップと、

第 1 のデータ項目に対するロックを保持するトランザクションが実際にコミットされていた場合、ロックを保持する前記トランザクションのコミット時点、前記データ項目の鮮度時点として使用するステップとを行なうことによって鮮度時点を識別するステップを前記プロセッサが行なうための命令をさらに含む、請求項 1 0 または 1 1 に記載のコンピュータ読取可能媒体。

【請求項 1 5】

データ項目の第 1 のバージョンがロックされているかどうかを判断するステップと、

データ項目の第 1 のバージョンがロックされている場合、データ項目の第 1 のバージョンに対するロックを保持するトランザクションが実際にコミットされたかどうかを判断するステップと、

第 1 のデータ項目に対するロックを保持するトランザクションが実際にコミットされて

10

20

30

40

50

いなかった場合、第1のデータ項目に対するロックを保持するトランザクションが、データ項目を必要としているトランザクションであるかどうかを判断するステップと、

第1のデータ項目に対するロックを保持するトランザクションが、データ項目を必要としているトランザクションではない場合、データ項目の第1のバージョンはトランザクションにより使用可能ではないと判断するステップとを行なうことによって鮮度時点を識別するステップを前記プロセッサが行なうための命令をさらに含む、請求項10または11に記載のコンピュータ読取可能媒体。

【請求項16】

データ項目の第1のバージョンは、トランザクションにより使用可能なデータ項目のバージョンである複数の候補のうちの一候補として確立され、

トランザクションに関連するMUST-SEE時点は、トランザクションに関連するMUST-SEE時点とデータ項目に関連する鮮度時点とのうちの大きい方に変更され、

トランザクションに関連するCANNOT-SEE時点は、トランザクションに関連するCANNOT-SEE時点とデータブロックに関連するEXCLUDE時点とのうちの小さい方に変更され、

前記コンピュータ読取可能媒体は、複数の候補のうち前記トランザクションに提供する候補に、MUST-SEE時点およびCANNOT-SEE時点により制限される範囲を最も広くする候補を選択するステップを前記プロセッサが行なうための命令を含む、請求項11に記載のコンピュータ読取可能媒体。

【請求項17】

データ項目の第1のバージョンは、トランザクションにより使用可能なデータ項目のバージョンである複数の候補のうちの一候補として確立され、

トランザクションに関連するMUST-SEE時点は、トランザクションに関連するMUST-SEE時点とデータ項目に関連する鮮度時点とのうちの大きい方に変更され、

トランザクションに関連するCANNOT-SEE時点は、トランザクションに関連するCANNOT-SEE時点とデータブロックに関連するEXCLUDE時点とのうちの小さい方に変更され、

前記コンピュータ読取可能媒体は、複数の候補のうち前記トランザクションに提供する候補に、最大のCANNOT-SEE時点をもたらし候補を選択するステップを前記プロセッサが行なうための命令を含む、請求項11に記載のコンピュータ読取可能媒体。

【請求項18】

トランザクションは、マルチバージョンングデータベースシステム内で実行されるトランザクションであり、

データ項目は、前記データベースシステム内のテーブルの行である、請求項10または11に記載のコンピュータ読取可能媒体。

【発明の詳細な説明】

【技術分野】

【0001】

優先権の主張

この出願は、その開示全体がここに十分に述べられているかのごとくここに引用される、「データ項目の使用可能バージョンの提供」(Providing a Useable Version of a Data Item)と題され、発明者としてアミット ガネーシュ (AMIT GANESH)、シュリーニバース・ベムリ (SRINIVAS S. VEMURI) およびロジャー・J・パンフォード (ROGER J. B. AMFORD) の名が挙げられた、2002年5月10日出願の仮特許出願連続番号第60/379,899号の優先権を主張する。

【0002】

発明の分野

この発明はデータベースシステムに関し、より具体的には、データベースシステム内のトランザクションがデータ項目の或るバージョンを使用できるかどうかを判断するための方法に関する。

【背景技術】

【0003】

10

20

30

40

50

発明の背景

典型的なデータベースシステムでは、ユーザは、データベースサーバにコマンドをサブミットすることによって情報を記憶、更新および検索する。データベースサーバによって正しく処理されるには、コマンドは、データベースサーバによりサポートされているデータベース言語に準拠していなければならない。或る人気の高いデータベース言語は、構造化照会言語 (SQL: Structured Query Language) として公知である。

【0004】

「データ項目」という用語は、ここでは、任意の論理データタイプまたは構造に関連する任意の値、もしくは値の組を指す。リレーショナルデータベースシステム内では、典型的なデータ項目は、テーブル、行、行内のフィールド、インデックス、インデックス入力、ディスクブロックなどを含む。

10

【0005】

マルチバージョンデータベースシステムでは、同じデータ項目のバージョンがいくつか存在する場合がある。通常、データ項目の1つのバージョンが「最新の」もの（つまり、そのデータ項目にこれまでなされた変更をすべて反映しているもの）であり、他のバージョンは最新のものではない。最新のものではないバージョンは通常、以前の時点での最新バージョンの様子を反映している。

【0006】

マルチバージョンデータベースシステムにはさまざまなタイプがある。マルチバージョンデータベースシステムの中には、或るデータ項目の多数のバージョンを永続的に記憶するものもある。また、データ項目の1つのバージョンを永続的に記憶し、アンドゥーまたはリドゥー記録を適用して変更を除去または追加することによって他のバージョンを必要に応じて再現するマルチバージョンデータベースシステムもある。ここに説明される手法は、マルチバージョンシステムのどの特定のタイプにも限定されない。

20

【0007】

マルチバージョンデータベースシステムは通常、或る特定のレベルの細分性 (granularity) でバージョンングを行なう。このため、或るテーブル全体の別個のコピーを保存する（または生成する）よりもむしろ、マルチバージョンシステムは通常、或るテーブルの比較的小さな塊の別個のコピーを保存（または生成）する。説明のため、システムがバージョンングを管理する細分性レベルを表わす塊を、ここでは「データブロック」と呼ぶことにする。

30

【0008】

現在のデータベースシステムでは、データベースに含まれるデータブロックのコピーはしばしば揮発性メモリに記憶され、それは、必要なアクセス時間が不揮発性メモリよりも少なく、トランザクション処理性能を向上させる。マルチバージョンデータベースシステムでは、同じデータブロックの多数のコピーが揮発性メモリに保持され、並行して実行されるトランザクションがその中に含まれるデータ項目の異なるバージョンに同時にアクセスできるようになっている場合がある。

【0009】

データベースシステムの状況では、トランザクションとは、1つ以上のデータベース言語ステートメントから構成される作業の論理的な単位である。データベースシステムがトランザクションを実行する場合、トランザクションは、前のトランザクションの実行に回答して書込まれた、または更新されたデータ項目を読み出す、または更新する場合がある。したがって、任意の所与のトランザクションを実行することによってデータベースシステムにより戻される結果は通常、前に実行された1組のトランザクションによりなされた変更によって指示される。

40

【0010】

各トランザクションはデータベースを一貫した状態で見なければならぬため、或るデータ項目の全バージョンが必ずしも或るトランザクションによって使用可能であるとは限らない。たとえば、そのデータ項目の一貫したビューを見るトランザクションについて、

50

そのトランザクションが、データ項目になされた或る更新を見る必要があるかもしれない。しかしながら、データ項目を含むデータブロックのバージョンの中には、トランザクションによって見られなければならない更新を含まないものがあるかもしれない。他方、データ項目を含むデータブロックのバージョンの中には、トランザクションによって見ることができない更新を含むものがあるかもしれない。このため、データブロックの多数のバージョンがトランザクションにとって利用可能である状況においては、トランザクションによるデータ項目のバージョンへのアクセスは、データベース一貫性が維持されるように管理されなければならない。

【 0 0 1 1 】

説明のため、「トランザクション」という用語は、ここでは、データ項目の可能なバージョンのすべてではないもののいくつかを使用することを何らかの理由で許可された任意のエンティティを、より一般的に指すために用いられるものとする。

【 発明の開示 】

【 発明が解決しようとする課題 】

【 0 0 1 2 】

システムが多数のノードを含み、それらの各々がそれ自体のキャッシュを有する場合、データ項目の「使用可能なバージョン」をトランザクションに提供するための効率のよい手法を用いることがより一層重要である。たとえば、或る特定のノード上で実行中のトランザクションが、或るデータ項目の或る特定のバージョンを必要としていると仮定されたい。データ項目のその必要なバージョンがその特定のノードのキャッシュ内のデータブロックにある場合、データ項目の所望のバージョンは、最小限のオーバーヘッドで、そのデータブロックからトランザクションへ供給され得る。しかしながら、データ項目の必要なバージョンを含むデータブロックが（１）ディスク上のみに存在する、（２）異なるノードのキャッシュ内に存在する、または（３）ディスク上の、または異なるノードのキャッシュ内の情報に基づいて再構築されなければならない、という場合、必要なバージョンを提供する作業は極めて効率が悪くなり得る。

【 0 0 1 3 】

前述の事項に基づき、データ項目の使用可能バージョンをトランザクションに提供するための効率のよい手法を提供することが望ましい。

【 課題を解決するための手段 】

【 0 0 1 4 】

発明の概要

データ項目がデータブロックの多数のバージョン上に存在し得るマルチバージョンングシステムにおいて、データ項目をトランザクションに提供するための手法が提供される。この発明の一面によれば、その手法は、データ項目の第１のバージョンを含むデータブロックの第１のバージョンを、揮発性メモリ内に配置することを伴う。次に、データブロックの第１のバージョンがトランザクションにより使用可能かどうかにかかわらず、データ項目の第１のバージョンがトランザクションにより使用可能かどうか判断される。データ項目の第１のバージョンがトランザクションにより使用可能である場合、データ項目は、トランザクションに提供可能な候補として確立される。このため、ブロック内のデータ項目は、データ項目が存在するデータブロックのバージョンがさもなくばそのトランザクションにより使用可能ではない場合でも、トランザクションに提供されるべき候補と考えられ得る。データ項目の第１のバージョンがトランザクションにより使用可能ではない場合、トランザクションにより使用可能なデータ項目のバージョンは、第１のバージョンとは異なるデータブロックの第２のバージョンから取得される。

【 0 0 1 5 】

行の或るバージョンが使用可能かどうかを判断するために使用される情報がトランザクションに関連するスナップショット時点を含んでいる実施例が説明される。また、行の或るバージョンが使用可能かどうかを判断するために使用される情報がMUST-SEE時点（必見時点）とCANNOT-SEE時点（見えない時点）とを含んでいる実施例も説明される。多数の使

10

20

30

40

50

用可能な候補の中から選択するためのさまざまなアプローチも説明される。

【 0 0 1 6 】

以下の記載では、説明のため、この発明の完全な理解を提供するために特定の詳細が述べられる。しかしながら、この発明がこれらの特定の詳細なしで実践されてもよいことは明らかである。他の点では、この発明を不必要に不明瞭にしないために、周知の構造および装置はブロック図の形で図示される。

【 0 0 1 7 】

機能的概要

トランザクションがデータ項目の或る特定のバージョンを使用できるかどうかを判断するためのアプローチが提供される。以前のアプローチとは異なり、データ項目の特定のバージョンが使用可能かどうかの判断は、データ項目のその特定のバージョンを含むデータブロックのバージョンが一般にトランザクションによって使用できない場合でもなされる。具体的には、データブロックの或る特定のバージョンが、あまりに最近起ったばかりでトランザクションには見えない変更を含む場合でも、トランザクションが関与する特定のデータ項目がそのような変更をどれも含まないかもしれず、したがって依然としてトランザクションにより使用可能であり得る。

【 0 0 1 8 】

一実施例によれば、データ項目の或る特定のバージョンが使用可能かどうかを判断することは、データブロック内のデータ項目がロックされているかどうかを判断することを伴う。データブロックがロックされていない場合、データ項目のそのバージョンを更新する最後のトランザクションは既にコミットされており、したがってコミット時点を割当てられている。説明のため、データブロック内のデータ項目の或る特定のバージョンを更新した最新のトランザクションのコミット時点を、ここでは、そのデータ項目のそのバージョンの「鮮度時点」(freshness time)と呼ぶこととする。

【 0 0 1 9 】

なお、ブロック内のデータ項目の或る特定のバージョンを更新した最新のトランザクションは、データブロック自体を更新する最新のトランザクションではないかもしれない。実際、同じデータブロックが、アクティブなトランザクションによりロックされている他のデータ項目を有し、なおかつ、特定のデータ項目が、関与するトランザクションが使用するには十分古い鮮度時間を有することは起り得る。

【 0 0 2 0 】

データ項目がロックされていない場合、データ項目の鮮度時点は、トランザクションに関連するスナップショット情報と比較される。以下により詳細に説明されるように、トランザクションに関連するスナップショット情報は、データのどのバージョンがトランザクションには見えるかおよび見えないかを示す。スナップショット情報と鮮度時点との比較は、データ項目のその特定のバージョンがトランザクションにより使用可能かどうかを示す。データ項目のそのバージョンがトランザクションにより使用可能である場合、データ項目のその特定のバージョンは、次にトランザクションに提供されるか、または、トランザクションに実際に提供されるバージョンがそこから選択される候補バージョンのプールに追加されるかもしれない。

【 0 0 2 1 】

多くのシステムでは、コミットされたトランザクションにより更新されたデータブロックは、トランザクションがコミットされたことを示すためにすぐに更新されるわけではない。したがって、データ項目がロックされていることをデータブロック内の情報が示していても、ロックが実際には、コミットされたトランザクションによって保持されているかもしれない。このため、データブロック内の情報によればデータ項目がロックされているという場合、ロックが実際にはアクティブなトランザクションによって保持されているかどうか判断される。

【 0 0 2 2 】

ロックを保持しているトランザクションが実際にコミットされている場合、ロックを保

10

20

30

40

50

持っているトランザクションのコミット時点は鮮度時点として使用され、上述の比較が行なわれて、データ項目が使用可能かどうかを判断する。

【 0 0 2 3 】

ロックを保持しているトランザクションがアクティブである場合、ロックを保持しているトランザクションがデータ項目を使用しようとしているトランザクションと同じトランザクションであるかどうか判断される。ロックを保持しているアクティブなトランザクションがデータ項目を使用しようとしているトランザクションではない場合、そのブロックのデータ項目のバージョンは使用可能ではない。他方、ロックを保持しているアクティブなトランザクションがデータ項目を使用しようとしているトランザクションと同じトランザクションである場合、あまりに最近起ったためにトランザクションには見えない任意の変更をデータ項目が有するかどうか判断される。あまりに最近起ったためにトランザクションには見えない任意の変更をデータ項目が有していない場合、データ項目はトランザクションによって使用可能である。

10

【 0 0 2 4 】

トランザクションによって使用可能なデータ項目の多くのバージョンがある状況では、バージョンのうちのどれが実際にトランザクションに提供されるかにより、効率が影響され得る。使用可能な候補のプール内のデータ項目のどのバージョンが或る特定のトランザクションに提供されるべきかを判断するためのさまざまな手法をここに説明する。

【 0 0 2 5 】

ブロックレベルバージョン情報

20

情報の正しいバージョンをトランザクションに提供するマルチバージョンングシステムについては、このシステムは情報のバージョンを示す情報を保持しなければならない。通常、データベースシステムはそのようなバージョンング情報をデータブロックレベルの細分性で保持している。なぜなら、それはバージョンングが行なわれる細分性レベルであるためである。たとえば、データベースシステムの中には、データブロックの各バージョンについて INCLUDE TIME (包含時点) および EXCLUDE TIME (除外時点) を保持するものもある。INCLUDE TIME は、データブロックのバージョンにその変更が含まれている最近コミットされたトランザクションのコミット時点を特定する。EXCLUDE TIME は、データブロックの内容が「最新の」ものであった時点を特定する。

【 0 0 2 6 】

30

たとえば、変更をロールバックすることによって作成されたデータブロックの EXCLUDE TIME は通常、データブロックのバージョンからその変更が既に除去された最も古いトランザクションのコミット時点である。データブロックの或る特定のバージョンがデータブロックの最近のバージョンである場合、そのバージョンについての EXCLUDE TIME は、変更が除去されていないため、無限である。

【 0 0 2 7 】

たとえば、第 1 のトランザクションが、データブロックのコピーを揮発性メモリ内にロードさせ、データブロックの新しいバージョン (「第 1 のバージョン」) を作成するためにデータブロックのコピーを更新し、次に時点 T 1 0 でコミットする、と仮定されたい。この第 1 のバージョンについての INCLUDE TIME は時点 T 1 0 に設定され、EXCLUDE TIME は無限に設定される。次に、第 2 のトランザクションが、データブロックの第 2 のバージョンを作成するためにデータブロックの同じコピーを更新し、時点 T 3 0 でコミットする。データブロックの第 2 のバージョンの INCLUDE TIME は時点 T 3 0 に変更されて、データブロックの第 2 のバージョンが時点 T 3 0 でコミットされたトランザクションによりなされた変更を含むことを反映しており、一方、EXCLUDE TIME は、データブロックの第 2 のバージョンから変更が除去されていないため、無限のままである。

40

【 0 0 2 8 】

ここで、第 3 のトランザクションが時点 T 2 0 でデータブロックへのアクセスを要求すると仮定されたい。データブロックの第 2 のバージョンは、時点 T 3 0 でコミットされた第 2 のトランザクションによりなされた更新を含んでいるため、その現在の状態では使用

50

できない。したがって、第2のトランザクションによりなされた変更を除去することによって、データブロックの第1のバージョンが再構築される。再構築動作が一旦完了すると、新しいコピーのINCLUDE TIMEは時点T10に変更され、EXCLUDE TIMEは時点T30に変更されて、データブロックのこのバージョンが時点T10でコミットされた第1のトランザクションによりなされた変更を含むこと、および時点T30でコミットされた第2のトランザクションによりなされた変更が除去されたことを反映する。このため、時点T10から時点T30まで(時点T30を含まず)存在したようなデータブロックへのアクセスを要求するトランザクションはどれも、データブロックのこの再構築された第1のバージョンを使用できる。

【0029】

行レベルバージョン情報

典型的なシステムでは、バージョン情報はデータブロックの細分性で保持される。なぜなら、システムはデータブロックの細分性でバージョンを管理するためである。しかしながら、この発明の一局面によれば、バージョン情報はデータ項目細分性で保持され、単一のデータブロックが多数のデータ項目を含む場合がある。データブロックバージョン情報に加えて、および/またはデータブロックバージョン情報の代わりに、データ項目バージョン情報が保持されてもよい。

【0030】

テーブルの或る特定の行にアクセスしようとする、データベースシステム内のトランザクションの状況において、この発明の実施例をここに説明する。テーブルのその行についてのデータは通常、他の数行についてのデータを含むデータブロック上に存在する。このため、データ項目が行である例を挙げる。したがって、バージョン情報は行レベルの細分性に基づいて保持される。しかしながら、行は、マルチバージョンシステム内のデータブロック上に存在し得るデータ項目の単なる一例である。この発明のアプローチは、どの特定のデータ項目にも、またはどの特定のタイプのマルチバージョンシステムにも限定されない。

【発明を実施するための最良の形態】

【0031】

図1は、この発明の一実施例に従った、トランザクションがデータ項目の或る特定のバージョンを使用できるかどうかを判断するための行レベルバージョン情報を保持するデータベースシステム100のブロック図である。

【0032】

図1を参照すると、データベースシステム100は、揮発性メモリ102と不揮発性メモリ104を含む。不揮発性メモリ104は通常、データベースが記憶される、磁気ドライブまたは光ドライブなどの1つ以上の記憶装置を表わしている。データブロック106は不揮発性メモリ104に記憶されており、1つ以上のデータ項目107を含んでいる。例示された実施例では、データ項目107はD1、D2、D3およびD4として個々に識別される。

【0033】

データブロック106は、データ項目107上で最近動作したトランザクションについての情報も含んでいる。具体的には、入力110に列挙されたトランザクションは、データブロック106内のデータ項目の各々を更新する最近のトランザクションを含んでいる。そのようなトランザクションは、アクティブである(その場合、データブロックは通常、トランザクションがデータ項目に対するロックを保持していることを示している)か、またはコミットされているかのいずれかであり得る。たとえば、入力110は、トランザクションTX1およびTX2がそれぞれ時点10および時点5でコミットされ、一方、トランザクションTX3はまだコミットされていないことを示している。

【0034】

データ項目を最後に更新したトランザクションがコミットされると、トランザクションのコミット時点は行のバージョン番号と考えられ得る。このため、TX1がデータブロッ

10

20

30

40

50

ク 1 0 6 内のデータ項目 1 を更新した最後のトランザクションである場合、データブロック 1 0 6 内のデータ項目 1 のバージョンは「1 0」（T X 1 のコミット時点）と考えられ得る。このように、入力 1 1 0 は、データブロック 1 0 6 内のデータ項目についてのバージョン情報を、データ項目レベルの細分性で伝達する。以下により詳細に説明されるように、データ項目レベルバージョン情報を有することにより、データ項目は、そのデータ項目を含むデータブロックが一般にトランザクションにより使用可能ではない場合でも、トランザクションに提供されるようになる。

【 0 0 3 5 】

システムによっては、データブロック内の情報が、トランザクションが実際にコミットされた際にトランザクションがアクティブであることを示すことは、可能である。この現象は、トランザクションがコミットされる時点と、トランザクションにより修正されたデータブロックが更新されてトランザクションがコミットされたことを示す時点との間に起り得る遅延によるものである。以下により詳細に示すように、この現象は、データブロック内の情報によればアクティブであるトランザクションが実際にコミットされたかどうかを判断することによって勘案され得る。

10

【 0 0 3 6 】

システム 1 0 0 に示された構造は、マルチバージョンングシステムがいかにしてバージョン情報をデータ項目レベルの細分性で保持するかの単なる一例である。特定のバージョン情報、および情報を保持するための手法は、実現化例によって異なり得る。この発明は、バージョン情報をデータ項目レベルの細分性で保持するためのどの特定の構造または手法にも限定されない。

20

【 0 0 3 7 】

スナップショット情報

データベース一貫性を維持するためにトランザクションの実行を管理するための一アプローチは、トランザクションが（１）或る特定の組のコミットされたトランザクションによってデータベースにコミットされた変更、および（２）トランザクション自体の以前のステートメントによってなされた変更のみを見ることを確実にすることを伴う。或るステートメントにはその変更が見える、その組のトランザクションは、そのステートメントの「スナップショットセット」と呼ばれる。このため、トランザクションはデータベースの「スナップショット」を見なければならない。トランザクションのスナップショットは、そのスナップショットセット内のトランザクションによりなされたすべての変更を含み、そのスナップショットセットにはない任意のトランザクション（「除外されたトランザクション」）によってなされた変更は全く含まない。

30

【 0 0 3 8 】

一手法によれば、マルチバージョンングデータベースシステムは、トランザクションが実行を開始した時点でのデータベースの状態に対応するスナップショット時点を各トランザクションに割当てることによって、トランザクションがデータベースの一貫したスナップショットを見ることを確実にし得る。一貫した読出については、トランザクションは、トランザクションのスナップショット時点がそれより前にコミットされたトランザクションによってデータベースになされたすべての変更を見なければならず、割当てられたスナップショット時点の後でコミットされたトランザクションによってなされた変更を見ることはできない。

40

【 0 0 3 9 】

一貫した読出の概念は、１９９７年４月２３日出願の米国特許出願第 ０ ８ / ８ ４ ２ , １ ６ ９ 号「一貫した読出のためのスナップショットの共有」（SHARING SNAPSHOTS FOR CONSISTENT READS）、１９９７年４月２３日出願の米国特許出願第 ０ ８ / ８ ３ ８ , ９ ６ ７ 号「データベースのスナップショットの数を減少させるための手法」（TECHNIQUES FOR REDUCING THE NUMBER OF SHAPSHOTS OF A DATABASE）、および１９９７年４月２３日出願の米国特許出願第 ０ ８ / ８ ４ １ , ５ ４ １ 号「ダイナミックスナップショットセット調節」（DYNAMIC SNAPSHOT SET ADJUSTMENT）により詳細に記載されており、それらの各々の内容は

50

ここに引用により援用される。

【 0 0 4 0 】

単純な一例として、トランザクションが、T 1 5 の時点で存在したようなデータブロックの或るバージョンへのアクセスを要求すると仮定されたい（つまり、このトランザクションのスナップショット時点はT 1 5である）。トランザクションには、時点T 1 5またはそれ以前にコミットされたトランザクションによってなされたすべての変更を含むものの時点T 1 5の後でコミットされたトランザクションによってなされた変更は含まないデータ項目のバージョンが供給されなければならない。まだコミットされていないトランザクション（要求中のトランザクション自体は除く）、または時点T 1 5の後でコミットされたトランザクションによってなされた更新を含むデータ項目のバージョンは使用できない。

10

【 0 0 4 1 】

データ項目のバージョンが使用可能かどうかの判断

この発明の一実施例に従った、トランザクションがデータ項目の或るバージョンを使用できるかどうかを判断するための方法を、図 2 のフローチャートを参照してここに説明する。図 2 で識別されたステップは、トランザクションによって要求されるデータ項目がデータベースシステム内のデータブロックに位置する行である実施例に言及している。

【 0 0 4 2 】

一般に、その行のバージョンがトランザクションによって使用されるかどうかの判断は、その行に関連するバージョン情報を、その行を必要とするトランザクションに関連するスナップショット情報と比較することによってなされる。しかしながら、行のバージョン情報は、それがトランザクションのスナップショット情報と比較される前に判断されなければならない。したがって、図 2 に示したステップのうちのいくつかは、或る行の或るバージョンの鮮度時点を判断する作業に対応している。

20

【 0 0 4 3 】

図 2 を参照すると、ステップ 2 0 0 で、データブロックに関連するEXCLUDE時点がトランザクションに関連するスナップショット時点未満であるかどうか判断される。データブロックに関連するEXCLUDE時点がトランザクションに関連するスナップショット時点未満である場合、データブロックに含まれる行のバージョンは、トランザクションによって見られなければならない欠落した変更であるかもしれない。したがって、制御はステップ 2 0 2 へ渡り、そこでシステムは、行の正しいバージョンを、データブロックの別のバージョンから得る。

30

【 0 0 4 4 】

データブロックに関連するEXCLUDE時点がトランザクションに関連するスナップショット時点以上である場合、制御はステップ 2 0 4 へ渡る。ステップ 2 0 4 で、データブロック内の情報が、行がロックされていることを示すかどうかについて、判断がなされる。行がロックされている場合、制御はステップ 2 0 5 へ進む。その他の場合、制御はステップ 2 1 0 へ渡る。

【 0 0 4 5 】

ステップ 2 1 0 で、そのデータブロック内の行のバージョンを修正した最新のトランザクションのコミット時点が判断される。コミット時点は、たとえば、行を更新する最新のトランザクションについての情報を含むエントリ 1 1 0（図 1）内のエントリを検査することによって判断されてもよい。このように判断されたコミット時点は、行のそのバージョンの鮮度時点を構成する。

40

【 0 0 4 6 】

ステップ 2 1 2 で、鮮度時点がスナップショット時点未満、またはそれと等しいかどうか、判断がなされる。そうではない場合、トランザクションはデータブロック 1 0 6 のこのバージョンを使用できない。なぜなら、行のバージョンが、新しすぎてトランザクションには見えない修正を含むためである。この状況下では、制御はステップ 2 0 2 へ進み、データブロックの別のバージョンが使用される。システムによっては、新しすぎてトラン

50

ザクションには見えない変更を除去するためにアンドゥー記録をデータブロックに適用することによって、データブロックの他のバージョンが生成されてもよい。

【 0 0 4 7 】

鮮度時点がスナップショット時点未満、またはそれと等しい場合、制御はステップ 2 1 2 からステップ 2 1 4 へ渡る。ステップ 2 1 4 で、トランザクションを使用するために、行のバージョンがトランザクションに提供される。代替的な一実施例では、行のバージョンは、バージョンがトランザクションにより使用可能であると判断されたというだけの理由で、すぐさまトランザクションに提供されなくてもよい。むしろ、システムは、トランザクションにより使用可能な行のいくつかまたはすべてのバージョンを識別するよう進行してもよい。使用可能な候補のプールが一旦決定されると、システムは、行のどのバージョンを実際にトランザクションに提供するかを選択するためのさまざまな基準を使用してもよい。さまざまな選択基準は、以下により詳細に説明されるように使用され得る。

10

【 0 0 4 8 】

ステップ 2 1 8 で、トランザクションに関連するスナップショット情報は、トランザクションに供給された行の特定のバージョンを反映するように更新される。このステップは、たとえば、スナップショット情報がスナップショット範囲を識別する実施例において行なわれる。そのような実施例、およびスナップショット情報がいかに更新されるかの詳細を、以下により詳細に説明する。

【 0 0 4 9 】

ステップ 2 0 4 で、データブロックに含まれる情報が、行がロックされていることを示す場合、ステップ 2 0 5 で、ロックを保持するトランザクションが本当にアクティブかどうか判断される。データブロック内の情報が、行がロックされていることを示していても、ロックを保持するトランザクションが実際には既にコミットされたかもしれず、一方データブロックはまだトランザクションのコミット情報を用いて更新されていなかったために、このステップは行なわれる。

20

【 0 0 5 0 】

ロックを保持するトランザクションがアクティブでない場合、制御はステップ 2 1 0 に渡されて、ロックを保持していたトランザクション（データブロックのそのバージョンにある行のバージョンを修正する最新のトランザクション）のコミット時点が判断される。

【 0 0 5 1 】

30

ロックを保持するトランザクションがアクティブである場合、ステップ 2 0 6 で、ロックを保持するトランザクションが行を読出そうとしているトランザクションと同じトランザクションかどうか判断される。ロックを保持するトランザクションが行を読出そうとしているトランザクションと同じトランザクションではない場合、データブロック内にある行のバージョンは、トランザクションには見えない情報を潜在的に含んでおり、制御はステップ 2 0 2 へ渡る。

【 0 0 5 2 】

ロックを保持するトランザクションが行を読出そうとしているトランザクションと同じトランザクションである場合、ステップ 2 0 8 で、行のそのバージョンがトランザクションには新しすぎて見えない情報を含むかどうか判断される。この試験は、データベースシステムによっては、トランザクションが、或る特定の時点までにそれ自体がなした変更を見ることができるものの、その特定の時点の後でなされた変更は見ることができないために、行なわれる。

40

【 0 0 5 3 】

行のバージョンがトランザクションには新しすぎて見えない情報を含む場合、制御はステップ 2 0 2 へ渡る。その他の場合、制御はステップ 2 1 4 へ渡る。

【 0 0 5 4 】

スナップショット範囲

トランザクションについての単一のスナップショット時点の代わりに、データベースシステムの中には、MUST-SEE時点およびCANNOT-SEE時点を使用して各トランザクションにつ

50

いての依存性範囲を規定するものもあり、それは、読出すのに便利なバージョンを選択する上でより一層の柔軟性を提供し得る。このアプローチによれば、トランザクションは、コミット時点がMUST-SEE時点かそれより前の任意のトランザクションによってなされた変更を見なければならない。加えて、トランザクションは、コミット時点がCANNOT-SEE時点かそれより後の任意のトランザクションによってなされた変更を見ることができない。MUST-SEE時点およびCANNOT-SEE時点の概念は、上述のように引用により援用される出願に、より詳細に記載されている。

【 0 0 5 5 】

データ項目のバージョンの鮮度時点およびデータブロックのバージョンについてのEXCLUDE TIMEと併せて使用する場合、所与のトランザクションがデータブロックの或るバージョンを使用できるようになるには、2つの条件が満たされなければならない。第1に、所与のトランザクションのCANNOT-SEE時点は、データ項目のバージョンの鮮度時点よりも大きくななければならない。言い換えれば、データ項目のバージョンにその変更が反映された最新のトランザクションは、所与のトランザクションのCANNOT-SEE時点の前にコミットされていなければならない。第2に、所与のトランザクションのMUST-SEE時点は、データブロックのバージョンのEXCLUDE TIMEよりも小さくななければならない。言い換えれば、所与のトランザクションによって見られるべき変更を、データブロックのバージョンから除去することはできない。

【 0 0 5 6 】

たとえば、行の或る特定のバージョンが鮮度時点T 3 3を有し、行のそのバージョンが存在するデータブロックがEXCLUDE TIME T 3 9を有すると仮定されたい。MUST-SEE時点T 3 0およびCANNOT-SEE時点T 4 0を有するトランザクションが、データブロックへのアクセスを要求する。鮮度時点がCANNOT-SEE時点よりも小さく、EXCLUDE TIMEがMUST-SEE時点よりも大きい場合、データブロックのその特定のバージョンはこのトランザクションに供給可能である。

【 0 0 5 7 】

読出後のスナップショット範囲の更新

スナップショット範囲を使用する実施例では、範囲は、データ項目のバージョンがトランザクションに供給された後で調整されなければならない場合がある。たとえば、鮮度時点T 1 3のデータ項目のバージョンがトランザクションに提供される場合、トランザクションはその後、時点T 1 3の時点でコミットされた全データを見なければならない。したがって、トランザクションのMUST-SEE時点がデータ項目のそのバージョンを読出す前のT 1 2である場合、トランザクションのMUST-SEE時点は、データ項目の読出後、時点T 1 3に調節されなければならない。したがって、トランザクションについてのMUST-SEE時点は、現在のMUST-SEE時点と鮮度時点との大きい方に変更される。

【 0 0 5 8 】

加えて、コミットされた或る特定のトランザクションによって変更が除去されたデータを読出す場合、読出トランザクションはその後、その特定のトランザクションにおいてまたはその後でコミットされたトランザクションによって任意のデータになされたどの変更も見ることができない。したがって、読出トランザクションについてのCANNOT-SEE時点は、トランザクションの現在のCANNOT-SEE時点と読出されたデータ項目を含むデータブロックのEXCLUDE TIMEとの小さい方に変更される。

【 0 0 5 9 】

利用可能な候補からの選択

上述のように、データ項目の或るバージョンがトランザクションに提供されると、CANNOT-SEE時点およびMUST-SEE時点が調節される。この調節の効果は、トランザクションが将来の読出において見るかもしれないスナップショットの範囲を狭くすることである。データ項目の異なる使用可能バージョンは、異なる方法で範囲に影響を与え得る。したがって、データ項目の使用可能バージョンがいくつか存在する場合、システムの効率は、どの使用可能バージョンがトランザクションに提供されるべきバージョンとして選択されるかに

基づいて影響され得る。

【 0 0 6 0 】

一実施例によれば、候補のプールからの候補の選択は、候補を使用することが読出トランザクションのスナップショット範囲に対して有する影響に基づく。例示のため、候補バージョンのプールが以下を含むと仮定されたい。

【 0 0 6 1 】

【表 1】

| バージョン | 鮮度 | EXCLUDE TIME |
|-------|-----|--------------|
| V1 | T10 | T30 |
| V2 | T15 | T33 |
| V3 | T9 | T20 |

10

データ項目を必要とするトランザクションが、MUST-SEE時点 T 1 1 およびCANNOT-SEE時点 T 3 2 を有すると仮定されたい。この筋書きでは、バージョン V 1、V 2 および V 3 はすべて、使用可能な候補である。なぜなら、全 3 バージョンについて、鮮度時点はCANNOT-SEE時点よりも小さく、MUST-SEE時点はEXCLUDE TIMEよりも小さいためである。

【 0 0 6 2 】

20

V 1 を提供することから生じる [MUST-SEE, CANNOT-SEE] 範囲は [T 1 1, T 3 0] である。V 2 を提供することから生じる [MUST-SEE, CANNOT-SEE] 範囲は [T 1 5, T 3 2] である。V 3 を提供することから生じる [MUST-SEE, CANNOT-SEE] 範囲は [T 1 1, T 2 0] である。

【 0 0 6 3 】

一実施例によれば、トランザクションに提供するバージョンは、バージョンがトランザクションに提供された後に残る [MUST-SEE, CANNOT-SEE] 範囲の大きさに基づいて、使用可能な候補から選択される。そのような実施例では、V 1 がトランザクションに提供される。なぜなら範囲 [T 1 1, T 3 0] は他の範囲よりも広いためである。

【 0 0 6 4 】

30

代替的な一実施例によれば、トランザクションに提供するバージョンは、最新のCANNOT-SEE時点をもたらず候補に基づいて、使用可能な候補から選択される。そのような実施例では、V 2 がトランザクションに提供される。なぜなら、T 3 2 は、他の候補のいずれかを使用することから生じるCANNOT-SEE時点よりも遅いためである。この実施例は、たとえば、トランザクションにデータのより新しいバージョンを見させようとするので、有用である。

【 0 0 6 5 】

上述の実施例は単に、トランザクションに実際に提供される候補を候補のプールから選択するのに使用され得る基準の 2 例を例示している。代替的な実施例では、これらの例で規定された基準の代わりに、または基準に加えて、異なる基準が考慮されてもよい。さらに、実施例によっては、使用可能な候補のプールを識別するというよりもむしろ、識別される第 1 の使用可能な候補を単に提供するかもしれない。

40

【 0 0 6 6 】

ハードウェア概要

図 3 は、この発明に従った一実施例が実現され得るコンピュータシステム 3 0 0 を示す。コンピュータシステム 3 0 0 は、情報を通信するためのバス 3 0 2 または他の通信メカニズムと、情報を処理するためにバス 3 0 2 と結合されたプロセッサ 3 0 4 とを含む。コンピュータシステム 3 0 0 はまた、プロセッサ 3 0 4 により実行されるべき命令および情報を記憶するためにバス 3 0 2 に結合された、ランダムアクセスメモリ (R A M) または他のダイナミック記憶装置といったメインメモリ 3 0 6 も含む。メインメモリ 3 0 6 は、

50

プロセッサ 304 により実行されるべき命令の実行中に一時的な変数または他の中間情報を記憶するためにも使用されてもよい。コンピュータシステム 300 はさらに、プロセッサ 304 用の命令およびスタティック情報を記憶するためにバス 302 に結合された読出専用メモリ (ROM) 308 または他のスタティック記憶装置を含む。磁気ディスクまたは光ディスクといった記憶装置 310 が、情報および命令を記憶するために提供され、バス 302 に結合されている。

【0067】

コンピュータシステム 300 は、情報をコンピュータユーザに表示するためのブラウン管 (CRT) などのディスプレイ 312 に、バス 302 を介して結合されていてもよい。英数字キーおよび他のキーを含む入力装置 314 が、情報およびコマンド選択をプロセッサ 304 に通信するためにバス 302 に結合されている。ユーザ入力装置の別の種類は、方向情報およびコマンド選択をプロセッサ 304 に通信し、ディスプレイ 312 上のカーソルの動きを制御するための、マウス、トラックボール、またはカーソル方向キーといったカーソル制御 316 である。この入力装置は通常、2つの軸、つまり第1の軸 (たとえば x) および第2の軸 (たとえば y) において2つの自由度を有しており、それによりこの装置は平面における場所を特定することができる。

10

【0068】

この発明は、トランザクションがデータ項目の或るバージョンを使用できるかどうかの判断のためにコンピュータシステム 300 を使用することに関する。この発明の一実施例によれば、トランザクションがデータ項目の或るバージョンを使用できるかどうかの判断は、プロセッサ 304 がメインメモリ 306 に含まれた命令のシーケンスを実行するのに応答して、コンピュータシステム 300 により提供される。そのような命令は、記憶装置 310 などの別のコンピュータ読取可能な媒体からメインメモリ 306 に読込まれてもよい。メインメモリ 306 に含まれる命令のシーケンスの実行により、プロセッサ 304 は、ここに説明された処理ステップを行なうようになる。プロセッサ 304 の代わりに多重処理構成を使用して、処理ステップを行なってもよい。

20

【0069】

代替的な実施例では、この発明を実現するために、ソフトウェア命令の代わりに、またはソフトウェア命令と組合わせて、配線接続回路が使用されてもよい。このため、この発明の実施例は、配線接続回路とソフトウェアとのどの特定の組合せにも限定されない。加えて、プロセッサ 304 の代わりに多重処理システムが採用されてもよい。

30

【0070】

ここで用いられるような用語「コンピュータ読取可能な媒体」とは、プロセッサ 304 に命令を実行用に提供することに関与するあらゆる媒体を指す。そのような媒体は、不揮発性媒体、揮発性媒体、および通信媒体を含むもののそれらに限定されない多くの形態を取り得る。不揮発性媒体はたとえば、記憶装置 310 などの光ディスクまたは磁気ディスクを含む。揮発性媒体は、メインメモリ 306 などのダイナミックメモリを含む。通信媒体は、バス 302 を構成する配線を含む、同軸ケーブル、銅線および光ファイバを含む。通信媒体はまた、無線周波数および赤外線データ通信中に発生するものなどの音波または光波の形も取り得る。

40

【0071】

コンピュータ読取可能な媒体の一般的な形態は、たとえば、フロッピー (登録商標) ディスク、フレキシブルディスク、ハードディスク、磁気テープ、または任意の他の磁気媒体、任意の他の光媒体、パンチカード、紙テープ、孔のパターンを有する任意の他の物理的媒体、RAM、PROM、EPROM、FLASH-EPROM、任意の他のメモリチップまたはカートリッジ、いかに説明するような搬送波、または、コンピュータがそこから読取可能な任意の他の媒体を含む。

【0072】

コンピュータ読取可能な媒体のさまざまな形態は、プロセッサに命令のシーケンスを実行用に搬送することに関与していてもよい。たとえば、命令はまず、遠隔コンピュータの

50

磁気ディスク上で搬送されてもよい。遠隔コンピュータは、命令をそのダイナミックメモリにロードし、電話回線を通してモデムを用いて命令を送信することができる。コンピュータシステム 300 にとってローカルなモデムは、電話回線上のデータを受信し、赤外線送信機を用いてデータを赤外線信号に変換することができる。バス 302 に結合された赤外線検出器は、赤外線信号で搬送されたデータを受信し、データをバス 302 上に配置することができる。バス 302 はデータをメインメモリ 306 に搬送し、そこからプロセッサ 304 は命令を検索して実行する。メインメモリ 306 によって受信された命令は、プロセッサ 304 による実行の前または後のいずれかで、記憶装置 310 上に随意に記憶されてもよい。

【0073】

コンピュータシステム 300 はまた、バス 302 に結合された通信インターフェイス 318 も含む。通信インターフェイス 318 は、ローカルネットワーク 322 に接続されたネットワークリンク 320 に双方向データ通信結合を提供する。たとえば、通信インターフェイス 318 は、データ通信接続を対応する種類の電話回線に提供するデジタル相互サービス網 (ISDN) カード、またはモデムであってもよい。別の例として、通信インターフェイス 318 は、データ通信接続を互換性がある LAN に提供するローカルエリアネットワーク (LAN) カードであってもよい。無線リンクも実現され得る。任意のそのような実現化例では、通信インターフェイス 318 は、さまざまな種類の情報を表わすデジタルデータストリームを搬送する電気信号、電磁信号、または光信号を送信および受信する。

【0074】

ネットワークリンク 320 は通常、1 つ以上のネットワークを介して、他のデータ装置にデータ通信を提供する。たとえば、ネットワークリンク 320 は、ローカルネットワーク 322 を介して、ホストコンピュータ 324 に、またはインターネットサービスプロバイダ (ISP) 326 により運営されるデータ装置に、接続を提供してもよい。ISP 326 は次に、現在一般に「インターネット」328 と呼ばれている全世界的パケットデータ通信ネットワークを介して、データ通信サービスを提供する。ローカルネットワーク 322 およびインターネット 328 は双方とも、デジタルデータストリームを搬送する電気信号、電磁信号または光信号を使用する。コンピュータシステム 300 へ、またはコンピュータシステム 300 からデジタルデータを搬送する、さまざまなネットワークを通る信号、ネットワークリンク 320 上の信号、および通信インターフェイス 318 を通る信号は、情報を運ぶ搬送波の例示的な形態である。

【0075】

コンピュータシステム 300 は、ネットワーク、ネットワークリンク 320 および通信インターフェイス 318 を介して、メッセージを送信し、プログラムコードを含むデータを受信する。インターネットの例では、サーバ 330 は、アプリケーションプログラム用の要求されたコードを、インターネット 328、ISP 326、ローカルネットワーク 322、および通信インターフェイス 318 を介して送信してもよい。この発明によれば、そのようなダウンロードされたアプリケーションは、トランザクションがデータ項目の或るバージョンを使用できるかどうかを判断することに備える。

【0076】

受信されたコードは、受信された際にプロセッサ 304 によって実行されてもよく、および/または、後の実行用に記憶装置 310 または他の不揮発性記憶装置に記憶されてもよい。このように、コンピュータシステム 300 は、搬送波の形をしたアプリケーションコードを取得し得る。

【0077】

前述の明細書において、この発明を、その特定の実施例を参照して説明してきた。しかしながら、この発明のより幅広い精神および範囲から逸脱することなく、様々な修正および変更がそれになされてもよいことは明らかである。明細書および図面はしたがって、限定的な意味というよりは例示的な意味で考えられるものとする。

10

20

30

40

50

【図面の簡単な説明】

【 0 0 7 8 】

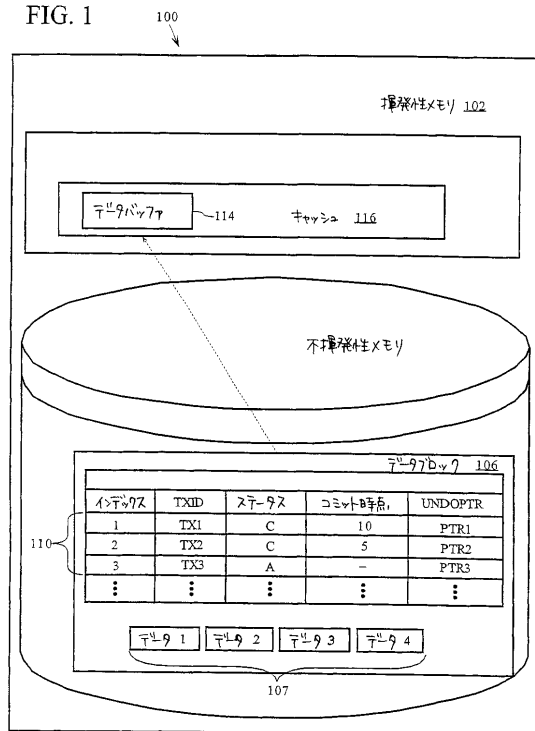
【図 1】この発明の一実施例に従った、トランザクションがデータ項目の或る特定のバージョンを使用できるかどうかを判断するための行レベルバージョン情報を保持するデータベースシステム 100 のブロック図である。

【図 2】この発明の一実施例に従った、トランザクションがデータ項目の或るバージョンを使用できるかどうかを判断するための方法を示すフローチャートである。

【図 3】この発明に従った一実施例が実現され得るコンピュータシステム 300 を示す図である。

【 図 1 】

FIG. 1



【 図 2 】

FIG. 2

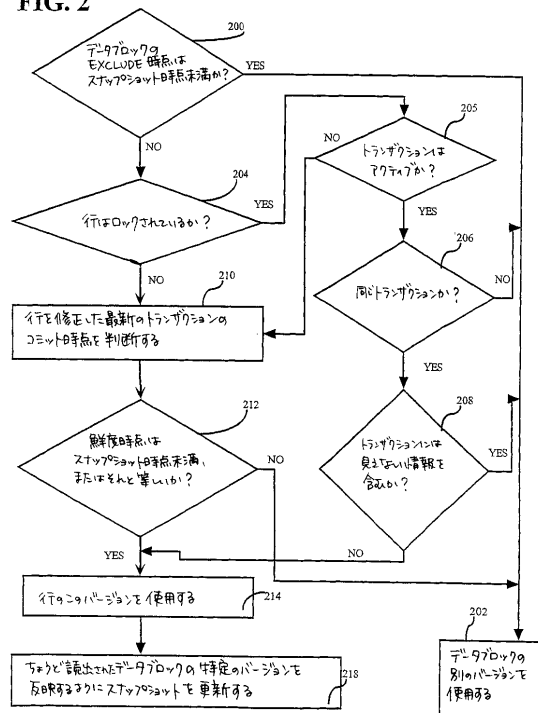
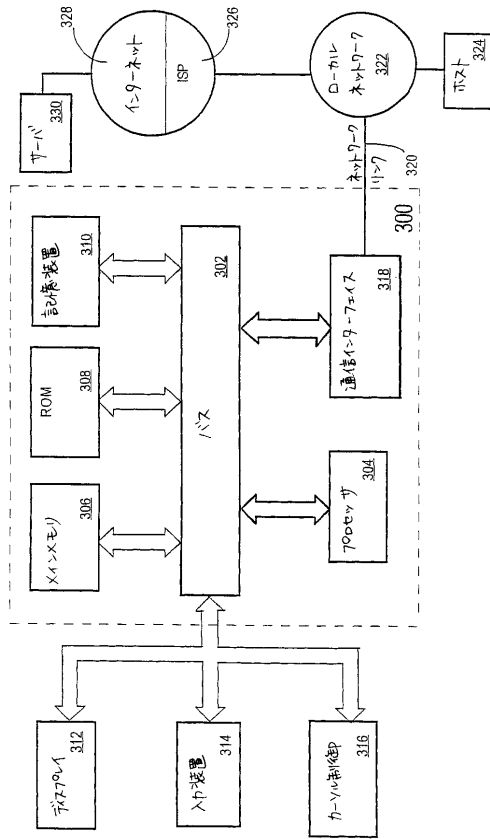


FIG. 3



フロントページの続き

(74)代理人 100098316

弁理士 野田 久登

(74)代理人 100109162

弁理士 酒井 將行

(72)発明者 ガネーシュ, アミット

アメリカ合衆国、9 5 1 2 9 カリフォルニア州、サン・ノゼ、ブルックグレン・ドライブ、1 4 2 6

(72)発明者 ベムーリ, シュリーニパス・エス

アメリカ合衆国、9 4 4 0 4 カリフォルニア州、フォスター・シティ、カタマラン・ストリート、6 3 5、ナンバー・3

(72)発明者 バンフォード, ロジャー・ジェイ

アメリカ合衆国、9 4 0 6 2 カリフォルニア州、ウッドサイド、マンザニータ・ウェイ、5 5 5

審査官 田川 泰宏

(56)参考文献 米国特許第0 6 1 9 2 3 7 7 (U S , B 1)

米国特許第0 5 8 0 6 0 7 6 (U S , A)

米国特許第0 5 8 7 3 0 9 8 (U S , A)

(58)調査した分野(Int.Cl. , D B 名)

G06F 12/00