5G－9

データベース指向ＯＳ ＸＥＲＯにおける永続的キャッシング技術

登内敏夫 加藤和彦 益田隆司

東京大学 理学部 情報科学科

1．はじめに

データベースを指向したオペレーティングシステムXEROでは，永続的記憶装置上のデータを、オブジェクト識別子（OID）によって一意に指定される永続オブジェクトとして管理している[加藤89a][成田90][脇田89]。この方式では、オブジェクトが他のオブジェクトのOIDをもつことで参照関係を実現している。

参照関係をもつオブジェクトをもちいる場合，参照関係を高速にたぐる技術が望まれる．またXEROでは、OIDをオブジェクトの物理位置とは独立な値としているために、OIDからオブジェクトの物理アドレスを得る効率の良い変換が望まれる。

本稿では、参照する側のオブジェクトに、参照される側のオブジェクトの複製やその物理アドレスを保持することにより、二次記憶上のオブジェクト間の参照関係をたぐるコストを軽減する方法について述べる。

2．アルゴリズムの概要

二次記憶上に２つの永続オブジェクトＡ，Ｂがあり、ＡからＢにリンクが張られていたとする（図１）。Ａの指している（リンクが張られている）オブジェクトＢの内容を得る場合を考える。通常は、Ａを二次記憶から読み出すことにより、Ａが指しているオブジェクトＢの識別子O_Bを得る。そののちに、O_Bを使ってＢを読み込む。

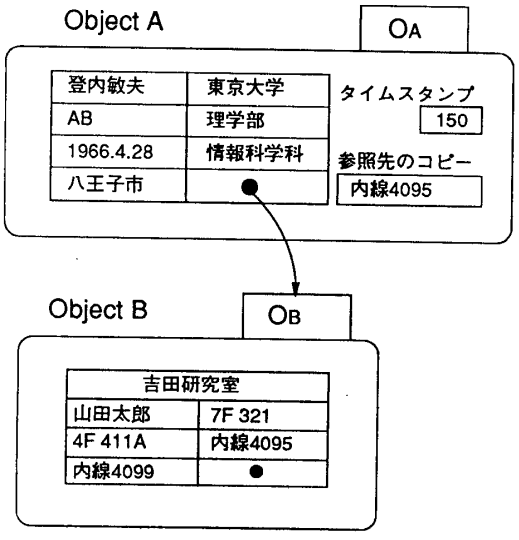

図１ 永続オブジェクト

これに対し、Ｂの複製をあらかじめオブジェクトＡと同じディスクページにもてば、Ｂを二次記憶から読み出すまでもなく、Ａをアクセスするだけでの値を得ることができる。

この複製を二次記憶（永続記憶）上にオブジェクトの値がキャッシュされているとみなせるので、永続キャッシングと呼ぶ[Kato89b]。

複製をもちいる場合は、オブジェクト自体の値とその複製の値との一貫性が問題となる。一貫性を保つためにオブジェクト本体が更新された時点で二次記憶上のすべての複製を更新する方法が考えられる。しかし、この方法では更新の負荷が重い。そこで、複製の更新を本体の更新時にまとめて行なうのでなく、それぞれの複製を参照する時点まで遅らせることにより更新時の負荷を減らす。つまり、オブジェクトの更新時に更新時刻を記録しておく。そして、複製を参照するときに、オブジェクトが最後に更新された時刻と、その複製が最後に更新された時刻を比較することで一貫性を検査する。もし、一貫性が破れていれば複製の更新を行なう。

各々のオブジェクトが最後に更新された時刻を記録するためＣＶＴ（Consistency Validity Table）という表を設ける。ＣＶＴは、高速にアクセスするために、主記憶上に置かれる。ただし、二次記憶空間上にある全オブジェクトと１対１対応できる巨大な表を主記憶上につくることは不可能である。そこで、異なるオブジェクトであっても、そのOIDに対してハッシュ関数を適用して同じハッシュ値を持つならば、同じＣＶＴの項目に対応させる。通常、オブジェクトが異なれば最後に更新された時刻は異なるが、そのときはその項目に対応するオブジェクトの中で更新時刻が最新のものを記録する。

3．並行実行環境での永続キャッシング

並列実行環境ではオブジェクト間の一貫性を保つことが重要である。ここでは永続的キャッシング技術で直列可能性を保つ方法を述べる．

方針としては2-フェーズ・ロックにより直列可能性を維持し，ロックの解除はトランザクションが終了するときに行なう。ロックの管理はロックマネージャーが一元的に行ない、ロックマネージャーがデッドロックの検出とトランザクションの再実行を行なう。

アルゴリズムを述べるにあたって次のことを仮定する．

- オブジェクトには一意なOIDが付いている。リンクは、参照しているオブジェクトが参照されているオブジェクトのOIDを持つことで実現される。
- ロックは占有ロックと共有ロックを用い、更新される可能性のあるオブジェクトには占有ロックをかけ、更新が行なわれないオブジェクトには共有ロックをかける。
- トランザクションを直列可能性の単位とする。
- ＣＶＴに対する読み書きは不可分命令である。

以下，順を追ってアルゴリズムを説明する．

Persistent Caching Technique of the XERO Operating System
Tonouchi TOSHIO, Kazuhiko KATO, and Takashi MASUDA
Dept. of Information Science, Univ. of Tokyo.

3．1．トランザクションの開始宣言

ユーザはトランザクションの開始を宣言せねばならない。これには、直列性を維持する単位をはっきりさせることと、ロールバックの戻り先を決定するという2つの意味がある。

トランザクションの開始時、私的な作業領域を用意する。占有ロックがかかったオブジェクトへの処理はこの領域においてなされる．これはロールバックのコストを下げるためである。作業領域の内容の二次記憶への掃き出しはトランザクションの終了時のコミット処理により行なう。

3．2．参照読み込み

2つのオブジェクトA，Bがあり、Aの指しているオブジェクトBをトランザクションTが得る場合を考える。ただし、Aの識別子をO_A、Bの識別子をO_Bとし、AにはBの複製B'とその複製が最後に更新された時刻$t_{B'}$が付随しているとする（図2）。この処理を行なう以前はO_Bはまだわかってない。

［オブジェクトAの読み込み］

まずTはAに対する必要なロックをロックマネージャーに要求する。ロックが認められたらAを二次記憶から読み込み、参照先であるオブジェクトBの識別子O_Bを得る。このとき、Bの複製B'とそのタイムスタンプ$t_{B'}$も同時に読み込む。

［正当性の検証］

TはBに対して必要なロックをロックマネージャーに要求する。ロックが認められればBに対応するＣＶＴの項目C_Bに記録されているタイムスタンプt_Bを得る。t_BはBが最後に更新された時間、$t_{B'}$はBの複製B'が最後に更新された時間を表わすので、$t_B \leq t_{B'}$ならば複製B'とその実体Bとの一貫性は保たれていることがわかる。このときはBをアクセスするまでもなくB'の値をBの値としてもちいてよいことが保証される。

［複製の更新］

一方、$t_B > t_{B'}$のときはB'はもはや古くなっている可能性があるので、TはBを直接読み込んで実際の値を得る。

このとき、B'と$t_{B'}$も新しい内容を反映しなくてはならない。この作業はAに対するすべてのロックが解除された時点でロックマネージャーが行なう。TはAのもつBの複製B'が書き換わるまで待つ必要がない。

3．3．書き込み処理

トランザクションTが識別子がすでにわかっているオブジェクトBに書き込み処理を行なうことを考える。このとき、Tはロックマネージャーに B に対する占有ロックを要求する。競合するロックがなければ，トランザクションT用の作業領域に書き込み処理を行ない，対応するＣＶＴの項目のタイムスタンプを現在の時刻に変更する。

3．4．トランザクションの終了

トランザクション終了はユーザが明示的に宣言する。このとき私的な作業領域に格納したオブジェクトの内容を二次記憶に反映させる（コミット処理）。続いて、トランザクションがオブジェクトにかけたロックの解除をロックマネージャーに要求する。最後に作業領域を解放する。

3．5．ロックマネージャー

ロックマネージャーはロック要求の受け付け、ロックの解除、デッドロックの検出を行なう。

［ロック要求受け付け］

ロックの要求を受けたら、競合するロックが存在しているかどうか調べる。ロックが競合しなければ、ロック情報に新たなロックを追加する。

［デッドロックの検出］

ロックが競合する場合、トランザクションを待ち状態にし、ロック待ち情報に加える。このとき、デッドロック状態に陥ってないかを検査する。

ロックマネージャーはロックに関するすべての情報を有している。この情報をもとに待ちグラフを作り、デッドロックを検出する。デッドロックが発見された場合、最後にロックを要求したトランザクションをロールバックさせる。

4．おわりに

今後、並行処理での評価をおこなう必要がある。将来の課題として、この方式を分散環境にも適応できるよう拡張することが挙げられる。

参考文献

[加藤89a]加藤，猪原，脇田，益田，"データベース処理を指向した分散オペレーティング・システムXEROの構想,"電子情報通信学会コンピュータシステム研究会 CPSY89-29,1989.

[Kato89b]K.Kato and T.Masuda,"Persistent Cashing: An Implementantation Technique for Complex Objects with Object Identity," Univ. of Tokyo, Dept. Information Science Technical Report 89-021, 1989.

[成田90]成田、加藤、益田、"データベース指向OS XEROにおける複合オブジェクト管理,"情報処理学会第４０回全国大会講演論文集、1990.

[脇田89]脇田、加藤、益田、"データベース処理を指向した分散OS XEROの永続オブジェクト管理,"情報処理学会第３８回全国大会講演論文集、1989.

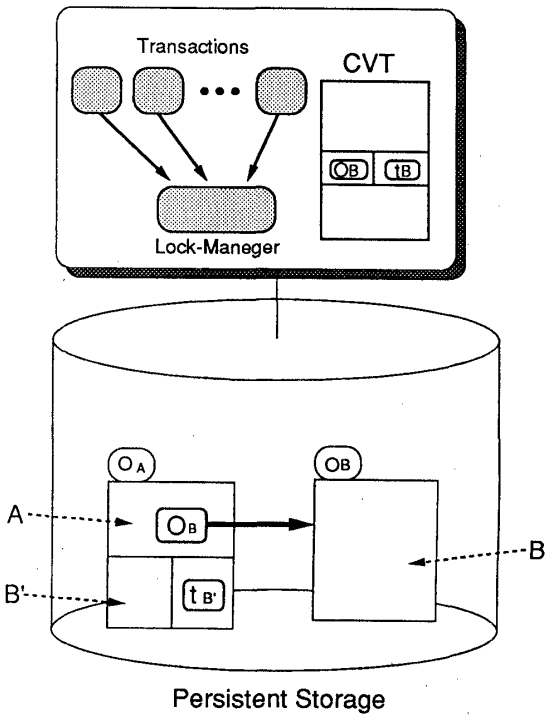

図 2 永続キャッシング-アルゴリズム