1H−6

バッファ管理に対するマルチバージョンロックの応用

林克己　三谷政昭　下雅意義徳　岡田敏治
(富士通㈱沼津工場)

1．はじめに

マルチバージョンロックプロトコル[1]を採用してトランザクション処理多重度の向上を計った商用DBMSが一般化しつつある．このプロトコルは，DBMSのハウスキーピング処理であるバッファ書き戻し処理とトランザクション処理間の逐次化にも適用できることを見いだした．これによって，トランザクションのレスポンスタイムの相当の改善が期待される．

2．従来のバッファの書き戻し処理方式

最も一般的に採用されているREDO/UNDOのためのログ取得にWAL(Write Ahead Log)を採用するトランザクションシステム[2]では，バッファの書き戻しとログの不揮発化の厳密な前後関係の成立が要請される．このため，従来，新たなログの発生を伴うバッファ内容の変更は，バッファの書き戻し処理開始から完了までの間には禁止されていた．この結果，直列化されたログの書き出し時間とバッファ書き戻し時間の合計が，書き戻しに続いて，このバッファに関わる処理を実行しようとするトランザクションのレスポンスタイムに加算されることになる．最も単純にバッファを二次記憶へ直接書き戻すならば10^2 msのオーダの書き戻し待ち時間は不可避である．

この関係を図1に示す．複数のトランザクションA，B，C …と，これらに共用されるバッファ領域(ページ 0〜7 からなる)，二次記憶上のデータベースからなる．ページ1の書き戻し経路を示す(A)は従来方式．(A1)(A2)の書き戻し経路は後述の新方式である．トランザクションA，B，C …がページ1をアクセスしている状態を示している．

従来方式は次のとおり．

∽∽∽∽∽∽∽∽∽∽従来の手順∽∽∽∽∽∽∽∽∽∽

ページ1のバッファ書き出し契機で（詳細説明は割愛），

(1) このページ1に関わるトランザクションA，B，C …からのログ収集及びページ1の変更を禁止する．

(2) 次にログを不揮発媒体に取得後，

(3) (A) の経路で二次記憶データベースへ書き戻す．

(4) 完了を待って，再びトランザクションからのページ1へのアクセスの再開を許可する．

∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽

ところで，バッファの書き戻しを無限に延期することによって，この様なタイミング の発生を抑止できるように思われるかもしれないけれども，特に無停止運転状態では現実的には不可能である．実際，システムクラッシュ やメディアフェーリャ 発生時の回復時間をリーズナブルな範囲に止めるために，無闇にAIを蓄積するわけにはいかない．ホットスポット でも適時書き戻し処理を実行し，差分ダンプ を取得しておくことによって各種のリカバリに関与するログ量を大きくしない運用が必須である．

これを改善するためにはログ，データベースの書き戻しの絶対時間を短縮することが考えられる．

実際，ログはデータベースに比べて少量であるので，バッテリーバックアップ のついたRAM を不揮発記憶媒体として採用することで，書き出し時間をμsのオーダ に短縮することが行われている．

一方，比較的安価な磁気ディスク等を採用せざる得ない大容量のデータベースでは，キャッシュメモリを採用するなどの間接的な方法が採用される．しかし，この種の方式ではコスト 高になったり，付帯する複雑な制御が信頼性を損ねる原因となる場合があった．

3．マルチバージョンロックプロトコルを応用した新方式

バッファ書き戻しの際，WAL の原理に従って，ログ取得が先行し，このログと対になるバッファのスナップショットが正しく二次記憶に反映することが目標である．

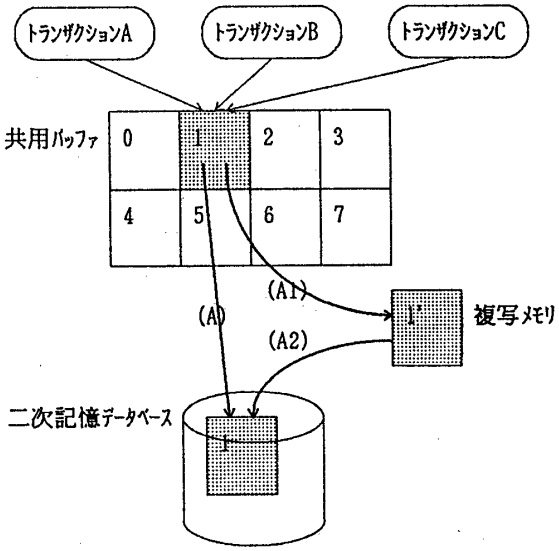

図1．共用バッファの書き戻し方式

An Application of Multi Version Lock for Buffer Management
Katzumi HAYASHI, Masaaki MITANI, Yoshinori SHIMOGAI, Toshiharu OKADA*
FUJITSU Ltd.

再び図1を用いて説明する．(A) の経路でなく，(A1)，(A2)の経路が利用される．

新方式では次のとおり．

∽∽∽∽∽∽∽∽∽∽∽新方式∽∽∽∽∽∽∽∽∽∽∽∽∽

ページ1のバッファ書き出し契機で（詳細説明は割愛），

(1) このページ1に関わるトランザクションA，B，C …からのログ収集及びページ1の変更を禁止する．

(2) 次にログを不揮発媒体に取得後，

(3) (A1)の経路で複メモリ にページ1のコピー が取得される．

(4) 完了を待って，再びトランザクションからのページ1へのアクセスの再開を許可する．ただし，書き戻し(A2)完了までは再び書き戻しシーケンス は開始されない．

(4)' トランザクションのアクセスと並行して，(A2)の経路で二次記憶データベースへ書き戻す．

∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽∽

新方式では最も時間を要する(4) のフェーズが(4)' と並列化されるので，トランザクションのレスポンスタイムへの影響が緩和されることになる．DBMSにとっての書き戻しのための複写メモリ とオリジナル の共用バッファ間でマルチバージョンの状態を構成している．複写メモリ のバージョン は取得済ログと完全に一致しており，書き戻し完了まで誰もアクセスしないので，実効的に同期して(A) の経路で書き戻したのと同じ状況と成っている．これはシステム的にはキャッシュ として主記憶を利用したものと解釈してもよい．キャッシュ に比べて遙に直接的であり単純である．

従来方式におけるトランザクション処理時間の遷移と，新方式におけるトランザクション処理時間の遷移を図2に示す．

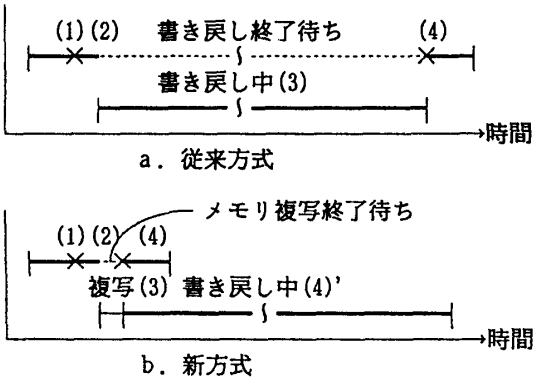

図2．トランザクション処理時間の遷移

4．実施例

最後に具体的な適用例で新方式の有効性を実証する．

データベース自身の領域管理データ を格納するビットマップデータ を保持するページ はホットスポット の一つの典型的な例である．データ 量の増減に応じて頻繁にアクセスされる．

図3の構成を示す．頻繁にアクセスされるため共用バッファ上に領域管理のためのビットマップとして実質的に常駐している．各ビット は一つの連続領域の使用，未使用を表示している．この原簿は二次記憶上のデータベースとして保持される．

ビット 単位でトランザクションからアクセスされトランザクションに同期して切りだし，返却が実行される．

従来方式で，ビットマップを二次記憶データベースに書き戻している期間，複数のトランザクションが待ち状態になるとすると非常に多くのトランザクションが同期処理に参与することになり，これに伴ってコンボイ現象による連鎖的な停止トランザクションへの影響を考えると性能劣化の一大要因となる．

高速のログ取得と，新方式の採用によって書き戻しによる影響を大幅に解消できる．

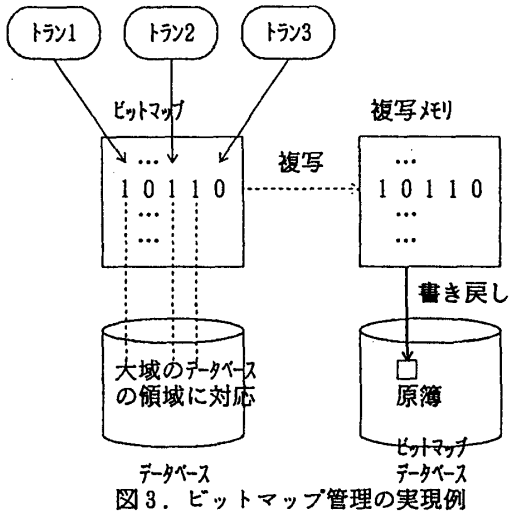

図3．ビットマップ管理の実現例

5．おわりに

マルチバージョンロックプロトコルを応用したバッファ書き戻し方式の実現可能性と有効性について述べた．

キャッシング方式との比較，主記憶階層が深い場合への適用等今後更に検討を継続して行く．

【参考文献】

[1] C.Papadimitriou, "The Theory of Database Concurrency Control", Computer Science Press, 1986.

[2] P.A.Bernstein et al.,"Concurrecy Control and Recovery in Database Systems", Addison-Wesley, 1987.

本稿の標題はプログラムと相違しています．