

## オンラインとバッチトランザクションの 混在実行のためのバッファ管理法

4 R-1

佐藤 哲司 平野 泰宏 井上 潮  
NTT 情報通信網研究所

### 1. はじめに

ネットワーク管理等の24時間サービスを提供するデータベース応用では、従来バッチで処理していた統計処理等をオンライン処理を中断することなく実行できることが望まれる。本稿では、オンラインとバッチの混在実行を狙いとして、トランザクションの持つアクセス特性を反映した優先度付きバッファ管理法を提案し、シミュレーション結果を示す。

### 2. 従来のバッファ管理の問題点

ネットワーク管理等のデータベースシステムでは、予め作成されたインデックスを用いて処理対象とするデータ(行)を高速にアクセスして検索・更新するオンライントランザクションと、テーブルのフルスキャンを伴う複雑な条件での検索やジョイン、ソート等のバッチトランザクションを混在して処理することが要求される。オンライントランザクションは、複数ユーザが同時に実行できる高い並行性とユーザをほとんど待たせない高い応答性が要求され、これらの性能が低下しないようにバッチを混在実行しなければならない。

このようなシステムは、一般に大規模でかつ高い信頼性を要求されることから、安定な記憶媒体である磁気ディスク装置にデータベースを格納しており、頻繁に使用されるデータを主記憶上にバッファリングすることで処理性能を向上させている。一般に、オンライントランザクションは、データベース中のごく一部のデータを頻繁にアクセスするためバッファリング効果が高いが、バッチトランザクションは、バッファ容量を超える大量データを処理するためバッファリング効果は得られにくい。また、大量の中間結果を一時的に保持するジョインやソートは、中間結果を個別に格納管理するよりも共有バッファ上に格納する方が、プログラム作成の容易性、並列処理への適用性、主記憶の利用効率を向上できる。

オンラインとバッチの混在実行におけるバッファ管理では、バッチの実行によってホットスポットページがバッファから追い出されないようにすると同時に、大量の中間結果であっても近い将来にアクセスされるデータは高い確率でバッファリングしなければならない。しかし、従来のバッファ管理法では、過去のアクセス経緯、たとえばアクセス順序や頻度

などの間接的な統計情報を用いて追出し制御を行なっているため、テーブルスキャンのように将来使用される可能性が低いデータと、バッチの中間結果のように近い将来に必ず使用されるデータを識別できず、誤った追出しや不当な滞留が避けられなかった。

### 3. 優先度付きバッファ管理法

#### 3.1 アルゴリズム

バッファ割当て法は、バッファを一括管理するグローバル管理法、トランザクション毎に割り当てるローカル管理法、データ属性に基づいて割り当てるタイプ別管理法に分類できる[1]。ここでは、以下の観点から、グローバル管理法を基本として、トランザクションのアクセス特性とデータ属性とからバッファ置換の優先度を決定する優先度付きバッファ管理法を提案する。なお、文献[2]では、提案法に基づくバッファ管理の実現法と並列処理方法を示した。

(1) 複数トランザクションを混在実行する環境では、トランザクション毎にバッファを割り当てるローカル管理法は、バッファの分割損が大きく割当てオーバヘッドも無視できない。インデックスやテーブル等のデータ属性でバッファを割り当てるタイプ別管理法では、オンラインとバッチの負荷変動に対する最適な割当ては困難である。データ属性に応じた優先度を与える優先度付きのグローバル管理法とすることで、負荷変動に追従した動的なバッファ割当てを容易に行なえる。

(2) トランザクション実行部がアクセスするデータの属性とアクセス特性に基づく優先度をバッファ管理部に指示することで、従来法の誤った追出しを回避する。トランザクション実行部は、インデックスやテーブルのページ単位アクセス(オンライントランザクション)には高い優先度を、テーブルの連続ページアクセス(バッチトランザクション)には低い優先度を与えてバッファ管理部にアクセス要求を出す。

(3) バッチが生成する中間結果は、図1に示すようにテーブルスキャンの進行に伴って選択結果を格納するために容量が増大(生成過程)し、その後ソートやジョイン準備(編成過程)を行ない、結果の出力を行なって消滅する(出力過程)。この中間結果は近い将来に必ず使用されるため、トランザクション実行

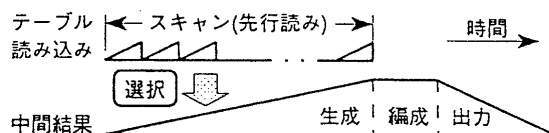


図1. バッチトランザクションのバッファ消費

部は高い優先度を与えてバッファに保持する。一方、テーブルスキャンで先行読み込みしたページは、選択処理が終了するまでバッファから追い出せない。この操作もバッファの優先度制御で行なえる。

(4) 更新ページを追い出す際は、ディスクへの書き戻しが必要となる。この書き戻し操作は、長時間を要するのでトランザクション処理とは非同期に、あるいは、エレベータアルゴリズム等を用いて複数ページを一括書き戻すことが考えられる。優先度付きのバッファ管理であれば、書き戻し中のページを最大優先度として他の操作を禁止する。

以上示したように、トランザクション実行部がデータ属性、アクセス特性に基づいて優先度を指定することで、オンラインとバッチの混在実行に適したバッファ管理を行なえる。

### 3.2 シミュレーション評価

複数のオンライントランザクションとバッチトランザクションの混在実行をモデル化し、上記提案法の(1),(2)を評価した。評価モデルとパラメータを図2に示す。複数のユーザが発行するオンライントランザクションは、インデックスとテーブルを正規分布に従ってアクセスする。バッチはランダムに与えられた初期ページから連続ページを順次アクセスするものとした。評価は、(a)優先度制御なし、(b)データ属性による優先度、(c)データ属性とアクセス特性を統合した優先度の3種類で行なった。(b)ではインデックスとテーブルの2段階の優先度を、(c)ではインデックスの単一ページアクセス、テーブルの単一ページアクセス、テーブルの連続ページアクセスの3段階の優先度を用いた。いずれの方式でも同一優先度内はLRUリストで管理する。

バッファ容量を変えて評価したオンライントランザクションのバッファヒット率を図3に示す。バッチを混在実行しない場合は、いずれのバッファ容量、管理方法でも、ユーザ数に依らず一定のヒット率が得られている。バッチを混在した場合には、優先度制御を行なわない(a)が最もヒット率が低下する。データ属性とアクセス特性を考慮した(c)では、バッチを実行した場合でもオンラインのヒット率は低下していない。これは、バッファ容量が500ページの場合にはインデックスのホットスポットが、2000ページの場合にはインデックスとテーブルのホットスポットページがバッファリングされ、バッチを実行してもホットスポットページが追い出されないからと考えられる。なおバッチは、スキャンしたページがバッファされていたとしても同一ページを再度アクセスする可能性は低いため、いずれのバッファ管理方式でも差は出なかった。

### 4. まとめ

本稿では、オンラインとバッチの混在実行のための優先度付きバッファ管理方法を示した。今後は、中間結果の格納を含めた評価、並行処理制御との組

合せによる更新を含めたバッファ管理法を検討する予定である。

### 参考文献

- [1] W. Effelsberg and T. Haerder: "Principles of Database Buffer Management", ACM Trans. on Database Systems, Vol.9, No.4, pp. 560-595, Dec. 1984
- [2] 佐藤, 片岡 他: メモリ共有型マルチプロセッサにおけるオンラインとバッチ処理の混在した実行制御法, データベースシステム研究会 89-18, pp.141-150, 1992. 7

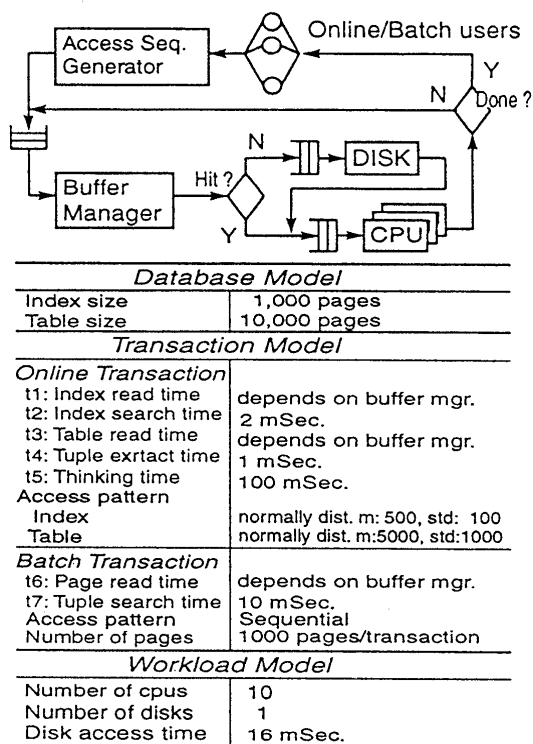


図2. シミュレーションモデル

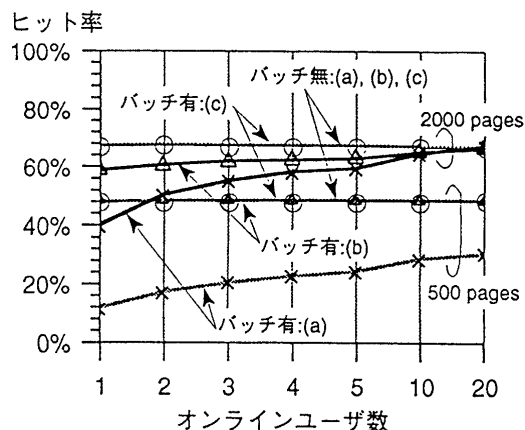


図2. オンラインのバッファヒット率