分散オブジェクトへの動的な問い合わせスケジューリング

5AA－9

江 允† 牧之内 顕文

†倉敷芸術科学大学ソフトウェア学科　九州大学大学院システム情報科学研究科

†jiang@soft.kusa.ac.jp,　akifumi@db.is.kyushu-ac.jp

1. はじめに

近年、計算機の低価格化と高速ネットワークの普及とに伴い、複数台のワークステーションをLANで接続した分散環境が広く使われつつあり、分散されている計算機資源をよく利用しようとする研究が盛んに行われている。一方、従来の問い合わせ処理技術では、このようなネットワーク分散環境における大量のオブジェクトに対する並列処理機能についてはあまり考えられていなかったため、我々は、分散環境におけるオブジェクト指向データベース(OODB)に対して並列問い合わせ処理を研究している。

本論文では、問い合わせ処理を行うための動的な並列問い合わせスケジューリングを提案し、そのアルゴリズムの設計と実現方法について述べる。

2. 分散並列問い合わせ処理

問い合わせ処理とは、データベース管理システムが、ユーザーによって与えられた問い合わせを処理することである。問い合わせ処理のコストを最小にするために、問い合わせ分解、問い合わせの最適化、問い合わせスケジューリングなど一連の操作を行わなければならない。OODBにおいて問い合わせ処理に関する基本概念は関係データベースとほぼ同じであるが、オブジェクト指向データモデルと関係データモデルとの基本的な違いのため、問い合わせ処理が異なる[2][1]。我々は、分散環境における並列問い合わせ処理のため、以下のような3つ方針で研究をしている。

- ワークステーションクラスタにおけるOODB

  大量なおかつ複雑なデータ構造をもつOODBの膨大なデータを１つサイトに置くだけではなく、ワークステーションクラスタ環境上で複数のサイトに分散配置する。

- 分散共有永続オブジェクト空間

  分散・並列処理を容易にするために、分散仮想共有永続オブジェクト空間を利用する。この分散仮想共有永続オブジェクト空間は九州大学で開発されているINADA分散ストレージシステムで提供されている[3]。

- 全域性並列問い合わせ処理

  複数のサイトに分散されている全てのオブジェクトに対して、問い合わせ処理を異なるサイトで並列的に行なう。

3. 並列問い合わせスケジューリング

問合わせのスケジューリングとは、データ処理の順序、データへのアクセス経路、データベース処理アルゴリズムの選択および中間テーブル構造などを決定することである。分散環境における並列問い合わせスケジューリングの基本の機能として、クエリー木における各結点の問い合わせ操作を複数のサイトで並列的に実行させることである。このため、我々は、以下のことを考えている。

[クエリー木分解の原子化]

まず、並列問い合わせスケジューリングに適応させるために、問い合わせ分解する時に、以下のような原子化されたクレリー木を求める必要がある。

(1) *Union* 操作結点での部分木が独立している。

(2) 各結点は単一な操作しかもっていない。

(3) 結点毎でのデータの処理量が適合である。

(4) 葉結点における操作は独立であり、なおかつ直ちに実行可能な状態になっている。

次に、並列問い合わせスケジューリングに使われている２つの集合の定義を与える必要がある。

[定義 1]: 問い合わせ木上の依存関係集合 $[qt]_{DR}$ クエリー木 qt 上の依存関係集合 $[qt]_{DR}$ は以下のように定義されている。

$$[qt]_{DR} = \{< x,y > | (\forall x)(\forall y)(x \in OP(qt) \wedge y \in OP(qt) \wedge x \; is \; directly \; dependent \; on \; y)\}$$

ここで、OP(qt) は問い合わせ木 qt 上での操作集合である。x と y はOP(qt) の要素である。<x,y> は問い合わせ木 qt 上での直接的な親子依存関係（y 結点は x の親結点である）をもつ操作ペアである。従って、$[qt]_{DR}$ は要素<x,y>からなる親子関係をもつ操作ペアの集合である。

A Dynamic Query Scheduling for Distributed Objects, Yun JIANG†, Akifumi, MAKINOUCHI, †Kurashiki University of Science and the Arts, Graduate School of Information Science and Electrical Engineering, Kyushu University

[定義 2]: スケジュールに編入可集合 $[qt]_{SS}$

問い合わせ木上の個々の操作を見て、直ちに実施できる操作があれば、この操作を「問合せスケジュールに編入可」(immediate-schedulable operation) 操作という。これらの操作の集まりは「スケジュールに編入可集合 $[qt]_{SS}$」という。

$$[qt]_{SS} = \{x | (\forall x)(x \in OP(qt) \wedge x \ is \ an \ immediateschedulable \ operation)\}$$

常に、スケジューリングの実行によって、直ちに実施できる操作が変わってくるので、集合 $[qt]_{SS}$ に属する要素も時間的に変わってくる。

並列問い合わせスケジューリングアルゴリズムは以下のようである。

[アルゴリズム 1]: 並列問い合わせスケジューリング

(1) To define local variables: $[qt]_{DR}$, $OP(qt)$、$[qt]_{SS}$; Initially $[qt]_{SS} = \phi$;

(2) To compute $[qt]_{SS}$ with respect to the current $[qt]_{DR}$;
For each operation $op_i \in OP(qt)$, if op_i just only appears on the left-hand side of tuples $< x, y > \in [qt]_{DR}$, then $[qt]_{SS} = [qt]_{SS} \cup \{op_i\}$;

(3) For each $x \in [qt]_{SS}$, schedule a process of a networked cluster for executing the operation x; at the same time, $[qt]_{SS} = [qt]_{SS} - \{x\}$ is needed;

(4) When the operation x is finished, process will do:

$[qt]_{DR} = [qt]_{DR} - \{< z, y > | < z, y > \in [qt]_{DR} \wedge z = x\}$;

if y just only appears on the left-hand side of tuples (e.g $< y, h >$) $\in [qt]_{DR}$, then $[qt]_{SS} = [qt]_{SS} \cup \{y\}$;

$OP(qt) = OP(qt) - \{x\}$;

(5) Any processor can deal with the opeartion $x \in [qt]_{SS}$, only if $[qt]_{SS} \neq \phi$;

(6) If $[qt]_{SS} = \phi$ and $[qt]_{DR} \neq \phi$,
any processor has to wait until a $[qt]_{SS} \neq \phi$, so as to fetch next operation;

(7) When $[qt]_{SS} = \phi$, $[qt]_{DR} = \phi$, and $OP(qt) = \phi$, all perations of a query tree have been performed and the scheduling algorithm comes to the end.

このスケジューリングによって、相互に関連のない葉結点での問い合わせ操作が直ちに実行できるので、全ての葉結点での操作を並列的に行うことになる。

4. 動的なクエリー木情報表

並列問い合わせスケジュールを簡単に実行するために、一つの鍵として、動的なクエリー木情報表というメカニズムを導入した。動的なクエリー木情報表は、問い合わせ木を処理する途中の状態を表記し、クエリー木と一対一に設けられて、以下のように設計されている。

- 動的なクエリー木情報表はネットワーククラスタ全域のプロセスに呼び出され書き込まれることができる。しかも、各プロセスがローカル的な情報表を持つような感じで使われる。
- 動的なクエリー木情報表は分散共有永続オブジェクト空間上で設けられるので、排他制御をサポートされている。
- 新生の葉結点を待つことはプロセス間の同期を取ることと統合的に考えられるので、並列問い合わせ処理のための同期機構を追加する必要がない。

5. おわりに

本論文では、ネットワーク分散環境上での分散オブジェクトに対して並列問い合わせスケジューリングの提案、アルゴリズムの設計と実現方法について述べた。OODB の並列問い合わせ処理システムの各段階の研究は今後の課題である。

参考文献

[1] A. J. Blakeley, J. W. McKenna, and G. Graefe, "Experiences Building the Open OODB Query Optimizer ", *Proc. of ACM SIGMOD'93 Conference on Management of Data*, May 1993.

[2] D. D. Straube and M. T. Ozsu, "Queries and Query Processing in Object-Oriented Database Systems", *ACM Transactions on Information Systems*, Vol.8, No.4 pp. 387-430, October 1990.

[3] G. Bai,and A. Makinouchi, "WAKASHI/D: a distributed paged-object server for storage management of new generation databases," *Proc. of the International Symposium on ADTI*, pp.137-144, October. 1994.