分散共有メモリを用いたデータベース処理の評価

7D－8

今井英貴　斎藤章一　中村 素典　大久保 英嗣
立命館大学理工学部情報学科

1　はじめに

分散データベースシステムや分散ファイルシステムでは，利用者に対して分散を意識させないことが重要であり，分散されたオブジェクトに対してさまざまな透過性（transparency）に関わる機能を提供する必要がある．しかし，このことは開発者にとって大きな負担となるものである．そこで我々は，分散されたオブジェクトへのユーザのインタフェースを従来の集中型システムの自然な拡張とするために，分散共有オブジェクトリポジトリ DSO-Base の研究を行っている．

DSO-Base は，分散共有メモリ（DSM）を中核に置き，分散オブジェクトの統合管理を行なうシステムである．分散データベースや分散ファイルシステムなどに必要な，データオブジェクトに対する位置（location），移動（migration），分割（fragmentation），重複（replication），障害（failure）に対する透過性の機能を提供することを目的とするシステムである．システム構築のための予備検討として，今回，DSM を用いてデータベース処理に関する実験を行なった．本稿では，この実験の結果を示し，DSO-Base の構築に向けての考察を述べる．

2　実験に用いた DSM の概要

本実験で用いた DSM サーバ [1] は，Mach オペレーティングシステムの外部ページング機能を利用したもので，そのページサイズは４Ｋバイトである．書き込みは，write-invalidate プロトコルによって行われ，分散メモリ間の一貫性制御には，sequential consistency を採用している．

A Performance Evaluation of Database Processing by Distributed Shared Memory
Hideki Imai, Shoichi Saito, Motonori Nakamura, Eiji Okubo
Department of Computer Science, Ritsumeikan University

3　実験結果

3.1　オブジェクトの配置

分散共有メモリ上のオブジェクトの配置場所によるアクセス時間への影響を調べるための実験を行った．それぞれ，分散共有メモリ上に配置されているローカルのメモリ上，ローカルのディスク上，リモートのメモリ上のデータについて選択率１％のセレクト演算を行い，その処理時間を測定した．実験に用いたデータは，属性が一つでランダムな並びのリレーションであり，データの１要素は８バイトである．実験結果を図１に示す．実験に用いたマシンは，PC-AT 互換機，CPU は Intel486DX2 66MHz，OS は Mach3.0 で，メモリは 16M バイトである．

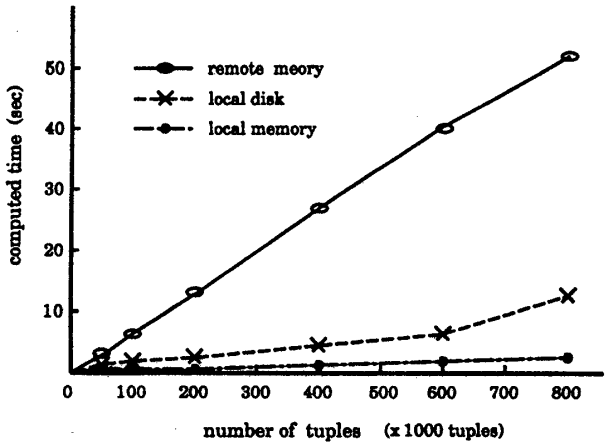

図 1: オブジェクトの配置と処理時間の関係

ローカルのディスク及びメモリ上にデータを配置した場合の処理時間とリモートの処理時間の差は，分散共有メモリがリモートデータにアクセスするときに発生するネットワークワイドなページフォルトと外部ページングのコストの大きさを表している．また，ローカルメモリ上のデータの処理時間は，ディスク上のデータの処理時間に比べて短くなっており，データのキャッシングが有効であることが確

認できた.

DSO-Base を効率の良いシステムにするには, 分散オブジェクトの配置, 特にオブジェクトの重複や分割について, 以上の実験結果をよく考慮して設計する必要がある. 例えば, あるホストが, 同一のリモートオブジェクトに頻繁にアクセスすることがわかった場合, 動的にそのオジェクトを重複して持つことで効率化を図ることが可能となる.

3.2 Wisconsin Benchmark を用いた評価

分散環境において, データベースの並列演算を行なった際に DSM の一貫性制御やネットワークワイドな外部ページングによるデータ移動のコストが, 処理効率にどのような影響を及ぼすのかを調査するために, セレクト演算とジョイン演算の実験を行なった. 評価には Wisconsin Benchmark の演算の中から 1%select と JoinABprime を用いている. それぞれのリレーションは各ホスト上にタプルワイズに等分割されており, 演算スケジュール表, ジョイン演算のハッシュバケットや演算結果とともに DSM 上に配置されている. 1 つのリレーションの属性の数は 13 個で, データの 1 要素は 8 バイトである. 1%select と, JoinABprime の処理時間を図 2 に示し, 速度向上を図 3 に示す.

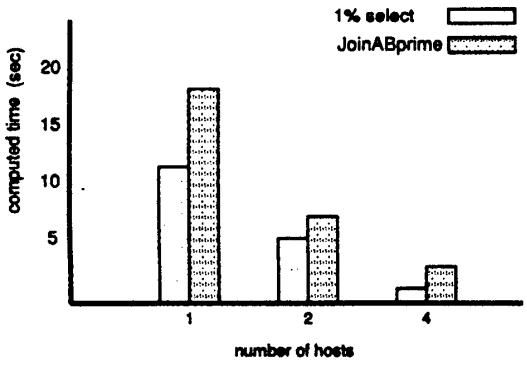

図 2: 演算処理時間

(1)1%select

ランダムな並びの 200,000 タプルのデータに対して選択率 1 %のセレクト演算を行う. 速度向上が線形よりも良くなっているのは, 1 ホスト当たりのデータ量が大きいほど頻繁にページイン, ページアウトを起こし, それが処理時間を増加させているからである. この結果から, 並列処理による演算時間

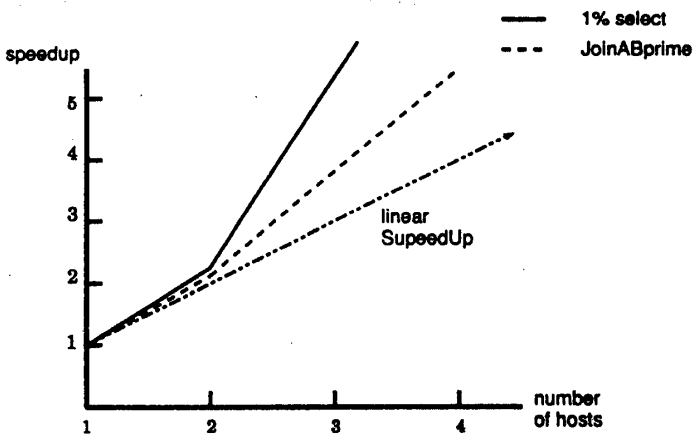

図 3: 速度向上

の縮小に加えて, 分散配置によるディスク I/O コストの大幅な削減が期待できることが分かる.

(2)JoinABprime

200,000 タプルのリレーション A と 20,000 タプルの Bprime をジョインする. ジョインは, 各リレーションを Bprime の要素数個のハッシュバケットに分割し, それをホスト台数で等分割して処理するアルゴリズムを用いている. 速度向上は, 1%selectA 程ではない. これは, ジョイン演算の場合は, ネットワークワイドなデータ移動が避けられないので, そのコストが影響しているものと思われる.

以上の結果から, 分散処理は, 演算の並列化による負荷分散の他に, I/O 処理の効率化にも有効であることが分かった.

4 おわりに

本稿では, DSO-Base の設計を行う前段階として, オブジェクトの配置場所に関する実験と演算を分散処理した場合の実験を行った. 2 つの実験結果から, リモートのオブジェクトへのアクセスコストの大きさと, 分散処理による I/O コストの大幅な削減を確認できた. 今後は, DSO-Base の基本設計を完成させ, 実装を進めていく予定である.

参考文献

[1] 斎藤彰一：分散共有メモリサーバの構成と大規模分散並列処理への応用. 立命館大学情報工学科修士論文 (1994).