6F－1

共有メモリ型計算機上での トランスポーズドファイルを用いた 並列関係問合せ処理の実装方式とその評価

武藤 精吾　田村 孝之　中野 美由紀　喜連川 優
東京大学 生産技術研究所

1　はじめに

意思決定支援システムでは大容量のデータに対して非定型問合せ処理が行なわれるため、膨大な処理時間を要する。非定型問合せ処理では事前にアクセスパターンを予測することが困難であるため、大容量のデータを扱うにもかかわらず索引を十分に利用することができず、ディスク入出力コストが高い。一方、CPUコストの方はCPUの著しい性能向上に加え、並列計算機の実用化などにより並列処理による高速化が可能となったため、ディスク入出力コストにくらべ低くなっている。トランスポーズドファイルを用いれば、CPUコストは多少増加してもディスク入出力コストを減らすことことができるため性能向上が期待できる。そこで実際に共有メモリ型の並列計算機上において問合せの処理系を実装し、性能評価を行うことによりトランスポーズドファイルの有効性を明らかにする。

2　トランスポーズドファイル

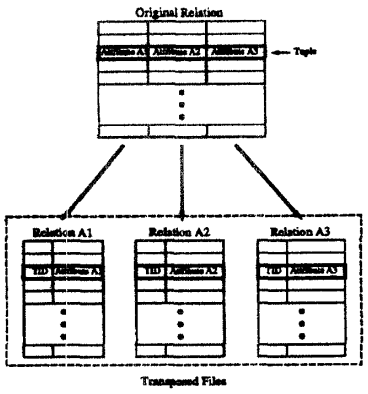

図 1: トランスポーズドファイル

トランスポーズドファイルとは図1に示すようにリレーションをアトリビュートごとに分割したものである。これによりアトリビュート単位でデータを読み出すことが可能となるため、問合せに必要とされないアトリビュートへのアクセスを避けることができ、ディスク入出力コストを減少させることができる [1]。

トランスポーズドファイルでは分割前のデータ間のつながりを保つためにタプルID(TID)と呼ばれるアトリビュートを新たに付与する。実際の処理ではディスクからデータを読み出したあとにこのTIDをもとにしてアトリビュート同士の結合処理を行い、リレーションを再構成する。そのため、CPUコストは増大することになる。

また、トランスポーズドファイルの変形としてBATと名付けた2項関係を用い、オブジェクト指向データベースによるGISを対象とした研究もある [2]。本研究では関係データベースの意思決定支援システムの高速化を目的としている。

3　実装方式

問合せ処理では主に射影演算、選択演算、結合演算、集約演算がよく用いられる。このうち射影演算、選択演算の実装は単純であるが、結合演算に関しては処理負荷の高い演算であるため、ネストループ方式、ソートマージ方式、ハッシュ方式などさまざまな実装方式が研究されている。今回はその中でもとりわけ高い性能を示しているハッシュ結合演算方式を用いる。なお、集約演算の実装は今回は行っていない。

ハッシュ結合演算方式は、一方のリレーションの各タプルに対して結合演算の対象となるアトリビュートにハッシュ関数を適用し、ハッシュテーブルを生成するビルドフェイズと、他方のリレーションに対して同様のハッシュ関数を用いてハッシュテーブルを検索し、結合処理を行なうプローブフェイズからなる。

今回の実装ではHP社の分散共有メモリ型並列計算機Exemplar SPP1600の1ノードを共有メモリ型並列計算機として用いる。CPUはPA-RISC(120MHz)8台、メモリ(512MB)、ディスク(1GB)1台からなる。メモリの一部をディスク入出力のためのリードバッファ、ライトバッファに割り当て、残りの部分をハッシュテーブルに用いる。

ビルドフェイズにおいてタプルをハッシュテーブルに挿入する操作では、異なるCPU同士による同一エントリへの同時書き込みを防ぐため、排他的に行なわなければならない。また、バッファへのアクセスも排他制御が必要である。しかし、それ以外の大部分の操作は並列に

Implementation and Evaluation of Parallel Relational Query Processing Using Transposed Files on Shared Nothing Multiprocessors

Seigo Muto, Takayuki Tamura, Miyuki Nakano and Masaru Kitsuregawa
Institute of Industrial Science, University of Tokyo
Roppongi 7-22-1, Minato, Tokyo, 106 Japan

行なうことが可能である。

ハッシュ結合演算方式における処理はディスクとの読み書きを管理する入出力処理とCPUでハッシュ関数の適用やハッシュテーブルの検索、タプルの比較などを行う結合処理の2つに大きく分けることができる。今回はそれぞれの処理ごとに独立したプロセスを用いて実装し、入出力プロセスを1台のCPUに割り当て、残りの各CPUに結合プロセスを割り当てることとした。

4 性能評価

4.1 射影演算のみを行なう場合の性能評価

最初の性能評価として、1つのリレーションに対して射影演算のみを行なう場合の測定を行なった。図2に射影されるアトリビュート数の変化に対する実行時間の様子を示す。通常の手法を用いた場合には、射影されるアトリビュート数に関係なくリレーション全体がディスクから読み込まれるため、実行時間は常に一定の値を示している。トランスポーズドファイルを用いた場合には射影されるアトリビュート数の増加にともなってディスク入出力時間も増加しており、また、結合演算数も増加するためにCPUコストも増加していることがわかる。しかしながら、全体としての処理時間は、射影されるアトリビュート数が全体のアトリビュート数の一部だけである場合には、トランスポーズドファイルを用いた場合の方が優れた性能を示していることがわかる。

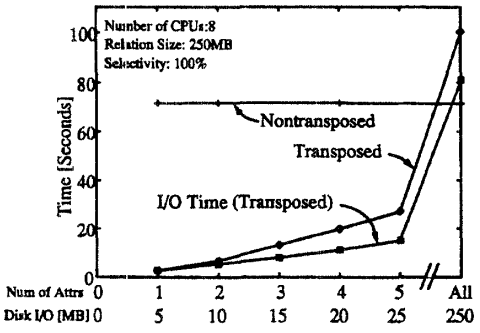

図 2: 射影アトリビュート数に対する射影演算の実行時間

4.2 TPCベンチマークDによる性能評価

続いて意思決定支援システム用のベンチマークであるTPCベンチマークDを用いて性能評価を行った[3]。全部で17の問合せが提供されており、ここではそのうちの問合せ3(スケールファクタ0.5)の測定結果を示す。選択演算、結合演算に用いられるアトリビュートのデータ型にはすべて整数を用いた。表1に問合せ3においてトランスポーズドファイルを用いることによるディスク入出力量と結合演算数の変化を示す。ディスク入出力量は減少し、結合演算数は増加しているのがわかる。

ディスク入出力量		結合演算数	
NTP	TP	NTP	TP
515MB	139MB	2	9

表 1: ディスク入出力量と結合演算数の変化

図3に問合せ3において結合プロセス数を変化させたときの実行時間の様子を示す。図3から通常の手法を用いた場合では結合演算プロセス数に関係なく実行時間は一定であり、入出力バウンドとなっていることがわかる。トランスポーズドファイルを用いた場合にはプロセス数の増加にともない、実行時間が減少しており、CPUバウンドとなっている。結合プロセス数が1つの場合には通常の手法を用いた場合の方が実行時間が短いが、2つ以上になるとトランスポーズドファイルを用いた場合の方が優れた性能を示していることがわかる。

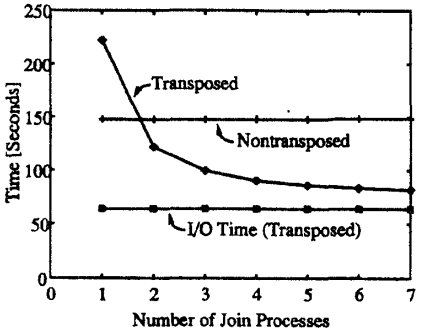

図 3: 結合プロセス数の変化に対する実行時間 (TPC-D 問合せ 3)

5 まとめ

トランスポーズドファイルを用いた問合せ処理について、共有メモリ型の並列計算機上に処理系を実装し、性能評価を行った。結合演算処理を並列に行うことにより、トランスポーズドファイルの有効性を引き出すことができることを明らかにした。

参考文献

[1] D. S. Batory, "On Searching Transposed Files", ACM Transactions on Database Systems, Vol. 4, No. 4, December 1979, pp. 531-544.

[2] P. Boncz and M. L. Kersten, "Monet: An impressionist sketch of an advanced database system", Proceedings of IEEE BIWIT workshop, San Sebastian Spain, July 1995.

[3] "TPC Benchmark™ D (Decision Support) Standard Specification Revision 1.1", Transaction Processing Performance Council (TPC), San Jose, CA 95112, 19 December, 1995.