

並列パイプライン方式データベースマシンの性能評価

5H-6

柿本篤巳* 津田和幸* 平川正人** 田中稔** 市川忠男**
 * 広島大学大学院 ** 広島大学工学部

1. はじめに

データベースを効率よく処理する試みの1つとしてデータベース処理専用のハードウェアを用いるデータベースマシンがある⁽¹⁾。筆者らは既に関係代数演算モジュールを用いた並列パイプライン方式データベースマシンの提案を行なった⁽²⁾。本研究では、大規模データベースへの適用ならびに問合せに対する演算モジュールの割当て方式についての検討を行ない、シミュレーションを通して、提案するデータベースマシンの性能評価を行なった。

2. 並列パイプライン方式データベースマシン

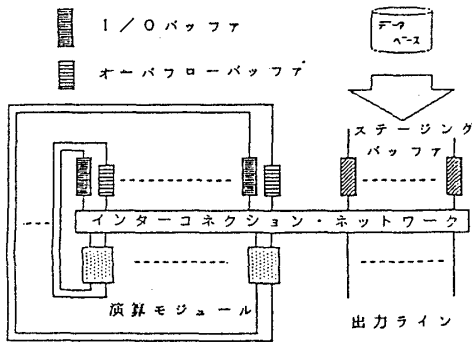


図1 問合せ実行システムの構成

筆者らの提案するデータベースマシンのシステム構成を図1に示す⁽²⁾。システムは、複数の演算モジュール、I/Oバッファ、オーバーフローバッファ、ステージングバッファ、およびこれらを接続するためのインターコネクション・ネットワークから構成されている。

演算モジュールは、セレクションを除く7つの全ての関係代数演算を効率よく実行する。問合せが与えられ、それに応じてインターコネクション・ネットワークの切り換えが行なわれ、演算モジュール間の接続関係が規定される。この切り替えによって、システムはどのような問合せに対しても柔軟に対応できる。さらに、各演算モジュールではそれぞれ独立に演算の実行を行なうことができ、しかも演算モジュール間では処理の時間的重ね合せがタプル単位で可能となっている。

3. メモリオーバーフロー

入力リレーションの大きさが演算モジュール内のメモリ容量を越えると、メモリオーバーフローが発生する。これに対処するための方法として以下に示す2つ

を用意する。

(a) 1つの演算の実行に複数の演算モジュールを用いる方法:

オーバーフローが発生したモジュールに別のモジュールを連結し、それら2つのモジュールで1つの演算処理を行なう。それらのモジュールに再びオーバーフローが生じた場合には、また新たに空きモジュールを割りつけることによって処理が継続される。

(b) ネスティッドループアルゴリズムに基づき、ディスクを介して処理を行なう方法:

入力リレーションの一方(Sリレーション)をページ単位で分割し、それらの各ページごとにもう一方のリレーション(Tリレーション)の全てのタプルとの比較を行なう。Tリレーションの各タプルはオーバーフローバッファを介してディスクに書き込まれ、Sリレーションの次のページに対応するタプルとの比較に際して再びディスクから読み込まれる。

使用されていないモジュールがある間は(a)の方法によってオーバーフロー処理が行なわれる。この場合には、処理速度の低下が引き起こされることなく演算の実行が継続される。(b)の方法では、ディスクのI/Oアクセス回数が増大による処理効率の低下が引き起こされるが、限られたモジュール資源の下で処理を継続することができる。

4. モジュール割当て方式

与えられた問合せに対する演算モジュールの割当て方式として、以下に示す3つを考える。

(a) S I S D割当て方式

問合せを構成している各関係代数演算を順次処理する方式で、一度には高々1つの演算しか実行されない。実行可能な(入力リレーションが全て生成されている)演算の1つに空きモジュールを全て割当てる。

各演算に対する入力リレーションが大きい場合には、演算の実行に使用するモジュールの数を多くとれるため、メモリオーバーフロー時におけるネスティッドループアルゴリズム採用による演算処理効率の低下をまねくことなく処理を行なうことができる。その反面、入力リレーションが小さい場合はモジュールの使用率が低くなる。

(b) インストラクション割当て方式

全ての実行可能な演算に対してモジュールの割当てを行なう方式で、同時に複数の演算が実行される方式である。

Performance Evaluation of A Parallel Pipelined Database Machine

Atsumi KAKIMOTO, Kazuyuki TSUDA, Masahito HIRAKAWA, Minoru TANAKA, and Tadao ICHIKAWA

HIROSHIMA UNIVERSITY

この方式は、(a)ならびに次に説明する(c)の両方の特徴を合わせ持っている。

(c)パイプライン割当て方式

問合せの実行に先だて、問合せを構成する全ての演算に対してモジュールを割当てる。モジュール間ではタプルを単位として、パイプライン的に処理が進められる。

この方式は前述の(a)、(b)とは異なり、演算の途中結果をディスクへ書き込むことなしに処理を行なうことができる。演算の並列パイプライン処理により、処理効率の向上が期待される。しかしながら、同時に複数の演算が実行されるため、各演算処理に割当てられるモジュール数が少なく、入力リレーションが大きい場合は、ネスティッドループ方式によるオーバーフロー処理の影響を受けて処理効率の低下が考えられる。

5. 性能評価

問合せに対する処理時間評価のためにVAX 11/750上にシミュレーションシステムを構築し、提案したデータベースマシンの性能評価を行なった。ここで、処理時間とは、ディスクからデータを転送し始めてから、演算結果を全てディスク上へ書き込むまでの時間としている。

評価を行なうにあたり、モジュールのメモリサイズを1024(タプル)、モジュール数を10と仮定する。また、1ページを16kバイト(256タプル)、モジュールとバッファ間で1タプルの転送に要する時間を64μsec、タプルの比較に要する時間を128μsec、モジュール内でのタプルのシフトに要する時間を0.7μsec、ディスクから1ページ分のタプルを読み込む時間を28.75msec、ディスクへ1ページ分のタプルを書き込む時間を33.75msecとする。

まず、ジョイン演算について、CS=CT=CRとした時の処理時間の評価結果を図2に示す。ここで、演算の2つの入力リレーションのタプル数をCS、CTとし、出力リレーションのタプル数をCRとする。メモリオーバーフロー時のネスティッドループ処理におけるディスクI/O回数の増加に伴って、処理時間の急激な増加が起こる。

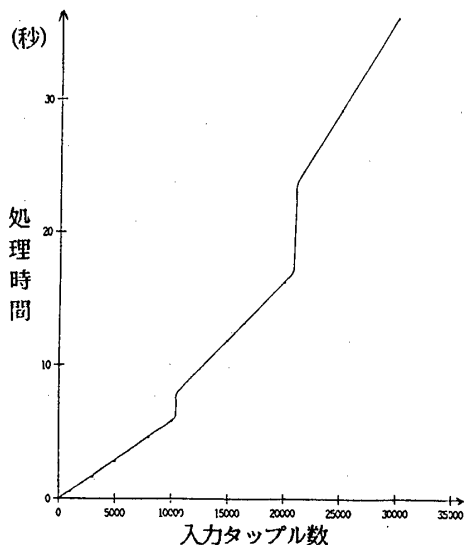


図2 ジョイン処理時間

次に、図3に示す問合せの処理時間を評価した結果を図4に示す。ジョイン演算においてはCS=CT=CRとし、プロジェクション演算においてはCR=0.7×CSとしている。入力リレーションの小さい範囲ではパイプライン割当て方式が優れている。これは、生成された中間リレーションの各タプルを直ちに次の演算の入力タプルとしてあてがうことによって処理の時間的重ね合せを行なうとともに、余分なディスクアクセスの時間を削減できるからである。しかしながら入力リレーションのサイズが大きくなると、パイプライン割当て方式では、演算当りに割当てられるモジュール数が少ないのでネスティッドループ方式によるオーバーフロー処理の影響が大きくなり、処理時間が大きくなる。データベースが大きくなるにつれて、インストラクション割当て方式、SISD割当て方式が有効となる。

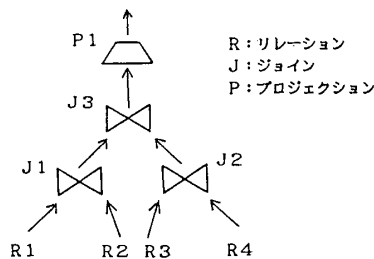


図3 問合せ例

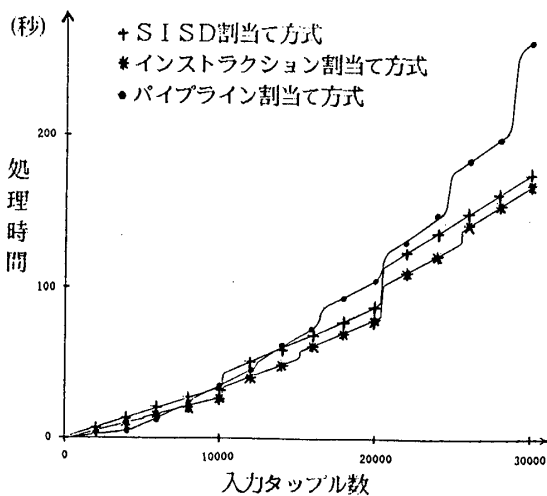


図4 問合せ処理時間

6. おわりに

本研究では、筆者らの提案するデータベースマシンの性能評価を行なった。今後は、オーバーフロー時の処理時間に大きく影響するディスク系について検討を行なうとともに、データ圧縮技法を用いた処理効率の改善、ハードウェアコストを考慮したシステムの評価などを行なう必要がある。

参考文献

(1) G.Z.Qadah; "Database machines: A survey," Proc., AFIPS NCC Conf., Vol.54, pp.211-223, 1985
 (2) 平川、津田、中山、田中、市川;"並列パイプライン方式データベースマシンを指向した関係代数演算モジュール"、信学技報 EC85-23(1985-8).