

SMASH システムの並列実行性能のベンチマーク評価

2G-6

佐藤 聡 清水 誠二 清木 康

筑波大学 電子・情報工学系

1 はじめに

我々は、データベースの多様な応用分野に適応可能な関数型並列データベース・システム SMASH の設計および実現を行っている [2][3]. このシステムの特徴は、データベース処理に関数型計算の概念を適用した点にある。これにより、応用分野に適したデータ構造と、そのデータ構造を対象としたデータベース演算を関数として柔軟に定義することができる。また、関数計算を並列に処理することによって、データベース処理を高速に行う。

本稿では、OO1 ベンチマーク [1] を用いて、SMASH システムの並列処理性能の評価を行い、その結果を示す。

2 SMASH システム

SMASH システムにおいて、データベース設計者は、それぞれの応用分野に適したデータ構造、および、そのデータ構造を対象とする任意のデータベース演算を関数として記述する。データベース利用者は、定義された関数を組み合わせた式として問い合わせを構成し、それを発行する。そして、システムの並列処理系は、問い合わせを並列に処理する。

我々は、関数群から構成されるデータベースの問い合わせを並列処理する方式として、ストリーム指向型並列処理方式を提案している。この方式において、データベース本体および、問い合わせの中間結果は、ストリームとして表現される。問い合わせは、ストリームを引数とする関数として定義されたデータベース演算を組み合わせることにより構成され、要求駆動型評価によって並列に処理される。ストリーム指向型並列処理方式により、問い合わせ処理に内在する、関数の引数間の独立性による並列性、および、ストリーム型並列性を抽出する。我々は、ストリーム指向型並列方式を実現するソフトウェア・アーキテクチャの基本操作としてプリミティブ・セットを設定している [2]。現在、プリミティブ・セットの解釈実行を行う並列処理実行系は、分散メモリ型並列処理環境(汎用の単一プロセッサを LAN で結合した環境)および共有メモリ型並列処理環境(汎用の共有メモリ型マルチプロセッサを用いた環境)に実装されている。

Evaluation of parallel query processing in SMSAH system by bench marks

Akira Sato, Seizi Shimizu and Yasushi Kiyoki
Institute of Information Sciences and Electronics, University of Tsukuba, Tsukuba city, Ibaraki 305, Japan

3 ベンチマーク・テスト

OO1 ベンチマークは、エンジニアリング・データベースの性能を測定するために設計されたベンチマークである [1]. このベンチマークは、2つの論理レコードが定義されている。

```
Part:RECORD[id:INT,Type:STRING[10],x,y:INT,
Built:DATE]
Connection:RECORD[from:Part-id,to:Part-id,
type:STRING[10],length:INT]
```

Part データベースは、それぞれユニークな識別子をもつ2万個のデータからなる。Connection データベースは、6万個のデータからなる。それぞれの Part は、ランダムに選択された3つの Part と Connection を介して接続されている。Part の接続は、局所性が存在する。すなわち、90% の Part は全体の1%の範囲内の Part と接続されている。

OO1 ベンチマークは、以下の3種の操作を測定に用いる。

Lookup 任意に選択した1000個の識別子に対応する Part を検索する。

Traversal 任意に選択した1個の識別子に対応する Part, またはそれに接続されている Part 等7重まで接続された全ての Part を検索する。

Insert 100個の Part と各々3個の Connection を挿入する。

4 実現方法

前節において述べた OO1 ベンチマークを用いて、分散メモリ型並列処理環境および共有メモリ型並列処理環境上に実現された SMASH システムの並列実行性能を測定する。分散メモリ型並列処理環境として、3台の SUN 社の SparcStation を Ethernet で結合した環境を用いた。また、共有メモリ型並列処理環境として、OMRON 社の LUNA88K(4CPU) を用いた。双方の環境で用いた計算機には、直接、ディスクが接続されている。ここでは、OO1 ベンチマークで定義された3種の操作の中で、Traversal 操作に着目し、さらに検索範囲を表す Part の接続数(以後、検索段数とよぶ)を変化させて、測定を行う。そして、プロセッサを1台使う場合と3台使う場合の問い合わせの実行時間を測定し、速度向上を調べる。

はじめに、データ構造を定義する。PartとConnectionを別のレコードとして定義し、各々別のファイル内に格納する。このファイルは、処理の高速化のため、親子リンクを用いて実現した。すなわち、Connectionレコードは、toおよびfromで接続されるPartのレコード識別子、および、同じtoまたはfromに接続されているConnectionのレコード識別子を含んでいる。また、Partレコードは、toおよびfromの最初のConnectionのレコード識別子を含んでいる。

これら2種のデータ構造に対して、Partのレコード識別子を入力ストリーム要素とし、そのレコード識別子が示すPartに接続されているPartの各々のレコード識別子を出力ストリーム要素とする2種の演算を関数として定義した。この2種類の関数は、入力ストリーム要素として受け取ったPartのレコード識別子からPartレコードを検索し、そのレコード内のConnectionレコード識別子からConnectionレコードを検索し、そして、そのPartに接続されているPartレコード識別子をストリームとして出力する。これら2種類の関数の相違点は、出力ストリームの数にある。すなわち、一方の関数（以後、関数Aとよぶ）は、検索したPartのレコード識別子をすべて同じストリームに対して出力し、もう一方の関数（以後、関数Bとよぶ）は、3つのストリームに分配して出力する。

Traversal操作を実現する問い合わせは、これらの関数を用いて構成する。使用するプロセッサが1台の場合、問い合わせは、検索段数と同じ数の演算Aを接続して構成した。使用するプロセッサが3台の場合、問い合わせは、1段目の検索には演算Bを、それ以降の段目の検索には演算Aを接続して構成した。分散メモリ型並列処理環境において、分割した後の各々の検索を行う演算群ごとに1台の計算機を割り当てた。さらに、分散メモリ型並列処理環境では、各々の計算機のディスクにPartおよびConnectionのデータベースのコピーを配置した。

各々の環境において使用した各計算機は、2つのファイルを全てキャッシュできる大きさ以上の主記憶を有する。

5 測定結果

各々の実行環境において、接続段数を7段から13段まで変化させ、ディスク・キャッシュがクリアされている状態における問い合わせの実行時間(cold)および、同じ問い合わせを10回行った中で最も短い実行時間(warm)を測定した。各々の環境における、検索段数に対する問い合わせ処理の速度向上の関係を図1に示す。

分散メモリ型並列処理環境について考察する。キャッシュがcoldの場合、検索段数が7段から10段までは、ディスクへのアクセス時間が問い合わせの実行時間の多くを占めている。この環境においては、ディスク・アク

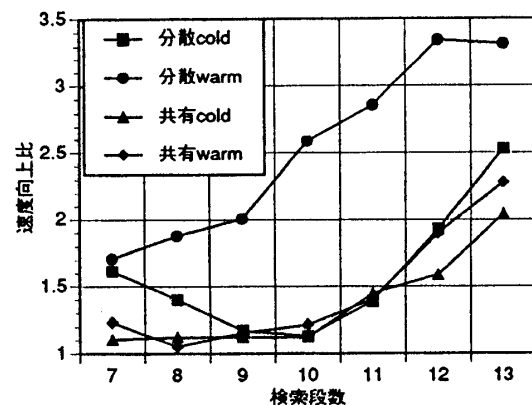


図1: 検索段数に対する速度向上

セスが分散されているが、段数の増加に伴い、各々の計算機におけるアクセス回数も増大している。よって、台数効果が表れない。検索段数が10段を越えると、ほぼ全てのデータがキャッシュに載る。よって、キャッシュ空間内のアクセス時間が実行時間の多くを占めているため、台数効果が表れている。キャッシュがwarmの場合、検索するPartのデータのほぼ全ては、キャッシュに載っている。検索段数の増加に伴い、全体の時間に占めるアクセス時間の割合が増加する。よって、検索段数の増加に伴い、台数効果が表れている。

共有メモリ型並列処理環境について考察する。図1より、キャッシュの状態の変化に対して、速度向上の変化はほとんどない。これは、共有メモリ空間へのアクセスが逐次化しているためである。検索段数が10段を越えると、Partのデータに対するアクセス箇所が分散するため、台数効果が表れている。

6 おわりに

本稿では、OO1ベンチマークを用いて、SMASIIシステムの並列処理性能を検証した。今後の課題としては、分散メモリ型並列処理環境上で、より多い台数を用いた場合の並列処理性能の検証が考えられる。

参考文献

- [1] R. G. G. Cattell and J. Skeen. Object Operations Benchmark. In *ACM Trans. Database Syst.*, pp.1-31, March 1992.
- [2] Y. Kiyoki, T. Kurosawa, K. Kato, and T. Masuda. The software architecture of a parallel processing system for advanced database applications. In *Proc. 7th IEEE Int. Conf. on Data Engineering*, pp. 220-229, Apr. 1991.
- [3] 佐藤, 清木. 関数型計算に基づくデータベースシステムの並列処理実行系の実現方式. 情報処理学会データベースシステム研究会研究報告 92-DBS-89, 1992.