

データベースの多様な応用分野に対応可能な関数型並列処理システム SMASH  
 - データ構造および演算の定義系と実行系 -

3H-7

黒沢貴弘<sup>†</sup> 清木 康<sup>†</sup> 加藤和彦<sup>†</sup> 益田隆司<sup>†</sup>  
<sup>†</sup>筑波大学 <sup>†</sup>東京大学

## 1. まえがき

データベースに関する研究の進歩にともない、データベース技術の様々な分野への応用が試みられている。多様な分野に対応するために、データベース・システムは、新しい演算および複雑なデータ構造を柔軟に追加する機能を提供する必要がある。また、データベースの応用分野の多様化につれて処理対象のデータ量も増大を続けている。このため、データベース処理の性能の向上は、データベース・システムの実現における主要な課題の一つとなっている。多様化するデータベースの応用分野に対応しようとする場合、対象のデータ構造および演算が固定化できないため、特定のアルゴリズムを対象とした専用アーキテクチャによる直接的な効率向上を図ることが難しい。したがって、我々は、任意のデータ構造および演算を対象とした並列処理技術の適用による高速化を目指している。

このような背景に基づき、我々は、データベースの多様な応用分野および知識ベースに柔軟に対応できる並列処理方式および並列処理システム SMASH の設計、実現を行っている。

多様な応用分野に対応するための拡張性の支援は、データベースの分野におけるもっとも重要な課題の一つと考えられる。本処理系では、基本演算およびデータ型に関する拡張性の支援により、その解決を試みている。本稿では、SMASHにおけるデータ型の記述、および、それらを処理系で実現するための機能について述べる。

## 2. 並列処理システム SMASH

SMASHシステムの特徴は、以下のようにまとめられる。

- (1) 応用分野に応じたデータ構造および独自のデータベース演算を定義し、それらをシステム内に組み込むための機能を提供する。
- (2) 関数型計算の概念を適用し、複雑なデータ構造を対象とする任意のデータベース演算を並列に処理する。関数型計算モデルは、任意の操作に内在する並列性を抽出するのに適した計算モデルである。
- (3) 汎用の並列処理計算機あるいは分散処理環境の上に構築する。これは、ハードウェアの進歩を柔軟に反映し、最新の汎用ハードウェアに対応するためである。

(2), (3) を満たす並列処理環境を実現するためにプリミティブ・セットを設定した。プリミティブ・セットは、処理対象となる任意の応用分野の基本演算を記述できるように、基本演算に依存しないレベルに設定され、また、関数計算のための機能をハードウェアに依存しない水準で記述できるように、ハードウェアから独立なレベルに設定されたソフトウェア・アーキテクチャである。

応用分野の基本演算間で受渡されるデータは、ストリームとして扱われる。したがって、対象分野の各基本演算は、関数型プログラミングの枠組みの中で、ストリームを

引数とする関数として記述される。このようにして記述された関数群に内在する並列性が、要求駆動型評価による関数型計算によって抽出される。具体的には、基本演算を単位として記述された各関数は、ストリームを対象とする関数計算を実行するためのプリミティブ群を含む逐次的なオブジェクト・コードに変換される。そして、各基本演算は、関数を単位としてプロセッサへ割り当てられ、要求駆動型評価により並列に実行される。その結果、関数を粒度とした並列性が抽出される。

## 3. データ構造および基本演算の記述機能

本処理系の応用分野は、知識ベースあるいはCADのように、複雑な構造を持つデータを大量に扱う分野である。一般に、これらの分野では、木構造やグラフ構造で表現されるデータを扱う。複雑なデータ構造を表現し、操作するためには、柔軟にデータ構造を定義し、操作する機能が必要である。そこで、本処理系では、多様なデータ構造を定義、合成し、それらのデータ構造を操作するデータベース演算を記述できる言語を提供する。

## 3.1 データベース処理記述言語

一般的なプログラミング言語との比較において、本言語の特徴は、データベース処理に適する型の導入を行い、それらの型に対して、並列処理系における効率的な支援を行っている点にある。

データベースでは、大量のデータを同様の構造を持つデータ群として表現できる場合が多く、それらに対して効率的な操作を施す機能を実現することが重要である。本処理系では、同一の型を持つデータ群を、ストリームとして規則的に扱い、ストリーム型並列処理[1]を実現する。本言語では、ストリームを対象とした関数定義を記述するための枠組みを用意している。

## 3.2 型構成子

本言語では、タプル型(tuple)、列型(sequence)、id型(id)、選択型(variant)の4つの型構成子を用意している。これらの中で、列型およびid型の扱いに特徴がある。関数間で受渡されるデータの型を列型に設定した場合には、ストリーム型並列性が抽出される。すなわち、列型の引数を持つ関数では、その関数の実引数の生産者関数との間で、ストリーム型並列性を抽出できる。同様に、関数の計算結果(関数値)が列型の場合、その計算結果の消費者関数との間で、ストリーム型並列性を抽出できる。また、関数間で受け渡されるデータの型を列型として、統一的に表現することにより、部品化された関数をつなぎあわせ、新しい機能を実現することが容易となる。

データ定義の中にid型を含む場合、複雑な構造を持ったデータの転送に際して、関数間で受け渡されるデータの量を最小化でき、また、関数間でのデータの共有も可能となる。id型により表現されたデータに関しては、そのデータへの参照が起こった時点で、はじめて、そのデータの実値が取り出される。したがって、関数内で参照されない無駄なデータの生成は必要なくなるので、関数間でのデータ転送量が最小化される。この点は、複雑なデータ構造に対す

A Functional Parallel Processing System SMASH  
 supporting a wide variety of database applications  
 -- Definition and processing system  
 for data structures and operations --  
 Takahiro Kurosawa, Yasushi Kiyoki,  
 Kazuhiko Kato, Takashi Masuda  
 University of Tsukuba, University of Tokyo

る参照に関して、関数型計算における遅延評価の概念を応用したものとして位置づけられる。さらに、id型では、強い型付けにより、そのidがポイントしているデータの型が特定されている。

4. 並列処理システムのソフトウェア・アーキテクチャ

本処理系の構成を図1に示す。この処理系は、関数およびデータ構造を定義、登録するための定義系、定義されたデータ構造および関数群を用いて問い合わせ処理を行うための問い合わせ処理系、および、関数のインスタンスを生成しデータベース処理を行う並列処理実行系により構成される。

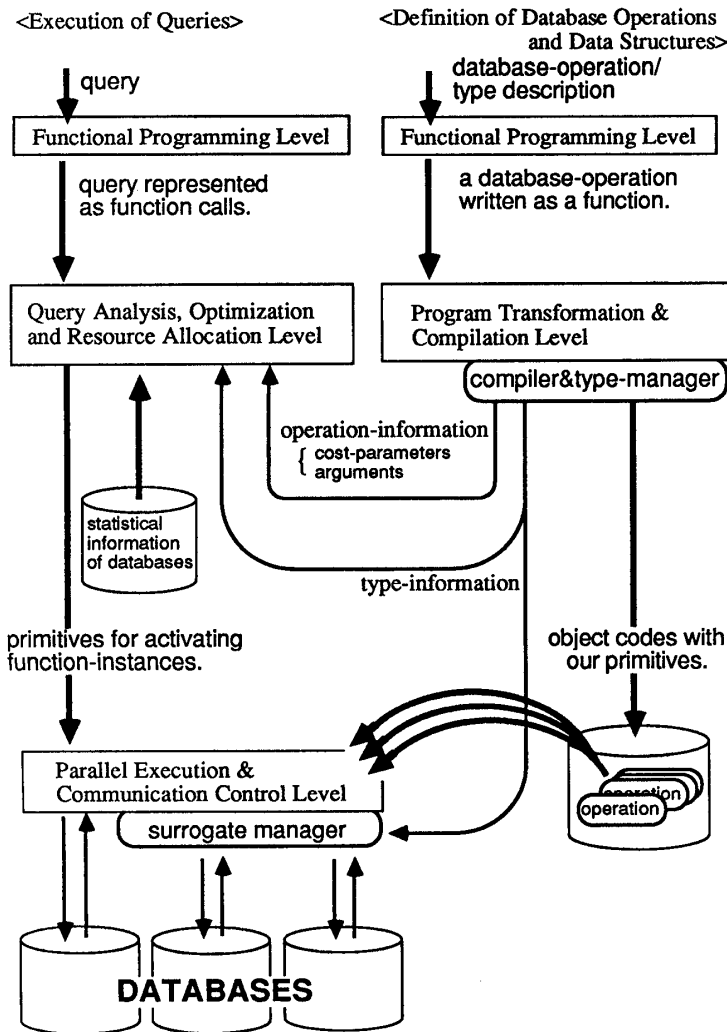


図1. SMASHの構成

4.1 データベース定義系

まず、特定の応用分野の基本演算およびデータ構造が関数プログラムとして定義される。ここで、対象分野の基本演算は、ストリーム・データを受渡しする関数として記述される。各関数は、SMASHのアーキテクチャ・レベルであるプリミティブを含む逐次実行されるオブジェクト・プログラムに変換される。このオブジェクト・プログラムは、関数本体内の処理（繰り返しの制御、変数への単一代入）に関しては、手続き的なコードにより記述される。また、関数型計算における引数評価および関数適用に関する部分、すなわち、関数計算の中断および起動、関数間でのデータ

の送受に関しては、プリミティブにより記述される。このように、プリミティブ・セットは、関数を起動・中断し、関数間のデータの受渡しを制御する基本操作群として設定されている。

4.2 問い合わせ実行系

問い合わせ処理時には、問い合わせを構成する各関数は、並列処理の単位となる関数インスタンスとして、並列プロセッサ群の1プロセッサに配置される。異なるプロセッサに配置された関数インスタンス間において、並列処理が行われる。関数型計算の基本的操作である関数引数評価および関数適用は、プリミティブによって実現される。このレベルでは、多様なデータ型をストリーム処理するためのプリミティブ、さらに、システムの提供するデータ型構成子に関するオペレータをランタイム・ルーチン（ライブラリ）としてサポートしている。

二次記憶に格納されているデータベースの実データへのアクセスに関しては、surrogate managerを介して、idを実値へ変換する場合（ランダム・アクセス）と、関数が、実引数としてデータベースを直接読む場合（逐次アクセス）がある。前者は、プリミティブとして用意され、後者は、実データをストリームとして並列処理システムへ送り込むための機能が関数として設定される。この関数は、アクセス・パスの選択を行った後に、二次記憶上の実データを読み込み、ストリームとして出力する。本システムでは、二次記憶上のデータベースの更新については、関数の定義が変更されるのではなく、関数に適用される実引数値が更新されるものとしてみなされる。

5. まとめ

本稿では、データベースを対象とした拡張性のある関数型並列処理システムについて、データ構造の記述、および、その支援のための機能について述べた。

現在、プリミティブ・セットを、SunワークステーションをLANにより結合した環境の上に実現しており、本稿で述べた関数型言語を用いて記述した演算により、問い合わせ処理の実験を行うことができる。

今後は、複雑なデータ構造を対象とする任意のデータベース演算を処理するために、ストリーム指向型処理を対象とした計算機資源の割り当て方式[3]の拡張、問い合わせ言語の処理系の実現、二次記憶系におけるアクセス・メソッドの設計を行っていく。また、これらの機能を加えた並列処理システムを、実際のデータベース応用（具体的には、CAD、テキスト処理など）に適用することを検討している。

[参考文献]

[1] Y. Kiyoki, K. Kato and T. Masuda, "A relational database machine based on functional programming concepts," Proc. 1986 ACM-IEEE Computer Society Fall Joint Computer Conf., pp. 969-978, 1986.  
 [2] Y. Kiyoki, T. Kurosawa, P. Liu, K. Kato and T. Masuda, "Implementation and Experiments of an Extensible Parallel Processing System Supporting User Defined Database Operations," Proc. of International Symp. on Database Systems for Advanced Applications, 1989.  
 [3] P. Liu, Y. Kiyoki and T. Masuda, "Efficient algorithms for resource allocation in parallel processing environments," Proc. of the IEEE International Conf. on Distributed Computing Systems, 1989.