

\* 筑波大学 電子・情報工学系  
\*\* 東京大学 理学部 情報科学科

### 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、データベース技術を様々な分野に応用しようとする試みがなされている。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している。そこで、高い拡張性を備え、かつ、高速処理能力のあるデータベース・システムの実現が重要な課題となっている。

このような背景の中で、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[1][2][3]。この方式の特徴は、次のようにまとめられる。

(1) データベースおよび知識ベースを対象とする任意の演算を柔軟にシステム内に統合し、それらを並列に処理する拡張性の高い環境を実現する。データベースおよび知識ベースを対象とする任意の操作に対して並列処理を実現するために、関数型計算の枠組みの中でそれらの操作を記述し、実行する。本処理方式では、多様化するデータベースの応用に対し柔軟な並列処理環境を実現するために、関数型計算モデルの概念を、データベース処理に適用している。

(2) ハードウェアの急速な進歩に対応するために、汎用のハードウェア資源を用いて実現する。

(1), (2) を満たす並列処理環境を実現するために、基本プリミティブを設定した[1][2]。基本プリミティブは、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し、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した。

本処理系において、関数は、応用分野の基本演算をシステム内に統合する際の単位となっている。すなわち、本処理系の対象となる各基本演算は、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、関数型計算の枠組みの中で、抽出される。基本演算を単位として記述された各関数は、ストリームに対する関数計算を実行するための基本プリミティブを含む逐次的なオブジェクト・コードに変換される。それらの基本演算は、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る関数型計算の枠組みの中で並列に実行される。その結果、関数を粒度とした並列性が抽出される。

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である。本システムでは、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、それらをシステム上で実現するための機能について述べる。

### 2. データ型記述の機能の位置づけ

データ型記述の機能の実現は、データベース・システムの拡張性を高める。この機能により、アプリケーションに依存する様々なデータ構造をシステム内に組み込むことができる。

SMASHのソフトウェア階層は図1に示すように、4階層からなる。

#### ① 関数プログラミング・レベル

処理対象となる基本演算を、ストリーム・データを受渡しする関数として記述する環境を実現する。

#### ② プログラム変換レベル

関数プログラミング・レベルで記述された各関数を、逐次的に実行されるオブジェクト・コードに変換する。

#### ③ 実行・通信制御レベル

要求駆動型評価方式のもとで関数の実行を行う。このレベルでは、基本プリミティブの解釈が行われ、また、関数間のストリーム・データおよびデマンドの通信が制御される。

#### ④ アーキテクチャ・レベル

汎用プロセッサを利用するため、利用するプロセッサのアーキテクチャとのインタフェースを構成する。

データ型を記述し、これを支援するための機能は、

#### ③ 実行・通信制御レベルで実現される。

それらの機能は次の2項目により実現される。

- (1) 多様なデータ型を支援するための基本プリミティブの拡張
- (2) システムの提供するデータ型に関するオペレータの設定

### 3. 基本プリミティブの拡張

本実現方式では、すべての基本演算を、ストリームを入出力する関数により表現し、その関数を単位として並列性を引き出す。以下、実行時に起動される関数を関数インスタンスと呼ぶ。関数インスタンスは、1つ、または、複数のストリームを入力し、1つのストリームを出力する。この際、関数インスタンス間のストリーム要素およびデマンドの受け渡しは、すべてチャンネルを介して行う。したがって、チャンネルはストリーム要素の型を把握する必要があり、このための基本プリミティブが必要となる。

具体的には、defineTypeを基本プリミティブ・セットに追加し、チャンネルを生成する基本プリミティブ

A parallel processing system SMASH for a wide variety of database applications

-- A manipulation method for complex data structures --

Yoshihiro TAKAI\*, Takahiro KUROSAWA\*, Yasushi KIYOKI\*, Kazuhiko KATO\*\* and Takashi MASUDA\*\*.

\* University of Tsukuba, \*\* University of Tokyo

ブchannelの拡張を行う。

定義されたデータ型はストリーム要素の型として用いられる。これを実現するために、データ型をシステムに登録するための機構が必要である。この型の登録に関しては、基本プリミティブdefineTypeを新たに設定する。defineTypeに型定義を引数として渡すと、その型定義をシステムに登録し、システム内で一意のtype\_idを返す。

チャンネルは各関数インスタンスにとっては引数に相当する。そこで、関数インスタンス間で受渡されるストリーム要素の型を各チャンネルについて宣言する。関数インスタンスは、この型を参照することによりストリーム要素のデータ構造を把握する。ストリーム要素の型宣言は、type\_idを基本プリミティブchannelに引数として渡すことにより行われる。

以上により、システムの提供する型だけでなく、新たに定義される任意のデータ型を要素とするストリームを扱うことが可能になる。

#### 4. 型定義の機能

本システムにおいて、支援するデータ型は以下の3種類がある。

##### (1) atomic type

システムには、基本的なデータ型がatomic typeとして用意されている。これらを利用して新たな型定義を行うことができる。

atomic typeには、整数、実数、文字などがある。

##### (2) 型構成子

データの記述能力に拡張性をもたせるため、以下に示す汎用のデータ型構成子を導入する。

##### • tuple

複数の型の直積を定義域とする型。

tuple型の記述形式

$[n_1:t_1, \dots, n_n:t_n]$

※ $n_i$ は属性名、 $t_i$ はその型を示す。

(例) [age:int, gender:char]

##### • set

同一定義域内のデータの集合を示す型。

set型の記述形式

{type}

※typeは集合要素の型を示す。

(例) {int}

##### • ordered-set

集合内の各データに順序をつけたもの。

この中では重複が許される。

ordered-set型の記述形式

<type>

(例) <char>

##### (3) defined type

上記のatomic typeと型構成子を用いて型定義を行うことができる。さらに、その型に名前をつけることにより、別の型定義においてそれを参照することができる。

(例)

child: [name:<char>, age:int, gender:char]

emp: [name:<char>, sal:int, children:{child}]

本システムでは、さらに、各型構成子に対しオペレータを設定する。オペレータは、selector, constructor, 比較演算子などからなる。ordered-setのオペレータを図2に示す。

#### 5. まとめ

本稿では、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における、データ型の記述、およびその支援のための機能について述べた。

現在、新たに定義されたデータ型に関するデータ操作における型チェックの方法、およびデータ型に固有なデータ操作を組み込むための抽象データ型に基づく方法を検討している。

また、この機能を加えた並列処理システムを、複雑なデータ構造を必要とするデータベース応用(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] 黒沢, 清木, 加藤, 益田, "関数型計算モデルに基づく並列型問い合わせ処理系の実現方式," 情報処理学会データベース・システム研究会報告, No.68-7, Nov.1988.

[3] 劉, 清木, 益田, "ストリーム指向型関係演算処理におけるバッファ資源割り当ての計算方式," 情報処理学会論文誌, Vol.29, No.8, pp.770-781, Aug.1988.

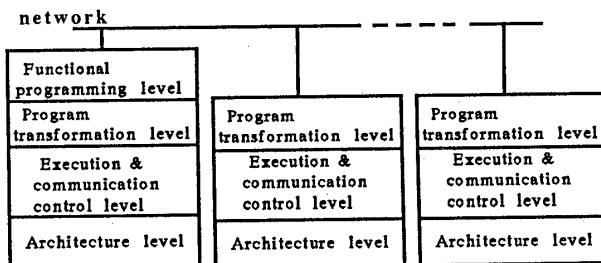


図1 システム構成

オペレータ	機能
<b>selector</b> nth sublist count	要素の参照 部分列の参照 要素数の参照
<b>constructor</b> compose	ordered-setの構成
<b>modifier</b> insert delete	要素の挿入 要素の削除
比較演算 in? equal?	包含関係 等価性

図2 ordered\_setのオペレータ