3H－9 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 －CADデータベースへの適用－

岡田健一　関安宏　清木康
筑波大学

1．はじめに

我々は、データベースの多様な応用分野を対象とした、関数型並列処理システム SMASH を開発中である[1]. SMASH では、データベース処理に対し、関数型計算の概念に基づくストリーム指向並列処理を適用している．現在、単純なデータ構造に対する処理機構に加え、グラフ構造や木構造のような複雑なデータ構造を処理するための機構を、設計している[2].

現在までに、SMASH のアプリケーションとして、機械系 CAD データベースの検討を行なってきた[3]. このアプリケーションを実現することで、複雑なデータ構造を扱う応用分野への SMASH の適応性を実証する．

本稿では、このアプリケーションの実現方式について述べる．

2．CAD データベース

CAD データベースの研究分野は、トランザクション、版管理、データモデルなど多方面に及んでいる[4]. ここでは、機械系 CAD データベースで扱うデータ構造と、そのデータベースに対する検索系の実現について検討する．

2．1 機械系 CAD で扱われる情報とデータ構造

機械系 CAD において、製品を表現するための製品情報には、形状情報、部品属性情報、配置情報、接続情報、製品属性情報などがある[3]. これらの製品情報を扱うためのデータ構造として、図1に示す木構造を設定した．形状情報は solid節、部品属性情報、配置情報は parts 節、接続情報は group 節、製品属性情報は product 節で管理する．

2．2 問い合わせと基本機能

データベースの実際の問い合わせ例として、「製品 A を構成している部品名を求めよ」「部品 P を使用している製品名を求めよ」 などを設定した．これらの問い合わせは“木の選択”，“節の選択”，“節情報の取り出し”の操作の組み合せで実現できる．そこで、表1に示すような、それぞれの操作に対応する基本機能を設定した．

表1　基本機能一覧

基本機能	説明
get-attribute-value	節に含まれる属性値を取り出す.
select-node	節を選択する.
get-all-children	節の全ての子節を取り出す.
get-all-parents	節の全ての親節を取り出す.
get-tree	データベースから木を取り出す.

A Functional Parallel Processing System SMASH supporting a wide variety of database applications -- Applying SMASH to CAD databases --
Kenichi Okada, Yasuhiro Seki and Yasushi Kiyoki
University of Tsukuba

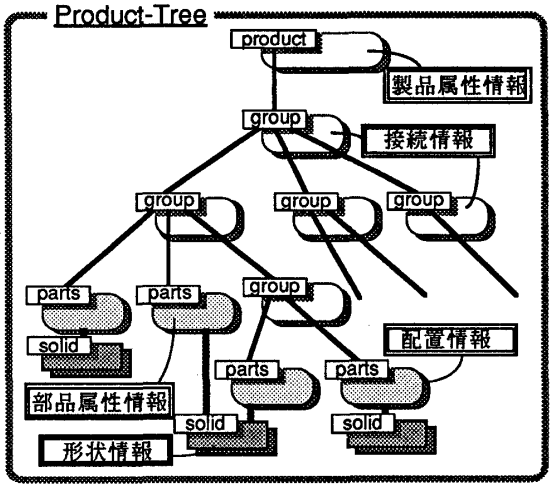

図1　製品を表現する木構造

様々な問い合わせは、これらの基本機能を組み合せることで実行される．

これらの基本機能は、ストリームを対象とした木操作となっている．例えば、一般の木操作における節の選択は、次のように実現する．基本機能 get-all-children で全ての節を取り出し、それらをストリームの要素として流し、基本機能 select-node で条件に合う節を選択する．

3．SMASH における実現

木構造を表現する一方式として、関係モデルによる方式、複合オブジェクトを直接扱う方式、および、両者を組み合わせた方式について、検討を行なった．

関係モデルによる方式では、データ構造が単純であり、それらのデータに対し一様な処理を行なう．また、一つの木構造が、いくつかの表に分散して格納される．

複合オブジェクトを直接扱う方式では、木構造をそのままデータ定義するため、データ構造は複雑になるが、一つの木構造を一まとまりの単位として処理できる．

3．1 複合オブジェクトを直接扱う方式

複合オブジェクトを表現するために、データ型として id 型を用いる[2]. これは、プログラム言語におけるポインタ型に相当する．この id により参照されるデータが、データベースの入出力の単位となる．また、入出力のための SMASH のプリミティブとして、fetchと store が用意されている[2]. id 値は、データを store プリミティブを用いて、データベースに格納した時、システムから与えられる．また、fetch プリミティブを用いて、データをデータベースから取り出す時、その値を指定する．

id 型を用いることにより、データベースの問い合わせ処理では、データの実体を流すのではなく、id をストリームの一要素として流す．各関数では、必要なときに fetch プリミティブを用いて、データベースからデータの実体を取り出す．

3．2 データ定義による実現方式の分類

SMASH 上に実現する方式を、図２に示すような、データ定義により分類して検討した．

まず、関係モデルによる方式（図２①部分）、複合オブジェクトを直接扱う方式（図２②部分）、そして、両者を組み合わせた方式（図２③,④部分）を設定した．

次に、木構造を構造情報と節情報に分けて考え、節情報を構造情報に統合して管理する方式（図２Ⓐ部分）と、節情報を構造情報とは別々に管理する方式（図２Ⓑ部分）に分類した．構造情報とは、節間の接続関係の情報であり、節情報は節に含まれる属性情報である．

さらに、構造情報と節情報の id を付ける単位により、細かく分類した（図２ⓐ,ⓑ部分）．

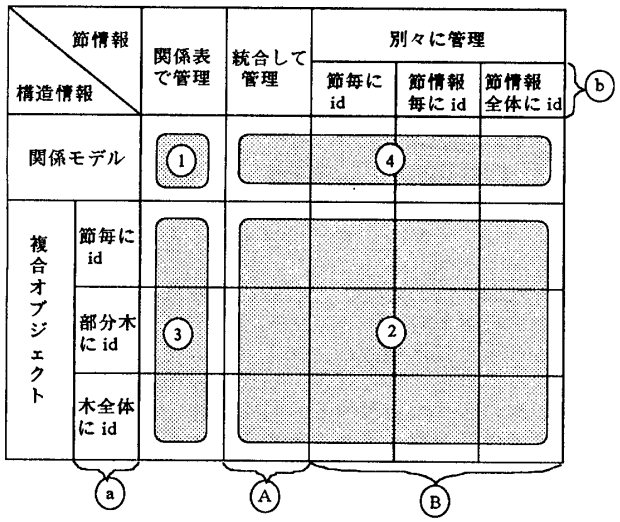

図２　実現方式の分類

3．3 基本機能の実現

基本機能については、データ定義によって処理内容が異なるため、前節で分類したデータ定義毎に実現する必要がある．

関係モデルによる方式の場合、データベース演算として関係演算を SMASH に組み込む．基本機能は、関係演算を単独、もしくは、組み合わせることにより実現する．この場合、並列性は、関係演算間で抽出される．

複合オブジェクトを直接扱う方式の場合、データベース演算として、木操作を行なう関数を記述し、SMASH に組み込む．この関数の内部では、木操作を行なうために fetch プリミティブを用いる．一般的に、各基本機能は、一関数に対応しており、並列性は、関数間で抽出される．

前述した二方式を組み合わせた方式の場合、データベース演算として、関係演算と木操作の関数を SMASH に組み込む．基本機能は、必要に応じて、関係演算と木操作の関数を単独、もしくは、組み合わせて用いることにより実現する．この場合、並列性は、関係演算を表す関数、木操作の関数の間で抽出される．

4．各実現方式に対する考察

それぞれの実現方式について考察する．

・構造情報に着目した場合

構造情報を関係モデルで実現する場合（図２①,④部分）、木探索のために、関係間の結合演算処理が必要となる．構造情報を複合オブジェクトとして実現する場合（図２②,③部分）、木探索は fetch プリミティブを用いて行なわれる．すなわち、木探索の性能は、fetch プリミティブのオーバヘッドに関係する．

・節情報に着目した場合

節情報を構造情報とは別々に管理して実現する場合（図２Ⓑ部分）、木探索の必要のない、すなわち節情報の検索だけの問い合わせでは、直接、節情報を検索できるため、木探索を行なう場合に比べ、fetch プリミティブを実行する回数が少ない．また、特に、関係表で管理する場合（図２③部分）、全木に対する節情報の検索では、関係演算を用いて一括して検索できる．節情報を構造情報に統合して管理して実現する場合（図２Ⓐ部分）、木探索と節情報の検索を同時に行なう問い合わせでは、構造情報から節情報への参照に fetch プリミティブを実行する必要がないため、節情報を別々に管理する方式に比べ、fetch プリミティブを実行する回数が少なくなる．

・id を付ける単位に着目した場合

id を付ける対象データの単位は、関数内で必要となるメモリ量と fetch 回数に影響を与える．一般に、その単位が小さい場合には、fetch 回数が増えるが、メモリ使用量は少なくなる．

5．おわりに

本稿では、SMASH において、CAD データのような複雑なデータ構造を、ストリーム指向の並列処理に適用する実現方式について検討した．実際に、機械系 CAD データベースを対象にアプリケーションの開発を進めている．

今後の課題を次に示す．

・id を付ける単位によって、アプリケーションの性能が大きく左右される．データの単位の設定に関する実験を行ない、最適な id 付けのための指針を得る．

・並列処理の単位、すなわち関数の粒度について検討を行ない、より高い並列性を抽出するための、最適な関数の粒度を設定する指針を得る．

・SMASH では、ストリーム型並列性だけでなく、独立な引数間の並列評価による並列性を抽出できる．問い合わせ処理において、この並列性の抽出による効果について検討を行なう．

［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,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 --,情報処理学会第40回全国大会, March 1990.

[3] 関,関島,清木,並列処理システム SMASH における機械系 CAD データベース実現のための一考察,情報処理学会データベースシステム研究会, 89-DBS-72, pp.177-184, July 1989.

[4] 大保, CAD データベースの動向,アドバンスドデータベースシステム・シンポジウム, pp.73-82, December 1988.