

Kappa-P の中間言語処理部

5 R-7

澤部直太¹, 西山聡¹, 富樫泰子¹, 河村元夫²

1: (株) 三菱総合研究所 情報科学部

2: (財) 新世代コンピュータ技術開発機構

1 はじめに

第 5 世代コンピュータプロジェクトでは、大規模知識処理を目的として、並列推論マシン PIM[1] 及びその上で動作する並列知識処理システムの研究開発を行なっている。これらの知識処理システムでは、複雑でかつ大量の知識データを効率的に処理できるデータベース管理システム (DBMS) が必要とされる。Kappa-P[2] は、この目的のために研究開発された並列データベース管理システムであり、現在、その最初の版が PIM 上で動作している。Kappa-P 中間言語処理部は、関係代数演算処理を並列に実行する機能を提供するものである。中間言語処理部では、関係代数演算処理の対象とするデータ構造として、テーブル、集合、ストリームの 3 種類を用意しており、並列処理の処理効率を考慮してデータ型の選択が行なえるようになっている。本稿では、Kappa-P の中間言語処理部の設計方針ならびに提供機能について報告する。

2 Kappa-P の言語階層

Kappa-P はユーザからの問い合わせ処理を行なうために 5 段階の言語階層をもっており、中間言語はその 4 番目のレベルに相当する (図 1)。以下に、各レベルの特徴を説明する。

ユーザ記述言語: ユーザが問い合わせを記述するための言語で、拡張関係代数に基づく問い合わせ言語である。

関係代数列: ユーザ記述言語で書かれた問い合わせから抽出した関係代数列である。ユーザ記述言語の記述内容とほぼ一対一に対応する。

部分関係代数列: 関係代数列の内容に対して水平分割情報の内容を考慮して最適化を行なうことによって得られる、ローカル DBMS レベルで閉じた関係代数列である。

中間言語: 部分関係代数列に対してローカル DBMS 内での最適化を施した結果から生成される。中間言語を生成する上で行なわれる主な最適化は、検索条件を変形することによってデータベースに対する読み出しの回数を減少させるという静的なものである。

原始コマンド: レコード ID 情報によるポインタ操作に基づく演算であり、非正規関係に対する効率的な操

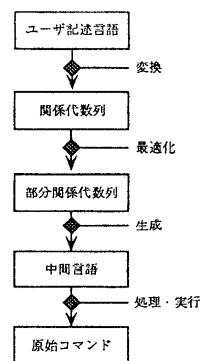


図 1: Kappa-P の言語階層

作を提供している。

3 中間言語処理部における実行制御

Kappa-P の言語階層で特徴的なところは、ユーザは処理の並列性を考慮せずに問い合わせを記述するのに対して、部分関係代数列から中間言語を生成する際には、関係代数演算の流れの中である程度並列性を考慮した中間言語列の生成を行う必要がある点である。

中間言語処理部における基本的な実行制御方式は、処理対象レコードをパイプラインで処理プロセス (個々の関係演算処理、ならびにデータベースに対するアクセス処理) を流すことによって行なっている。しかしながら、いくつかの処理プロセスに関しては、パイプライン処理によって制御することができないため、これらのものについては、処理の正常終了を表すステータスを受け渡すことにより、実行制御を行なうこととした。ステータスの受け渡しによって実行制御を行なうものとして、以下のものがある。

関係の生成処理: 生成した新しい関係がデータベース上に正常に定義される前に、その関係に対してレコードの追加処理を行なってもその結果は保証されない。したがって、新たに作られた関係に対してレコードの追加処理は、生成処理の実行結果ステータスの到着を待って実行されなければならない。

関係の更新処理: 関係に対するレコードの追加、および削除といった更新処理を並列に実行することはデータベースの整合性を損なう原因となるため、逐次処理を行なう必要がある。したがって、ある関係に対して更新処理を行なう場合は、その関係に対する全ての更新前の処理に関する実行結果ステータスの到

Internal Language Processor for Relational Algebra on Kappa-P
Naota SAWABE¹, Satoshi NISHIYAMA¹, Yasuko TOGASHI¹,
Moto KAWAMURA²

1: Mitsubishi Research Institute, Inc. 2: Institute for New Generation Computer Technology.

着を待って更新処理を実施しなければならない。また、更新した関係を対象とした処理についても、更新処理に関する実行結果ステータスの到着を待って実行されなければならない。

4 中間言語の構造

4.1 中間言語が提供するオブジェクト

中間言語処理部では、その処理対象として次の5種類のオブジェクトを用意している。

テーブル：「関係」に相当するオブジェクトで、レコードの読み出し、ならびに書き込みを行なうことができる。

集合：テーブルから抽出したレコード群を、対応するレコードID情報列の圧縮形式で表現したものである。集合指定によるレコード群の読み出しや、集合同士による和・差・積演算処理を行なうこともできる。

整列集合：集合に対して属性名をキーとして整列化を行なったものである。

ストリーム：レコード、またはレコードとそのレコードID情報の対のリストである。ただし、ストリームの個々の要素を複数のレコード(たとえば、100レコード)を格納できるバッファとし、このバッファを要求駆動形式で受け渡すことにより、実行時のレコードの読み出しのオーバー・ヘッドを小さくするとともに、主記憶の効率的な利用を実現する。

ハッシュ表：テーブルに対して作られるハッシュ表(テーブル・ハッシュ)と、ストリームに対して作られる一時的なハッシュ表(ストリーム・ハッシュ)とがある。テーブル・ハッシュには、ハッシュ・キーとレコードID情報の対が蓄えられ、ストリーム・ハッシュにはキーとレコードの対が蓄えられることができる。

4.2 中間言語が提供するコマンド

中間言語処理部が提供するコマンドは大きく分けると、次の5つのグループに分けることができ、関係代数演算に関しては各オブジェクト(とその組合せ)に対応するものが用意されている。

関係代数演算：選択処理、射影処理、和演算処理、積演算処理、差演算処理、結合演算処理、直積演算処理、整列処理

オブジェクト・タイプの変換：ハッシュ表の生成、ストリームからテーブルへの変換、集合からテーブルへの変換

関係の更新処理：レコードの追加、レコードの更新、レコードの削除

関係の構造操作：中間言語レベルでのネスト/アンネスト処理用コマンド

外部プロセスとのインタフェース用コマンド：スキーマの変更コマンド、テーブル・プロセスの生成コマ

ド

その他：関係に対するレコードの読み出しコマンド、リモート・アクセス・コマンド

4.3 中間言語の生成に関する基本方針

中間言語が提供する各オブジェクトの特徴から、部分関係代数列から実行効率の良い中間言語列を生成するための基本方針として以下のものを考えている。

- 関係演算のうち、和演算や直積演算は実行中に与えられた関係の全てのレコードを必要とするが、差演算、積演算、および結合演算はそうではない。特に、後者の演算については、オブジェクトの特性を活かすことにより、実行効率の良い中間言語列を生成することができる。
- ストリーム同士の演算の場合は、処理対象として与えられた全てのストリームを同一の属性で整列することにより、マージ処理として扱うことができる。
- ストリーム、集合、ならびにテーブルが混在している演算の場合は、以下の観点から、レコードをテーブルから読み出す回数を可能な限り少なくするようにする。

- テーブル(集合と対になっているテーブルを含む)に索引属性がある場合は、ストリームを流れてくるレコード中の索引属性値を利用して、テーブルの索引読みを行なう。
- テーブルにハッシュ表が作られている場合は、まずストリームを流れてくるレコード中の索引属性値を使ってハッシュ表からレコードID情報を取り出した上で、そのレコードID情報で指定されたレコードをテーブルから直接取り出す。
- テーブルが索引属性とハッシュ属性のどちらも持たない場合は、そのテーブルにハッシュ表を作成し、上述の方法を適用する。

5 おわりに

現在、Kappa-Pの中間言語処理部は、個々のコマンドに関してPIM上での実装が終了している。今後は、複数LDBMSに渡る処理に対する評価、LDBMS間のデータの受け渡しに関する評価、ならびにこれらの評価結果に基づく中間言語コマンドの見直しを順次行なっていく予定である。

参考文献

- [1] 瀧：“Parallel Inference Machine PIM” The International Conference on Fifth Generation Computer Systems '92, 1992.
- [2] 河村ほか：並列データベース管理システムKappa-Pの概要，第45回情報処全国大会，5R-3，1992-10.