

3H-10

リレーショナルDBMSにおける 集合フェッチ方式の提案

茂木啓次

(日立マイクロコンピュータエンジニアリング(株))

武藤英男

(株)日立製作所システム開発研究所

1. はじめに

リレーショナルDBMS (Database Management System) の特徴の一つに集合操作がある。すなわち、リレーショナルDBMSの検索・更新の要求は、"何を"検索・更新するかの指定であり、指定した条件を満足する全てのタプルが処理の対象となる。しかしながら、ホスト言語インタフェースでの検索は、従来通りの1タプルずつの受け渡しである。このため、以下の問題が発生する。

- (1) プログラム内の配列変数等に、集合検索した結果をまとめて取り出すことができない。
- (2) UAP (User Application Program) と DBMS との間の転送オーバーヘッドが、タプル数が増えるのに比例して増大する。

上記問題点を解決するものとして、一回のフェッチで複数個のタプルの取り出しを可能にする集合フェッチ方式を提案する。集合フェッチ方式実現のために以下を行う。

- (1) 検索タプルの集合フェッチ方式の設定
- (2) ホスト言語インタフェース機能の拡張

```
EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 XENO          PIC S9(4) COMP.    [SQL文中で
01 XENAME        PIC X(16).         使用するホス
01 SQLCODE       PIC S9(4).         ト変数の宣言]
EXEC SQL END DECLARE SECTION END-EXEC.
EXEC SQL DECLARE CURSOR FOR        [カーソル
      SELECT ENO,ENAME FROM EMP    宣言文]
      WHERE DEPTNO=10 END-EXEC.

EXEC SQL OPEN CURSOR END-EXEC.     [オープン文]
FETCH.
EXEC SQL FETCH CURSOR              [フェッチ文]
      INTO :XENO,:XENAME END-EXEC.
IF SQLCODE=100 THEN GO TO FINISH.
PERFORM OUTPUT.
GO TO FETCH.
OUTPUT.
[1個の検索タプルの出力処理]
FINISH.
EXEC SQL CLOSE CURSOR END-EXEC.    [クローズ文]
EXIT PROGRAM.
```

図1 従来のプログラム例(COBOL)

2. 現状のホスト言語インタフェース

現状のホスト言語インタフェースでの検索の例を図1に示す。データベース言語として、ISOでの標準化が進められているSQLを使用する。SQLでは、FETCH文により検索タプルを一個ずつ取り出す。

従来のプログラム・インタフェースでは、タプル取り出し要求(FETCH文)に対し、図2のようにタプルを転送していた。一個タプルを転送する度にUAPとDBMSとの間で制御の受け渡しが発生する。このオーバーヘッドは、タプル数が増えれば増えるほど性能への影響が大となる。

このオーバーヘッドの削減のため、一回の取り出しで複数個のタプルの取り出しを可能とする集合フェッチ方式を提供する。

3. 集合フェッチ方式

集合フェッチ方式の概要を図3に示す。UAP側には、UAP自身の他に、UAPとDBMSとの間のインタフェース部分に位置するPIM(Program Interface Module)が存在する。PIMとDBMSとの間で集合フェッチを行い、データ転送時間の短縮を図る。

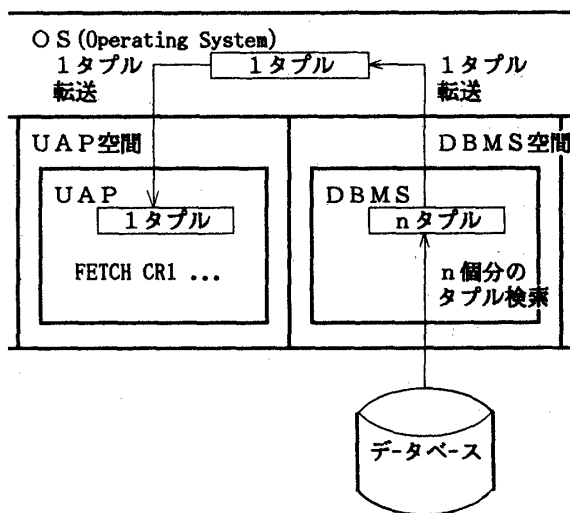


図2 従来のタプル転送手順

Multiple Tuple Fetch Facility in Relational DBMS

Keiji MOKI¹, Hideo MUTOU²

1) Hitachi Microcomputer Engineering Ltd. 2) Systems Development Laboratory, Hitachi, Ltd.

集合フェッチ方式では、タプルの集合を受け渡すするために、PIMに次の機能を追加する。

- (1) 集合フェッチで受け取るテーブル数の決定、およびDBMSへの通知機能
- (2) DBMSから受け取ったテーブル群の管理、およびUAPへの転送機能
- (3) 集合フェッチで受け取るテーブル群を格納する領域の管理機能

集合フェッチするテーブル数は、テーブル長や、PIMで用意できる受け取り領域長、空間間のデータの受け渡しに使用する領域長により、決定する。

また、DBMS側には、次の機能を追加する。

- (1) 集合フェッチ要求時の複数テーブル転送機能
- (2) 転送した有効テーブル数の連絡機能

4. ホスト言語インタフェースの拡張

検索結果を集合として扱うようなUAP、例えば検索結果を、そのまま他のUAPや計算機に転送するようなUAPの場合には、検索結果を一個ずつ受け取るよりは、まとめて受け取った方が処理しやすい。このようなUAPに対して集合フェッチ機能を提供するためにホスト言語インタフェースを拡張する。

UAPで集合フェッチを利用するためには、前述のPIMと同等の機能が必要になる。集合フェッチ機能は、SQLのフェッチ文の仕様を拡張して実現する。

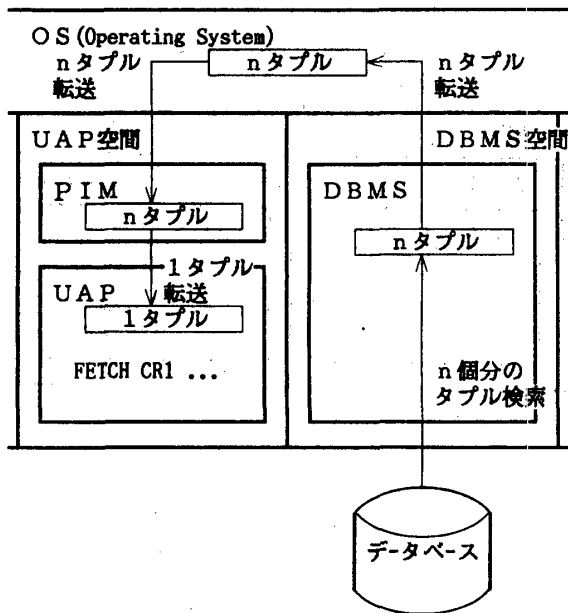


図3 集合フェッチ方式の概要

拡張したフェッチ文を使用したプログラム例を図4に示す。UAPがPIMと同等の機能を実現できるように、従来のフェッチ文に以下の仕様を追加する。

- (1) 一回の集合フェッチで受け取るテーブル数の連絡
カーソル名の後に、最大転送テーブル数を指定する。転送テーブル数が1の場合は、省略可能であり従来のフェッチ文と同じ形式になる。実際に転送されたテーブルの個数は、FOR句で指定したプログラム変数に取得する。
- (2) テーブル単位の受け渡し
アトリビュート単位ではなく、テーブル単位の受け渡しとし、受け取り領域にはテーブル群を配列形式で受け取る。

5. 終わりに

集合フェッチ方式の実現により、次のような効果を期待できる。

- (1) 大量のテーブル検索時の転送オーバーヘッドの削減
 - (2) ユーザの使い勝手の向上
 - (3) ハードウェア支援への適合による性能向上
- 本稿では、テーブルの検索についてのみ示したが、テーブルの追加についても同様である。

参考文献

- 1) ISO: Database Language SQL (March 1986)
- 2) 福嶋,他: リレーショナルデータベース管理システム“RDB1”, 日立評論, 66, 8, 37-42 (昭59-8)

```

EXEC SQL BEGIN DECLARE SECTION END-EXEC.
01 XEMP OCCURS 200. [SQL文中で
02 XENO PIC S9(4) COMP. 使用するホス
02 XENAME PIC X(16). ト変数の宣言]
01 ECNT PIC S9(4).
01 SQLCODE PIC S9(4).
EXEC SQL END DECLARE SECTION END-EXEC.
EXEC SQL DECLARE CR1 CURSOR FOR [カーソル
SELECT ENO,ENAME FROM EMP 宣言文]
WHERE DEPTNO=10 END-EXEC.

EXEC SQL OPEN CR1 END-EXEC. [オープン文]
FETCH.
EXEC SQL FETCH CR1(200) [集合フェッチ
INTO :XEMP FOR :ECNT END-EXEC. 文]
IF SQLCODE=0 OR SQLCODE=100
THEN PERFORM OUTPUT.
IF SQLCODE=100 THEN GO TO FINISH.
GO TO FETCH.
OUTPUT.
[ECNTに返された個数分の検索テーブルの出力処理]
FINISH.
EXEC SQL CLOSE CR1 END-EXEC. [クローズ文]
EXIT PROGRAM.
    
```

図4 集合フェッチ機能を利用したプログラム例(COBOL)