

機能ディスクシステム (FDS-R) への QUELサブセットの実装

5H-8

中野 美由紀, 喜連川 優, 高木 幹雄

東京大学 生産技術研究所 機能エレクトロニクス研究センタ

1. はじめに

現在、我々は、関係データベース指向の超高速二次記憶システム：機能ディスクシステム (Functional Disk System: FDS-R) の開発を行っている。今までに、試作機を作成し、基本的な関係データベース演算による性能評価を行ってきた¹⁾。この結果、FDS-Rが他のシステムと比較して、高い性能をもっていることがわかった。ただし、この性能評価は、試作機上で直接C言語を用いて、それぞれの関係データベース演算を単独に記述し、実行したものであり、問い合わせ言語・インタフェースは提供されていなかった。今回、FDS-Rの機能を踏まえて、新規に試作機上で問い合わせ言語・インタフェースを開発している。本稿では、この新しく開発したシステムの機能、実装方式、実装上の課題について述べる。

2. QUELサブセット・システムの設計

新規に開発した関係データベース・インタフェースは、INGRES¹⁾で用いられている問い合わせ言語QUELのサブセットとなっている。これをQUELサブセット・システムと呼ぶ。QUELサブセット・システムはFDS-R上での問い合わせ言語・インタフェースとして、大きく、つぎの機能があげられる。まず、ユーザ・インタフェース部分に関しては、FDS-Rとしての特別な仕様はなく、かなり制限された形ではあるが、QUEL言語のサブセットが提供されている。次にユーザから指定された問い合わせ (QUERY) をFDS-Rに特有なインタフェースに合わせたQUERYパケットに変換する機能、そして、QUERYパケットを実行するためにFDS-Rを制御する機能がある。後の2つはFDSに固有な機能を活かすために従来のデータベース・システムとは異なった機能が必要である。

2.1 FDS-Rの機能とその特徴

システムに構築にあたって、FDS-Rに特有な機能について考察する。FDS-Rではディスクの制御を高速かつ強力に行うために、新しく開発された高機能ディスクコントローラIDC (Intelligent Disk Controller) を有している。FDS-Rは、このIDCを通じてディスクに対し、通常のトラック、セクタによる制御ではなく、より高水準のデータ構造による制御を行っている。具体的には、レコードのクラスタ化、フィールドの抜き出し、数値比較等の指定された条件によるレコードの取捨選択等である。これらの処理は、ディスクからデータを取り出す際にIDCがハードウェア・フィルタとして機能することにより、実現される。条件等の設定は、FDS-Rのマスタ・プロセッサからシステム・バスを通してIDCのマイクロプロセッサに送られる。IDCは、条件に従い、ディスク内のデータを大容量のステージング・バッファに書き出す。

次にFDS-Rではステージング・バッファ上に取り出されたクラスタ化されたデータをマルチプロセッサで並列処理する。つまり、ステージング・バッファ上のクラスタ化されたデータは、個々のプロセッサによって並列にプロセッサ内のメモリに取り込まれ、クラスタ単位で処理される。

2.2 QUELサブセット・システムの設計

今回FDS-R上で実装されたQUELサブセット・システムは、

前述の特徴を有効に利用するために以下の機能が必要である。1. ユーザからのQUERYを解析する時点でIDCに対するフィルタ条件を抽出する。2. QUERYを解析する時点で並列性に対する条件を抽出する。3. 指定されたQUERYに従って、効率の良いIDC制御、マルチプロセッサ制御およびステージング・バッファ (共有メモリ) の利用を行う。

特にQUERYの解析に伴うIDCフィルタ条件の抽出は、注意しなければならない。QUERYで指定されるデータ取り出し条件からIDCが実行できる単位のフィルタ条件に変換する際には、効率の良い組合せの単位にすることが必要である。

また、IDCの制御とマルチプロセッサ制御が共にステージング・バッファを資源として利用する (データの書き込み、プロセッサ間通信等) ため、ステージング・バッファの管理も必要である。

3. QUELサブセット・システムの構成

QUELサブセット・システムは、前回報告したFDSコミュニケーション・システム²⁾の上で、図1に示す構成をしている。

このシステムは、2つのプロセスから構成され、それぞれ、QUELサブセット・プロセスとプロシジャ・ハンドラと呼ばれている。

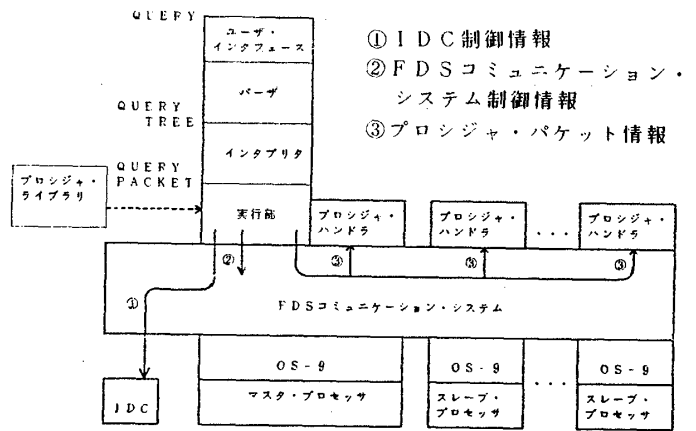


図1. QUELサブセット・システムの構成

3.1 QUELサブセット・プロセス

QUELサブセット・プロセスは、マスタプロセッサ上にあり、ほとんどのシステムの制御と資源の管理を行っている。図からわかるように大きく4つのコンポーネントと1つのライブラリから成っている。

▷ユーザ・インタフェース部：ユーザからのQUERYを受け取るための会話型のタスクである。文単位でQUERYの設定が行なえる。

Implementation of QUEL subset on FDS-R

Miyuki Nakano, Masaru Kitsuregawa, Mikio Takagi

Institute of Industrial Science, The University of Tokyo

▷パーザ部: ユーザが指定したQUERYの文を解析し、問い合わせ木(QUERY TREE)を作る。図2の(b)にQUERY TREEの例を示す。

▷インタプリタ部: QUERY TREEを解釈することにより、実際にFDSとして動作すべき情報に変換したパケットを生成する。このとき、データを実際にプロセッサ上で処理する手順をプロシジャ・パケットに設定し、データを取り出す際のIDCに対するフィルタ条件をIDC制御パケットとして作成、マルチプロセッサの並列制御あるいはステージング・バッファ(共有メモリ)管理のための情報をシステムテーブル制御パケットとして作成する。図2の(c)にQUERY TREEとQUERYパケットの対応を示す。

▷実行部: 実行部は、パケットに従い、システムに必要な資源や制御情報を定められた順序で必要に応じて、各システム制御部(IDC, FDSコミュニケーション・システム, プロシジャ・ハンドラ)に転送する。また、実行部は並列処理のためのプロセッサとIDCの同期、ステージング・バッファの管理をFDSコミュニケーション・システムを通して行う。

▷プロシジャ・ライブラリ: 今回のシステムで提供されている集計演算, AND/OR/NOT等のデータ処理を行うためのプログラム・モジュールのライブラリである。

3.2 プロシジャ・ハンドラ

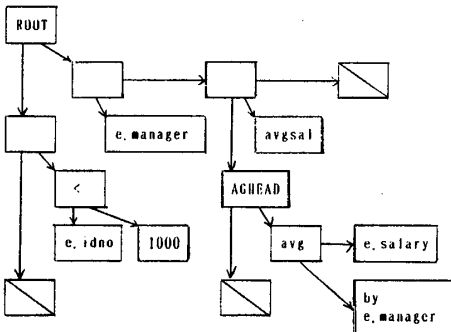
プロシジャ・ハンドラは各プロセッサ上にあるプロセスでステージング・バッファ上のデータを並列処理を行う。データの処理は、プロシジャ・パケットに記述されているプログラム・モジュールに従う。また、プロシジャ・パケット内の情報に従い、中間リレーションをステージング・バッファの指定された位置に書き込みを行ったり、あるいは結果リレーションの表示を行う。

4. インタプリタの動作とFDSインタフェース

今回のQUELサブセット・システムの最も重要な部分はインタプリタ部である。ここで、IDC制御情報と並列処理の可能性について解析する。今回は、最適化については考慮しておらず、IDCの機能を最大限利用することを目的にパケットを構成することとした。

```
range of e is employee
retrieve (e.manager,
         avgsal=(e.salary by e.manager))
where e.idno < 1000
```

(a) QUELによるデータの記述



(b) QUERY TREEの構成例

IDC 制御
パケット

システム・
テーブル
制御
パケット

プロシジャ
・
パケット

クラスタリング・フィールド	e.manager/2 byte
フィールド取り出し情報	e.manager/2 byte, e.salary/4 byte
レコード選択条件	e.idno/4 byte < 1000
並列処理情報	プロシジャ終了までプロセッサは全て有効
共有メモリ管理情報	プロシジャ終了までFDSモード
リレーション情報	total 6 byte, 2 アトリビュート
アトリビュート情報	e.manager/2 byte/integer, avgsal/4 byte/floating
関数パラメタ情報	avg(0, 2 byte, 2, 4 byte)
モジュール・ポインタ	&(avg())
復帰情報	nクラスタ, mレコード, lバイト処理情報の返却
有効共有メモリ番地	300000H + #プロセッサ*200H
有効共有メモリサイズ	200H

(c) QUERY PACKETの構成例

簡単にQUERY TREEの解析の流れについて説明する。

- (1) WHERE 節内の条件でIDC制御パケットを一端作成する。ここで抽出された条件が優先される。
 - (2) (1)でIDC制御パケットとして作成されなかった条件(複数の比較条件, 集計演算を含んだ条件等)をプロシジャ・パケットとして作成する。
 - (3) TARGET LIST内の条件によってIDC制御パケットを作る。ここで対象となるのは、グループ単位の集計演算である。この時、(1)で作成されたパケット情報を付加する。
 - (4) TARGET LISTのRESULTアトリビュートの情報をリレーション・テーブルを元にプロシジャ・パケットに付加する。また、集計演算に対応するモジュールをプロシジャ・ライブラリから検索し、プロシジャ・パケットに設定する。
 - (5) 今までに作成されたプロシジャ・パケットをまとめて、IDC制御パケットとの実行順序を決め、共有メモリの使用情報、復帰情報等を設定して、最終的にプロシジャ・パケットを完成させる。
 - (6) 作成されたIDC制御情報のクラスタリング情報とプロシジャ・パケットの共有メモリ情報を元に、並列処理のフラグとステージング・バッファの制御順序をシステム・テーブル制御パケットに設定する。
- このようにして作成されたIDC制御パケット、システム・テーブル制御パケット、プロシジャ・パケットを組にして、QUERYパケットと呼ぶ。QUERYパケットは、1つのQUERYを満足するまでいくつか作成され、実行部に渡される。

5. 終わりに

簡単にQUELサブセット・システムについて説明した。本システムは、FDSの問い合わせ言語インタフェースとして、試作機上で実装されている。現在、その機能拡張及び性能評価を行っている。今後、よりFDSに適合するようにQUERY最適化とシステムの強化を行いたい。

[参考文献]

- 1) Stonebraker M, et al. The Design and Implementation of INGRES, ACM Transaction on Database Systems, Vol. 1, no. 3, Sep. 1976
- 2) 喜連川優, 中野美由紀, 高木幹雄 機能ディスクシステムに於けるシステムソフトウェアの設計, 第32回情報処理全国大会5S-3(1986)
- 3) 喜連川優, 原田リリアン, 高木幹雄 関係データベース処理に於ける機能ディスクシステムの性能評価, 第32回情報処理全国大会5S-4(1986)

図2. QUERY TREEとQUERY PACKETの構成例