情報処理学会第37回(昭和63年後期)全国大会

7Q-3

機能ディスクシステム第2版における結合演算処理の考察

中野　美由紀，平野　聡，喜連川　優，高木　幹雄

東京大学　生産技術研究所

1．始めに

機能ディスクシステム（Functional Disk System with Relational database engine：FDS-R）は，プロセッサ本体と二次記憶システム間のI／Oボトルネックの問題に着目し，二次記憶システムを単なる記憶媒体ではなく，それ自身がデータに対して高レベルな処理機能を持つことによりデータベース・システムとしての二次記憶の性能向上を図ったものである。特に，大容量のステージング・バッファを導入することで，大規模データ処理を効率良くかつ高速に行う新しいデータ関係代数演算を支援する専用ディスク・コントローラを開発し，専用ハードウェアによる"機能"も実装している。すでにFDS－R第1版の試作機を開発し，基本性能についての計測を行い，一般の商用データベース・システムと比較して高い性能を得られることを確認した(1)。続いて第1版で得られた知見をもとに新たにFDS－R第2版を開発し，大規模リレーションに対する関係代数演算の処理方式の実装を行ってい，200MB程度のリレーションを用いた性能評価の結果，他のデータベース・マシンと比較して十分高い性能を確認した(3)。本報告では，FDS－Rで採用している大規模リレーションの処理方式:Nested Loop方式とData Partitioning方式について，第2版試作機上での結合演算の計測結果を用いて行った解析について報告する。

2．FDS－R第2版における処理方式の考察

2．1　測定環境

以下に述べる測定結果は，プロセッサは，MC68020 4台（ローカル・メモリ1MB），ステージング・バッファ6MB，8インチディスク（NEC D2257）1台を用いて計測された。プロセッサ台数は特に記述しない場合には4台である。ステージング・バッファは，大きさを変化させて測定する。測定用のデータベースのレコード長は128バイト固定である。リレーションのタプル数などはパラメータとして測定毎に変化する。今回の測定では，結合演算の問合せを試作機上に実装されている問合せ解析実行システムのQUELサブセット・システム(2)を用いてQUELステートメントとして実行している。

結合演算に用いてNested Loop 方式とData Partitioning 方式の違いを明らかにするために，それぞれのアルゴリズム毎にリレーション・サイズを変化させて以下の問合せについて測定した。

```
range of e is relationR
range of f is relationS
retrieve into tmp (e.all,f.all) where e.a1 = f.a1 and e.a1 <= α
```

ここで用いる2つのリレーション・サイズは等しい。また，フィルタリング・ファクタはαによって指定する。

2．2　結合演算における解析

図1，図2にフィルタリング・ファクタ（α）が0.1の場合のリレーション・サイズの変化によるNested Loop 方式とData Partitioning 方式の全処理時間と処理時間の内訳を示す。図3，図4にフィルタリング・ファクタが1の場合のNested Loop 方式とData Partitioning 方式の処理時間を示す。ステージング・バッファは512KB，クラスタ・サイズは平均10タプルに設定されている。

Nested Loop 方式の全処理時間は，ディスクからのリレーションの読出し時間とデータ処理時間（データに対する演算処理時間と結果リレーションの書き戻し時間）から構成されている。さらにフィルタリング・ファクタが小さい場合には内側リレーションのフィルタリング後の書き戻しが行われるため（文献(3)参照），内側リレーションの書き戻し時間が入る。今回の計測では図1のフィルタリング・ファクタ=0.1の場合に内側リレーションの書き戻しが行われている。Data Partitioning 方式の全処理時間は，ディスクからのリレーションの読出し時間，I／Oクラスタ生成時間，I／Oクラスタ読出し時間，データ処理時間（データに対する演算処理時間と結果リレーションの書き戻し時間）から構成されている。I／Oクラスタ生成時間は，データをクラスタ化するための前処理時間，データ処理時間，I／Oクラスタの書き戻し時間から構成される。また，いずれの方式でもデータの処理は4台のプロセッサで並列に処理されるため，データに対するプロセッサ上での演算処理と結果リレーションの書き戻しは並行し

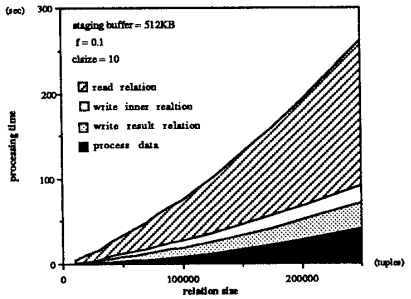

図1．結合演算：リレーション・サイズによる処理時間の変化 (f=0.1)　Nested Loop 方式の場合

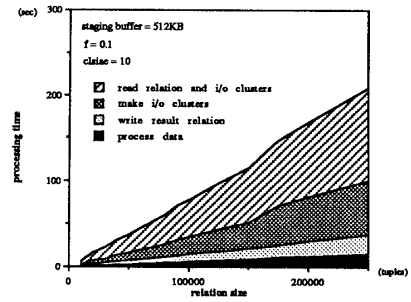

図2．結合演算：リレーション・サイズによる処理時間の変化 (f=0.1)　Data Partitioning 方式の場合

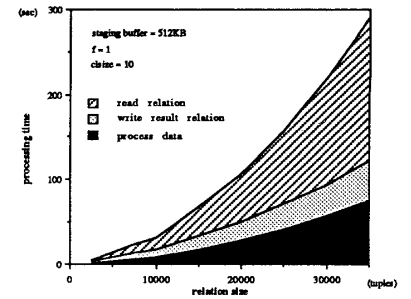

図3．結合演算：リレーション・サイズによる処理時間の変化 (f=1)　Nested Loop 方式の場合

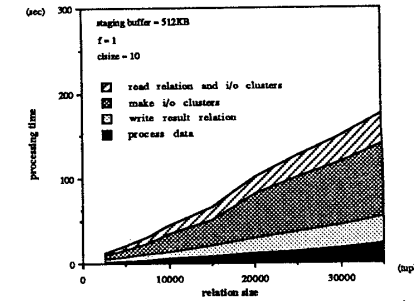

図4．結合演算：リレーション・サイズによる処理時間の変化 (f=1)　Data Partitioning 方式の場合

Join Query on FDS-R2
Miyuki NAKANO, Satoshi HIRANO, Masaru KITSUREGAWA, Mikio TAKAGI
Institute of Industrial Science, University of Tokyo

て行われる。今回の図1，2，3，4では，参考のために結果リレーションの書き戻し時間を除いたデータに対する演算処理時間を黒色で示している。

Nestd Loop方式では図1，3からわかるように，リレーションの読み出し時間はリレーションが大きくなるにつれてO(R*S) で増大する。特に図3のフィルタリング・ファクタが1の場合は明らかである。しかし，フィルタリング・ファクタが小さい場合には，2回目からの内側リレーションの読出しにはフィルタリング後のリレーションを用いるため，図1からリレーションの読出し時間の増加率が図3と比較して小さいことがわかる。また，データ処理時間もリレーションの読出し時間同様にO(R*S) で増加していることがわかる。

Data Partitioning 方式では，全リレーションとフィルタリング後のI／Oクラスタの読出し時間を加えたものをディスクからのリレーション読出し時間として図2，4に示している。Nested Loop 方式と異なりリレーションの読出し時間はほぼ，リレーション・サイズに比例している。今回の計測では，フィルタリング・ファクタが1の図4の場合に明らかであるが，I／Oクラスタ生成時間が全処理時間の中で占める割合が高い。これは，一つには，今回のシステムにおいてディスクの読出し時間より書き戻し時間がほぼ4倍程度かかるためである。つぎにリレーション・サイズが大きくなるにつれて，I/O クラスタを生成するためのオーバーヘッド時間が増大するためである。つまり，I/O クラスタを生成するためには，ソース・リレーションを分割し，それぞれの分割片をI/O クラスタ数分ずつディスクに書き戻すため，書き戻しのオーバーヘッド時間はO(ソース・リレーションの分割数*I/Oクラスタ数) で増大するためである。図5にステージング・バッファ・サイズを変化させた場合のData Partitioning 方式の処理時間を示す。この図からわかるように，ステージング・バッファ・サイズが小さくなるとI/O クラスタ数とソース・リレーションの分割数がともに増加することによりI/O クラスタ生成処理時間が増大している。この計測ではリレーション・サイズは一定であるからオーバーヘッド時間の増大によるものであることがわかる。一方，データ処理時間は，Nested Loop 方式と異なり，リレーション・サイズにほぼ比例して増加している。これは，Data Partitioning 方式ではすでにI/O クラスタとしてリレーションが分割されているため，Nested Loop 方式のように何回も同じデータを検索する必要がなく，一回のデータのロードで処理できるためである。

以上のことより図2，4からわかるように全処理時間はData Partitioning 方式の場合にはリレーション・サイズにほぼ比例して増加している一方，図1，図3よりNested Loop 方式の場合にはリレーションが大きくなるに従って急激に処理時間が増加しているといえる。さらに今回システムの制約によりディスクへの書き戻しが読出し時間に比較して遅くなっているのを読出し時間と同じに補正したものを図6,7,8,9 に示す。図6,8 から明らかなようにNested Loop 方式の全処理時間は，ディスクの読出し時間とデータ処理時間がほとんどのため，それほど効果がでないが，図7,9 からわかるようにData Partitioning 方式では全体処理時間が小さくなり，大規模リレーションの処理におけるData Partitioning 方式の有効性が確認できる。

3．おわりに

機能ディスクシステム（ＦＤＳ－Ｒ）第２版試作機上における結合演算の性能評価について報告した。すでに提案したＦＤＳ－Ｒ上での関係代数演算処理方式を用いることにより，ステージング・バッファを越えるリレーションの結合演算を測定し，Nested Loop 方式と Data Partitioning方式の処理コストを考察した。

今後，性能評価の一環として大規模データを用いた集計演算，射影演算等の評価を行う予定である。システムの拡張をめざし，ディスクを複数台用いたＦＤＳ－Ｒのアーキテクチャについても検討を行う予定である。

［参考文献］

(1)M.Kitsuregawa, et.al.,"Functional Disk System for Relational Database,"Proc. of 3rd Int. Conf. on Data Engeneering,pp.88-95,1987
(2)中野　美由紀，他：機能ディスクシステム（ＦＤＳ－Ｒ）における問合せ処理方式　電子通信学会データ工学研究会，DE86-25(March,1987)
(3)中野　美由紀，他：機能ディスクシステム第２版における結合演算処理方式　第３６回情報処理全国大会，5E-1(1988)

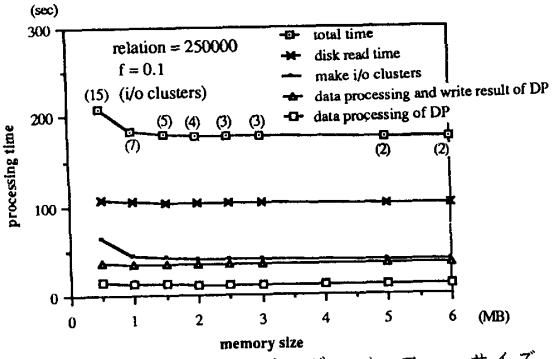

図５．結合演算：ステージング・バッファ・サイズによる処理時間の変化 Data Partitioning 方式の場合

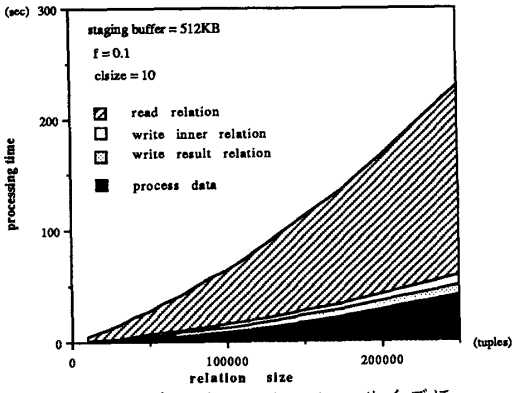

図６．結合演算：リレーション・サイズによる補正した処理時間の変化 Nested Loop 方式の場合 (f=0.1)

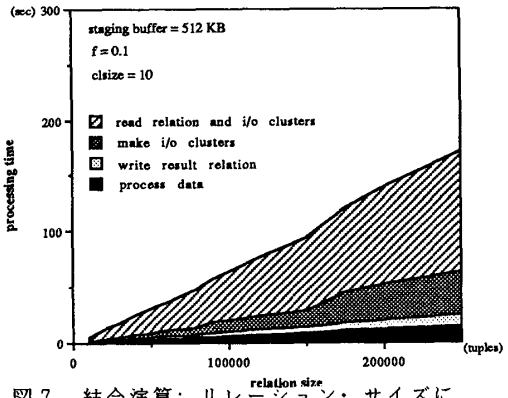

図７．結合演算：リレーション・サイズによる補正した処理時間の変化 Data Partitioning 方式の場合 (f=0.1)

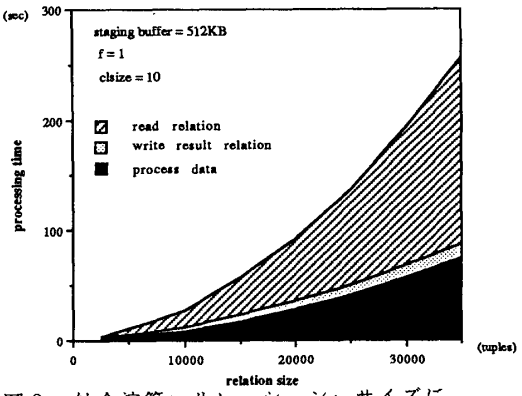

図８．結合演算：リレーション・サイズによる補正した処理時間の変化 Nested Loop 方式の場合 (f=1)

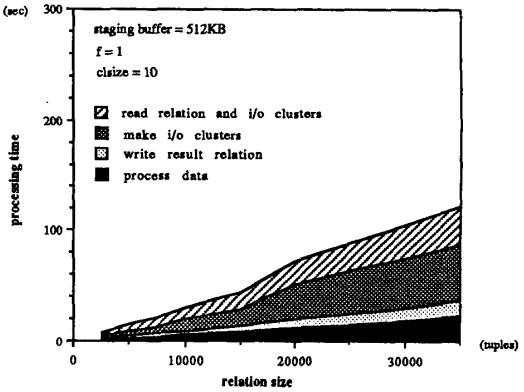

図９．結合演算：リレーション・サイズによる補正した処理時間の変化 Data Partitioning 方式の場合 (f=1)