7Q-8 不均一分布データに対する動的デステージング方式の有効性について

中山雅哉，喜連川優，高木幹雄
東京大学生産技術研究所

1．はじめに

我々は，クラスタリング技法（Hash）に基づく高速結合演算処理方式の提案を行ってきた［1，2］。本方式は，従来から用いられてきたNest Loop 方式や，Sort Merge方式に比べて，特に大容量のデータベースに対して有効であることが，文献［3］等にも述べられている。本稿では，Hybrid Hash 方式等，他のクラスタリング技法による結合処理方式にはみられない，分割バケットの動的な処理順序決定機構（動的デステージング方式）を取り入れた場合の処理性能について考察を行っている。静的に処理順序を決定する方式では，分割バケットのデータ分布により処理性能が大きく左右されていたが，本手法を用いることで，バケットのデータ分布によらず，ほぼ一定の性能を得ることが確認した。

2．本稿における演算処理方式と記号表記について

本稿では，ユーザから発行される一般の問合せ式のうち，制約処理を施した2つのリレーションRとS（R＜Sとする）に対して結合処理を施し，結果をディスクに書き戻す操作のみを対象として取り扱っている。複雑な問合せ式の中間ノードとして働く場合に相当する，結合結果に対する制約演算処理はここでは扱わない。

クラスタリング技法に基づく結合演算処理方式では，対象リレーションR（及びS）が演算処理に利用できる主記憶サイズMより大きい場合，各リレーションを一旦H個の部分リレーション（バケットと呼ぶ）に分割しておき（ステージングフェイズ），次に各バケットを順次取り出してHash操作を用いて結合演算処理を実行していく方法を取る（プロセシングフェイズ）。更にバケット自身がMより大きい場合は，再帰的にバケットを分割していくことで処理を実行するようにしている。

試作システムでは，ディスクとの入出力負荷を抑える為に，主記憶空間を入出力単位である2048バイトの固定長ページに分割，管理している。全体（N＋2）ページのうち，Nページをリレーションのステージング処理に用いており，リレーションSの読み込みと，結果リレーションの書き出し様に1ページずつ割り当てている（図1）。

また，性能評価の際に用いたリレーションR，Sは，共にタプル長64バイトで，結合属性は，キー重複のない4バイト整数フィールドを用いている。

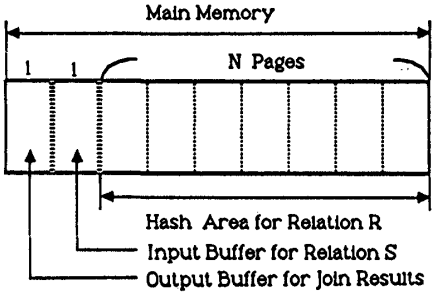

図1　主記憶の利用構成

3．クラスタリング技法を用いた演算処理方式

Nest Loop 方式や，Sort Merge方式との比較は，他の文献［2，3］に譲り，本稿では，前章で概説したクラスタリング技法を用いた結合演算処理の方式について詳細に述べる。

3．1．ステージングフェイズにおける処理

まず，ステージングフェイズにおける分割バケット数Hについては，以下の様な理由から最小になる様に選ばれることが多い。

Ⅰ．プロセシングフェイズにおいては，各バケットを処理する毎にハッシュテーブルの初期化が必要となる。

Ⅱ．Hを大きくすると，各バケットに配分されるタプル数が減少する為，入出力バッファ内の無効領域が増大することが多い。

Ⅲ．分割バケット数を増やすことで，ディスクへの書込みのための入出力バッファ数が増加し，最初の処理に利用できるページ数が減少してしまう。

Hを決定するにあたり，従来から提案されてきた方法では，各バケットのサイズがほぼ均一にNページになると仮定するものが多い。例えば，Hybrid Hash 方式では，

$$H-1 = \left\lceil \frac{|R|-|M|}{|M|-1} \right\rceil$$

$$|R_1| = |M|-H+1$$

$$|R_i| = \frac{|R|-|R_1|}{H-1} \qquad (2 \le i \le H)$$

としてHを決定している（図2）。特にこの手法では，最初に処理すべきバケットR_1と，それ以外のバケットを区別し，R_1の処理は，リレーションSのステージングフェイズとオーバーラップして実行することで，入出力コストの削減を図っている。

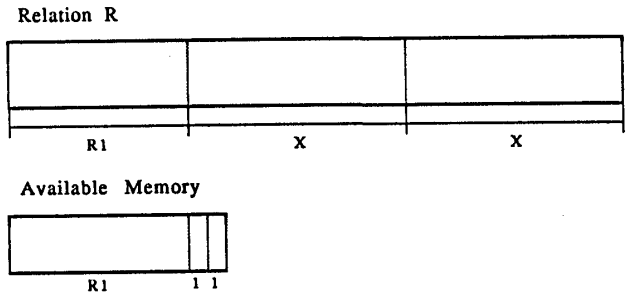

図2　Hybrid Hash 方式におけるリレーションの分割方法

3．2．プロセシングフェイズにおける処理

プロセシングフェイズでは，各バケットに属するタプルを極力分割して，結合演算処理のコストを削減する様にする。このため，ステージングフェイズでは，バケットをペ

Effectiveness of Dynamic Destaging Strategy for Unballanced Data Distributions

Masaya NAKAYAMA, Masaru KITSUREGAWA, Mikio TAKAGI

Institute of Industrial Science, Univ. of Tokyo

ージ単位に連結した集りと考えるが，ここでは，タプル単位に連結した集りと考えて処理を行う。即ち，ステージングフェイズでは，H個の空間に分割して処理し，プロセシングフェイズでは，３２＊Ｎ（＝2048＊Ｎ／64）個の空間に分割して処理することになる。

Hybrid Hash 方式では，各バケット毎に結合演算処理が実行されている。

4．動的デステージング方式による結合演算処理

動的デステージング方式［２］は，従来とられてきた各手法と次の２点で大きく異なっている。

① 従来は，リレーションＲのサイズをもとにして分割バケット数Ｈを最小とする様にしてきたが，本方式ではＲによらず一定の値（Ｎ／２）をとる様にしている。

② プロセシングフェイズにおける各バケットの処理順序は演算処理を実行する前に決定されてきたが，これをデータの処理に伴い，動的にスケジュールする方法をとっている。

前者は，前章で述べた様な欠点を持っているため，従来は取り入れられなかったものであるが，Ⅰ．Ⅱ．に関しては，ステージング処理の終了時に，各バケットのサイズをもとにスケジューリングを行い，小さなバケットはまとめてひとつの処理単位（プロセシングクラスタ）としてプロセシングフェイズにおいて一度に処理する様にアルゴリズムの改良を行うことで解決している。Ⅲ．に関しては，対処できる方法はないが，最初のプロセシングクラスタの処理におけるメモリ利用効率のみが低下するだけなので，大容量のデータに対する処理全体の中で占める割合は低い。これに対し，より多くのバケットで分割することで，少々のバケットサイズの変動に対しても，各々が主記憶容量を超えないで済むことに伴うオーバーヘッド軽減の効果は高いといえる。

後者は，従来の方式では考慮されていなかった各バケットのサイズが不均一となる場合に対応するための機構である。主記憶空間が足りなくなった時点で最もサイズの大きいバケットは，優先的にデステージングされて後で処理するバケットとして扱われる。この方式を用いることで，事前に予測したバケットの分布と実際のデータによるバケットの分布が異なる場合でも，最初に処理されるバケットや，その後の処理バケットのスケジューリングを行うことができる様になる。

5．動的デステージング方式による演算処理の性能評価

２章でも述べた様に，動的デステージング方式をとる試作システムをワークステーション SUN3/260 上に作成し，不均一なバケット分布をとる２つのリレーションを結合した時の演算処理性能を測定した。本章はその結果についてまとめている。

不均一なバケット分布のリレーションとは，図３に示す様に，第ｋバケットのサイズR_kが，第１バケットのサイズR_1のｋ倍になる（三角分布）データ分布をとるリレーションを指している。ここで，各バケットのサイズＲ　は，$\Sigma R_k = R$をもとにして計算される。

まず，Hybrid Hash 方式と，動的デステージング方式について，バケット分布が均一の場合と不均一の場合の性能を比較すると，図４に示す様になる。Hybrid Hash 方式では，バケット分布により大きく性能が変動するのに対し，動的デステージング方式では，不均一なバケット分布に対しても殆ど性能低下が見られないことが分る。これは，先に述べた様に，分割バケット数を大きくして，動的にバケットの処理順序をスケジュールすることで，プロセシングクラスタのサイズをほぼＮにすることが可能となるためである。これは，図５の様にＨ＝500 に固定した時，主記憶サイズＮを変化させてもそれほど性能低下を招かないことからも明らかである。

この他にも，バケット分布を図６の様にした場合も，ほぼ同様の結果を得ており，データ分布の変動に対して柔軟に対応できることが明らかとなった。

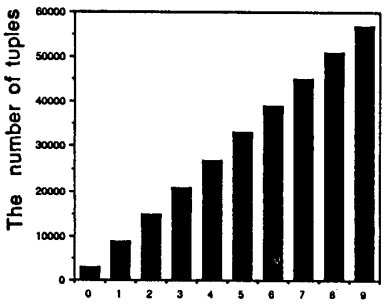

図３　不均一なデータ分布例１

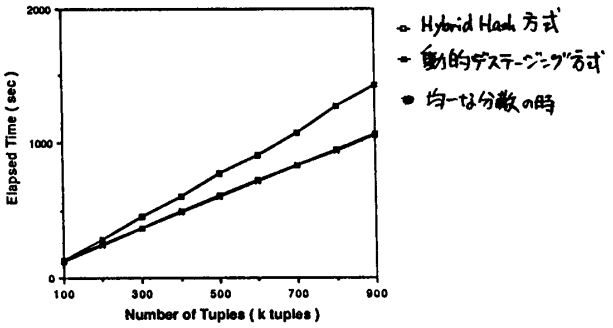

図４　各方式の性能評価結果

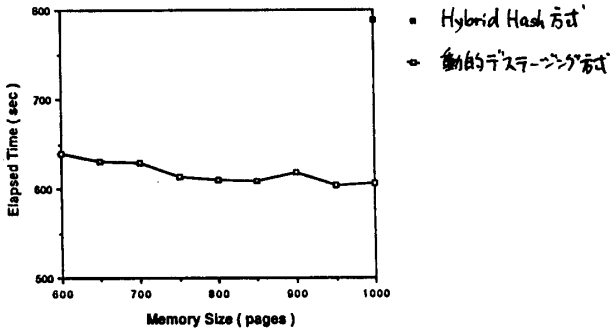

図５　主記憶サイズを変動させた時の性能

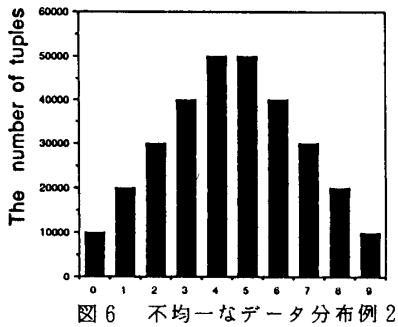

図６　不均一なデータ分布例２

6．まとめ

本稿では，動的デステージング方式を用いた結合演算処理方式について，従来から提案されてきたHybrid Hash 方式との比較を行い，提案する方式の柔軟性について述べてきた。また，ワークステーション上に実装した試作システムを用いて，分割バケットのデータ分布が不均一な場合の演算処理性能について，測定結果をまとめている。この結果，データの分布特性によらずほぼ同程度の性能が得られることが示されている。

参考文献

［１］中山他，「クラスタリング技法に基づく大規模リレーションの結合演算処理方式とその評価」，36回情報全大，3F-5，1988

［２］中山他，「動的クラスタリング技法を用いた結合演算処理の性能評価」，信学技法，DE87-21，1988

［３］D. DeWitt, et al.,「Multiprocessor Hash-Based Join Algorithms」, Proc. of Conf. on VLDB 1985