論　文

UDC 681.322.078

関係代数マシンGRACEにおけるバケット収集網

正員　坂井　修一[†]　　正員　喜連川　優[††]
正員　田中　英彦[†††]　　正員　元岡　達[†††]

Interconnection Network for Bucket Collection on Relational Algebra Machine GRACE

Shuichi SAKAI[†], Masaru KITSUREGAWA[††], Hidehiko TANAKA[†††] *and* Tohru MOTO-OKA[†††], *Members*

あらまし　現在，大規模データベースに対する複雑な問合せを高速に処理するデータベースマシンへの需要が急速に高まってきている．我々はこうした需要に応える高性能データベースマシンを開発中であり，その関係代数処理部をＧＲＡＣＥと名づけている．ＧＲＡＣＥは多モジュール構成の並列マシンであるため，各構成要素間を結ぶ相互結合網の役割は重要である．ＧＲＡＣＥの相互結合網は機能上，ステージング空間へデータを送り込むバケット分配網と，プロセッサ群へデータを送り込むバケット収集網に分類される．本論文は，関係代数マシンＧＲＡＣＥにおけるバケット収集時のデータ流の制御方式と，バケット収集網の実現方式を提案し，シミュレーションによる評価・検討によってその正当性を確認したものである．ＧＲＡＣＥの処理方式の特徴を活かすため，結合網として多段結合網の一種である間接キューブ網を採用し，きわめて低い転送オーバヘッドを得た．さらに複数の演算による結合網のパーティショニング問題を検討した．

1. まえがき

データベースマシンの研究開発は，１９８０年代に入ってオフィスを中心としたビジネス・ユースを目的とする商用マシンの誕生をみたが，一方で大容量データベース（10^{13}～^{15} Byte）を効率良く処理することの可能なマシンへの需要が急速に高まってきている．また，応用分野によっては，従来のような単純な検索ばかりでなく，複雑な関係代数演算が支配的な環境，いわゆるジョイン・ドミナントな環境への対応能力を要求されるものも多い．

すなわち高速で大容量のデータベースマシンの実現が課題であるが，こうしたマシンは，一般に複数台のプロセッサ，多バンク化されたステージング空間および複数の二次記憶装置から構成され，データベース処理が内包する並列性の抽出により性能向上をはかっている．このようなシステムでは，各構成要素間の通信オーバヘッドが全体の処理能力を決定すると考えられ，高い転送レートのモジュール間結合網の実現が課題となる．

我々は現在，大容量データベースに対して高速の関係代数処理を行うマシンＧＲＡＣＥ[(1)～(3)]を開発中である．本論文ではＧＲＡＣＥに関して，その処理方式に適合した結合網およびデータ転送の制御方式を提案・評価・検討した結果を報告する．今回は本マシンのモジュール間結合網のうち，データをプロセッサ群に送り込むバケット収集網に関して述べる．結合網はおもに多段結合網を対象とし，シミュレーションによるデータ転送時間の評価を行った．

2. 関係代数マシンＧＲＡＣＥ

ＧＲＡＣＥはハッシュとソートを用いた高速並列処理関係代数マシンである．その著しい特徴として，ＳＥＬＥＣＴＩＯＮやＵＰＤＡＴＥなどの処理負荷の軽い操作のみならず，負荷の重い関係代数演算・集合演

† 東京大学大学院工学系研究科，東京都
Graduate School of Engineering, The University of Tokyo, Tokyo, 113 Japan
†† 東京大学生産技術研究所，東京都
Institute of Industrial Science, The University of Tokyo, Tokyo, 106 Japan
††† 東京大学工学部電気工学科，東京都
Faculty of Engineering, The University of Tokyo, Tokyo, 113 Japan

算に対して高い処理能力を持つことがあげられる．本マシンにより実際的な大きさのリレーションに対するJOIN, PROJECTIONなどは，$O(m)$（m：メモリページサイズ）で実現される．本章では，GRACEの処理方式とシステム構成の概略を示し，モジュール間結合網の役割に関する整理を行う．

2.1 関係代数演算の処理技法

GRACEでは，当該リレーションの動的なクラスタリングのためにハッシュを用いる．今，リレーションA（大きさN）とリレーションB（大きさM）のJOINを取ることを考えれば，最も単純な方法では$O(N\times M)$時間必要であるが，JOINアトリビュートに関して両リレーションにハッシュを施し，全体をS個のバケットに分割したとすると，異なるハッシュ値をもつバケットどうしはJOINの取れる可能性が無いため，処理時間Tは，

$$T \propto \sum_{i=1}^{S} (n_i \times m_i) \quad \left(N=\sum_{i=1}^{S} n_i,\ M=\sum_{i=1}^{S} m_i\right)$$

に軽減される．次にこうして生成されたバケットを多数のプロセッサによって並列処理する．各プロセッサは，内部に$O(n)$時間でソートを完了するハードウェア・ソータを備えており，バケットをその大きさに比例した時間で処理することができる．さらにステージング空間のバンクパラレリズムを反映させることが可能であるため，K台のプロセッサを用いてリレーションを並列に処理することにより，処理時間は$1/K$となる．以上のごとく，本マシンでは全体として$O(m)$($m=(N+M)/K$）の処理を実現している．これはリレーションの大きさによらない，メモリページサイズに比例した一定時間での処理である．

ハッシュ関数を用いたクラスタリングでは，その性質上，ハッシュ後のデータの分布は必ずしも均一にならず，バケットの大きさに偏りが生じてしまう．さらにプロセッサの容量に収まらず，バケットオーバフローを生じることもある．この分散不均一性はハッシュを利用する場合には避け難い特徴であり，一方ハードウェア・アーキテクチャの側から見れば，処理対象の大きさの変動は，リソースの使用効率の低下・性能低下の要因となる．

GRACEではこの問題を以下の方法で解決した．1バケットは，一つのメモリに格納するのではなく，各バケットをそれぞれ複数のメモリに均等に分配し，メモリ内では各タプルをハッシュ値と連結して記憶しておく．このようにして1メモリバンクのオーバフローを防ぎ，かつマルチストリーム間の偏りを無くす．さらに，あらかじめ細かなハッシュを行ってから，バケットを複数個集めてプロセッサの処理能力に合わせるように統合するバケットサイズ・チューニングを行い，バケットサイズの偏りを抑え，プロセッサの使用効率を高めるとともに，モジュール間パイプラインの擾乱を抑える．

また，本マシンはMIMD方式を採り，同時に複数の関係代数演算を効率良く実行できるようにする．

2.2 GRACEのシステム構成

本関係代数マシンの構成を図1に示す．マシンは，4種のモジュールおよびこれらを結合する相互結合網から成る．すなわち，プロセシングモジュール（PM），メモリモジュール（MM），ディスクモジュール（DM），コントロールモジュール（CM）とプロセシング網，ステージング網である．リレーション・データは下のステージング網を介してDMからMMにステージングされ，その後上のプロセシング網を介してPMに送られ，ソート及び関係代数処理が施される．CMはDM－MM，MM－PMの両方のモジュール間に存在し，全体の実行制御をつかさどる．

2.3 GRACEの処理方式と相互結合網

本マシン上でのリレーションの処理を概説する．

DM内のディスクに記憶されている当該リレーションは，同モジュール内のフィルタ・プロセッサによってSELECTIONやPROJECTION（重複除去は含まない）を施され，JOINアトリビュート（あるいはPROJECTIONアトリビュート）に

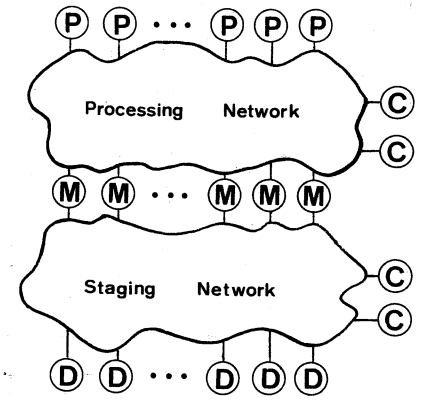

図1 GRACEアーキテクチャ
Fig.1 GRACE architecture.

関してハッシングされた後，MM群に移される．その際，**2.1**で述べた理由から，各バケットはそれぞれMM間で均等な大きさになるように分配される．この段階をステージングフェイズと呼ぶ．

一方，PMによって処理が行われる段階をプロセシングフェイズと呼ぶ．プロセシングフェイズの実行手順を以下に示す．MMに移されたリレーション・データには，バケットサイズ・チューニングが施され，適切なスケジューリングがなされた後，PMに送出される．一つのPMには一つのバケットが対応し，MMを順番にアクセスして，分配された一つのバケットを収集し，ソートを施した後に関係代数処理を行う．処理を終えたPMは再び新たなバケットの処理に取りかかり，全てのバケットを複数のPMにより処理していく．この処理は図2に見られるごとくパイプライン化して

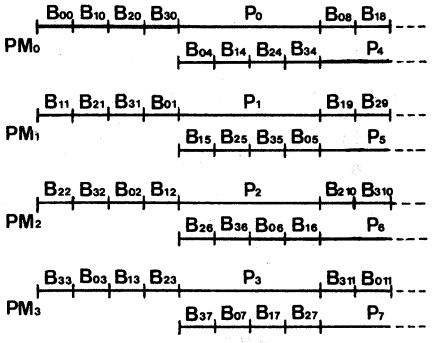

図2　バケット処理パイプライン
Fig.2　Bucket processing pipeline.

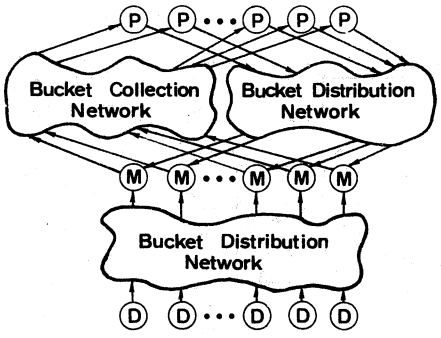

図3　GRACE上の2種類の結合網
Fig.3　2 kinds of networks on GRACE.

行われるが，これをバケット処理のパイプラインと呼ぶ．結果リレーションは，次のオペレーションに用いられる時は再びハッシュを施され，生成されたバケットはMM間に均等に分配される．

以上の関係代数処理を，バケットの流れの点から見直すと以下のようになる．DM・PMからMMへのデータ転送においては，ハッシングによるバケットの生成に続いてその分配が行われている．また，MMからPMへは，分配されたバケットをバケット処理のパイプラインのもとに収集するように，タプルが移動する．先にリレーションの処理をステージングとプロセシングの2フェイズに分けたが，機能的には以上の二つの動作に分類される．このうち，DM・PMからMMにデータを送る動作をバケット分配，MMからPMにデータを送る動作をバケット収集と呼び，前者に携わるモジュール間結合網をバケット分配網，後者に携わるモジュール間結合網をバケット収集網と呼ぶことにする．すなわち，バケット分配網とバケット収集網は図3に示されるものである．

本論文では，バケット収集の制御方式とバケット収集網を対象として評価・検討を行う．

3. バケット収集網

3.1　バケット収集の制御方式

バケット収集の詳細な手順を以下に示す．

(1)　バケット分配が終わった時点では，バケットはPMの容量に比して小さくなるように細かなハッシュが施されているので，これを適切な大きさになるよう統合する（バケットサイズ・チューニング）．

(2)　バケットサイズの昇順にバケットをソートし，この順を転送順序とする．これは，当該演算にかかわる全PMの，あるサイクルでの処理負荷をならし，アイドル時間を減らすためである．

(3)　実際に転送を行う．N台のMMにN台のPMが結合し，各PMがそれぞれサイクリックに当該バケットを取り込んでいく．あるバケットの転送およびソートは，前のバケットの関係代数処理に重畳化して行われる．

理想的なバケット収集処理の様子は，先の図2で表わされる．実際には，バケットサイズの擾乱その他によってPMにアイドル時間が生じる場合がある．

図に示されたように，バケット転送はサイクリックな順序づけがなされているため，バケット収集時のMM・PM両モジュールの対応関係は，基本的には，

$P_N=\{(MM_i, PM_{(i+j) \bmod N}) \mid 0 \leq i, j<N\}$
と表現される規則的なクラス（サーキュラ・シフト）に入る．

3.2 間接キューブ網の導入

バケット収集網の候補として，多段結合網の一種である間接キューブ網[8]（図4）を適用することを検討する．間接キューブ網は多段結合網としては最も段数の少ない $\log_2 N$ 段の網である．各段は，2×2のクロスバスイッチおよびその制御部を含むスイッチング・ユニット（SU）と結線部を持つ．任意の入力モジュール・出力モジュール間で通信のパスを張ることが可能であるが，すべての入出力写像が可能であるわけではなく，閉塞を起こす場合が多いのが難点である．ルーティングは，ソースの番地とデスティネーションの番地（あるいはデスティネーションの番地のみ）から計算されるデスティネーション・タグの1ビットを各段のSUが参照しつつ，スイッチの切替えを行う，いわゆるセルフ・ルーティングが用いられる．

3.3 間接キューブ網における入出力写像の実現

Lawrie[9]はオメガ網に関して，入出力写像の実現可能性を検討したが，以下に間接キューブ網について同様のことを示す．

〔定理1〕 ある入出力写像 $P_N=\{(S_i, D_j) \mid 0 \leq i, j<N\}$（$i$は入力モジュールの番地，$j$は出力モジュールの番地）について，

間接キューブ網上で P_N が閉塞無しに実現可能
$\Leftrightarrow {}^{\forall}(S_i, D_j), {}^{\forall}(S_k, D_l) \in P_N$ について，
$(i \neq k)$ かつ $(i \underset{m}{\equiv} k \Rightarrow j \underset{m}{\not\equiv} l)$
ただし，$m=2^h$（$1 \leq h \leq \log_2 N$），div は整数型の除算とした（証明は略す）．

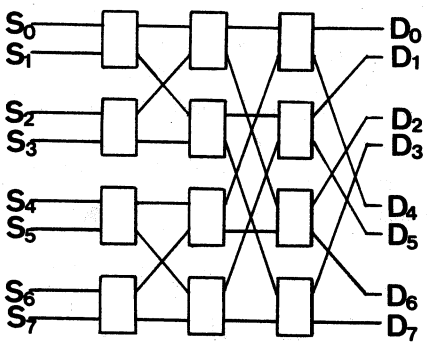

図4 間接キューブ網（3段）
Fig.4 Indirect binary 3-cube network.

定理1は同値条件であるが，これを満足する写像のクラスをいくつか示すことができる．

〔定理2〕 間接キューブ網上では，閉塞無しにサーキュラ・シフト（3.1参照）が実現可能である（証明は付録に記した）．

定理2により，GRACEのバケット収集網として間接キューブ網の適用が有利であると推測される（3.1.4.でその評価に関して述べる．

3.4 間接キューブ網のパーティショニング

一つのマルチプロセッサ系で複数のたがいに独立した演算が行われている時には，相互結合網上で異なる演算どうしの干渉が起こらないようにすることが重要である．

間接キューブ網上では，これはパーティショニング問題[10]として表される．パーティショニングとは，ある相互結合網をたがいに独立な規模の小さい複数の結合網に分割して用いることで，特にもとの網の結合規則が分割してできた網にそのまま保存されている場合をいう．図5に，三つのオペレーションが間接キューブ網をパーティショニングして使用している例を示した．

次に，間接キューブ網をパーティショニングした状態で用いるための十分条件を述べる．

〔定理3〕 ポート数Nの間接キューブ網で結合された多重計算機システム上で，すでにいくつかの独立した演算が実行中であるとする．今，$N/2^k$ 個のソース・モジュール $\{S_i\}$ と，$N/2^k$ 個のデスティネーション・モジュール $\{D_j\}$（$0 \leq k \leq n$，$n=\log_2 N$）が使

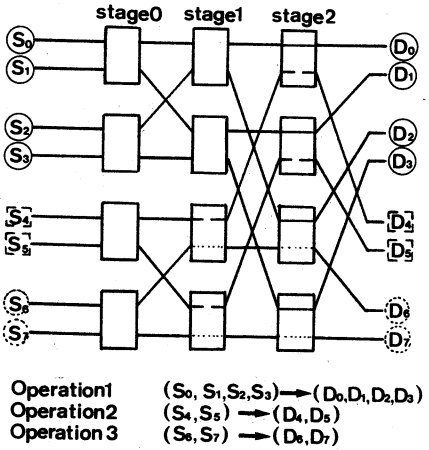

図5 3演算によるパーティショニング
Fig.5 Partitioning by 3 operations.

用可能の時，$\{S_i\}$と$\{D_j\}$の結合がパーティショニングされた網上で実現されるための十分条件は，以下の(1)，(2)で与えられる．

(1) $\forall S_i$，$\forall S_j \in \{S_i\}$，$\forall D_{i'}$，$\forall D_{j'} \in \{D_j\}$，$i^p = j^p$，$i'^p \neq j'^p$ を同時に満足するようなpがk個存在し，$k+1$個以上は存在しない．これをp_1，p_2，…，p_kとする．ただし，i^pはiを二進表示した時の2^pの桁の数字（0か1）である．

(2) $\{S_i\}$，$\{D_j\}$を割り付ける以前の網上で，$\{S_i\}$，$\{D_j\}$の結合に関与するp_l（$1 \leqq \forall l \leqq k$）段目のスイッチの状態が，

$i^{p_l} = i'^{p_l}$のとき状態0，状態1，状態2のどれか

$i^{p_l} \neq i'^{p_l}$のとき状態0，状態3，状態4のどれか

である．ただしp_lは(1)で求められたものとし，各状態（State）は図6で示されるものとした（証明の概略を付録に記す）．

4. バケット収集網のシミュレーションによる評価

本章では，一つの関係代数演算の実行時における，バケット収集の動作に注目する．この時，MMとPMの結合が基本的にはサーキュラ・シフトになることを**3.1**で，サーキュラ・シフトが間接キューブ網上で閉塞無しに実現可能であることを**3.3**で述べた．本章では実際に，バケット収集網として間接キューブ網を用いた時の評価を，シミュレーションによって行う．

バケットの収集時では，1回の結合で転送する単位は，各MM間に分割されたバケットであり，これは普通数KB以上であることから，回線交換方式を採用することにする．

4.1 シミュレーション

図2で表されるように，バケットサイズが均一で各バケットが完全に均等にMMに分配されている場合には，結合はすべての時点でサーキュラ・シフトになるため，間接キューブ網を用いた時，閉塞による転送のオーバヘッドは生じない．実際には，バケットサイズ・チューニングの限界などから，バケット処理のパイプラインは擾乱をきたす．これによって，間接キューブ網の閉塞が生じることもある．この効果を，以下の各場合についてシミュレーションによって測定することにした．

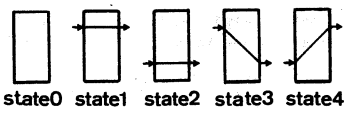

図6 スイッチの状態
Fig.6 States of a switch.

ただし，以下で「バケットサイズの乱れ」とは各バケット間の大きさの擾乱を意味し，「MM間のバケットサイズのゆらぎ」とは，一つのバケットのMM間での分割の不均等性を意味するとする．

(1) MM間のバケットの分割は完全に均等とし，バケットサイズの乱れと転送オーバヘッドの関係を求める．

(2) MM間のバケットの分割は完全に均等とし，バケットサイズの乱れがある時の，網の大きさと転送オーバヘッドの関係を求める．

(3) 全体としてのバケットサイズの乱れは無いとし，MM間のバケットサイズのゆらぎと転送オーバヘッドの関係を求める．

(4) 全体としてのバケットサイズの乱れは無いとし，MM間のバケットサイズのゆらぎがある時の，網の大きさと転送オーバヘッドの関係を求める

(5) バケットサイズの乱れがある時，MM間のバケットサイズのゆらぎと転送オーバヘッドの関係を求める．

ただし，(1)，(2)，(5)はバケットサイズに関するソート（**3.1**の(2)）を施した場合とそうでない場合について調べた．

4.2 結　果

シミュレーションによる(1)～(5)の測定の結果は，それぞれ図7～図11のグラフで表される．この結果をまとめて以下に示す．

(1) バケットサイズに関するソートを施した場合，バケットサイズの乱れは，ほとんど転送オーバヘッドをもたらさない（図7）．

(2) バケットサイズに関するソートは大幅な転送オーバヘッドの軽減をもたらす．場合によっては30％以上の効率向上をみることがある（図7，図8）．

(3) MM間のバケットサイズのゆらぎは，比較的大きな転送オーバヘッドをもたらす（図9，図10，図11）．

(4) 網の大きさは，転送オーバヘッド率（バケット分配が理想的な場合との転送時間の比）にほとんど影響をおよぼさない（図8，図10）．

(5) 間接キューブ網（各図の実線）と閉塞の無い網（各図の破線）の転送時間の差は，ほとんど無い（図7～図11）．

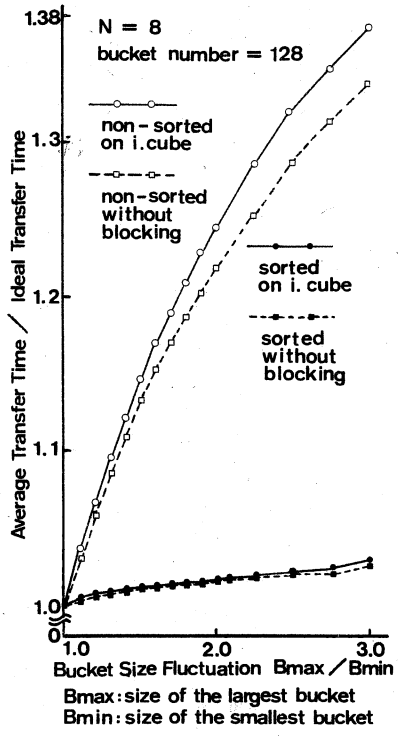

図7 バケットサイズの乱れと転送オーバヘッド
Fig.7 Transfer overhead vs. bucket size fluctuation.

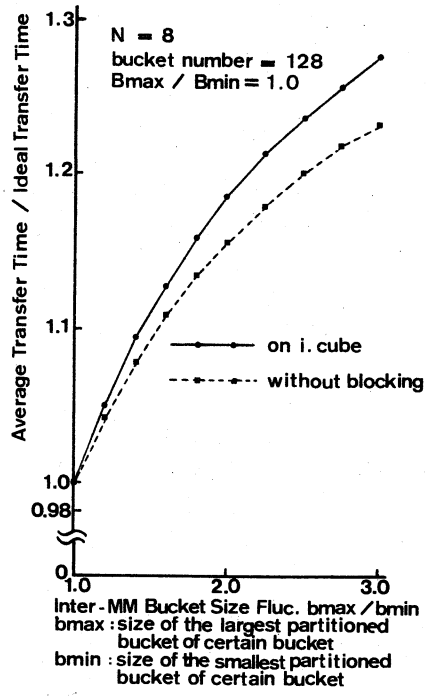

図9 MM間のバケットサイズのゆらぎと転送オーバヘッド(i)
Fig.9 Transfer overhead vs. inter-MM bucket size fluctuation (i).

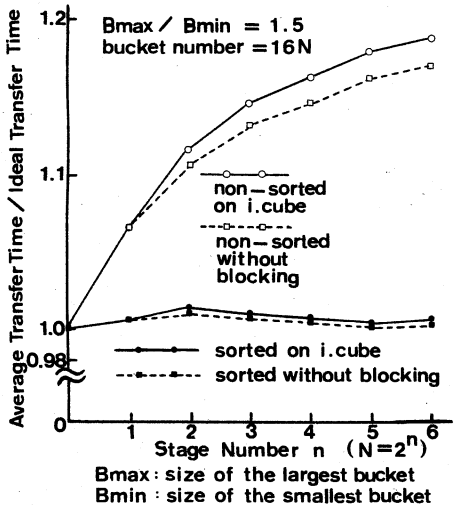

図8 網サイズと転送オーバヘッド(i)
Fig.8 Transfer overhead vs. network size (i).

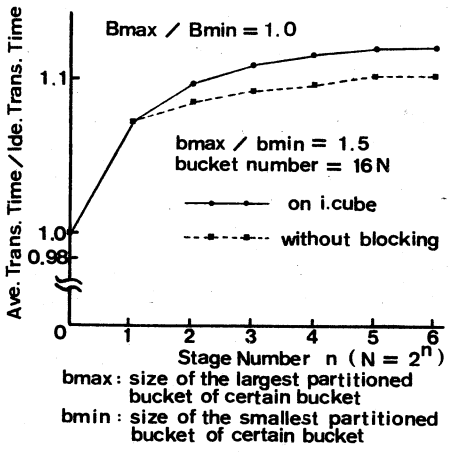

図10 網サイズと転送オーバヘッド(ii)
Fig.10 Transfer overhead vs. network size (ii).

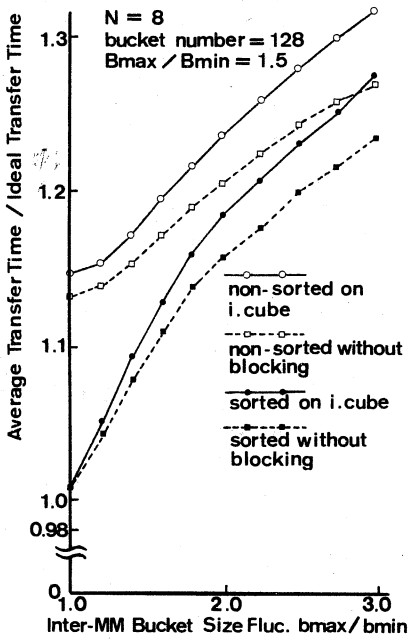

図11　MM間のバケットサイズのゆらぎと転送オーバヘッド(ii)

Fig.11　Transfer overhead vs. inter-MM bucket size fluctuation.

5. 評価・検討

前章の結果より，バケット収集網としては，クロスバスイッチ（接点数N^2，Nはモジュール数）のような大規模な網を用いず，間接キューブ網（接点数$2N \times \log_2 N$）を用いるのが，コスト的に有利であり，性能の面からも適当であることが確認された．バケット転送の際には，バケットサイズに関するソートを施すことが重要である．このスケジューリングによって，バケットサイズの乱れがバケットのパイプライン処理に及ぼす悪影響を低く抑えることができる．したがって，バケットサイズ・チューニングは厳密に行う必要はない．逆に，MM間のバケットサイズのゆらぎは，比較的大きなパイプラインの擾乱をきたす．これを防ぐため，バケット分配時に1バケットの均等な分割を達成するようなタプルの行先制御が必要となる．

以上は一つの演算のみに着目した評価であったが，GRACEでは，同時に複数のたがいに独立な演算が行われ，モジュールの割り付けも動的に行われる．そのため，間接キューブ網を用いる場合，**3.4**で述べたパーティショニングが考慮されねばならない．

バケット収集網として，間接キューブ網以外に，時分割多重チャネル方式の光ファイバ・リングバスを検討している．この方式は高速のデータ転送を行うことが可能であるが，接続されるモジュール台数が大きくなった場合，容量の点で問題が生じる．

GRACEは，データ流に追随した処理を行うマシンであり，その動作速度は，ディスクの転送レートによって規定される．現在最大の転送レートを持つディスクのそれは3MB/sec程度であり，本論文で提案した方式のモジュール間結合網は，1ポートあたりの実効的な転送レートとして十分にこれを達成することができる．

6. むすび

ハッシュとソートを用いた高性能関係代数マシンGRACEにおけるバケット収集網と，それに関係する制御方式に関して述べた．その結果，モジュール間結合の規則性の点から，$\log_2 N$段の多段結合網である間接キューブ網を用いた実現が有利であることを示し，また動的なモジュール割り付けを行う環境のもとでの結合網のパーティショニングに関して検討した．

今後，バケットの分配方式およびそれに携わる相互結合網の検討とともに，バケット収集網で用いられるスイッチング・ユニットの設計や各モジュール内の網インタフェース部の設計・試作を行うことなどが課題である．

文　献

(1) 喜連川，鈴木，田中，元岡：“Hashと Sortによる関係代数マシン”，信学技報，EC81-35（1981-10）．

(2) Kitsuregawa, M., Tanaka, H. and Moto-oka, T.: “Application of Hash to Data Base Machine and Its Architecture”, New Generation Computing, 1, pp.63-74 (1983).

(3) Kitsuregawa, M., Tanaka, H., Moto-oka, T.: “Relational Algebra Machine GRACE”, Lecture Notes in Computer Science, 147, pp.191-214, Springer-Verlag (March 1983).

(4) 坂井，喜連川，田中，元岡：“GRACEに於けるモジュール間結合方式”，第25回情処全大，4P-2（1982-10）．

(5) 坂井，喜連川，田中，元岡：“GRACEに於けるモジュール間結合網とその評価”，第26回情処全大，4F-2（1983-03）．

(6) 坂井，喜連川，田中，元岡：“データベースマシンGRACEに於けるモジュール間結合網”，信学技報，EC83-14（1983-06）．

(7) Feng, T.: “A Survey Interconnection Networks”, IEEE Computer, 14, 12, pp.12

-27 (Dec. 1981).
(8) Pease, M. C. : "The Indirect Binary n-Cube Microprocessor Array", IEEE Trans. Comput., C-26, 5, pp. 548-573 (May 1977).
(9) Lawrie, D. H. : "Access and Alignment of Data in an Array Processor", IEEE Trans. Comput., C-24, 12, pp. 1145-1155 (Dec. 1975).
(10) Siegel, H. J. : "The Theory Underlying the Partitioning of Permutation Networks", IEEE Trans. Comput., C-29, 9, pp. 791-801 (Sept. 1980).

付　　録

1. 定理2の証明

サーキュラ・シフト $CS_N(a)$ は，次の式で定義される．

$CS_N(a)=\{(S_i, D_{(i+a) \bmod N}) \mid 0 \leq i < N\}$

今，$(S_i, D_j), (S_k, D_l) \in CS_N(a)$ $(i \neq k,\ i \operatorname{div} m = k \operatorname{div} m,\ m \mid N)$ とすれば，

$i \bmod m \neq k \bmod m$

i.e. $(i-k) \bmod m \neq 0$

$m \mid N$ ゆえ，

$\{(i+a) \bmod N - (k+a) \bmod N\} \bmod m \neq 0$

$\therefore\ j \underset{m}{\neq} l$　　(q.e.d.)

2. 定理3の証明手順

$\{S_i\}$, $\{D_j\}$ が(1), (2)を満足したとする．間接キューブ網の幾何学的な性質から，この結合に関与する p_l $(1 \leqq^{\forall} l \leqq k)$ 段目のSUは，すべてスイッチングが固定される (${}^{i}p_l = {}^{i'}p_l$ のとき平行，${}^{i}p_l \neq {}^{i'}p_l$ のとき交叉)．この時，(2)より，他の演算に携わるモジュール間の通信経路との干渉は，p_l 段目では回避される．

次に，他の $n-k$ 段のSUに着目する．今，このようなSUで，一方の入力ポートが $\{S_i\}$ と $\{D_j\}$ の結合に，他方の入力ポートが別の結合に関与することが可能なものが存在すると仮定し，そのうち段数の最も低いもの (q 段目) に着目する．すると，一方に結合可能なソース・モジュールは，すべて $\{S_i\}$ に属し，他方に結合可能なソース・モジュールは，すべて $\{S_i\}$ に属さないことになる．ところが，間接キューブ網の幾何学的な性質により，後者のアドレスは，すべて前者のアドレスの 2^q の桁の数字を逆転させたものである．$\{S_i\}$ の位数の点から，これは $\{S_i\}$ に属していなくてはならず，したがって矛盾が生じた．同様のことが出力ポートに関してもいえるため，$\{S_i\}$ と $\{D_j\}$ の結合に関する，p_l 段以外のSUは，この結合のみに関与することが示された．

以上によって，$\{S_i\}$ と $\{D_j\}$ の結合が，他の結合によって干渉を受けないことが示された．さらに，p_l 段目のSUのスイッチングを固定しても，ルーティングの規則はそのまま保存されるため，分割後の小規模な結合網も間接キューブ網の性質を保つ．

ゆえに，$\{S_i\}$ と $\{D_j\}$ の結合は，パーティショニングされた網上で実現される．

(昭和59年5月7日受付，8月3日再受付)