データベースマシンGRACEに於けるモジュール間結合網

Inter-Module Networks on Database Machine GRACE

坂井 修一[+] 喜連川 優[++] 田中 英彦[+] 元岡 達[+]
S. Sakai M. Kitsuregawa H. Tanaka T. Moto-oka

[+] 東京大学 工学部
Faculty of Engineering, The University of Tokyo
[++] 東京大学 生産技術研究所
Institute of Industrial Science, The University of Tokyo

1. はじめに

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^{8}~10^{16}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、応用分野に依っては、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、所謂ジョイン・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。

即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及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抽出の点から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に致命的な影響を及ぼす危険があり、高い転送レートのモジュール間結合網の実現が、マシンの成否を握る鍵となる。

本稿では、当研究室で考案され、開発中の高性能関係データベースマシンGRACE［1］に関して、その処理方式にフィットした結合網及び関係する制御方式を提案・評価・検討した結果を報告する。結合網は主に多段スイッチ網を対象とし、シミュレーションによるスループット・データ転送時間の評価を行い、また光ファイバ・リングバスによる実現方式との比較その他を行った。

2. データベースマシンGRACE

GRACEはハッシュとソートを用いた高速並列処理データベースマシンである。その著しい特徴として、SELECTIONやUPDATE等の処理負荷の軽いデータベース操作のみならず、負荷の重い関係代数演算・集合演算に対して極めて強力であることを挙げねばならない。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等は、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×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同士はJOINの取れる可能性が無い為、処理時間Tは、

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

に軽減される（図2．1）。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）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台のメモリモジュール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。以上の如く、ハッシュを用いたリレーションのクラスタリングによって、バケットレベルでO（S）処理とし、生成されたバケットをソータによってO（n）時間で処理し、さらにバンクパラレリズムの反映により、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内の処理である。

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。更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。ステージング時には、一般のハッシュの様にロケーションとハッシュ値を対応づけることをせず、各ハッシュ・バケットを複数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メモリ容量（プロセッサ・サイズ）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押え、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押える。

また、本マシンでは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。

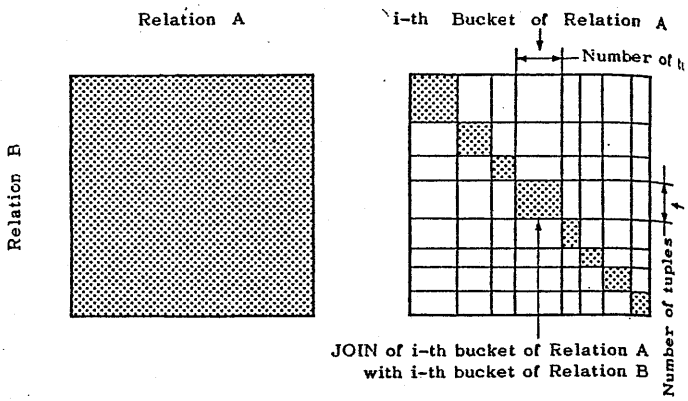
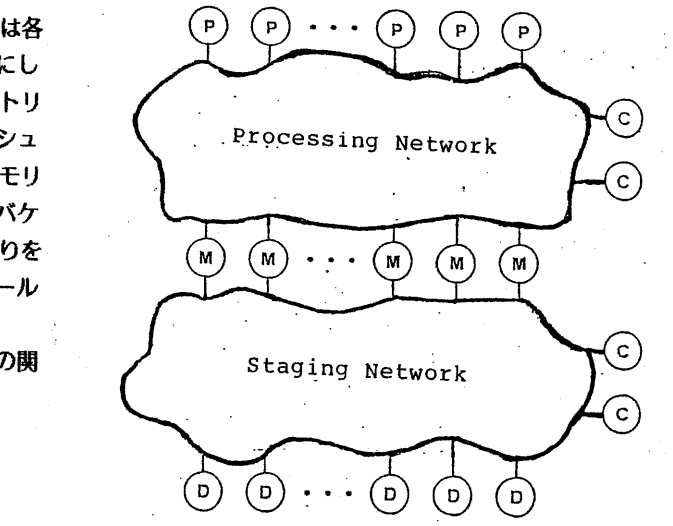


Fig. 2.1 Clustering Effects By Hashing In JOIN Operation

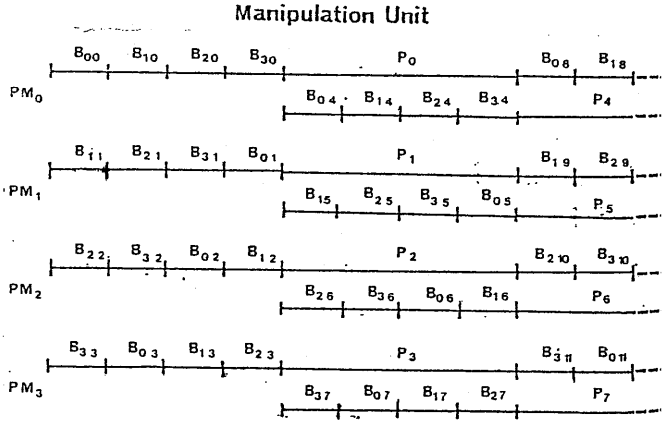
2．2 GRACEのシステム構成

本データベースマシンのデータ操作部を図2．2に示す。データ操作部は、4種のモジュール及びこれらを結合する相互結合網から成る。即ち、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、全体の実行制御を司どる。また、データ操作部の上位には、キュアリ・アナライザ、セキュリティ・エンフォーサ、インテグリティ・エンフォーサなどが存在し、CMには関係代数木に展開されたキュアリが渡される。



P： Processing Module　C： Control Module
M： Memory Module　D： Disk Module

Fig. 2.2 Global Architecture Of Data Manipulation Unit



P_i : processing of i-th bucket
B_{ij}: input of j-th bucket in MM_i

Fig. 2.3 Bucket Processing Pipeline

2．3 GRACEの処理方式と相互結合網

2．3．1 GRACEの処理方式

本マシン上でのリレーションの処理を概説しよう。

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(或いはPROJECTIONアトリビュート)に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、各ハッシュ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、適当なスケジューリングが為された後、PMに送出される。一つのPMには一つのバケットが対応し、関係する全MMから順番に当該バケットを受け取って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は再び次の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により順次処理していく。この処理は図2．3に見られる如くパイプライン化して行われるが、これをバケット処理レベル・パイプラインと呼ぶ。(GRACEでは、バケット処理レベル・パイプラインの他にも、二つのレベルのパイプライン処理が行われている。オペレータレベル・パイプライン及びソータ内マージソートのパイプラインがこれであるが、詳細は文献[1]を参照されたい。)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、ハッシュ・バケットは関係するMM間に均等に分配される。

2．3．2　バケット分配網とバケット収集網

前節で述べた関係代数処理を、ハッシュ・バケットの流れの点から見直す。DM・PMからMMへのデータ転送に於ては、ハッシュ・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レベル・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。前節では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。即ち、バケット分配網とバケット収集網は図2．4に示されるものである。

以下、本稿を通じて2種のモジュール間結合網の各々を対象とした評価・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ハッシュ・バケットをソートし、この順を転送順序とする。これは、当該オペレーションに関わる全PMの或るサイクルでの処理負荷を均し、アイドル時間を減らす為である。

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吸い上げていく。或るバケットの転送及びソートは、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図2．3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。

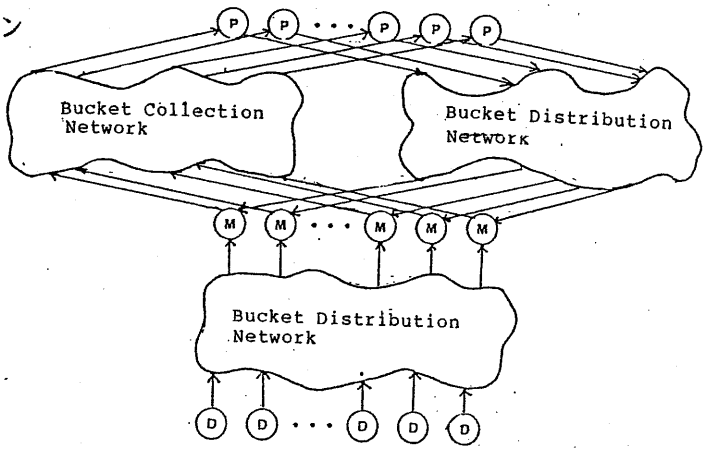
図に示された様に、バケット転送時間はサイクリックな順序づけがなされている為、バケット収集時のMM・PM両モジュールの対応関係は、基本的には、

$$P_N = \{ (MM_i , PM_{(i+j) \bmod N}) \mid 0 \leq i , j \leq N-1 \}$$

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と理論的考察

バケット収集網の候補として、多段スイッチ網の一種である間接キューブ網[5](図3．1)を適用することを考える(理由は後出)。間接キューブ網は多段スイッチ網として



P : Processing Module　D : Disk Module
M : Memory Module

Fig.2.4 2 kinds of networks on GRACE

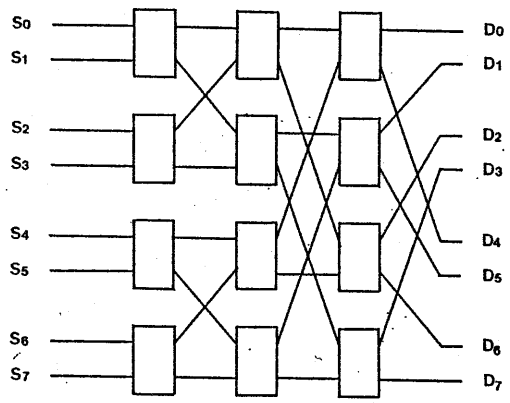


Fig. 3.1 indirect binary 3-cube network

は最もクラスの低い$\log_2 N$段の網である。各ステージは、2×2のクロスバスイッチ及びその制御部を含むスイッチング・エレメント（SE）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、すべての入出力マッピングが可能であるわけでは無く、閉塞を起こすケースが多いのが難点である。ルーティングは、ソース・アドレスとデスティネーション・アドレス（或いはデスティネーション・アドレスのみ）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SEが参照しつつ、スイッチの切替を行う、所謂セルフ・ルーティングが用いられる。

3．2．1　入出力マッピングの実現

Lawrie［6］はオメガ網に関して、入出力マッピング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。

Theorem. 1

ある入出力マッピング$P_N=\{(S_i, D_j) \mid 0\leq i, j \leq N-1\}$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
$\Longleftrightarrow {}^{\forall}(S_i, D_j)、(S_k, D_l) \in P_N$について、
$(i \neq k) \wedge (i \ \mathrm{div}\ m = k \ \mathrm{div}\ m) \Rightarrow j \neq l$
但し、$m = 2^h \quad (1\leq h \leq \log_2 N)$

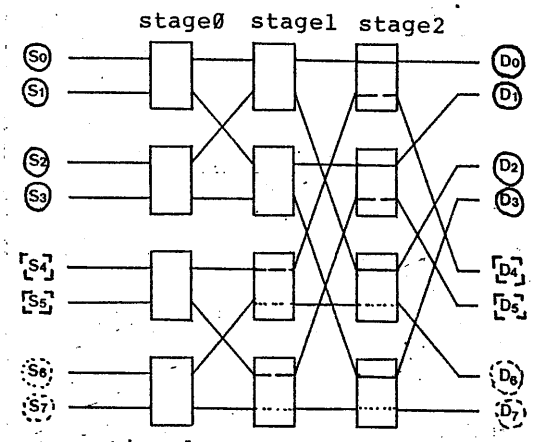
（証明略）

Theorem. 1は同値条件であるが、これを満たすマッピングのクラスをいくつか示すことができる。

Theorem. 2

間接キューブ網上では、閉塞無しにサーキュラ・シフト（前節参照）が実現可能である。

（証明略）



Operation 1.
$\{S_0, S_1, S_2, S_3\} \rightarrow \{D_0, D_1, D_2, D_3\}$
Operation 2. $\{S_4, S_5\} \rightarrow \{D_4, D_5\}$
Operation 3. $\{S_6, S_7\} \rightarrow \{D_6, D_7\}$
Fig. 3.2 Partitioning by 3 operations

Theorem. 2により、GRACEのパ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）、3．3でその評価に関して述べる。

3．2．2　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互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同士の干渉が起き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として表わされる。パーティショニングとは、ある相互結合網を互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そのまま保存されている場合をいう。図3．2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っている例を示した。

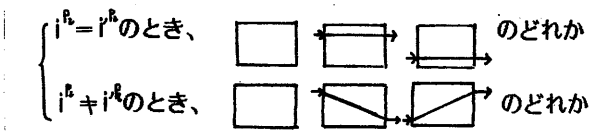
次に間接キューブ網をパーティショニングした状態で用いるための必要十分条件を述べる。

Theorem. 3

N×N間接キューブ網で結合されたマルチプロセッサ・システム上で既にいくつかの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、$N/2$個のデスティネーション・モジュール$\{D_j\}$（$0\leq k \leq n$，　$n=\log_2 N$）が使用可能の時、$\{S_i\} \rightarrow \{D_j\}$の結合が論理的にパーティショニングされた網上で実現されるための必要十分条件は、以下の1）2）で与えられる。

1）．${}^{\forall}S_i, S_j \in \{S_i\}$，${}^{\forall}D_{i'}, D_{j'} \in \{D_j\}$
$i^p=j^p$，$i'^p=j'^p$を同時に満足するようなpが少なくともk個存在する。これをp_1，p_2，p_3…… p_kとする。但しi^pはiを二進表示した時の2^pの桁の数字である。

2）．$\{S_i\}$，$\{D_j\}$を割り付ける以前の網上で、ステージ$n-p_l$［$1\leq l \leq k$］の関係するSEの状態が、



である。但しp'は、1）で求められたものとした。

（証明略）

3．3　バケット収集のシミュレーション

一つの関係代数オペレーション実行時における、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3．2．1で述べた。本節では実際に、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。

収集時では、1回の結合で転送する単位は、各MM間に分割されたハッシュ・バケットであり、これは普通数KB以上あることから、回線交換方式を採用することにした。

図2．3で表わされるように、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、結合のクラスは全ての時点でサーキュラ・シフトになる為、間接キューブ網を用いた時、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その他から、バケット処理レベル・パイプラインは擾乱をきたす。これによって、間接キューブ網上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依って測定することにした。

（1）MM間のバケット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）トータルなバケットサイズの乱れは零。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。

（3）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。

但し、（1）（3）はバケットサイズに関するソート（3．1参照）を施した場合とそうでない場合について調べた。

測定の結果は、それぞれ図3．3、3．4、3．5のグラ

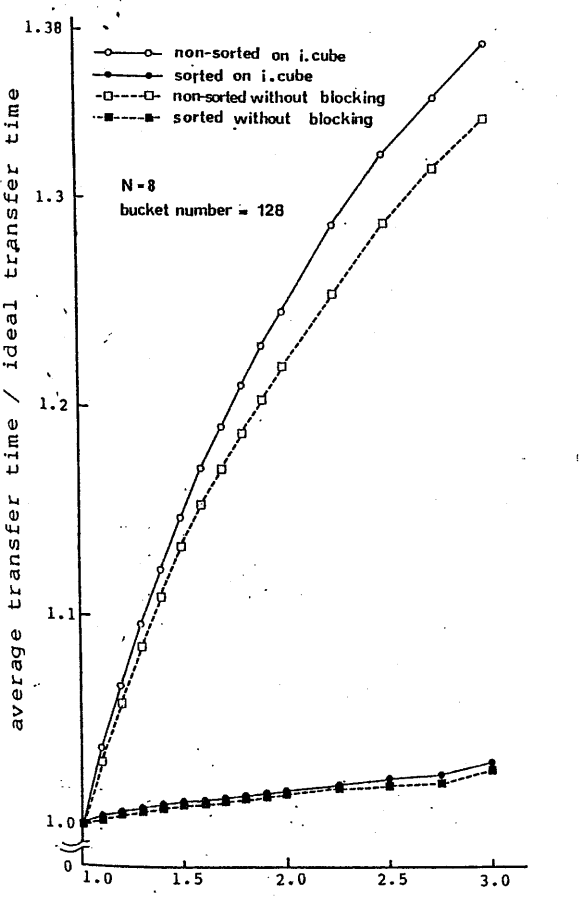


Fig3.3Data transfer time of 1 operation v s. bucket size fluctuation

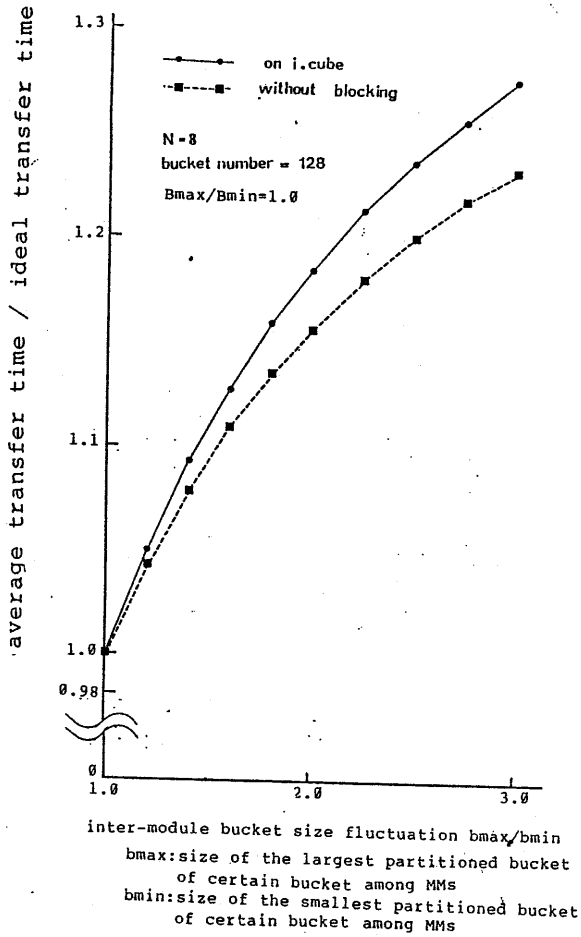


Fig3.4Data transfer time of 1 operation v s. inter-module bucket size fluctuation(1)

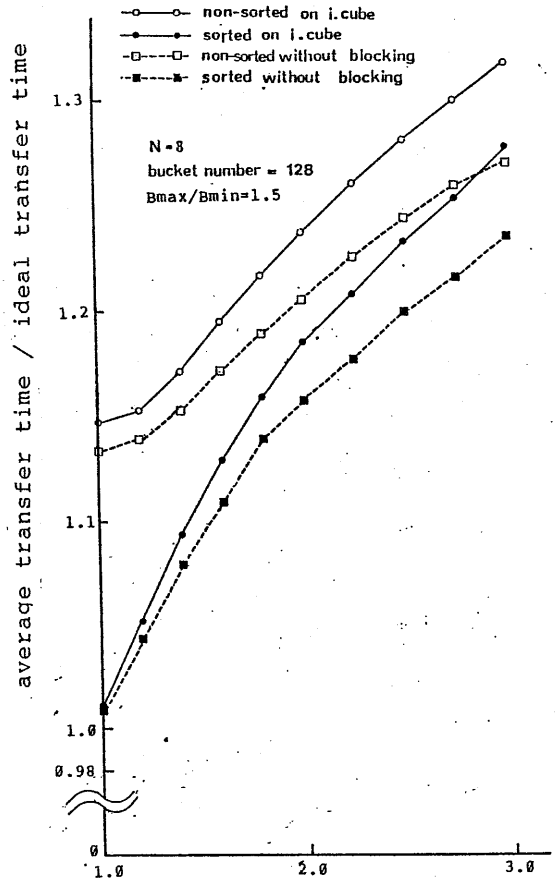


Fig. 3.5 Data transfer time of 1 operation vs. inter-module bucket size fluctuation (2)

フで表わされる。この結果をまとめて以下に示す。

結果1　バケットサイズ・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。従ってバケットサイズのチューニングは必要以上に厳密にやらなくてもよい。（図3．3）

結果2　バケットサイズ・ソートは大幅なオーバヘッド軽減をもたらす。場合によっては30%以上の効率向上をみることがある。（図3．3）

結果3　MM間のバケットサイズのゆらぎは比較的大きなオーバヘッドをもたらす。従ってバケット分配時に均等な分配の為の制御が重要となる。（図3．4、3．5）

結果4　間接キューブ網（図の実線）と閉塞の無い網（図の破線）との差は殆ど無い。即ち、クロスバ・スイッチの様な大規模な結合網を用いなくても、間接キューブ網で十分である。（図3．3～3．5）

（1）～（3）のシミュレーションは8×8の結合網に関してであったが、別のシミュレーションによって網のサイズは転送のオーバヘッド・レイショにほとんど影響を及ぼさないことがわかった。

以上は1オペレーションのみに着目した評価であった。GRACEでは同時に複数の互いに独立した演算が行われる為網の効率良い使用のために、モジュール割り付け時に、3．2．2で述べたパーティショニングが考慮されねばならない。パーティショニングは、大きな処理負荷のオペレーションの終了後に複数の小さな処理負荷のオペレーションが来た時は容易だが、逆の場合には不可能なケースがある。この時には、パーティションできるまで次のオペレーションを待たせるか、閉塞を承知で実行させるかを、演算の優先順位等によって、CMが判断せねばならない。

4．バケット分配網

4．1　バケット分配の制御方式

バケット分配は、DM・PMからMMにリレーション・データを転送する動作である。その際、各ハッシュ・バケットはMM間で均等に分配されねばならない。従って、そのために各タプルの行先を決定する制御が必要となる。これをバケット分配に於けるタプル行先制御と呼ぶ。

タプル行先制御のために必要な情報は、当該バケットが現在MM群に如何に分散されているか、というものである。転送すべきタプルが発生した時、基本的には行先制御装置はこの情報を参照して、当該バケットのタプル数が最小のMMにこれを転送すれば良い。

行先制御は、直前のタプルの転送と重畳化して行い、そのオーバヘッドをなくす。従って、1回の転送量（1タプルの大きさ）が大きい程、複雑な行先制御が可能となる。

タプル行先制御は、制御装置の実現形態に依って、以下の三種に分類される。

（1）集中行先制御　：　セントラル・コントローラによる制御
（2）分散行先制御　：　ソース・モジュール（DM・PM）一つ一つによる制御
（3）分散取込制御　：　デスティネーション・モジュール（MM）一つ一つによる制御

MM間でのバケットの均等な分割は、実際にはバケット分配終了時に達成されていれば良い。また、転送の効率上、同時に送出するタプルの行先が衝突するのは好ましくない。従って、セントラル・コントローラでこれを検知・回避する方法が考えられる。これを、

（1）' 1対1集中行先制御　：　ソース対デスティネーションの対応がかならず1対1になるようにする集中行先制御

と呼ぶ。（1）'では、1回のタプル転送時に、タプルが当該バケット最小のMMに行くとは限らないが、分配終了の時には、バケットは十分に平坦化される。

以上4種の行先制御方式の内、制御に要する時間は（1）'が一番大きく、次いで（1）で、（2）（3）はほぼ同じである。また、MM間でのバケットサイズのゆらぎは、（1）では零だが、他のものは若干生じるおそれがある。シミュレーションに依って調べたところ、（1）'（2）（3）ともに数%のゆらぎであり、3．3の結果からバケット収集時に性能低下を引き起こすことは無いと言える。

尚、（1）（1）'にかかる時間の短縮は、並列化によって果たすことができる。

実際には、タプルは十分に大きいと考えられるので、（1）'の方法が有効である。

4．2　バケット分配のシミュレーション

或るタプルが如何なるハッシュ値を持って出現するかは、全く予想のつかない事象である。従って、バケット分配時には、収集時のようなモジュール間結合の規則性が期待できず、これを利用した結合網の簡単化は不可能である。

バケット分配網では、結合に局所性が無い為、ソースとデスティネーションの結合力がどの組み合せを取っても等しい結合網でなければならない。候補としてクロスバ・スイッチ網、バイトニック・ソート網、ベネス網、間接キューブ網、時分割多重チャネル方式光ファイバ・リングバス等が考えられる。

各方式の結合網に関する検討は、段数の少いものから、即ち、間接キューブ網から行うことにした。。以下のモデルのもとに、バケット分配に要する転送時間のオーバヘッドの大きさを、シミュレーションによって調べる。まず、行先アドレスは一様乱数で与え、各モジュールは1000タプルを送出するものとする。これは、ハッシュ値の不規則性と、転送されるリレーション・サイズの概算によって根拠づけられる。

タプルは、ソース・モジュールに於いてデータ発生率 m（0<m ≦1）で発生し、結合網を介してデスティネーション・モジュールへ送られる。ソース・モジュール内には長さCのキューがあり、また、デスティネーション・モジュールは、到着したデータを必ず受け取るものとする。

（1）回線交換間接キューブ網

間接キューブ網を回線交換方式で用いたケースについての測定を、次の各場合について行う。

（1－1）データ発生率　m ＝1とした時の、網サイズと転送時間の関係を求める。

（1－2）データ発生率　m を変化させた時の、発生率とキューに受入れられる確率の関係を求める。（C＝1）

（1－3）1対1行先制御を施した時の、網サイズと転送時間の関係を求める。

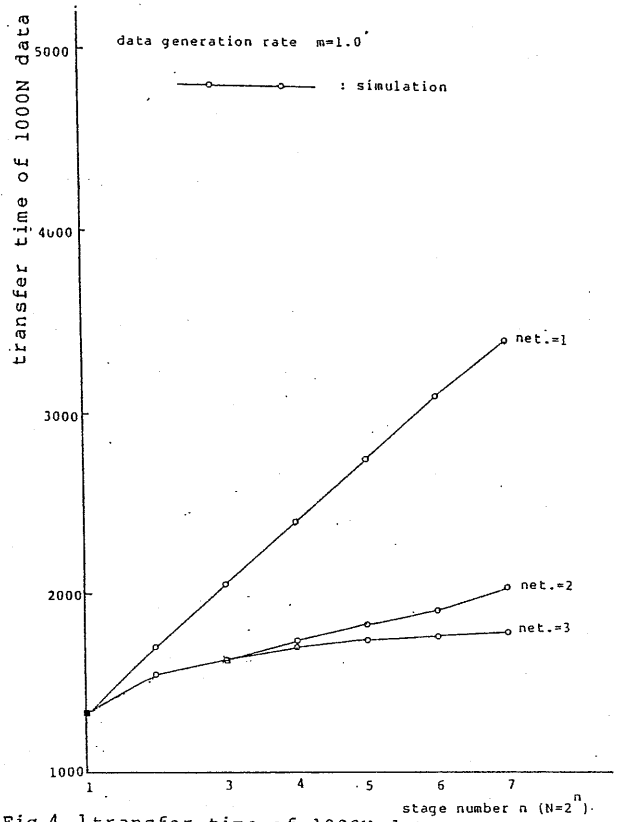


Fig.4.1transfer time of 1000N data v s. network size (indirect binary n-cube) circuit switching (without buffer)

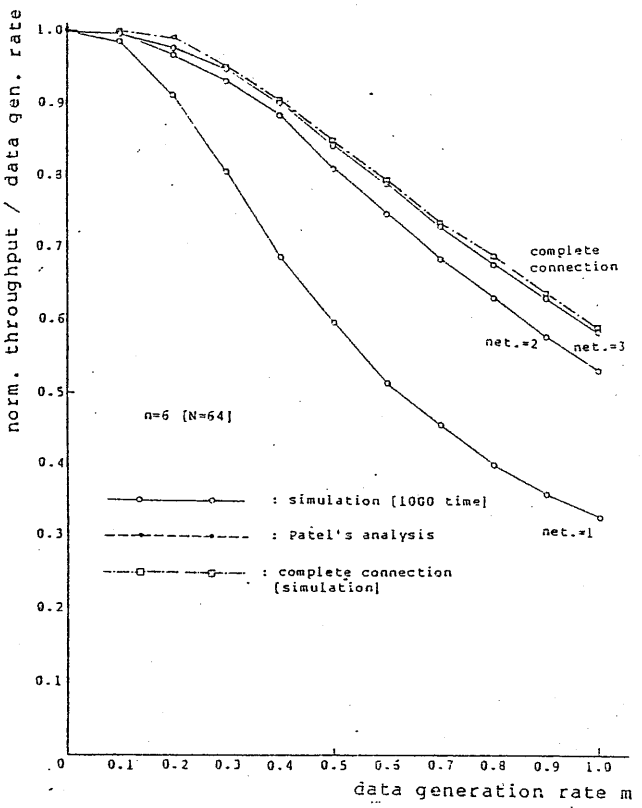


Fig.4.2norm. throughput / data gen. rate v s. data gen. rate (indirect binary n-cube) circuit switching (without buffer)

測定の結果は、それぞれ図4．1、4．2、4．3のグラフで表わされる。この結果をまとめて以下に示す。

結果1．1　網サイズが大きくなるにつれ、転送時間は増大する。（図4．1、4．3）

結果1．2　網の枚数を増すことで、性能向上が可能である。特に、間接キューブ網3～4枚で、ほぼ閉塞の無い網と同じ性能を実現できる。（図4．1～4．3）

結果1．3　1対1行先制御の効果は大きく、間接キューブ網を3枚用いれば、転送オーバヘッドをほぼ零にすることが可能である。（図4．3）

結果1．4　データ発生率 mの低い時には、間接キューブ網2枚でも十分な性能が得られる。（図4．3）

（2）蓄積交換間接キューブ網

蓄積交換方式は、網内に設けたバッファにより、結合網の転送性能を向上させる方式である。ここでは、間接キューブ網の各SEの各入力ポートにバッファを挿入することを考える。シミュレーションによる測定は、次の各場合について行った。

（2－1）データ発生率 m ＝1とした時の、網サイズと転送時間の関係を求める。

（2－2）データ発生率 m を変化させた時の、発生率とキューに受入れられる確率の関係を求める。（C＝1）

（2－3）1対1行先制御を施した時の、網サイズと転送時間の関係を求める。

それぞれの測定の結果をグラフで表したものが、図4．4、4．5、4．6である。結果の示すものを以下に列挙する。

結果2．1　網サイズが大きくなるにつれ、転送時間は増大する。（図4．4、4．6）

結果2．2　バッファサイズを大きくして性能向上を計ることが可能である。サイクル・タイムが異なるので簡単な比較はできないが、バッファサイズ4は、ほぼ回線交換方式間接キューブ網3枚に相当する。（図4．4～4．6）

結果2．3　1対1行先制御は、結合網のサイズが小さい時には有効だが、網が大きくなるとその意味を失う。（図4．6）

結果2．4　データ発生率 mの低い時には、バッファサイズ2でも十分な性能が得られる。（図4．5）

（1）（2）を比較して、1対1行先制御が可能な場合（タプルが十分に大きく、データ転送と1対1行先制御が十分に重畳化可能な場合）には、回線交換間接キューブ網を3枚程度用いるのが良く、そうでない場合にはバッファサイズ10程度の蓄積交換間接キューブ網を用いるのが良いと結論さ

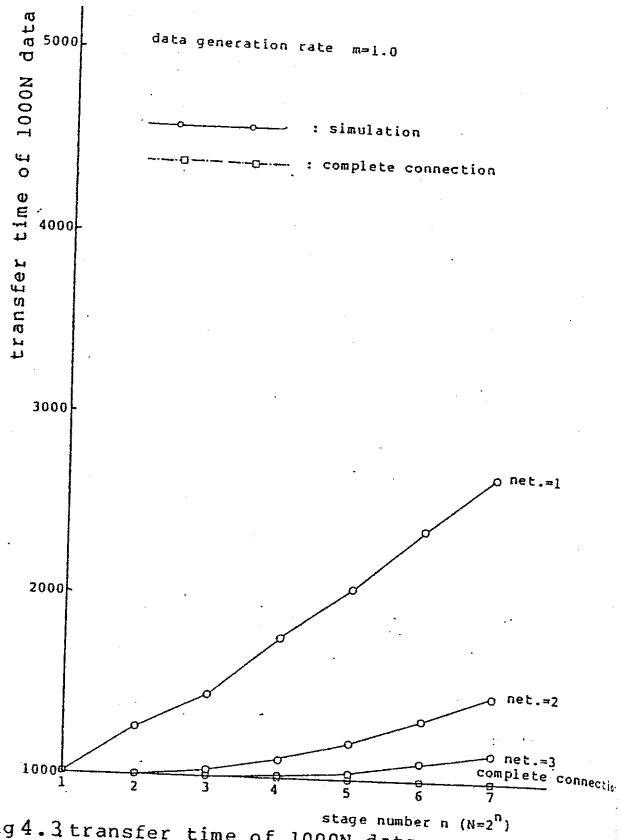


Fig 4.3 transfer time of 1000N data v s. network size (indirect binary n-cube) circuit switching (without buffer) mapping modified(1:1)

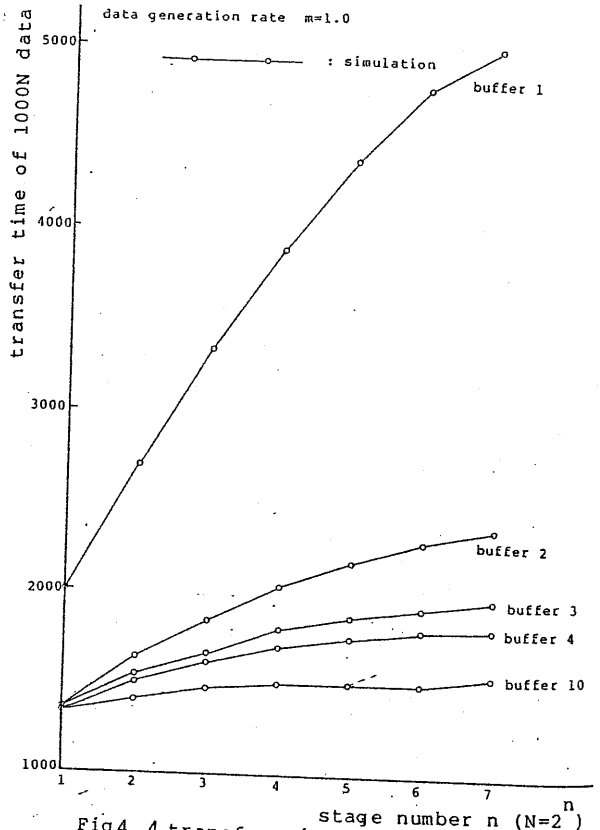


Fig.4.4 transfer time of 1000N data v s. network size (indirect binary n-cube) packet switching (with buffer)

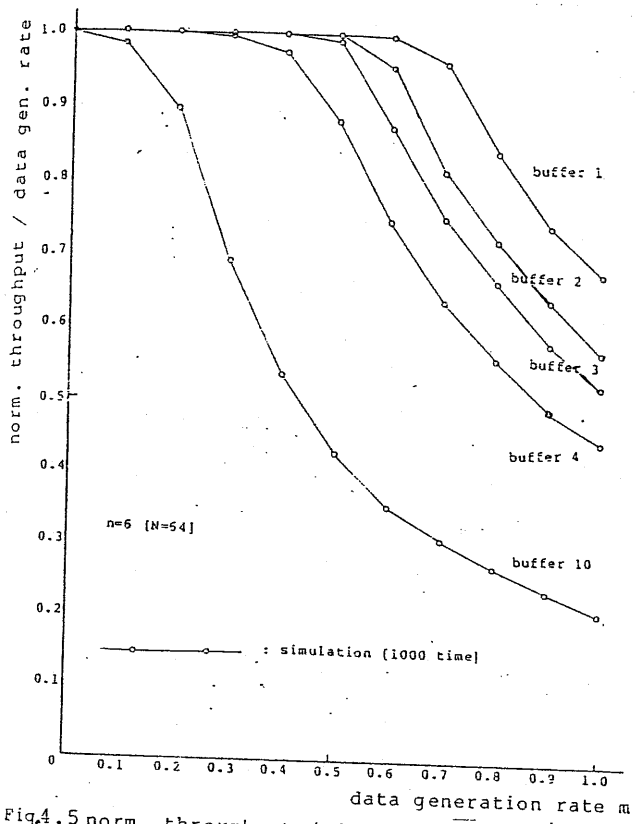


Fig.4.5 norm. throughput / data gen. rate v s. data gen. rate (indirect binary n-cube) packet switching (with buffer)

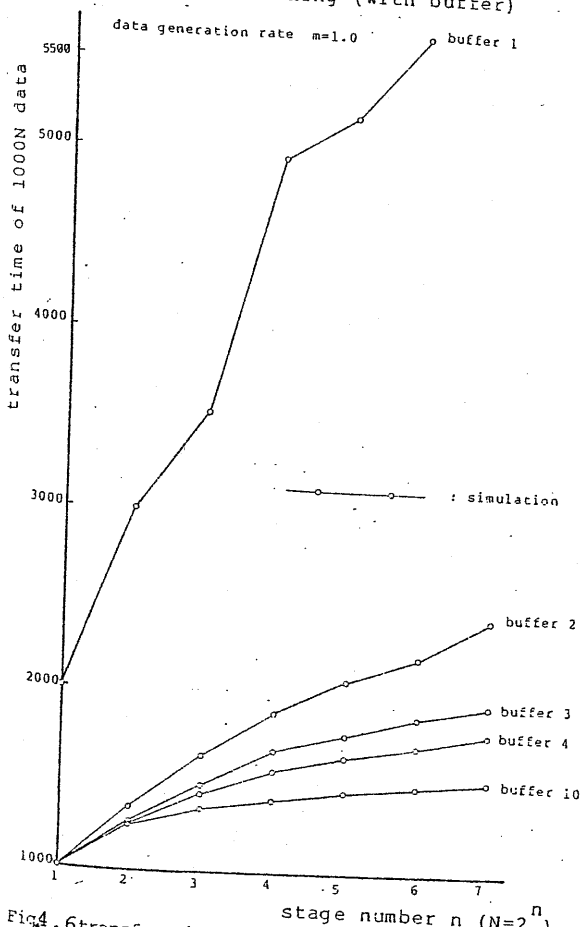


Fig.4.6 transfer time of 1000N data v s. network size (indirect binary n-cube) packet switching (with buffer) mapping modified(1:1)

れる（図4．1、4．3、4．4、4．6）。実際にはタプルは十分に大きく、1対1行先制御は十分に可能であると考えられるため、前者の方式を採用することにする。但し、トランスポーズド・ファイルを用いる場合等では、1回の転送量が小さいため、後者を採用するケースも有り得る。

（3） その他の結合網を用いた実現

ベネス網、バイトニック・ソート網、クロスバ・スイッチ網等による実現が考えられる。これらは図4．2で“complete connection ”で示される強い結合力を持つ。しかしベネス網はルーティングの手間の点で、またバイトニック・ソート網やクロスバ・スイッチ網等は、網のサイズが大きくなると、ハードウェアのコストの点で実現が難しい。

時分割多重チャネル方式の光ファイバ・リングバスも、高い結合能力を有するが、容量の点・インタフェース部のコストの点で問題が在る。

5．総合的なシミュレーション

3章ではバケット収集網の実現方式を、4章ではバケット分配網の実現方式を検討した。本章では、その結果として求められた結合網を用いて、バケット分配からバケット収集までのデータ転送をシミュレーションによって評価する。測定は、タプルが十分に大きい場合と、トランスポーズ等によって、転送されるタプルが小さくされている場合に分けて行うことにする。

5．1 転送するタプルが大きい場合（普通の場合）

バケット分配網として回線交換方式の間接キューブ網を3枚、1対1行先制御のもとで用いる。バケット収集網として回線交換方式間接キューブ網1枚を用いる。

バケット分配時の転送オーバヘッドはほぼ零である（図参照。32×32の網で 0.2%程度）。転送されたバケット・データをPM群が収集するシミュレーションを行ったところ、こちらのオーバヘッドもほぼ零であった。これは、1対1集中行先制御に由来するMM間のバケットサイズのゆらぎが、バケット処理のパイプラインを乱すに到らなかった為である。

5．2 転送するタプルが大きい場合（トランスポーズド・ファイルを用いる場合）

バケット分配網としてバッファサイズ10の蓄積交換方式間接キューブ網1枚を、分散行先制御のもとで用いる。バケット収集網として回線交換方式間接キューブ網1枚を用いる。

バケット分配時の転送オーバヘッドは例えば16×16の網の場合約38%、32×32の網で約41%である（図4.4参照）。一方、バケット収集の転送オーバヘッドをシミュレーションで測定

したところ、ほぼ零であった。これは、分散行先制御に由来するMM間のバケットサイズのゆらぎが、バケット処理のパイプラインを乱すに到らなかった為である。

6．おわりに

現在開発中の高性能関係データベースマシンGRACEに於けるモジュール間結合網と、関係する制御方式に関して検討した。その結果として、$\log_2 N$段の結合網である間接キューブ網を用いた実現方式を示した。

その他、光ファイバ・リングバスを時分割多重チャネル方式で用いる方法が考えられるが、現在のところ、容量の点で問題が生じがちである。このような場合、多重バス構造にして容量を増す方法を採るが、インタフェース部にかかるコストが大きくなる危険がある。

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、本稿で提案した方式のモジュール間結合網は、十分にこの転送レートを達成することができる。

今後の課題としては、フィルタリング等によるデータ発生率の見積り、実装面のより詳細な検討（ＬＳＩ化を考慮した結線構造や１スイッチング・エレメントの構成、故障への対処等）、さらにはタプルの行先制御装置や各モジュール内インタフェース部の設計等を行うことが挙げられる。

<<参考文献>>

［1］ 喜連川優、鈴木重信、田中英彦、元岡達：Hash と Sort による関係代数マシン、信学技法、EC81−35、1981

［2］ 坂井修一、喜連川優、田中英彦、元岡達：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

［3］ 坂井修一、喜連川優、田中英彦、元岡達：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

［4］ Feng，T.，A Survey of Interconnection Networks，COMPUTER，Dec.，1981

［5］ Pease，M. C.，The Indirect Binary n-Cube Microprocessor Array，Trans. on Computers，vol．C−26，No．5，May 1977

［6］ Lawrie，D. H.，Access and Alignment of Data in an Array Processor，Trans. on Computers，vol．C−24，No．12，Dec. 1975

［7］ Patel，J. H.，Processor−Memory Interconnection for Multi−Processors，The 6th Annual Symposium on Computer Architecture，April 23-25，1979

［8］ Dias，D. M.，Jump，J. R.，Packet Switching Networks for Modular Systems，COMPUTER，Dec.，1981