論　文

関係代数マシンGRACEにおけるバケット分配網

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 Distribution on Relational Algebra Machine GRACE

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

あらまし　高速大容量のデータベースマシンへの期待が高まってきている．我々は現在，ハッシュとソートを用いた高性能関係代数マシンGRACEを開発中であるが，本マシンは多モジュール構成の高並列計算機であるため，各構成要素間の通信に携わる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が，本論文では前者に関する研究結果を報告する．すなわち，ハッシュにより生成されたバケットをステージング空間に分配する手法を検討し，バケット分配網の実現方式を提案する．結合網は主に多段結合網の一種である間接キューブ網を対象とし，さらにこれを多ルート化する改良を検討する．評価は大規模なシミュレーションによって行い，要求される転送性能とバケット分配が得られることが確認された．また，他の結合網による実現との比較を行った．

1. まえがき

高速かつ大容量のデータベースシステムに対する産業社会からの需要は増加の一途をたどっており，近年データベース管理の諸機能をハードウェアにより実行する専用計算機，データベースマシンの研究が活発になってきている．1980年代に入って，その研究成果の一部は商用化され，定型業務主体のオフィスを中心とする市場を開拓している．しかし一方で，より複雑な不定型業務に対応する高性能マシンへの需要は高く，その実現が待たれているのが現状である．

高性能データベースマシンは，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なる並列処理システムの形態をとると考えられ，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における通信系の役割は重要であり，相互結合網の性能はマシンの全体性能を決定する．

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)〜(6)]を開発中である．本論文ではGRACEの相互結合網と，それに関係するデータ流の制御方式に関して報告する．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が，後者に関しては文献(6)で述べた．ここでは前者すなわちバケット分配網に関して述べる．結合網としては多段結合網の一種である間接キューブ網を採用し，これを多ルート化する改良などを検討した．データ転送時間の評価は大規模シミュレーションによって行い，本方式による実現の正当性を確認するとともに，他の結合網による実現方式との比較・検討を行った．

2. GRACEにおけるバケット分配

2.1 GRACEのシステム構成と処理方式

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り，JOINやPROJECTION

† 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

の処理を$O((N+M)/K)$（N, Mは当該リレーションのタプル数，Kはメモリバンク数）の時間計算量で実現する．これは，リレーションがステージング空間と比較して小さいときは，メモリページサイズに比例した一定時間での処理であり，大きいときは，リレーションの大きさに比例した時間での処理である．

ＧＲＡＣＥの全体構成を図１に示す．本マシンはプロセシングモジュール（ＰＭ．図中のＰ），メモリモジュール（ＭＭ．図中のＭ），ディスクモジュール（ＤＭ．図中のＤ）とバケット分配網，バケット収集網からなり，さらに図中には示されていないが，コントロールモジュール（ＣＭ）が全体の実行制御を司っている．

次に，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）に関してハッシングされた後，ＭＭ群に移される．このとき，ハッシュ値により分割されたリレーションの各部分をバケットと呼ぶ．本マシンでは，リレーション間のＪＯＩＮを，ハッシュ値の等しいバケット間のＪＯＩＮに帰着させることで，処理の高速化を実現する（ＰＲＯＪＥＣＴＩＯＮも同様）[6]．したがって，バケットは本マシンの処理対象であり，転送の対象である．

ＭＭ群にリレーションをステージングする際，従来の直接編成法でみられるごとくタプルの格納アドレスとハッシュ値を対応づける（１ＭＭに１バケットを格納する）ことをすれば，バケットの大きさの偏りによって，リソースの使用効率の低下・性能低下をまねく危険が生じる．

本マシンではこの問題を，文献(6)で述べた方法で解決した．ここではその概要を簡単に記す．

ステージング時には，ＭＭとハッシュ値を対応づけることをせず，各バケットを複数ＭＭに均等に分配し，ＭＭ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１ＭＭでのオーバフローを防ぎ，かつ後に述べるバケット処理のパイプラインの乱れをなくす．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ＰＭ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抑え，ＰＭの使用効率を高める．

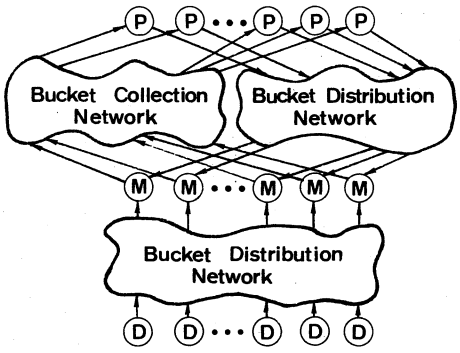

図１　ＧＲＡＣＥの全体構成
Fig.1　Overview of GRACE.

続いてＰＭによる関係代数処理の手順を以下に示す．ＭＭに移されたリレーション・データは，バケットサイズ・チューニングの後，適切なスケジューリング[6]を施され，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．この処理はパイプライン化して行われるが，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，次の演算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関係するＭＭ間に均等に分配される．

以上の関係代数処理の手順より，ＧＲＡＣＥにおけるモジュール間のデータ転送は２種に大別されることが示された．すなわち，ＤＭ・ＰＭでハッシュによって生成された各バケットをＭＭ間において均等に分配するデータ転送と，ＭＭからのバケットをＰＭにおいて収集するデータ転送である．前者をバケット分配，後者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（図１）．

2.2　バケット分配網の役割

前節で述べたように，ＭＭにバケットを転送する際に，各バケットはＭＭ間で均等な大きさになるように分割して分配する．この分配にゆらぎが生じると，バケット処理のパイプラインが擾乱をきたし，処理の性

能低下をまねく．シミュレーションによってこの効果を調べたところ，場合によっては３０％以上のオーバヘッドが生じることがわかった[6]．

したがって，バケット分配では，

(1) バケットの転送オーバヘッドが小さいこと

(2) バケットのMM間での分配が十分均等であること

の２点が重要である．

3. バケット分配の制御方式

3.1 タプルの行先制御方式

DM・PMで生成されたバケットを，それぞれMM間で均等な大きさになるように分配するためには，転送するタプルの行先を決定する制御が必要になる．これをバケット分配におけるタプルの行先制御と呼ぶ．

タプルの行先制御のために必要な情報は，当該バケットが現在MM群にいか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生成された時，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少のMMにこれを転送すればよい．

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吸収させる．したがって，１回の転送量（１タプルの大きさ）が大きいほど，複雑な行先制御を施してよいことになる．ふつうタプルは，数＋Bから数KBと考えられ，行先制御には十分な余裕がある．

タプルの行先制御は，制御装置の実現形態によって次の３種に分類される．

(1) 集中行先制御

(2) 分散行先制御

(3) 分散取込制御

(1)は，バケット転送の行先となる全MMにおけるバケットの分散情報を，１台のセントラル・コントローラが保持していて，転送のソースであるPM（あるいはDM）でタプルが生成されるたびに，その行先を決定する方式である．(2)は，ソース側のモジュール（DM・PM）のそれぞれが，おのおの各バケットのタプルをどのMMにいくつ転送したかという情報を保持していて，他のソースモジュールとは独立に行先制御を行う方式である．(3)は，特にリングバスのような結合網を用いたときに有効であり，それぞれのMMが，自メモリ内の各バケットの大きさに関する情報を保持していて，ソース側から送出されたタプルが通信路上を巡回するうちに，分割された当該バケットの最も小さいMMに取り込まれるよう制御する方式である．

３種の行先制御のうち，制御に要する時間は(1)が最も大きく，(2)，(3)は比較的小さい．一方，転送終了時のMM間におけるバケットサイズのゆらぎは，(1)ではないが，(2)，(3)では生じるおそれがある．

3.2 １対１制御

前節に述べたタプルの行先制御では，バケット分配の均等性の点のみが考慮されていた．しかしこの均等性は，バケット分配の終了時に達成されていればよく，各時刻で厳密に守られねばならぬものではない．ここで，バケットの転送効率を考慮に入れて行先決定機構を見直せば，同時に複数のソースモジュールから送出されるタプルの行先が衝突するのは好ましくない．したがって，行先制御装置で行先MMの衝突を回避し，転送時のオーバヘッドを軽減する方法が考えられる．このような，行先MMの再調整は集中制御でのみ可能であり，これを，

(4) １対１集中行先制御

と呼ぶことにする．(4)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に転送されるとは限らないが，分配終了の時には，バケットは十分に平坦化されると予想される．ただし，制御時間は(1)より大きくなる．

3.3 バケット平坦化のシミュレーション評価

本章で述べたタプルの行先制御方式のうち，(2)，(3)，(4)は，MM間でのバケットサイズのゆらぎが発生する可能性があり，その大きさによっては，バケット収集処理のパイプラインの擾乱をきたす危険がある．このゆらぎの大きさをシミュレーションにより測定した．

網の大きさ（ポート数２から１２８）とタプルの生成率をパラメータとして測定したところ，バケットサイズのゆらぎはこれらのパラメータには関係なく，(2)，(3)，(4)とも数％にとどまることが示された．すなわち，(2)におけるゆらぎは最大７％程度であり，(3)，(4)におけるゆらぎは最大４％程度であった．いずれの場合も，バケット収集処理時における処理の擾乱は十分小さい[5],[6]．

4. バケット分配網

本章では，バケット分配網の実現方式に関して検討する．

あるタプルがいかなるハッシュ値をもって出現するかは，全く予想のつかない事象である．前章で述べたように，バケット分配時のタプルの行先は，このハッ

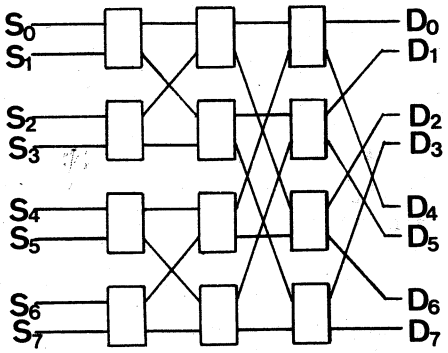

図2　間接キューブ網（3段）
Fig.2　Indirect binary 3-cube network.

シュ値をもとに決定されるため，DM（PM）からMMへの結合は不規則であり，しかも局所性がない．したがって，バケット分配網は，DM（PM）とMMの結合力が，どの組合せをとっても等しい結合網でなくてはならない．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，間接キューブ網などのさまざまなクラスの多段結合網や，時分割多重チャネル方式のリングバスなどが考えられる．

本章ではまず，段数が少なくハードウェア量の小さい多段結合網である間接キューブ網[8]（図2）をバケット分配網に適用することを検討する．最初に，間接キューブ網を回線交換方式で用いた場合のシミュレーション評価を行い，さらにこれにスイッチング・ユニット（SU）を付加し，多ルート化することによって得られる性能向上を評価する．次に，間接キューブ網を蓄積交換方式で用いた場合のシミュレーション評価を行う．さらに，他の結合網による実現方式との比較・検討を行い，与えられたデータベース環境に対していかなる網をバケット分配網に適用すべきかを論じる．

4.1　シミュレーション・モデル

本章の転送シミュレーションは，以下のモデルのもとに行われる．

まず，全体構成としてN台のDM（PM）とN台のMMが相互結合網を介して結合されているとする．各DM（PM）はそれぞれ1000タプルを送出するものとし，行先MMは一様乱数を用いて決定する．前者は転送されるリレーションの大きさの概算からくるものであり，後者は生成されるタプルが持つハッシュ値の不規則性からくるものである．

タプルは，DM（PM）においてデータ生成率m（$0<m\leqq 1$）で生成され，結合網を介してMMに送られる．

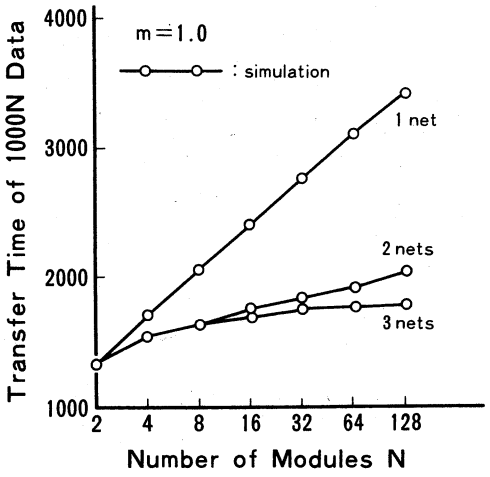

図3　回線交換間接キューブ網を用いた転送（測定1）
Fig.3　Data transfer by circuit-switching indirect binary n-cube(Measurement 1).

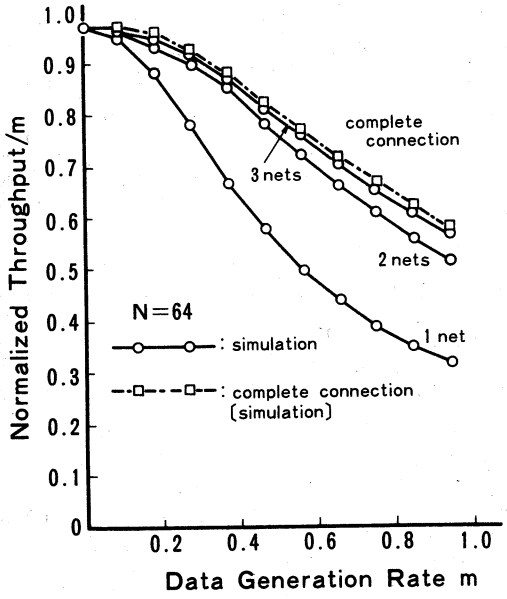

図4　回線交換間接キューブ網を用いた転送（測定2）
Fig.4　Data transfer by circuit-switching indirect binary n-cube (Measurement 2).

DM（PM）内には，長さCのFIFOメモリがあるとし，またMMは到着したタプルを必ず受け取るものとする．

測定結果は次の二つのものを記す．

〔測定1〕　$m=1$としたときの，モジュール数Nと転送時間（1タプルの転送時間を1とする）の関係．

〔測定2〕 mを変化させたときの，mとFIFOメモリに受け入れられる確率の関係．ただし，$N=64$，$C=1$．

4.2 回線交換方式の間接キューブ網

間接キューブ網を回線交換方式で用いた場合，行先の衝突だけでなく，網内の閉塞によって転送性能が大幅に低下することが予想される．本節では網を複数(≦3)並置することで，これに対応することを考え，前節のモデルにもとづくシミュレーションを行った．ここでは，1タプルの転送に際し，1枚目の網から順番に受理されるまで転送要求を出していく，というように並置された網を使い分けることにする．また，比較のために閉塞のない結合網を用いた場合の評価も示した．

測定結果は，図3(測定1)，図4(測定2)に表わされる．結果をまとめて以下に示す．

〔結果1〕 Nが大きくなるにつれ，転送時間は増大する(図3)．

〔結果2〕 並置する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枚で，閉塞のない網とほぼ同じ性能を実現できる(図3，図4)．

〔結果3〕 データ生成率mが小さいときには，間接キューブ網1・2枚でも十分な性能が得られる(図4)．

以上は，行先MMが単に一様乱数によって与えられる場合であったが，次に，**3.2**で述べた1対1制御を施した場合も，同様にシミュレーションを行った．測定結果は，図5(測定1)，図6(測定2)に示される．

〔結果4〕 1対1行先制御の効果は大きく，間接キューブ網を3枚並置すれば，転送オーバヘッドをほぼ0にすることが可能である(図5，図6)．

次に，1対1制御を施し，さらに間接キューブ網に1段分($\log_2 N/2$個)のSUを付加して網を多ルート化した場合に関するシミュレーションを行った．結果は図7(測定1)，図8(測定2)に示される．

〔結果5〕 1対1行先制御と網の多ルート化によって，著しいスループットの向上が得られる．例えば，$N=128$，$m=1.0$，網1枚の場合，図3に比較して，約30%の転送時間の短縮が得られる(図7，図8)．

4.3 蓄積交換方式の間接キューブ網

蓄積交換方式は，網内に設けたバッファにより，結合網のスループットを向上させる方式である．今回は，間接キューブ網の各SUの各入力ポートに，タプルp

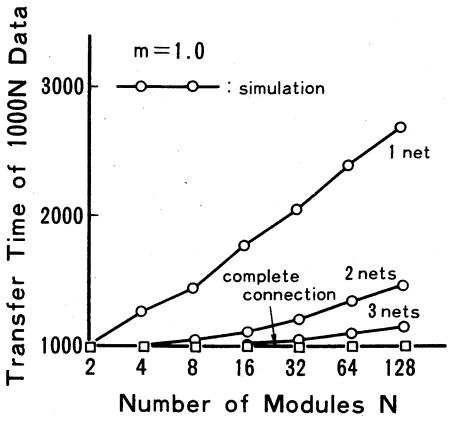

図5 1対1制御を施した場合の転送(測定1)
Fig. 5 Data transfer with one-to-one mapping control (Measurement 1).

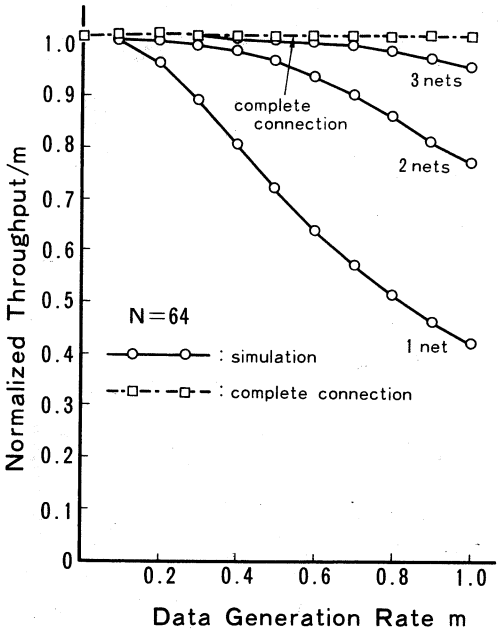

図6 1対1制御を施した場合の転送(測定2)
Fig. 6 Data transfer with one-to-one mapping control (Measurement 2).

個分の容量を持つFIFOバッファを挿入したものに関して検討した．**4.1**のモデルにもとづいて得られた測定結果は，図9(測定1)，図10(測定2)で表わされる．結果の示すものを以下に列挙する．

〔結果6〕 Nが大きくなるにつれ，転送時間は増大する(図9)．

〔結果7〕 バッファの大きさp(図中ではQ. len.)

を大きくして性能向上が可能である．転送の単位時間が異なるので単純な比較はできないが，$p=4$は，ほぼ回線交換方式の間接キューブ網を3枚並置したものに相当する（図3，図4，図9，図10）．

〔結果8〕 データ生成率mが小さい時には，$p=2$でも十分な性能が得られる（図10）．

また，別のシミュレーションによって，バッファの大きさがある程度になると転送性能が飽和すること，1対1行先制御やSUの付加による多ルート化はあまり有効でないことが示された．

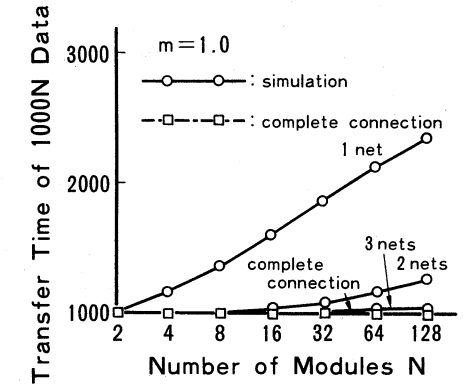

図7 多ルート化した網を用いた転送（測定1）
Fig.7 Data transfer by multi-route network (Measurement 1).

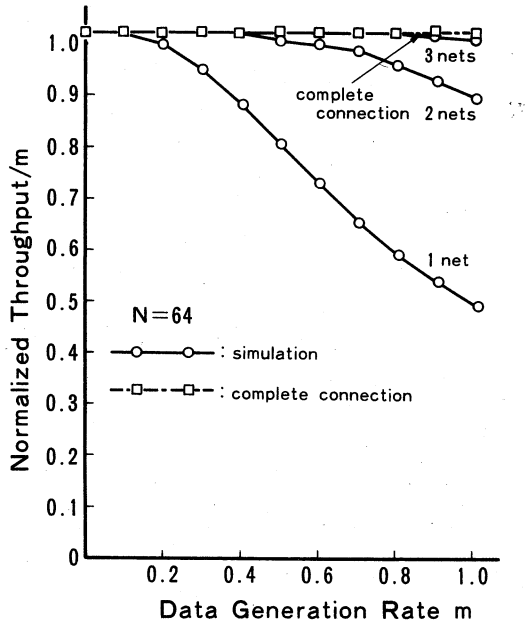

図8 多ルート化した網を用いた転送（測定2）
Fig.8 Data transfer by multi-route network (Measurement 2).

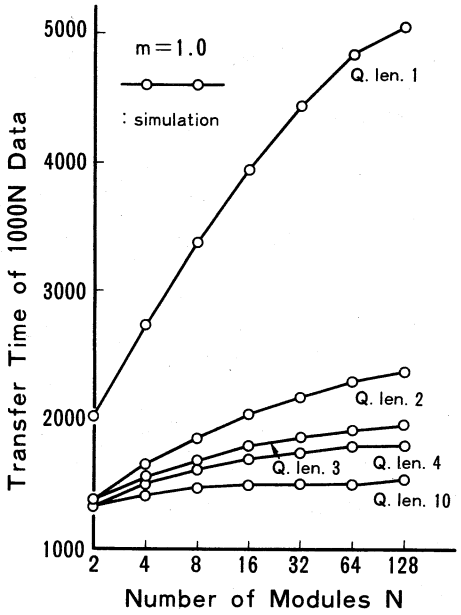

図9 蓄積交換間接キューブ網を用いた転送（測定1）
Fig.9 Data transfer by buffered indirect binary n-cube (Measurement 1).

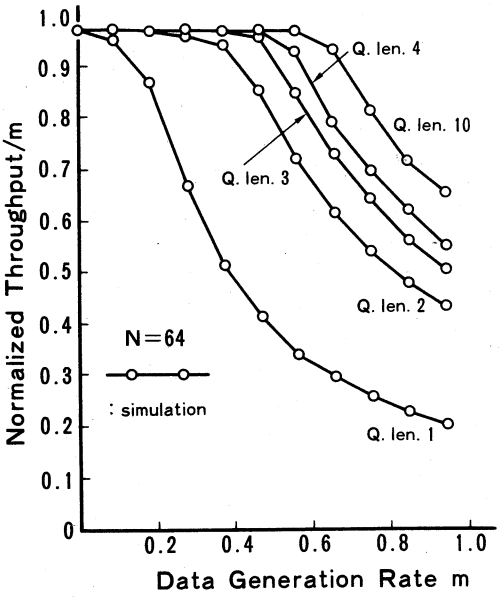

図10 蓄積交換間接キューブ網を用いた転送（測定2）
Fig.10 Data transfer by buffered indirect binary n-cube (Measurement 2).

4.4 その他の結合網を用いた実現

間接キューブ網以外の多段結合網として，ベネス網，バイトニック・ソート網，クロスバ・スイッチ網などによる実現が考えられる．これらは，図4～図8で“complete connection”で示される高い結合能力をもつ．

ベネス網は，間接キューブ網のほぼ2倍のハードウェア量で閉塞のない結合網を実現するが，ルーティングの手間と網の再構成（rearrange）の問題から，適用が困難である．また，バイトニック・ソート網やクロスバ・スイッチ網などは，それぞれ$O(N(\log_2 N)^2)$，$O(N^2)$（Nはモジュールの台数）のハードウェアが必要であり，Nが大きくなるとコストの点で実現が難しい．

時分割多重チャネル方式のリングバスも高い結合能力を有する．この場合には，**3.1**で述べたように分散取込制御が適当である．リングバスを用いた実装方式・評価については別途報告する．

4.5 評価・検討

前節までの結果から，バケット分配網の実現形態として適切な方式を定める．

1対1行先制御が可能な場合，すなわちタプルが比較的大きく，データ転送と1対1行先制御が重畳化可能な場合には，多ルート化した回線交換方式間接キューブ網を1～2枚用いるのがよく，そうでない場合にはバッファの大きさ10タプル分程度の蓄積交換方式間接キューブ網を用いるのがよい．実際にはタプルは十分大きく（数十B以上），1対1制御が可能であるため，前者の方式を採用することにする．ただし，トランスポーズド・ファイルを用いる場合などでは，1回の転送量が小さいため，後者を採用することになる．

5. 考　察

3.，**4.**の結果より，バケット分配網としては，多ルート化された間接キューブ網を回線交換方式で1～2枚用いるのが有利であり，その時，タプルの行先制御は1対1集中行先制御を用いるべきであることが示された．また，トランスポーズなどによってタプルの大きさが小さくされた時には，蓄積交換方式間接キューブ網（バッファの容量は10タプル程度）を適用するのが有利であり，制御時間の点から分散行先制御を採ることになる．

本論文では，SUはポート数2のものを扱ったが，ポート数4のSUを用いれば，さらに結合能力が向上すると予想される．現在この点に関するシミュレーション評価を行っている．

タプルの行先制御は，各モジュールやセントラル・コントローラが行う方式を検討したが，SU自体が簡単な行先決定機構を持つ方式も考えられる．

GRACEは，複数本の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もつディスクのそれは，3MB/sec程度であり，本論文で述べた方式のバケット分配網は，十分にこの転送レート（1ポートあたり）を達成することができる．

6. むすび

現在開発中の高性能関係代数マシンGRACEにおけるバケット分配の制御機構と，これに携わるモジュール間結合網に関して述べた．その結果，$\log_2 N$段の多段結合網である間接キューブ網を改良した結合網による実現方式を示し，タプルの行先制御機構を検討した．

今後の課題として，DM・PMにおけるデータ生成率の見積りを行うこと，バケット分配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，タプルの行先制御装置の試作を行うことなどが挙げられる．

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

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

(4) M. Kitsuregawa, H. Tanaka and T. Motooka: “Architecture and Performance of Relational Algebra Machine GRACE”, Proc. Int. Conf. Par. Proc., pp. 241-250 (Aug. 1984).

(5) 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）．

(6) 坂井，喜連川，田中，元岡：“関係代数マシンGRACEにおけるバケット収集網”，信学論(D)，J68-D, 1, pp. 9-16（昭60-01）．

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

pp. 12-27 (1981-12).
(8) M.C. Pease : "The Indirect Binary n-Cube Microprocessor Array", IEEE Trans. Comput., C-26, 5, pp. 548-573 (May 1977).
(9) J.H. Patel : "Performance of Processor-Memory Interconnections for Multiprocessors", IEEE Trans. Comput., C-30, 10, pp. 771-780 (Oct. 1981).
(10) D.M. Dias and J.R. Jump : "Analysis and Simulation of Buffered Delta Networks", IEEE Trans. Comput., C-30, 4, pp. 273-282 (April 1981).
(11) G.B. Adams and H.J. Siegel : "The Extra Stage Cube : A Fault-Tolerant Interconnection Network for Supersystems", IEEE Trans. Comput., C-31, 5, pp. 443-454 (May 1982).

(昭和59年10月9日受付, 60年1月28日再受付)