3階層記憶データベースマシンアーキテクチャと性能評価†

清木 康††* 峰松彩子††** 相磯秀夫††

本論文では，すでに提案した関係演算処理方式（V-方式）の有効性がデータベースのI/O処理系（データステージング系）を含めたデータベース処理全体の効率に与える影響について解析し，V-方式による関係演算処理の効率向上が3階層記憶のデータベースマシンに対する問合せの処理効率の向上に有効であることを示す．また，データステージング処理のための諸技術を抽出し，各データステージング技術の有効性を評価するツールとなるデータベースマシン・シミュレーションモデルを設定する．さらに，そのモデルを用いて，実際に各データステージング技術の性能評価を行い，個々の技術がデータベース処理全体の効率に与える影響を考察する．ここで提示するシミュレーションモデルはSIMULAにより実装した．このシミュレーションモデルは，3階層データベースマシンを設計する場合に，そのアプリケーションに適したアーキテクチャを選択するためのツールとして実際に利用できるものである．

1. まえがき

関係データベースシステムの性能を改善するために，数々のデータベースマシンが提案されてきた．データベースマシンの研究の初期には，データベース本体を格納するディスク装置に直接，並列処理，連想処理機構を付加したlogic-per-track方式のデータベースマシン[8]が提案された．これらのデータベースマシンは関係演算である選択演算(selection)，制約演算(restriction)および更新操作(update)をディスク1～2回転で処理できるので有効であるが，結合演算(join)，射影演算(projection)，割算(division)などの複雑な関係演算に対しては，関係（リレーション）内のタプル数，あるいは種別数（演算対象属性中のアイテム値の種類の数）のオーダのディスク回転を必要とするので処理効率が悪い．そこで，現在では複雑な関係演算を高速メモリ上でマルチプロセッサにより実行する形態のデータベースマシン[4)~6),9),13),15),16)]が主流となってきた．

この形態のデータベースマシンでは，データベース格納媒体であるディスク装置からマルチプロセッサの主記憶となる高速メモリへのデータステージング処理と，マルチプロセッサによる関係演算処理が性能を決定するおもな要因である．これら二つの処理を分散して並列に実行するためには，ディスク装置とマルチプロセッサの内部メモリ間にステージング・バッファ(ディスク・キャッシュに対応)を置くと有効である．このようにメモリ構成を3階層とした場合，データベースマシンの性能を決定する主要な処理は次のように分けられる．

（1）ディスク上の処理対象データへのアクセス
（2）ディスクからステージング・バッファへのデータ転送
（3）ステージング・バッファからマルチプロセッサの内部メモリへのデータ転送
（4）マルチプロセッサでの関係演算処理

筆者らはすでに(4)について，新しい関係演算処理方式(V-方式：V-method)[17]を提案し，過去に提案された関係演算処理方式との間で(3)の処理を含めて性能比較を行い，提案した方式の有効性を明らかにした[7]．しかし，提案したV-方式がデータベース処理全体の効率に与える影響，すなわち，(1)，(2)のデータステージング処理を含めたデータベース処理全体の効率に与える影響については明らかではない．また，過去に提案された関係演算処理技術およびデータステージング処理技術がデータベース処理全体の効率に与える影響も明らかにされていない．

本論文では，すでに提案した関係演算処理方式，および過去に提案されたデータステージング処理のための諸技術が，データベース処理全体の効率に与える影響について評価を行う．そのために，現在まで明らかにされていなかったデータベース処理の諸技術の有効性を評価するツールとなるデータベースマシン・シミ

† Three-level Memory Hierarchical Database Machine Architecture and Its Performance Evaluation by YASUSHI KIYOKI, AYAKO MINEMATSU and HIDEO AISO (Faculty of Science and Technology, Keio University).

†† 慶應義塾大学理工学部電気工学科
* 現在 日本電信電話公社武蔵野電気通信研究所
** 現在 日本アイ・ビー・エム(株)

ュレーションモデルを設定する．そして，そのシミュレーションモデルを用いて，V-方式の有効性を明らかにし，さらに，各データステージング処理技術の有効性について考察する．

2. 関係演算処理系

関係演算処理系においては，高速メモリ上でマルチプロセッサにより複雑な関係演算を効率よく処理することが重要となる．すでに，文献7)で，V-方式と過去に提案された関係演算処理方式の分類およびそれらの性能評価を行っているので，ここでは関係演算処理系について詳細に述べることは避け，本論文で評価対象とする関係演算処理方式の概要を述べる．

2.1 関係演算処理方式

評価対象とする関係演算処理方式は3種類である．以下に，結合演算，射影演算を処理する場合を例にとり，各方式を説明する．

（1）分割ソート・マージ方式 (PSS: Partial Sort Merge Scheme)

結合演算の場合には，演算対象の二つの関係をそれぞれの結合演算対象属性上でソートした後，それらの両属性上で比較操作を行い，条件を満たしたタプル同士を結合する．射影演算の場合には，演算対象属性(あるいは属性群)を抽出した後，関係をそれらの属性上でソートし，重複を排除する．

（2）クラスタリング方式 (CLS: CLustering Scheme)

結合演算の場合には，演算対象の二つの関係それぞれの結合演算対象属性内の各アイテム値に対してハッシュ関数をかけ，対応するバケットに各タプルを振り分けた後，各バケット内で両方の関係内のタプル結合演算対象属性上で比較し，条件を満たしたタプル同士を結合する．射影演算の場合には，関係内の演算対象属性(あるいは属性群)を抽出し，そのアイテム値(あるいは複合アイテム値)に対してハッシュ関数をかけ，対応するバケット内に各タプルを振り分けた後，各バケット内で重複を排除する．

（3）V-方式 (V-method)

V-方式はビットマップ表現された転置インデックスやリンク（二つの関係間のタプルの結合を示すテーブル）を主要データ構造とし，それらのデータに対して定義した基本演算により問合せを処理する独特な方式である．V-方式は，他の方式で結合演算結果が大きくなり処理効率が劣化する状況で，逆に，処理効率が向上するという特徴をもつ．V-方式は，関係代数を実行することによって問合せを処理する方式とは異なるので，結合演算，射影演算等の基本演算を用意していない．しかし，それらに対応する演算として，Multiply-AIR (Multiply-II), Multiply-VTFG (Multiply-I) という基本演算を用意している．これらの基本演算については文献17)に詳しく述べているのでここでは省略する．

2.2 マルチプロセッサへのデータ配置

各関係演算処理方式を実現するアーキテクチャとしてさまざまな形態が考えられるが，ここでは，各方式の並列処理アルゴリズムを平等な環境で評価するために，各方式に適した特殊なアーキテクチャではなく，汎用プロセッサを一次元配列状に並べ，それらを共有バスでつないだ一般的なマルチプロセッサのアーキテクチャを想定する．このようなアーキテクチャのもとで，PSS, CLS, V-方式により関係演算を並列処理する場合のアルゴリズムはすでに文献7)に示した．そこでは，PSS方式については，関係全体をソートするのではなく各プロセッサ (PE) に分割配置された部分関係に対してのみソートを行うアルゴリズムを対象として検討している．

演算対象の関係をステージング・バッファからマルチプロセッサへ転送し，マルチプロセッサの内部メモリへ配置する場合，その配置方法は次の2種類に分類できる．

（1）等分割配置

演算対象の関係をPEの台数と同数の部分関係に分割し，各部分関係に1台のPEを割り当てる．このとき，各部分関係内のタプル数は等しくする．ステージング・バッファが各PEに対応してブロック化され，各PEとそれに対応するブロックとの間に転送ラインが用意されているならば，ステージング・バッファとPE群の間はPE台数分の並列データ転送を行える．この配置はPSSおよびV-方式で採用される．

（2）クラスタリング配置

演算対象の関係内のタプルを演算対象属性のキー値に従ってPEに分配する．したがって，同じキー値をもつタプルは同じPE内に格納される．ステージング・バッファからPE群へのデータ転送はネットワーク回路により並列に行う．その並列度は，データ転送時の衝突やネットワークのハードウェアの性質によって制限されるので，等分割配置の場合の並列度よりも小さくなる．この配置はCLSで採用される．

以上のような関係演算処理系アーキテクチャのもとで，PSS, CLS, V-方式により関係演算を処理する場合について，すでに文献 7), 18)で性能評価式を設定し，各方式の性能評価を行っている．本論文では，各方式，とくに V-方式の性能がデータベース処理全体の効率に与える影響について５章で解析する．

3. データステージング系

データベース本体を格納する記憶媒体としては，可動ヘッドディスク，固定ヘッドディスクが考えられる．現在の技術でデータベースのような大量データを格納するには，容量，価格の点で可動ヘッドディスクが現実的である．可動ヘッドディスクでは，データステージング処理をシーク (seek)，サーチ (search)，転送 (transfer) の３処理に分けて考えることができる．これらの３処理を効率よく行うために，以下(１)～(５)の技術を抽出する．

(１) トラック並列読出し機構

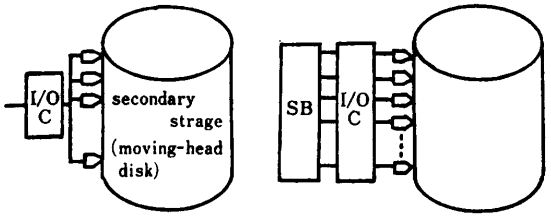
これは，DBC[2),3)]において採用されたもので可動ヘッドディスクの１シリンダ内の全トラックを並列に読み出し，ステージング・バッファに転送する機構である(図 1(b))．この機構を活かすために，関係を図２のように同一シリンダ内の各トラックに分けて格納する．これにより，１シリンダ内全データをディスク１回転で並列転送できる．

(２) 連想ディスク機構

可動ヘッドディスクに対しては，logic-per-some tracks 機構[3)]を付加できる．ディスクヘッドに連想処理機構を付加することにより，ディスクからステージング・バッファへのデータ転送量を軽減できるので，上位メモリのメモリ使用量の軽減，関係演算処理時間の軽減が可能となる．ただし，連想ディスクにおける連想処理はディスクの回転読出し中にデータの流れに追従して行われるので，データ転送時間を軽減することはできない．

(３) インデックス[1),2),12)](転置ファイル) の利用

関係を可動ヘッドディスクに格納する際に，関係内の特定の属性について同じアイテム値 (属性値) をもつすべてのタプルを同じシリンダ内に図３に示すように格納する．そして，ある属性値から，その値と同じ値をもつタプル群が格納されているシリンダ番号をひくインデックス(図４)を用意する．問合せ中に選択演算が含まれている場合，その演算対象属性に対応するインデックスを参照することによりディスクアクセス回数を軽減できる．さらに，データ転送量の軽減，上位



(a) serial-data-transfer (b) parallel-data-transfer
I/O C : I/O channel
SB : Staging Buffer

図 1 ディスク・ドライブの構成
Fig. 1 Organization of the disk drive.

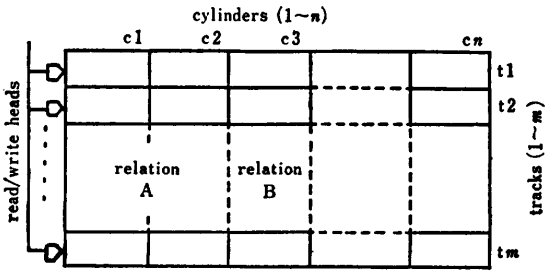


図 2 関係の格納法
Fig. 2 Allocation of relations to the disk drive.

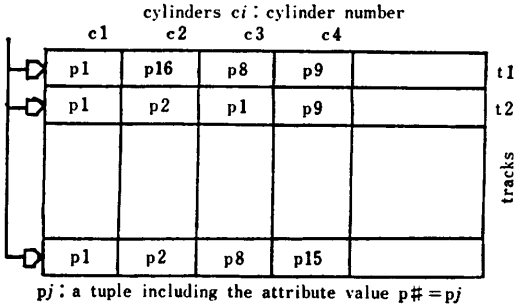


図 3 関係のクラスタ化格納
Fig. 3 Clustered allocation to the disk drive.

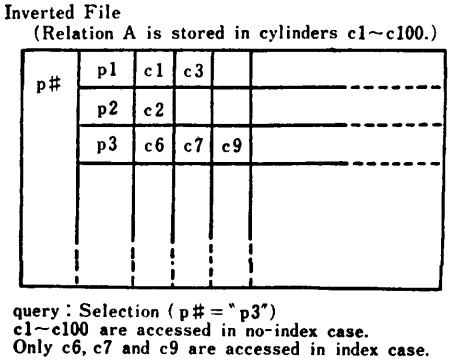


図 4 インデックスの構成
Fig. 4 Structure of the index.

メモリ使用量の節減，関係演算処理時間の短縮が可能となる．

（4）ディスクへの最適データ配置

可動ヘッドディスク・ドライブの動作のなかで，シーク動作はシリンダの移動という機械動作を伴うので，データステージング処理のなかで最も処理時間を費やす．そこで，シーク時間を軽減するために，ディスク上に関係を格納する場合に局所性をもたせる．すなわち，一つの関係内のページ群を同一シリンダ内に格納し，もしも1シリンダ内に納まらない場合には，入りきらないページを隣接のシリンダに格納する．さらに，頻繁に同時に用いられる複数の関係は，隣接のシリンダに格納しておく．

（5）ディスクヘッド・スケジューリング

関係データベースの問合せ処理の環境はディスクヘッド・スケジューリングを行うのに適している[10]．関係データベースにおける問合せには処理対象データが陽に示されているので，問合せのコンパイル時にそれらを一括把握することができる．また，問合せを構成する関係演算群の実行順序に対する制限は少ない．たとえば，5.2節で述べる問合せ（図6）Q1において，三つの演算対象の関係に対する各選択演算の間には実行順序の制約はない．そこで，それらの関係演算の演算対象の関係群をアクセスする場合に，ディスクヘッドの現在の位置に近いものからアクセスし，ステージング・バッファへ転送するようにする．これにより，シーク時間を軽減することができる．また，複数の問合せを同時に処理する多重問合せの環境では，より多くの処理対象データをディスクヘッド・スケジューリングの対象にできるので，シーク時間をさらに軽減できる．ディスクヘッド・スケジューリングアルゴリズムとしては，FCFS方式，SSTF方式，SCAN方式，CSCAN方式が代表的である[14),20)]．ただし，FCFS方式は，ディスクへのアクセス要求を先着順に受けつけるので，シーク時間を軽減することはできない．

以上のようなデータステージング技術を対象として，個々の技術がデータベース処理全体の処理性能に与える影響を5章で解析する．

4. シミュレーションモデル

関係演算処理とデータステージング処理を含めたデータベース処理全体の性能を評価するために，データベースマシンのシミュレーションモデルを設定する．**図5**にシミュレーションモデルを示す．以下に，このモデルにより問合せを処理する過程を示す．

（1）関係データベースシステムに到着した問合せの識別子 Q_i が query queue に入れられる．

（2）問合せ Q_i のコンパイル時に明らかとなった処理対象の関係の識別子 $R_{ij}(j=1,\cdots,m_i : m_i$ は処理対象の関係数）が，それらの関係を格納している各ディスク・ドライブの disk queue に入れられる．

（3）R_{ij} を構成する各ページの識別子 $P_{ijk}(k=1,\cdots,h_{ij} : h_{ij}$ はページ数）が，それらのページを格納している各シリンダに対応する cylinder queue に

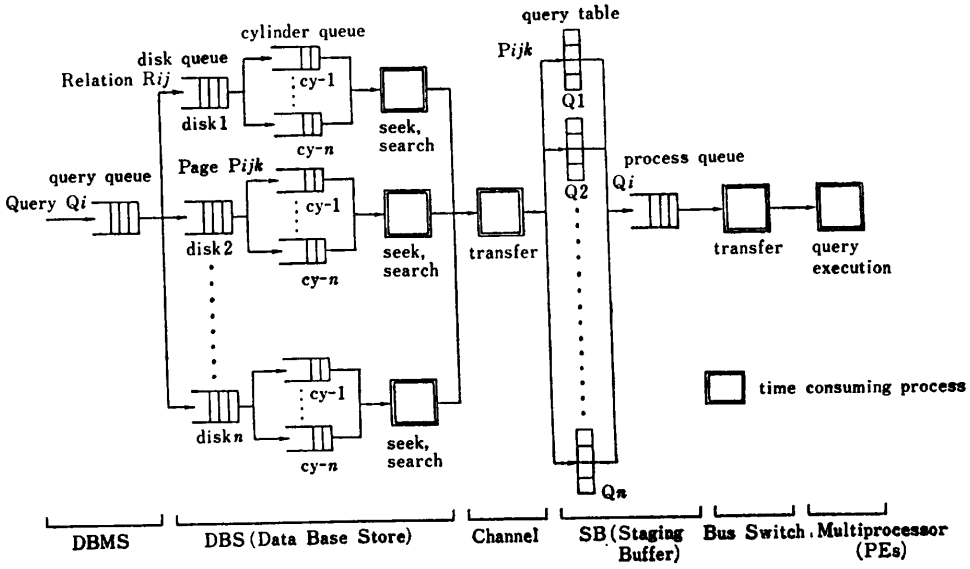


図5 シミュレーション・モデル
Fig. 5 Simulation model.

入れられる（R_{ij} を構成するページ群は，図2で示したように同じシリンダ内に格納するとデータステージング時間を軽減できる）．

（4）各ディスクドライブにおいては，FCFS 方式あるいは SCAN 方式のディスクヘッド・スケジューリングによりディスク上をディスクヘッドが走査していく（SSTF 方式については，ディスク上へのデータ配置により処理効率が大きく変動してしまうのでここでは対象としない）．各シリンダの cylinder queue を参照し，そのシリンダへの要求が存在している場合にはそのシリンダをアクセスする．このプロセスでは，各シリンダへのディスクヘッドの移動時間すなわちシーク時間が消費される．

（5）（4）でシーク動作が行われた後，チャネルが空き状態になるとサーチ動作が行われる．このプロセスではサーチ時間が消費される．平均サーチ時間はディスク半回転時間である．もしも他のディスクドライブがデータを転送中でチャネルを専有していたならば，チャネルが空くまでの待ち時間が消費される．

（6）（4），（5）でアクセスされたページ P_{ijk} がチャネルを介してステージング・バッファへ転送される．このプロセスでは，このデータ転送時間が消費される．

（7）（6）のプロセスによりステージングバッファに到着したページの識別子 P_{ijk} が query table へ登録される．

（8）Q_i が処理対象とするすべての関係を構成するのページ $P_{ijk}(j=1,\cdots,m_i,\ k=1,\cdots,h_{ij})$ が query table に登録されたならば，Q_i は process queue に入れられる．マルチプロセッサで Q_i の実行が開始されるまで，Q_i のすべての処理対象データはステージングバッファ上で待たされる．このプロセスではステージングバッファでの待ち時間が消費される．

（9）（8）のプロセスの後，Q_i の処理対象データ R_{ij} がステージングバッファからマルチプロセッサの PE 群の内部メモリへ転送される．このプロセスでは，このデータ転送時間が消費される．このデータ転送時間は，2.2 節で述べたように，関係演算処理方式によって異なる．

(10) マルチプロセッサにより問合せ Q_i の処理が行われる．このプロセスではマルチプロセッサでの問合せ処理時間が消費される．この処理時間は，2.1 節で述べたように関係演算処理方式によって異なる．

以上のようにデータベースマシンの問合せ処理過程をモデル化し，DEC 2020 (TOPS 20) 上で SIMULA によりこのモデルを実装した．なお，関係演算処理系のプロセス（9）と（10）の処理時間は，文献7）で設定した関係演算処理方式の性能評価式を用いて計算している．

5. 性能評価

5.1 アーキテクチャの評価環境

（1）ディスク・ドライブ

ディスク・ドライブの性能は表1に示すように IBM 3330 をもとに設定する．10 台のディスク・ドライブが一つのチャネルを介してステージング・バッファに接続されているものとし，各ディスク・ドライブには並列読出し機構および連想処理機構 (logic-per-some tracks) を付加できるものとする．ディスクヘッド・スケジューリング・アルゴリズムとして FCFS 方式および SCAN 方式を対象とする．

（2）ステージング・バッファ

ステージング・バッファは，システム内に存在している問合せが対象としている全データを格納するのに十分な容量をもつものとする．ステージング・バッファは半導体メモリを想定し，アクセス時間は 400 nsec/バイトとする．マルチプロセッサとステージング・バッファの間には，等分割配置用に 64 ライン，クラスタリング配置用に 8 ラインの並列データ転送路があるものとする．

（3）マルチプロセッサ

マルチプロセッサは一次元配列状に 64 台並列に結合されており，各プロセッサはおのおの内部メモリをもつ．マルチプロセッサの総内部メモリ容量は，一つの問合せで用いる処理対象データおよび中間結果のデータを格納するのに十分な容量をもつものとする．

表1 ディスク・ドライブの特性
Table 1 Disk parameters and settings.

Parameter	Value
#Cylinders/unit	404
#Tracks/cylinder	19
Track size	13,030 bytes
Track read/write time	16.7 msec
Average access time	38.6 msec
Transfer rate	15 Mbytes/sec (Parallel transfer of 19)
Total capacity	1 Gbytes

5.2 問合せ環境

性能評価に用いる問合せ[11]Q1を図6に示す．Q1は，代表的な関係演算である選択演算，結合演算，射影演算から成る．Q1を処理する場合には，まずデータベースに格納されている関係R, S, Tそれぞれに対して選択演算（R. AGE='26', S. COLOR='RED', T. DATE='JAN. 24'）を行い，おのおのの演算結果の関係R1, S1, T1を得る．次に，R1とS1の間の結合演算により関係RSを得た後，RSとT1の間で結合演算を行い関係RSTを得る．最後に，RSTの三つの属性（[CITY, PART, TOTAL]）に対して射影演算を行い，Q1の結果として関係OUTを得る．

データベース上の関係R, S, Tが，それぞれに対する選択演算の演算対象属性上でクラスタリングされてディスクに格納されている場合（図3）には，その属性についてのインデックス（図4）があれば，選択演算の結果となるタプル群だけをディスク上でアクセスできる．また，連想処理機構が付加されている場合には，インデックスがなくともディスク上で選択演算を実行できる．選択演算の結果のタプル数が，もとの関係R, S, Tのタプル数と同数，1/10，1/100となる場合を設定し，それぞれをno-index case, 1/10-index case, 1/100-index caseとよぶ．結合演算結果のタプル数については，関係RSのタプル数NRSはR1のタプル数NN0(NN0=NR1=NS1)の10倍，RSTのタプル数NRSはNRSと等しくなるものとする．

関係R, S, Tのタプル数は8,000～128,000，1タプルは64バイトとする．したがって，関係の大きさは約0.5～8Mバイト（約2～32シリンダ分）である．

本システムが同時にディスクヘッド・スケジューリングの対象とする問合せ数 m（問合せの多重度）は

relation	attribute	number of tuples
R S T	[NAME,CITY,AGE] [NAME,PART,COLOR,WEIGHT] [WEIGHT,DATE,TOTAL]	NR = N0 NS = N0 NT = N0 (N0 = 8000 ~ 128000)
R1 S1 T1	[NAME,CITY] [NAME,PART,WEIGHT] [WEIGHT,TOTAL]	NR1 = NN0 NS1 = NN0 NT1 = NN0 (NN0 = (N0/100,N0/10,N0))
RS	[NAME,CITY,PART,WEIGHT]	NRS = NR1*NS1*ALPHA = 10*NN0
RST	[NAME,CITY,PART, WEIGHT,TOTAL]	NRST = NRS*NT1*ALPHA2 = NRS (= 10*NN0)
OUT	[CITY,PART,TOTAL]	NOUT = NN0

The length of the tuple of R, S or T is 64 Bytes.

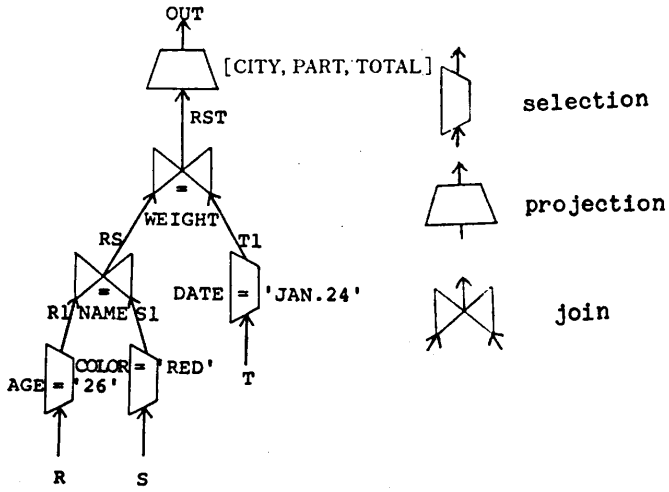


図6 問合せQ1
Fig. 6 Example query Q1.

1～40の間に設定し，各問合せはすべてQ1の構成で，処理対象の関係は各問合せごとに異なるものとする．関係演算処理系では一時に一つの問合せのみを処理するが，データステージング系では m 個の問合せの対象データ群を同時に扱う．

5.3 性能評価

5.1，5.2節の環境のもとで，関係演算処理系およびデータステージング系の技術に対して性能評価を行う．ここで用いる用語の定義は次のとおりである．

（1）応答時間 (response time)

データベースマシンに問合せ Q_i が到着して処理対象データがディスクヘッド・スケジューリングの対象となった時点から，マルチプロセッサでの関係演算処理により Q_i の結果が得られるまでの時間を応答時間とする．

（2）CPU 時間 (CPU time)

関係演算処理系において，一つの問合せ Q_i の処理対象データをステージング・バッファからマルチプロセッサへ転送し始めた時点から，マルチプロセッサでの関係演算処理により Q_i の結果が得られるまでの時間を CPU 時間とする．

（3）I/O 時間 (I/O time)

データステージング系において，問合せ Q_i の処理対象データがディスクヘッド・スケジューリングの対象となった時点から，Q_i のすべての処理対象データをステージング・バッファへ格納し終えるまでの時間を I/O 時間とする．

（4）待ち時間 (waiting time)

問合せ Q_i のすべての処理対象データがステージング・バッファにそろった時点から関係演算処理系で Q_i の処理が始まるまで，Q_i の処理対象データがステージング・バッファに滞在している時間を待ち時間とする．

5.3.1 関係演算処理系とデータステージング処理系の整合性

ここでは，関係演算処理系とデータステージング系の整合性をもとに，V-方式の有効性を検討する．図7，図8，図9に示す評価結果はトラック並列読出し機構，ディスクへの最適データ配置，SCAN 方式によるディスクヘッド・スケジューリング（問合せの多重度 $m=5$）を採用した環境のそれぞれのインデックス・ケースで，関係演算処理を PSS および V-方式により実現した場合の CPU 時間，I/O 時間，待ち時間を比較したものである．PSS，V-方式を比較すると，I/O 時間はほとんど同じであるが，CPU 時間は no-index case の 32 k タプル以上の場合を除いて V-方式が優れている．図7～図9でわかるように，CPU 時間が減少するとステージングバッファでの待ち時間も並行して減少する傾向がある．問合せの応答時間は CPU 時間，待ち時間，I/O 時間の合計値であり，CPU 時間の減少は待ち時間をも軽減させるので，3階層記憶データベースマシンでは CPU 時間の軽減が，データベース処理全体の効率を大きく改善することがわかる．したがって，従来の関係演算処理方式と異なる処理方式により CPU 時間を軽減する V-方式

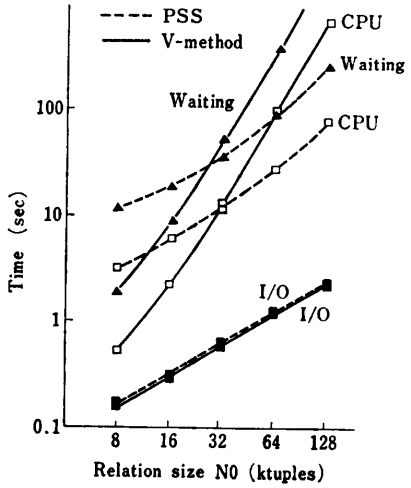


図 7 応答時間の内訳 (Q1: no-index case)
Fig. 7 Detail of response time (Q1: no-index case).

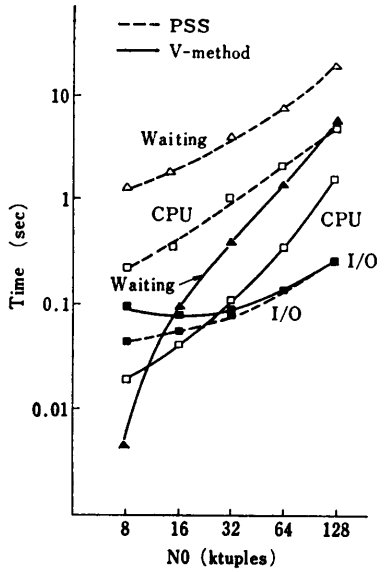


図 8 応答時間の内訳 (Q1: 1/10-index case)
Fig. 8 Detail of response time (Q1: 1/10-index case).

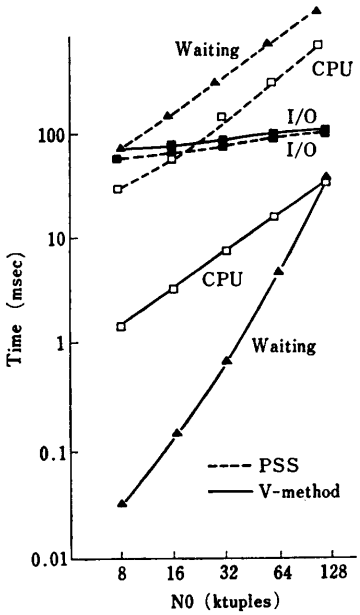


図 9 応答時間の内訳 (Q1: 1/100-index case)
Fig. 9 Detail of response time (Q1: 1/100-index case).

は，データベース処理全体の効率の点からも有効である．図7～図9に示したように，V-方式の有効範囲は広く，関係演算処理方式として有力であることがわかる．

5.3.2 データステージング技法の評価

ここでは，3章で示したデータステージング系の各技術の有効性を評価する．

(1) トラック並列読出し機構の有効性

図10，**図11**にそれぞれ，PSS，V-方式によりQ1を処理した場合の並列読出し機構の効果を示す．ここでは，処理対象の関係群は最適データ配置されている

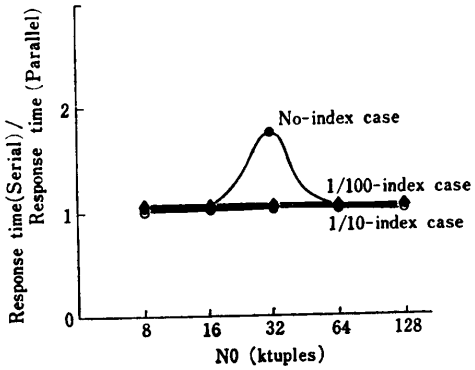


図 10 トラック並列読出し機構の効果 (PSS, Q1)
Fig. 10 Efficiency of the tracks-in parallel readout (PSS).

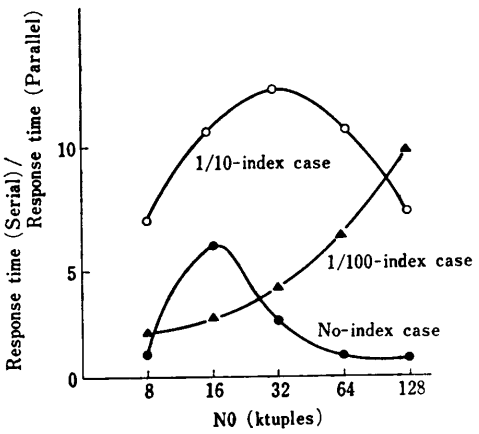


図 11 トラック並列読出し機構の効果 (V-方式, Q1)
Fig. 11 Efficiency of the tracks-in parallel readout (V-method).

ものとしている．PSSではトラック並列読出し機構により約1～2倍の応答時間の改善があるだけであるが，V-方式では約1～12倍の改善がある．この機構はI/Oネックの環境での効果が顕著であり，V-方式による関係演算処理効率の改善がこの機構の効果を引き出している．データ転送時間を1/19に軽減する並列読出し機構はデータベース処理全体の性能に与える影響が大きく，関係演算処理を効率よく処理するシステムでのデータステージング系の有効な技術であることがわかる．

(2) 最適データ配置とディスクヘッド・スケジューリングの効果

問合せの多重度mを5に設定した環境における平均シーク時間の推移を**図12**に示す．FCFS方式では，シーク時間はデータステージング要求数に無関係である．処理対象の関係のサイズが大きく，それらが多シリンダにわたって格納されている場合には，最適データ配置を行うことによってシーク時間が大きく改善されており，SCAN方式とほぼ同じシーク時間となっている．したがって，SCAN方式は関係のサイズが小さく，問合せの多重度が大きい環境で有効であることがわかる．図12に示すように，FCFS方式をSCAN方式に変えることで約10 msec，最適データ配置によりさらに約10 m sec，シーク時間を軽減できるので，30 msecのシーク時間は約1/3に軽減されている．しかし，トラック並列読出し機構によりデータ転送時間が1/19倍に短縮されることに比べるとこの効果は小さい．

(3) インデックスおよびディスクへの連想処理機

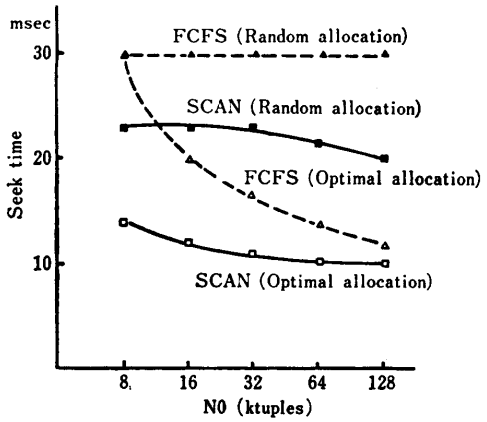


図 12 ディスクヘッド・スケジューリング，最適データ配置の効果 ($m=5$)
Fig. 12 Efficiency of the disk-head scheduling and the optimal data allocation ($m=5$).

構の付加による効果

インデックスが存在すれば，図 13 に示すように応答時間が 10～70% に軽減され，処理効率の改善は大きい．インデックスは，ディスクアクセス回数を軽減できる唯一の手段であり，I/O 時間，CPU 時間，ステージング・バッファの使用量を軽減する有効な技術である．しかし，データベースの内容に変更があった場合のオーバヘッドが大きいという欠点をもつ．

可動ヘッドディスクのディスクヘッドへの連想処理機構の付加による効果を図 14 に示す．応答時間は，40～90%に短縮されており，効果が大きい．この機構により CPU 時間，ステージング・バッファの使用量を減らすことはできるが，ディスクへのアクセス回数を減らすことはできないので I/O 時間は改善されない．したがって，インデックスに比べて効果は小さいが，データの更新に対するオーバヘッドは小さく[8]，データステージング系の有効な技術として位置づけることができる．

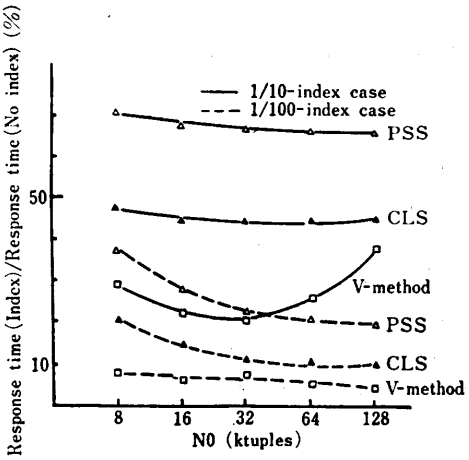


図 13 インデックスの効果 (Q1)
Fig. 13 Efficiency of the index.

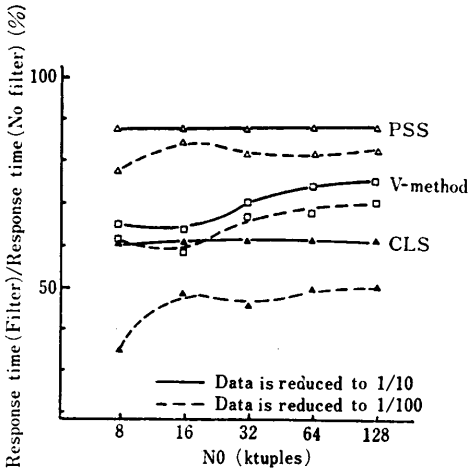


図 14 連想機構の付加による効果 (Q1)
Fig. 14 Efficiency by the attachment of the associative filter.

6. むすび

本論文では，すでに提案した関係演算処理方式 (V-方式) の有効性が，データステージング処理を含めたデータベース処理全体の効率に与える影響について解析し，V-方式による関係演算処理効率の改善が，3 階層記憶データベースマシンに対する問合せの応答時間の改善に大きく貢献することを明らかにした．また，データステージング処理のための諸技術を抽出し，現在まで明らかにされていなかった各技術の有効性を評価するツールとなるデータベースマシン・シミュレーションモデルを設定した．このモデルは SIMULA により実装されているので，3 階層記憶データベースマシンを設計する場合には，そのアプリケーションに合ったアーキテクチャを選択するためのツールとして実際に利用することができる．本論文では，このモデルを用いて実際に各技術の性能評価を行い，個々の技術がデータベース処理全体の効率に与える影響を考察した．その結果，トラック並列読出し機構，インデックスの利用，連想処理機構の付加は，データステージングの処理効率を大きく改善するが，それらは，関係演算処理系が十分な性能を有する場合のみ有効であることを示した．また，ここで提示したデータベース処理におけるディスクヘッド・スケジ

ューリングおよび最適データ配置は，インデックスの利用やトラック並列読出し機構に比べると効果が小さいことを示した．しかし，これらの技術は，現行のディスクドライブ上で簡単に実現できるので，データステージング系がネックとなるアプリケーションに対しては実現性の点で有効な技術となると考えている．

謝辞　本研究に当たり，有益なご助言をいただいた上林憲行博士（現富士ゼロックス（株)），瀬尾和男氏（現三菱電機(株)）に感謝いたします．

参考文献

1) Astrahan, M. M. et al.: System R: A Relational Database Management System, *IEEE Comput.*, Vol. 12, No. 5, pp. 42-48 (1979).
2) Banerjee, J., Baum, R. I. and Hsiao, D. K.: Concepts and Capabilities of a Database Computer, *ACM TODS*, Vol. 3, No. 4, pp. 347-384 (1978).
3) Banerjee, J., Hsiao, D. K. and Kannan, K.: DBC—A Database Computer for Very Large Databases, *IEEE Trans. Comput.*, Vol. C-28, No. 6, pp. 414-429 (1979).
4) Dewitt, D. J.: DIRECT—A Multiprocessor Organization for Supporting Relational Database Management Systems, *IEEE Trans. Comput.*, Vol. C-28, No. 6, pp. 395-406 (1979).
5) Kamibayashi, N., Kato, H., Kiyoki, Y., Ozawa, H., Seo, K. and Aiso, H.: SPIRIT: A New Relational Database Computer Employing Func tional-Distributed Multi-microprocessor Configuration, Proc. 1st International Conf. on Distributed Computing System, pp. 759-771 (1979).
6) Kiyoki, Y., Tanaka, K., Kamibayashi, N. and Aiso, H.: Design and Evaluation of a Relational Database Machine Employing Advanced Data Structures and Algorithms, Proc. 8th International Symposium on Computer Architecture, pp. 407-423 (1981).
7) Kiyoki, Y., Isoda, M., Kojima, K., Tanaka, K., Minematsu, A. and Aiso, H.: Performance Analysis for Parallel Processing Schemes of Relational Operations and a Relational Database Machine Architecture with Optimal Scheme Selection Mechanism, Proc. 3rd International Conf. on Distributed Computing Systems, pp. 196-203 (1982).
8) Ozkarahan, E. A., Schuster, S. A. and Smith, K. C.: RAP—An Associative Processor for Data Base Management, *Proc. AFIPS NCC*, Vol. 44, pp. 379-387 (1975).
9) Schuster, S. A., Nguyen, H. B., Ozkarahan, E. A. and Smith, K. C.: RAP. 2—An Associative Processor for Databases and Its Applications, *IEEE Trans. Comput.*, Vol. C-28, No. 6, pp. 446-458 (1979).
10) Seo, K., Kamibayashi, N., Minematsu, A. and Aiso, H.: A Look-ahead Data Staging Architecture for Relational Database Machines, Proc. 8th International Symposium on Computer Architecture, pp. 389-406 (1981).
11) Smith, J. M. and Chang, P.: Optimizing the Performance of a Relational Algebra Database Interface, *Comm. ACM*, Vol. 18, No. 10, pp. 568-579 (1975).
12) Stonebraker, M., Wong, E. and Kreps, P.: The Design and Implementation of INGRES, *ACM Trans. Database Syst.*, Vol. 1, No. 3, pp. 189-222 (1976).
13) Tanaka, Y., Nozawa, Y., and Masuyama, A.: Pipeline Searching and Sorting Modules as Components of a Data Flow Database Computer, Proc. IFIP-80, pp. 427-432 (1980).
14) Teorey, T. J. and Pinkerton, T. B.: A Comparative Analysis of Disk Scheduling Policies, *Comm. ACM*, Vol. 15, No. 3, pp. 177-184 (1972).
15) Uemura, S., Yuba, T., Kokubu, A., Ooomote, R. and Sugawara, Y.: The Design and Implementation of a Magnetic-bubble Database Machine, Proc. IFIP-80, pp. 433-438 (1980).
16) 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，EC-81-47 (1981).
17) 清木，田中，上林，相磯：関係データベースにおける関係演算の演算系とその実現方式，情報処理学会論文誌，Vol. 23, No. 6, pp. 653-664 (1982).
18) 清木，田中，峰松，磯田，小島，相磯：問題適応型関係データベースマシンのアーキテクチャと性能評価，情報処理学会データベース管理システム研究会，29-2 (1982).
19) 清木，峰松，相磯：3階層記憶データベースマシンアーキテクチャとその評価，電子通信学会電子計算機研究会，EC 83-46 (1984).
20) 宇都宮他：循環待ち行列モデルによるディスクヘッドスケジューリング方式の評価，信学論(D)，Vol. J 59-D, No. 9, pp. 636-643 (1976).

(昭和59年2月20日受付)
(昭和59年4月17日採録)