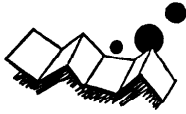


## 解説



# データベース・マシン†

田中 譲††

## 1. まえがき

1970年代に盛んに研究され、80年代を賑わすと考えられている専用マシンにデータベース(DB)マシンがある。1980年代に入って、西独 Software AG社が ADABAS マシンを発表し、米 Britton-Lee 社が IDM 500 を<sup>9)</sup>、米 Intel 社は FAST-3805 と呼ばれる半導体ディスクとデータベース・アシスト・プロセッサ DBAP を発表している。ADABAS は転置ファイルを用いた関係モデル型のデータベース管理システム(DBMS)である。ADABAS マシンは、IBM コンパチブルマシンを ADABAS 専用用に用い、IBM コンパチブルなホストコンピュータとこれとの結合を新開発のチャンネルを用いることにより可能にしたものである。IDM 500 は、関係モデル用の本格的な DB マシンで、16 bit のマイクロプロセッサの他に、ショットキ TTL を用いてパイプライン処理を行うデータベース・アクセラレータを付加することができる。付加時の性能は 10 MIPS に相当するという。半導体ディスクというのは、MOS メモリで構成され、IBM のディスク装置にプラグイン・コンパチブルにしたメモリ装置である。この他に、英 ICL 社は、レコードの連想検索機能を持ったディスク制御装置を開発している<sup>12)</sup>。この装置は CAFS と呼ばれている。

商品化の動きとは別に、関係の間の結合演算のような時間のかかる演算の高速化や、大規模データベースの管理と処理のための大容量化の研究も盛んになってきている。

さらに先進的な研究として、通産省の第5世代コンピュータの開発プランにみられるように、データベースのみならず、これとルールベースとを統合して、知識ベースを効率良く管理し処理するための知識ベースマシンの研究も始まらんとしている。

本解説では、DB マシンの開発の流れを振り返り、高速化と大容量化のための技術を整理し、代表的なマシンを比較評価し、今後の研究開発の方向を展望する。

## 2. DB マシンの分類と代表的 DB マシン

O. H. Bray らは、マシンを構成するプロセッサ数と、検索更新が2次記憶装置上で直接処理されるか否かによって、DB マシンを

- (1) SPIS 型  
(Single Processor Indirect Search)
- (2) SPDS 型  
(Single Processor Direct Search)
- (3) MPDS 型  
(Multiple Processor Direct Search)
- (4) MPIS 型  
(Multiple Processor Indirect Search)
- (5) MPSC 型  
(Multiple Processor Combined Search)

の5つのタイプに分類している<sup>1)</sup>。

SPIS 型の DB マシンには、関係モデル以外のモデルに基づくマシンもみられるが、他の型のマシンは基本的には関係モデルに基づいている。

### (A) SPIS 型

汎用プロセッサをホストコンピュータのバックエンド・マシンとし、データベースの管理と処理に専用する方式である。ファームウェアによって、データベース処理に適応させているマシンもある。

### (B) SPDS 型 (CAFS)

CAFS は処理速度が 4M バイト/秒で、数トラックを同時に検索可能である。普通のディスク装置を複数台制御するディスク制御装置であるが、連想アクセスのために、図-1に示すように、連想サーチ部とレコード部検索部を持ち、ディスクからの転送に遅れることなく、キーワードの照合を行うことができる<sup>10)</sup>。

† Database Machine by Yuzuru TANAKA (Faculty of Engineering, Hokkaido University).

†† 北海道大学工学部

検索要求: GET SUPPLIER, PRICE  
FOR (SUPPLIER="TEXAS" or PART="RESISTOR")  
and PRICE  $\geq$  3P.

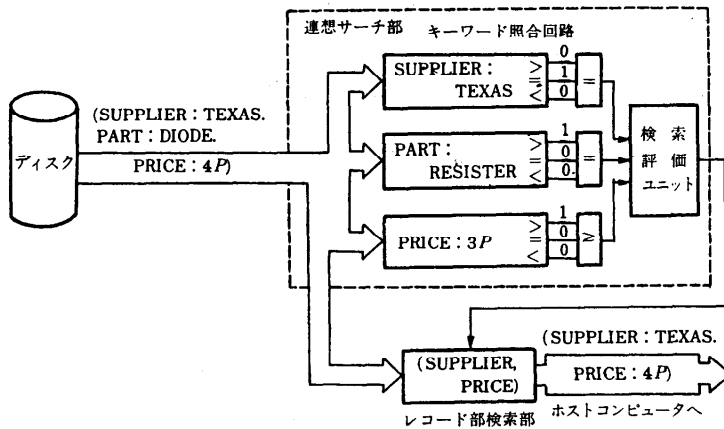


図-1 CAFS の基本アーキテクチャ

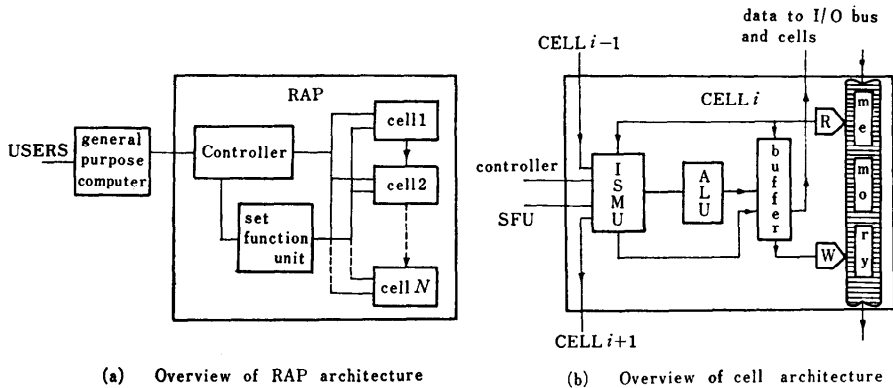


図-2 RAP. 1 のアーキテクチャ

更新機能は有していない。後に発表された **CAFS. 3** は 3 章に述べるハッシュ化ビットアレイを用いて、準結合演算\*もディスクからのデータの転送に合わせて処理する<sup>12)</sup>。

(C) **MPDS 型 (RAP. 1)**

トラックごとにヘッドを持つ特殊なディスクや、CCD や磁気バブル素子のような電子ディスクを用い、トラックごとに 1 プロセッサを持つ多重プロセッサ型の DB マシンである。RAP. 1 は図-2 のような構成を持つ<sup>10)</sup>。特殊ディスクの各トラックには 1 つの関係のタプルが順々に格納されている。選択 (selection) 演算、制限 (restriction) 演算は、タプルがバッファを通

過する間に照合器によってなされる。関係全体が特殊ディスク中にあるときは 1 回転でこれらの演算を処理する。トラック上のタプルごとに複数のマークビットが用意されており、照合結果に応じてタプルにマークを付けることができる。さらに、以下に述べるように、クロスマーク命令を用いて、 $R < A = B](S[C = 'v'])$  のような準結合演算を実行することができる。まず  $S$  の格納されているセルにおいて、 $C = 'v'$  を満足するタプルにマークを付ける。次に  $S$  の格納されているセルと  $R$  の格納されているセルを用い、 $S$  中のマークされたタプルの  $B$  属性の値を  $k$  個ずつ  $R$  の格納されているセルに送る。 $k$  はセル中の比較器の個数である。この  $k$  個の値のいずれかに等しい値を  $A$  属性としてもつ  $R$  のタプルにマークを付ける。 $S$  中のマークされたタプ

\* 準結合演算  $R < A = B]S$  は、結合演算  $R[A = B]S$  を、 $R$  に含まれる属性のみからなる属性集合に射影した関係を求める演算である。

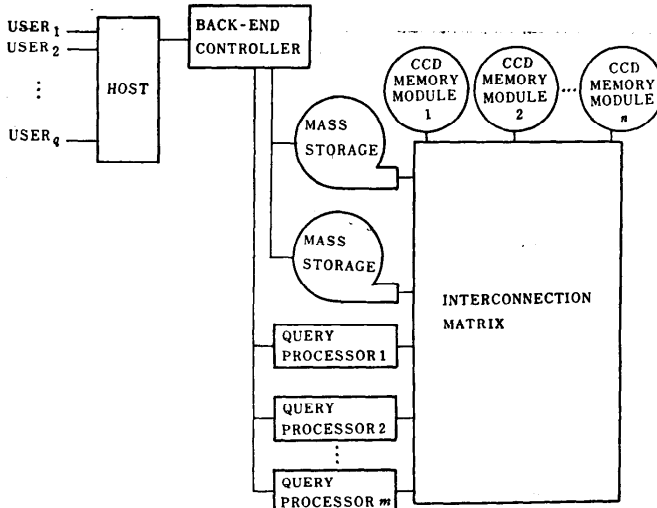


図-3 DIRECT のアーキテクチャ

ルすべてについてこの処理を行う。このように、 $S$ のマークを用いて $R$ にマークをつける演算をクロスマークと呼んでいる。各タプルは削除フラグを持ち、削除の結果生じるゴミを一か所に集めるために、バッファの途中から書き込みヘッドにデータを転送する機能が備わっている。

RAP. 1 では1トラックに1台のセルプロセッサを用い、1トラックには1種類の関係を格納した。この制限を緩めたり、関係を属性ごとに別々のセルに格納することを許すことにより種々の変化が可能である。

MPDS 型は、特殊ディスクや電子ディスクを用いるために、大容量化に問題がある。

(D) MPIS 型 (DIRECT)

MPDS 型の2次記憶装置をバッファメモリとみなし、より下位のメモリとして可動ヘッドディスクを用い、大容量化を図ったマシンを指す。バッファメモリには電子ディスクや MOS メモリが用いられる。

DIRECT は図-3に示す構成を持つ MIMD マシンである<sup>21)</sup>。バックエンド・コントローラには PDP 11/40 が用いられ、プロセッサ群の制御とメモリ管理を司る。ホストとは DMA で結合される。問い合わせプロセッサには PDP 11/03 が用いられ、コントローラによって割り当てられたページに対して、関係係数演算処理を行う。メモリモジュールは 16 kB の CCD メモリで、ディスクからのファイル転送のステージング・バッファとして用いられる。問い合わせプロセッサとメモリモジュールの間、メモリモジュールと2次

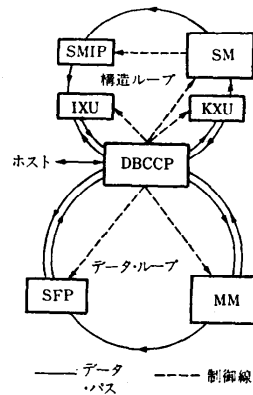
記憶装置の間はマトリクス・スイッチで結合される。結合ラインは 1 bit 幅である。タプルは固定長で、関係は固定長のページに分割して管理される。処理はページ単位に行われ、プロセッサには処理すべきページが動的に割り当てられる。中間結果は関係としてメモリモジュール上に保持される。ページアクセス、ページ割り当て、ページ処理等のページ単位の制御がすべてバックエンド・コントローラに集中し、このオーバヘッドが問題となる。

(E) MPDS 型 (DBC)

MPDS 型は、ファイルキー属性の値や、アクセス属性によってクラスタリングし、セグメントに分割する。

処理は、処理対象となるセグメントのセグメント番号を、セグメント・ディレクトリを検索してみつめることと、みつかったセグメント番号を持つ各セグメントの内容を処理することとに分けられる。セグメント・ディレクトリの検索処理は MPIS 型のアーキテクチャで行い、対象セグメントのセグメント内の処理は MPDS 型のアーキテクチャで行う。

DBC は図-4のような構成を持ち、データベース本体 ( $10^{10}$  バイト程度) をデータループのマス・メモリ (MM) で、ディレクトリ情報 ( $10^7 \sim 10^9$  バイト) は構



- DBCCP: 命令解釈および制御用プロセッサ
- KXU: キーワード変換部
- SM: 構造メモリ部
- SMIP: 構造情報プロセッサ
- IXU: インデックス変換部
- SFP: データ保全用アクセス制限プロセッサ
- MM: 大容量メモリ部

図-4 DBC のアーキテクチャ



どに長い。そこで、転送時間に重畳して処理を行い、処理のみに必要な時間をできるだけ短くするようなハードウェア・アルゴリズムがいくつか提案されている。これをストリーム処理方式と呼んでいる。代表的なものは、サーチエンジン、ソートエンジン<sup>27)</sup>、それに脈動アレイに基づいた DB マシン用 VLSI<sup>36)</sup> 等である。

(D) 符号化

データベース処理用の高性能機能モジュールの開発を難しくしている要因の1つは、比較や処理の対象データが可変長であるということである。多くの DB マシンでは、属性ごとに固定長とし、中には、キー属性はすべて一定長とするマシンもある。符号化の導入について検討し、順序を保存しない符号化関数による符号化を用いてデータベース処理が支障なく実行できるとの結論を得た報告もある<sup>29)</sup>。

この他、DBC<sup>30)-32)</sup>ではサブシステム間のパイプライン処理が導入されており、データフロー DIRECT<sup>25)</sup>や DFDBC<sup>27)</sup>、DSDBC<sup>29)</sup>にみられるように、データフロー制御の DB マシンも検討され始めている。

マシンの並列度と機能分散が進むにつれ、モジュール間を結ぶ結合ネットマークも、共有バス、近傍接続から、DIRECT によるマトリクス・スイッチの採用<sup>21)</sup>、GRACE によるリング・ネットワークの採用へと高機能化が進んでおり<sup>28)</sup>、DSDBC では、データ

の分配だけでなく、ソート能力をもったネットワークの使用が提案されている<sup>29)</sup>。

4. 大容量化の技術

図-6に DB マシンの記憶階層の移り変わりを示す。処理メモリは、処理部が直接アクセスし、処理時の作業用メモリとして用いるメモリである。バッファメモリは、2次メモリのバッファメモリとして、2次メモリのアクセス時間を短縮し、2次メモリとともに、大容量の仮想メモリ空間を提供するためのメモリである。ディスク・キャッシュは、ディスク制御装置や、ディスク駆動装置の内部に設けられたバッファメモリである。

1970年代後半は CCD や磁気バブルが DB マシンのメモリ・アーキテクチャに大きな影響を与えたが、1980年に発表された半導体ディスク FAST-3805に象徴されるように、MOS メモリの大容量化と低廉化に伴い、大容量のバッファメモリやディスク・キャッシュにも MOS メモリを用いたアーキテクチャの提案が今後増大するものと予想される。

メモリの階層化と大容量化は、ファイルのセグメンテーションを要請する。セグメンテーションでは、検索や更新が多数のセグメントに互って処理されることのないように、関係の各属性の値や、アクセス属性の種別に従ってタブルのクラスタリングが行われなければ

	処理メモリ	バッファメモリ	2次メモリ	DBマシン	クラスタリング
1972			disk	CAFS. 1	
3					
4					
1975	shift register		head qer track disk	RAP. 1	
6					
7					
8	electronic rotating disk			磁気バブル DBM	dynamic hashing <sup>40)</sup> virtual hashing <sup>40)</sup>
9	electronic rotating disk		disk	DIRECT	extendible hashing <sup>41)</sup>
1980			semiconductor disk	FAST. 3805	linear hashing <sup>41)</sup> DP network <sup>38)</sup>
1					
2	MOS RAM	MOS RAM	disk	DSDBC	Trie engine <sup>42)</sup>
198?	MOS RAM	MOS RAM	semiconductor disk disk cache moving head disk		

図-6 記憶階層の移り変わり

ならない。クラスタリングには、前もって行っておく静的クラスタリングと、処理の直前に処理内容に応じて行われる動的クラスタリングがある。静的クラスタリングは DBC で採用された<sup>31)</sup>。静的クラスタリングによるセグメンテーションでは、処理対象セグメントをみつけるのにセグメント・ディレクトリが用いられる。動的クラスタリングの 1 例に DP ネットワーク<sup>32)</sup>がある。これは、1 シリンダ上のトラックの並列読み書きが可能なディスクと、このヘッド数と同数のプロセッサからなる処理系とを結合する一種の多段スイッチネットワークで、各スイッチは比較機能を有している。DP ネットワークを用いると、シリンダ上に格納された関係を、ディスクの 1 回転の間に、任意の属性の値についてクラスタリングし、プロセッサ群に分配することができる。GRACE も動的クラスタリングの機能を持っている<sup>29)</sup>。R[A=B]S なる結合演算は、R が A 属性の値について、S が B 属性の値について静的クラスタリングされていないならば、R の 1 セグメントと S の 1 セグメントの結合演算を、(R のセグメント数) × (S のセグメント数) 回実行する必要がある。動的クラスタリングの機能があると、R と S を A と B で動的クラスタリングすることにより、セグメント間の結合演算の処理回数を (R のセグメント数) + (S のセグメント数) 程度にすることができる。これに対し、静的クラスタリングによるセグメンテーションは、選択演算や制限演算の処理の際に、アクセスしなければならないセグメント数を少なくすることを目的としている。動的クラスタリングにはこの効果はない。DSDBC では、転置ファイルを用意し、転置ファイルはキー属性の値で、関係はタプル ID で静的クラスタリングし、セグメンテーションを行っている<sup>29)</sup>。それ

とともに、処理メモリから処理メモリへのバッファメモリを介しての転送の際に、動的クラスタリングと、セグメント内のソーティングを同時に行う機能を持つ。

静的クラスタリングは適応型と非適応型に分類される。適応型は構築されるデータベースの各属性の値の分布に応じて自動的にセグメント・ディレクトリを作成し、更新によるセグメントのオーバフローやアンダフローにも自動的に対処して、セグメントの分割や融合を行い、ディレクトリも更新する方式である。これに対し、非適応型では、対象データベースにおける値の分布をあらかじめ想定しておき、これに準じてディレクトリを先に作成する。DBC は非適応型静的クラスタリングを用いている<sup>31)</sup>。適応型静的クラスタリングに関しては、動的ハッシング<sup>39)</sup>、仮想ハッシング<sup>40)</sup>、線形ハッシング<sup>41)</sup>、拡大可能ハッシング<sup>42)</sup>等の手法が DB マシンとは独立に提案されている。いずれも、ハッシュ関数をデータの挿入とともに動的に変化させ、オーバフローによるセグメントの分割や、アンダフローによる融合操作を行うものである。このハッシュ関数をディレクトリに用いることができる。上述の内、動的ハッシングと拡大可能ハッシングは順序保存関数を構成することもでき、属性値をいくつかの値域にクラスタリングすることができる。DSDBC での静的クラスタリングは適応型で、拡大可能ハッシングを用いている。DSDBC では、これを Trie エンジンと呼ばれる機能モジュールで管理・処理している<sup>29)</sup>。

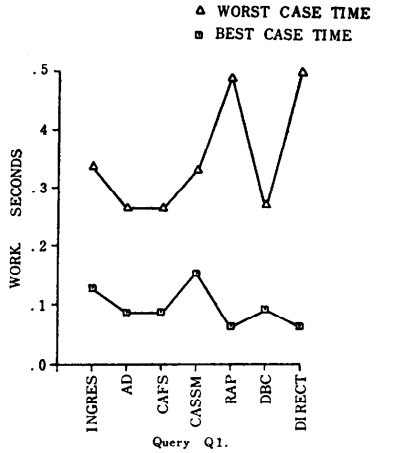
5. 性能評価

関係のタプル数を  $n$ 、(2つの関係を考える場合は  $n, m$ )、プロセッサ数を  $P$ 、1 プロセッサ内の比較器

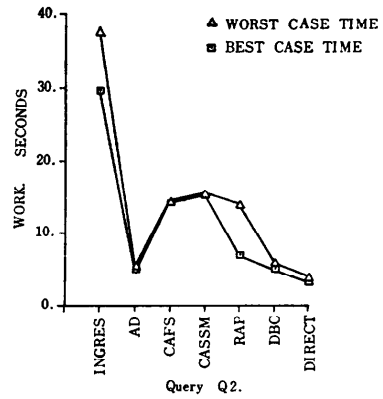
選択・制限演算		準結合演算		結合演算	
on-the-fly	$O(n)$	クロス・マーク	$O\left(\frac{n \cdot m}{P \cdot C}\right)$	クロス・マーク	$O\left(\frac{n \cdot m}{P \cdot C}\right)$
SIMD マシン	$O\left(\frac{n}{P}\right)$	ハッシュ化 ビットアレイ	$O(n+m)$	ページ内ソート	$O\left(\frac{n \cdot m \cdot \log S}{P \cdot S}\right)$
転置ファイルの利用	$O\left(\log \frac{n}{B} + \frac{B}{P}\right)$			$O(n)$ ソータ	$O\left(\frac{n \cdot m}{P \cdot S}\right)$
				動的クラスタリング	$O\left(\frac{n \cdot m}{P \cdot B}\right)$

$n, m$ : 関係のタプル数,  $P$ : プロセッサ数,  $C$ : 1 プロセッサ内の比較器の個数,  $S$ : セグメントの大きさ,  $B$ : バッファメモリの総容量

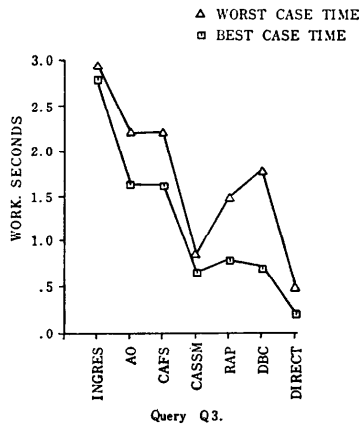
図-7 処理時間の移り変わり



A. 単一のリレーションについての問い合わせの場合  
 query Q1:  
 retrieve (QTRCOURSE. day., QTRCOURSE. hour)  
 where QTRCOURSE.instructor = "despain,  
 a.m."



B. 複数のリレーションにまたがる問い合わせの場合  
 retrieve (ROOMS. building, ROOMS. roomnum,  
 ROOMS. capacity,  
 COURSE. day, COURSE. hour)  
 where ROOMS. roomnum = COURSE. roomnum and  
 ROOMS. building = COURSE. building and  
 ROOMS. type = "lab"



C. アグリゲート関数を含む問い合わせの場合  
 Q3: retrieve (GMASTER. acct, GMASTER. fund,  
 encumb = sum (GMASTER. encumb by  
 GMASTER. acct, GMASTER. fund))

図-8 各種 DB マシンの性能比較 (P. Hawthorn 他<sup>49)</sup>)

の個数を  $C$ 、セグメントの大きさを  $S$ 、バッファメモリの総容量を  $B$  で表わす。各種関係代数演算の処理のみに要する時間は 図-7 のように移りかわっている。

ディスク・アクセスやオーバーヘッドをもある程度考慮した場合の、各種 DB マシンの総合性能の比較は、P. Hawthorn らによってなされている<sup>49)</sup>。比較されたのは、連想ディスク、CAFS、3、CASSM、RAP、DBC、DIRECT で、比較のために、改良による性能向上を見込んだ INGRES をソフトウェア・システムの代表として加えている。連想ディスクは、1 シリン

ダ上の各トラックのヘッドに比較機能を持つ論理セルを付け、並列サーチを行うように改造されたディスク装置である。単一の関係の選択演算では 図-8(a) のようになる。RAP、DIRECT で最悪ケースの性能が悪いのは、関係がバッファにない場合、この関係のすべてのセグメントを2次記憶から読み出さなければならないからである。したがって、転置ファイルを持つ INGRES よりも性能は悪くなる。一方が選択演算を受けて十分小さくなる2つの関係の結合演算の比較は 図-8(b) のようになる。連想ディスクはホストの支援

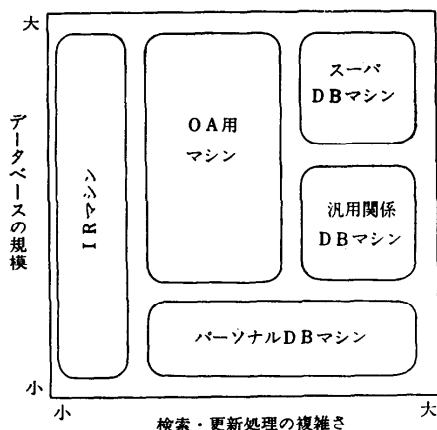


図-9 DBマシンの分化

を受ければ DIRECT と比べても遜色のない性能を示す。図-8(c)は、ある属性の値によって関係をグループに分け、グループごとに、数値をとる1つの属性の値を合計し、この値と、グループ分けに用いた属性の値との組からなる関係を求める処理を比較したものである。このような統計計算を含む検索では、処理モジュールが算術演算機能を持っているかどうかで性能を大きく左右する。

以上みたように、単純な検索では、連想ディスクや SPDS 型マシンが優れており、統計計算も含む検索では DIRECT のように処理モジュールとして汎用プロセッサを用いたものが優れていることがわかる。しかし、両方とも選択演算を含まない関係を結合するような処理に関しては、どのマシンも満足できる性能は示せないと予想できる。そのような機能をもつ超高性能 DB マシンを見指すものとしては、DSDBC、GRACE 等がある。

## 6. あとがき

今後 DB マシンの研究はどの方向に進んで行くであろうか、商用機は当分は SPIS 型であろう。MOS メモリを用いたディスク・キャッシュ・メモリや、大容量バッファメモリの採用、連想アクセス機能を持ったディスク制御装置の採用等は比較的早い時期に実現するかもしれない。

DB マシンの研究は今後分化してゆくものと思われる。図-9 に分化の方向を示す。図中の汎用 DB マシンやスーパー DB マシンでは、今後可動ヘッドディスクのアクセス時間を実際に、ないしは等価的に短縮することが重要な課題となる一方、機能の分散化は進み、

適応型の静的クラスタリングによるセグメンテーションが広く採用され、ディレクトリ処理のハードウェア化も進むであろう。特にスーパー DB マシンでは、ともに大きな2つの関係の結合演算を高速処理するために、動的クラスタリングの機能が重視されるようになるであろう。セグメント内処理では、専用 VLSI を用いたストリーム処理が、セグメントの管理にはオブジェクト指向の考えが、そして、処理の制御にはデータフロー・アーキテクチャが、今後の研究の方向にインパクトを与えるのではないかと考える。

性能の追求と同時に、DB マシンの信頼性の向上に関する基礎研究も今後望まれる研究である。高性能 DB マシンの研究は、ある意味では、始まったばかりなのかもしれない。

## 参考文献

### (A) 総論

- 1) Bray, O.H. and Freeman, H.A.: Data Base Computers, Lexington Books (1979).
- 2) IEEE Computer, Vol. 12, No. 3 (1979).
- 3) Hsiao, D.K.: Data Base Computers, in Advances in Computers, Vol. 19 (1980).
- 4) 植村, 前川: データベースマシン, 情報処理学会 (1980).
- 5) 田中: データベース・マシンの現状と未来展望, bit, Vol. 14, No. 4, pp. 38-50 (1982).

### (B) 各論

- 6) Canady, R.H. et al.: A Back-end Computer for Data Base Management, CACM, Vol. 17, No. 10, pp. 575-582 (1974).
- 7) Maril, T. et al.: The Data Computer-A Network Data Utility, Proc. AFIPS, Vol. 44, pp. 389-395 (1975).
- 8) Hakozaki, K. et al.: A Conceptual Design of a Generalized Database Subsystem, Proc. 3rd VLDB, pp. 246-253 (1977).
- 9) Epstein, R. et al.: The IDM 500-Communication Issues with Backend Processors, COMPCON, pp. 112-114 (1981).
- 10) Coulouris, G.F. et al.: Towards Content-Addressing in Data-bases, Computer Journal, Vol. 15, No. 2, pp. 95-98 (1972).
- 11) Leilich, H.O. et al.: A Search Processor for Data Base Management, 4th VLDB, pp. 280-287 (1978).
- 12) Babb, E.: Implementing a Relational Database by Means of Specialized Hardware, ACM TODS, Vol. 4, No. 1, pp. 1-29 (1979).
- 13) Copeland, G.P. et al.: The Architecture of CASSM: a Cellular System for Nonnumeric Processing, Proc. 1st Annu. Symp. Computer



- Architecture, pp. 121-128 (1973).
- 14) Ozkarahan, E. A. et al.: RAP-An Associative Processor for Data Base Management, Proc. AFIPS, Vol. 44, pp. 379-387 (1975).
  - 15) Lin, C.S. et al.: The Design of a Rotating Associative Memory for Relational Data base Applications, ACM TODS, Vol. 1, No. 1, pp. 53-65 (1976).
  - 16) Chang, H.: On Bubble Memories and Relational Data Base, 4th VLDB, pp. 207-229 (1978).
  - 17) Uemura, S. et al.: The Design and Implementation of a Magnetic-Bubble Database Machine, IFIP 80, pp. 433-438 (1980).
  - 18) Su, S. et al.: MICRONET: A microcomputer network system for managing distributed relational data bases, 4th VLDB, pp. 288-298 (1978).
  - 19) Hsiao, D.K. et al.: The Post Processing Functions of a Database Computers, Tech. Report, OSU-CISRC-TR-79-6, Ohio State Univ. (1979).
  - 20) Schuster, S. A. et al.: RAP. 2-An Associative Processor for Databases and Its Applications, IEEE Trans. Comput., Vol. C-28, No. 6, pp. 446-458 (1979).
  - 21) 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).
  - 22) Dewitt, D. J.: Query Execution in DIRECT, ACM SIGMOD pp. 13-22 (1979).
  - 23) Boral, H.B. et al.: Processor Allocation Strategies for Multiprocessor Database Machines, ACM TODS, Vol. 6, No. 2, pp. 227-254 (1981).
  - 24) Ofazer, K. et al.: RAP. 3-A Multi-Microprocessor Cell Architecture for the RAP Database Machine, Proc. Int'l Workshop on High Level Language Computer Architecture, pp. 108-119 (1980).
  - 25) Boral, H. et al.: Design Consideration for Data Flow Database Machines, ACM SIGMOD pp. 94-104 (1980).
  - 26) Wah, B.W. et al.: DIALOG-A Distributed Processor Organization for Data Base Machine, NCC pp. 243-253 (1980).
  - 27) Tanaka, Y. et al.: Pipeline Searching and Sorting Modules as Components of a Data Flow Data Base Computer, IFIP 80, pp. 427-432 (1980).
  - 28) 喜連川他: Hash と Sort による関係代数マシン, 信学技報, EC 81-35 (1981).
  - 29) Tanaka, Y.: A Data Stream Database Machine with Large Capacity, Proc. Int'l Workshop on Database Machines (1982).
  - 30) Baum, R.I. et al.: Database Computers-A Step toward Data Utilities, IEEE Trans. Comput., Vol. C-25, pp. 1254-1259 (1976).
  - 31) Banerjee, J. et al.: Concepts and Capabilities of a Database Computer, ACM TODS, Vol. 3, pp. 347-384 (1978).
  - 32) Banerjee, J. et al.: DBC-A Database Computer for Very Large Databases, IEEE Trans. Comput., Vol. C-28, No. 6, pp. 414-429 (1979).
  - 33) Berra, P.B.: Some Problems in Associative Processing Applications to Data Base Management, AFIPS Vol. 43, pp. 1-5 (1974).
  - 34) Oliver, E.J. et al.: RELACS: A Relational Associative Computer System, Proc. 5th Workshop on Computer Architecture for Non-Numeric Processing, pp. 108-114 (1980).
  - 35) Arden, B. et al.: A Single Relation Module for a Data Base Machine, COMPCON pp. 227-237 (1981).
  - 36) Kung, H.T. et al.: Systolic (VLSI) Arrays for Relational Database Operations, ACM SIGMOD pp. 105-116 (1980).
  - 37) McGregor, D.R. et al.: High Performance for Database Systems, 2nd VLDB, pp. 103-116 (1976).
  - 38) 小田他: データ分配ネットワークを用いたデータベース計算機の構成法, 信学技法 EC 80-72 (1980).
  - 39) Larson, P.: Dynamic Hashing, BIT 18 (1978).
  - 40) Litwin, W.: Virtual Hashing: A Dynamically Changing Hashing, 4th VLDB (1978).
  - 41) Litwin, W.: Linear Hashing: A New Tool for File and Table Addressing, 6th VLDB (1980).
  - 42) Fagin, R. et al.: Extendible Hashing, ACM TODS, Vol. 4, No. 3 (1979).
- (C) 性能比較
- 43) Hawthorn, P.B. et al.: Performance Analysis of Alternative Database Machine Architectures, IEEE Trans. on Software Engineering, No. 1, pp. 61-76 (1982).

(昭和57年6月18日受付)