解　説

データベース・マシン*

関野　陽** 植村俊亮***

1. はじめに

昨今の著しいデータベース技術の発展にともなって，最近データベース・システムをハードウェア面から積極的にサポートするアーキテクチャの研究が本格的に始まろうとしている．たとえば，昨年の米国NCCではデータベース・マシンなるセッションが設けられたし[1]，またこのようなアーキテクチャを研究発表する機会としてACMのSIGARCH, SIGIR, SIGMOD共催による Workshop on Computer Architecture for Non-numeric Processing も開催されるに至っている[2]．

本稿では，最近の機能分散型アーキテクチャをふまえて各地で研究されているデータ処理向きの計算機アーキテクチャの中からデータベース・マシンの研究動向について展望を試みる．

2. データ処理向きの新しいアーキテクチャ

2.1 ユーザの要求

データベース・システムに対するユーザの要求はますます多様高級化している．たとえば，ユーザはデータベース・システムの内部構造を一切知ることなくデータベースを種々のアプリケーションに対して容易に利用できることを望んでいる．この結果，データベース管理システム (DBMS) では，多数のユーザによる高水準非手続的言語による問い合わせ，高度のデータ独立性，各種アプリケーションにおける共用性，すぐれたコスト／パフォーマンス，高度の RASIS 機能 (RAS 機能及び Integrity/Security 機能) などが重要になる[3]．最近の DBMS では，データベース・システムは，(1)情報構造，(2)データ構造，(3)記憶構造，の3レベルに分離された階層構造をもつものとして捉えられることが多いようである[4]．

2.2 DBMS のインプリメンテーション

このような DBMS をつくり上げるには，少なくともつぎのインプリメンテーションが必要とされる．

(1) ユーザの問い合わせの最適翻訳，
(2) データ独立性の実現，
(3) データ構造に対する記憶構造，
(4) 記憶されたデータに対するアクセス法，
(5) 記憶構造及びアクセス法のシステムによる選択，
(6) 機密保護機構．

データベース・システムにおいては，これらのものをハードウェア，ソフトウェア，データベース管理者の間でどのように機能分担するかが問題になる．

2.3 機能分散型アーキテクチャ

上述のユーザ・アプリケーションの著しい多様化高級化に対応して，このところ機能分散型アーキテクチャが注目されている．これは近年進歩のめざましい LSI 技術と計算機システム技術の駆使によって複雑な計算機システムの効率的設計とコスト／パフォーマンスのよいシステム構成を目指すものである．

この基本的な考え方はシステム機能をシステム上に分散することであるが，分散の仕方には CPU の近傍，周辺装置／端末などを含めたシステム上，さらには計算機ネットワーク上と種々の程度がある．したがって，機能分散型アーキテクチャはユーザに近い周辺機能の強化の考えに一致し，またシステム構成の点では専用プロセッサの思想ともいえる．

2.4 データ処理向きアーキテクチャ

一方，従来のいわゆる von Neumann 型計算機は数値計算向きであって，データ処理などの非数値処理には不便な点が多いことが指摘されている．なかでも，

(1) データ処理効率****が低いこと，
(2) 2次メモリと主メモリ間のデータ転送量が過

---

* Database Machines by Akira SEKINO (Computer Engineering Division, Nippon Electric Co., Ltd.) and Syunsuke UEMURA (Computer Science Division, Electrotechnical Laboratory).

** 日本電気(株)コンピュータ技術本部
*** 電子技術総合研究所ソフトウェア部言語処理研究室

**** データ処理に必要とされる全主メモリ・サイクルのうちで実際のデータの主メモリ・リード／ライトに要するものの占める割合という定義がある．従来の計算機では 5～10% 程度になるといわれる[4]．

大になりやすいこと，

（3） データ処理に並列処理をとり入れていないこと，

が従来の計算機の問題点とされている[5]．

現在これらの問題点の解決をめざして各地で研究が進められているが，次第に上述の機能分散型アーキテクチャを指向するものがふえている．これらの研究を大まかに分類するならば，大略つぎのように考えることができよう．

（1） CPU のデータ処理機能強化

2個のデータ・ストリームをマージするハードウェア命令[7]とかディレクトリを高速サーチする連想メモリの採用[8]等の提案．

（2） ファイル処理プロセッサ

CPU から独立したマージ・プロセッサ[6]，連想プロセッサ[9),10)]，高級ディスク・コントローラ[11]，MSS コントローラ[12]などの提案や試み．

（3） データベース専用プロセッサ

CPU から独立してデータベース管理を専用に分担するプロセッサの採用[13)~15)]．

（4） 高級知能メモリ・デバイス

メモリ・デバイスに論理機能を組み込むことによって，データベースの基本論理操作をメモリ側で実行させるという提案[16)~21)]．

上記の（3）及び（4）によるデータベース管理のハードウェア・サポートは，（1）及び（2）の試みがデータベース向きにさらに発展したものと考えられる．また，この分類からもしられるように，データベース機能が徐々に周辺装置の方向に分散化される傾向が顕著である．このようなことから，データベース専用プロセッサや高級知能メモリ・デバイスは，計算機システムにおける通信制御用の前置プロセッサ (front-end processor) に対比して，**図-1** に示すように後置プロセッサ (back-end processor) とよばれるようになっている[13]．以下では，後置プロセッサ方式の代表的なデータベース・マシンを中心として述べることにする．

3. データベース専用プロセッサ

3.1 後置プロセッサ方式

後置プロセッサの概念は，ファイル処理プロセッサ

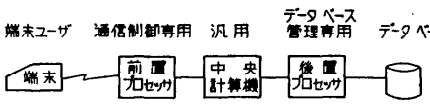

図-1 前置プロセッサ／後置プロセッサの概念

としての情報検索用マージ・プロセッサ[6]やSTARANのような連想プロセッサなど[9)~10)]においてもすでにみられる．しかしながら，後置プロセッサ方式のデータベース・マシンの長所短所は，次節の XDMS システムに関連して初めて本格的に検討されている[13]．この方式の主な長短所をまとめるとつぎのようになる．

（長所1） システム全体の経済性の向上が期待できる．なぜなら，後置プロセッサはデータベース専用であり，汎用計算機のような広範な高級諸機能を必要としないので低価格プロセッサ（ミニコン程度）ですむ．中央計算機はそれに適した処理に専念することができる．

（長所2） データベースを後置プロセッサごと複数の計算機システム間で共有することが比較的容易になる．中央計算機と後置プロセッサ間のインタフェースを高水準にして，かつこれを介する通信データ量を比較的小さくすることが可能であるからである．

（長所3） 後置プロセッサはデータベース専用であるため，汎用中央計算機よりも障害の発生拡大を防止しやすい．同様に，データベースの機密保護制御も容易になる．

（短所1） 後置プロセッサと中央計算機の間の負荷に不均衡を生じることがある．たとえば後置プロセッサの利用率がある程度低くてもがまんしなければならないであろう．

（短所2） 上記の二つのプロセッサに機能が分割された結果，端末ユーザへの応答時間が延びる可能性がある．

3.2 XDMS システム

米国ベル電話研究所では，後置プロセッサの概念のもとに XDMS (eXperimental Data Management System) とよばれる実験システムを開発した[13]．このシステムは CODASYL のデータベース用共通言語[22]を基礎にして，そのデータ操作命令 (DML) の解釈実行を後置プロセッサに分担させる構成をとっている．データベース全体の記述は固有データ構造記述言語（スキーマ DDL）によって中央計算機側で行われる．ユーザの応用プログラムは COBOL で書かれ，この中にデータベースを操作する命令がうめこまれる．

XDMS システムの中央計算機は**図-2**（次頁参照）に示すように UNIVAC 1108 であり，DMS 1100 という CODASYL 方式の DBMS をもつ．DMS 1100 はユーザのプログラムのデータベース操作命令をプリコン

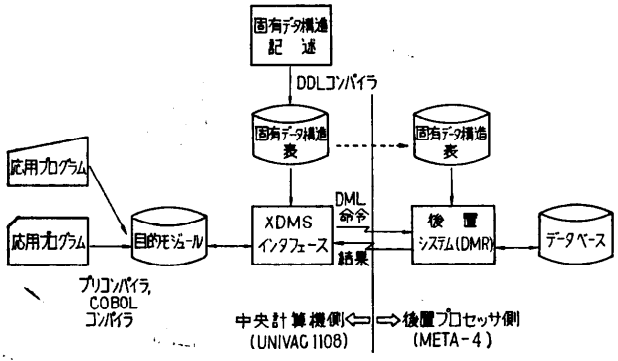

図-2 XDMS システムの概略

パイラによって COBOL の CALL 命令に変換する．COBOL コンパイラによる翻訳後のこの CALL 命令は通常は DMR (Data Management Routines) とよばれるデータベース操作サブルーチン群を呼び出すのであるが，XDMS では DMR 全体が後置プロセッサである META-4 (Digital Scientific 社) の上にあるので CALL 命令は XDMS インタフェースを介して後置プロセッサに送られる．その後，後置プロセッサ側でデータ操作を実行しその結果を中央計算機側に返答する．つぎに，このシステムの要点を示す．

中央計算機と後置プロセッサの接続：2000 ボーの通信回線で接続．

DBMS：中央計算機と後置プロセッサで機能分担．UNIVAC 1108 側は固有データ構造記述言語と応用プログラムの処理を分担し，XDMS インタフェースに 4.5 kW，バッファに 1 kW (1 W=36 bit) を使用する．一方，META-4 側はデータ操作命令の処理を分担し，DMR に 17 kW，バッファに 15 kW (1 W=16 bit) を使用する．

実際のこのシステムの開発は 1970 年 11 月に開始され，1972年 6 月には限定的な稼動に入っている．1974 年にはディスク容量とデータ転送路が改良されるならば実用に耐える状態に達した．この間に約 60,000 ドルの費用と約 6 人年の工数がかかったという．

性能的には，とくに中央計算機と後置プロセッサの間の 2000 ボーのデータ転送路が応答時間への大きな隘路になったようである．たとえば，典型的なデータ操作命令 1 個の伝送に約 250 バイトが必要とされるので，中央計算機からみた転送路上の往復だけで命令 1 個当り約 2 秒を必要とすることになる．一般に，ユーザの応用プログラムはいくつかのデータ操作命令を含むので全体としてはかなりの遅延時間が生じよう．両計算機をもっと高速の転送路で結合する必要があったようであり，また，データ操作命令のレベルが後置プロセッサのコマンドの水準として適当であるかどうかについても大いに検討の余地がある．全体的に XDMS システムの後置プロセッサ方式をみると，固有データ構造記述が両計算機に関係している点，後置プロセッサの中央計算機からの分離度を悪くしているという印象が残る．とくにその後これらの点の改良を行っているという様子はないが，XDMS システムはいずれにしろ最近のデータベース・マシンの先駆者として他のシステムの研究開発に大きな影響を与えたことは確かのようである．

4. 分散型データベースとデータベース・マシン

4.1 分散型データベース

機能分散型アーキテクチャに関連して，最近分散型データベースの必要性が議論されている[23),24)]．この目指すところは，地理的に離れたいくつかの計算機システムのもつデータベースを論理的に結合して一つのシステムの端末ユーザが通信回線などを介して他のシステムのデータベースを利用できるようにすることである．このようなシステムの技術的諸問題に対する研究は各地で始まっているが，その最大の問題は相異なる複数計算機システム間で一つのデータベースをいかにして共用できるようにするかということであろう．前章で述べた後置プロセッサ方式のデータベース・マシンはこの問題に対するシステム的解決のきわめて重要な一方法を与えている．しかしながら，XDMS システムのようにデータベース・マシンの中央計算機への依存度が高ければ高いほど異種計算機システム間のデータベースの共用は実質的に困難になる．次節では，米国 ARPANET における共用データベース・マシン Datacomputer について述べる．なお，分散型データベースでは，ユーザのデータベース・アクセスの地理的局所性を考慮したデータベース分割・分散法も重要な問題となるが，この点についてはとくに述べない．

4.2 Datacomputer

ボストン郊外にある Computer Corporation of America 社ではかねてから Datacomputer とよばれる後置プロセッサ型データベース・マシンを開発している[14)]．このシステムは ARPANET 上の多数の異種計算機システムのユーザに対して大型データベースの

オンライン・サービスを提供することを最終目的としている．すなわち，ILLIAC IV がネットワーク・ユーザに対して大型並列計算能力を提供する役割をもつのと類似して，Datacomputer はネットワーク上の多数の計算機システムの共通データベース・マシンとしてデータ・ユーティリティの役割をはたすことが期待されている．

このシステムは 1973 年暮から ARPANET におけるサービスを開始しているが，最近ではデータベースの大型化に対処できるように Ampex 社の大容量記憶システム (MSS) Tera Bit Memory[25] を接続している (図-3 参照)*．なお，プロセッサには DEC 社の PDP-10 を使用している．提供されるサービスには (1)ファイル管理機能と(2)データベース管理機能がある．つぎに，このシステムの特徴を列挙しよう．

(1) ユーザ側の計算機システムからは共通の datalanguage とよばれる高水準言語によって Datacomputer のデータベースにアクセスする．データベースを作成するばあいにもこの言語によってデータ構造記述を行う．

(2) Datacomputer はユーザ側の計算機システムから地理的に離れているので，両計算機間の通信量を最小にするように配慮されている．すなわち，ユーザからデータベースへのアクセス要求を受けると，Datacomputer はすべての処理を単独で実行しその最終結果のみを ARPANET を介してユーザに返答する．

(3) ユーザ側のシステムの多種多様なデータ表現形式やデータ構造に対応して Datacomputer はデータ変換機能をもつ．Datacomputer におけるデータ記録はデータベース作成者の使用している計算機のデータ・フォーマットでなされ，その後他のユーザがこのデータベースを読もうとするとそのユーザシステムに適したフォーマットにデータ変換しながら返答する．

(4) Datacomputer ではキーによる高速アクセスが必要な場合にインバーテド・ファイルが利用できる．

(5) Datacomputer のメモリ管理は，データベースの大きさや参照パターンを考慮した最適なステージング方法 (MSS/ディスク/主メモリ間) を用いる．

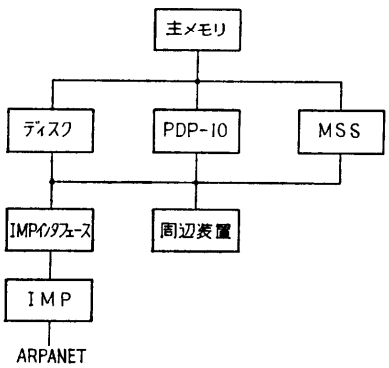

図-3 Datacomputer のハードウェア構成

* 現在は 22 GB (ギガ・バイト) の容量のようであるが，今後最大 350 GB まで拡張可能である．

上述のように，このシステムは使用目的・規模ともきわめて野心的なものとして今後の発展が注目されるが，現状では，実用的サービスはまだ(1)ARPANET 上の大学計算センターの計算機システムにファイル・メモリを一時的に貸すというファイル管理サービス，(2)ARPANET 上の全計算機システムの稼動状況に関するオンライン情報検索などに限定されているようである．しかし，今後の計画として，世界の地震情報の大型データベースや Stanford Research Institute による米国海軍船舶データベース管理システムへの利用が予定されているようである[26]．

5. 高級知能メモリ・デバイス

5.1 Logic Per Track Devices

従来から，メモリに論理機能を埋込んだ機能メモリ・デバイスの考え方があるが[27]，この考えを発展させて後置プロセッサ方式データベース・マシンの処理機能を回転型ファイル・デバイス (ドラム，ディスク，CCD メモリ，バブル・メモリなど) にまでもち込ませる提案が最近なされている．これらの提案では，とくに固定ヘッド型ディスク (各トラックに 1 対のリード／ライト・ヘッドをもつディスク) の各ヘッドにマイクロプロセッサを付加した形式の高級知能メモリ・デバイスが考えられていて，一般に logic per track devices とよばれる[28]．最近のいくつかの提案の特徴は，それが特定のデータベースを効率よく処理するデータベース・マシンとして検討されていることである．これらの提案には(1)CASSM システム，(2)RAP システム，(3)RARES システムなどがあり，2.4 で述べたデータ処理上の問題点を下記の共通的手法で解決している．

(1) 各マイクロプロセッサによる高速高級サーチ機能 (内容及び文脈による連想的サーチ)

(2) 各マイクロプロセッサの同時動作性 (並列処理)

(3) ファイル・デバイス上における直接的データ

処理（多量データ転送の回避）

いずれもこの数年間の新しい研究成果である．以下にこれらのシステムの現状と特徴点について述べる．

5.2 CASSM システム

米国フロリダ大学の G. J. Lipovski, S. Y. W. Su らによる試作中のシステムであり，CASSM (Context Addressed Segment Sequential Memory) システムとよぶ．スタンド・アロン型データベース・マシンであり，ディスクを主メモリとする計算機といった感じも強い．ユーザの高水準言語 (CASDAL) による問い合わせは中央計算機によって CASSM 命令のプログラムに翻訳される．CASSM は翻訳後のプログラムをディスク上に受けとり，一つずつプログラムで指定される操作をデータベースに対してほどこす．その後，最終結果は CASSM から中央計算機を介してユーザに返答される．

CASSM のディスク上の情報はそれぞれ3ビットのタグ，3ビットの状態表示，32ビットの情報部からなるタグ付き語の集合として記憶される．これらの情報は，タグによって，CASSM 命令語，待ち行列，削除語，データ(ポインタ，数値，ストリング，デリミタ)などに分類される．データの記憶構造は基本的に階層構造であり，階層構造は左右に展開された形式で記憶される．これに対して，各ヘッドに1個ずつあるデータ処理機能（マイクロプロセッサ）は**図-4** に示す構造をしていて，中央計算機との入出力，CASSM 命令の読出し，解釈・実行，さらには削除語に対するガーベッジ・コレクションなどを行う．CASSM 命令は大別すると Jump, Load PSW, Delimiter Mark, Q Search, Pointer Transfer, Rewrite, 待ち行列操作などに分類され，これらの組み合わせによって効率的な内容・文脈による連想的検索やデータの更新などが可能になる．さらに，図-4 のデータ処理機構のもつパイプライン処理能力（命令読出し，命令実行，命令後処理）などによってデータ処理速度が速められている．

5.3 RAP システム

カナダのトロント大学の E. A. Ozkarahan, S. A. Schuster らが提案しているリレーショナル・データベース[29]専用のデータベース・マシンであり，RAP (Relational Associative Processor) システムとよぶ．CASSM と同様に，中央計算機による翻訳後のプログラムを実行するスタンド・アロン型の logic per track device

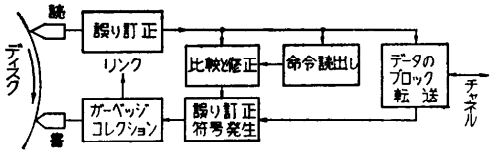

図-4 CASSM のマイクロプロセッサの構造

である．一部に中央計算機とのインタフェースに不明確なところがあるものの CASSM よりも進んだいくつかの特徴をもっている．

RAP のメモリ上には CASSM と異なってデータ（リレーショナル・データベース）のみが記憶される．各トラックにはたかだか1個のリレーションが収容され，リレーション（ファイル）を構成する各テューブル（レコード）は**図-5** に示すように1次元的に並べられて記憶される．トラックの始点にはヘッダ・ブロックが存在し，リレーション名及び各ドメイン名を記述している．後続する実際のテューブルは可変長データ・フォーマットをとり，各テューブルにはマーク用の4ビットが付加されている．一方，各ヘッドのデータ処理機構（マイクロプロセッサ）は RAP の中央制御のもとに同一命令を並列的に実行する機能や入出力機能，ガーベッジ・コレクション機能などをもつ．RAP 命令セットは少なくともリレーション代数と同等の能力をもつように設計されていて，基本機能として選択操作，結合操作 (implicit join)，集合操作，射影，自由変数，算術集合関数，算術的更新などを含んでいる．命令は原則としてつぎの形式をとる．

〈OPCODE〉[〈SPECIFICATION〉:
〈QUALIFICATION〉]

大部分の命令の実行時間はディスクの1～2回転程度の時間であり，この結果 RAP の性能は，リレーショナル・データベースを従来の計算機システム上でイン

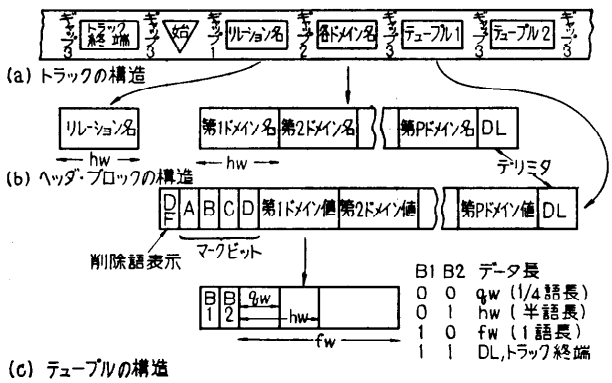

図-5 RAP の記憶構造

バーテド・ファイルを用いて処理するばあいに比べて単純な検索において 80～160 倍，結合操作において 0.4～200 倍，更新では 2,000～5,000 倍に高められるという[19] (200 トラック程度のシステムを想定).

しかしながら，RAP のもつファイル・メモリには約 10 メガバイトのデータベースしか収容できないので，最近ファイル空間を仮想メモリ化によって拡大する方法を提案している[20]. なお，ファイル空間は元来の固定ヘッド・ディスクと可動ヘッド・ディスクから成るものとし，ユーザの問い合わせに関係するすべてのリレーションは RAP 上でのデータ処理に先立ってあらかじめ可動ヘッド・ディスクから固定ヘッド・ディスクに一括的にロードするものとしている.

5.4 RARES システム

米国ユタ大学の C. S. Lin と Smith 夫妻らはリレーショナル・データベースを効率的に処理する RARES (Rotating Associative Relational Store) というデータベース・マシンを提案している[21]. このシステムは上記の 2 システムより中央計算機への依存度が大きい.

RAP と同様にリレーションをほぼ生の形で固定ヘッド・ディスクに記憶するのであるが，連続する複数トラックからなるバンド内でこれらのトラックに直交する方向にリレーションを記録するところに特徴がある. また，データ処理機構は各トラックに 1 個ずつ付加するのではなく，バンドに 1 個ずつですますようにしている. したがって，厳密には logic per track device ではないがその変形といえよう. このような構成をとることによって，(1)選択されたテープルの高速出力が容易，(2)データ処理機構の必要とするメモリ量が小さい，(3)ソート順によるテープルの出力が容易になるなどの利点を実現することができるとしている. 今後，このような詳細なインプリメンテーション上の諸問題の検討も重要になるものと思われるが，このシステムではまだ未検討のいくつかの大きな問題も残されているようである.

6. おわりに

以上最近のデータベース・マシンの研究の背景及び現在研究されているデータベース・マシンの中で代表的と思われるものについてその特徴を中心として記述した. 紙面の制限から，個々のデータベース・マシンの詳細にまで立ち入って説明することはできなかったが，この点については個々の参考文献を参照されたい. これらの研究に共通している特徴は，2.4 で指摘したデータ処理における従来の計算機アーキテクチャの問題点を解決するために，(1)データ処理効率を高める連想的処理機能の導入，(2)後置プロセッサ方式の採用による主メモリへの多量のデータ転送の回避，(3)可能な限り並列処理方式をとるという諸技術が検討されていることである. また，このような研究を通じて，ユーザの要求に合致した今後のデータベース・システムの実現にハードウェア面から巾広いサポートをしようとしている.

本稿では個々のデータベース・マシンの問題点を十分に記述する余裕もなかったが，読者自身でお考えいただけるようにデータベース・マシン設計上(評価上)の要点をつぎにまとめておこう.

(1) 中央計算機とのインタフェース: 中央計算機との機能分担，中央計算機からの独立性，コマンド・プログラムの水準，中央計算機との通信インタフェース (チャネル，通信回線).

(2) コマンド命令の解釈実行: ファームウェア，ハードウェア.

(3) メモリ・スペースの有効利用: データの符号化，データのメモリ割付.

(4) 内容あるいは文脈によるレコードへの効率的アクセス: ISAM, VSAM, B-tree, ハッシング，インバーテド・ファイル，連想的サーチ，並列サーチ，データ構造再構成.

(5) キーの大小順による効率的アクセス: ソートされたレコード，ハードウェア面の考慮.

(6) データベースの効率的更新: ハードウェアによるレコードの削除・追加・変更やガーベッジ・コレクション，(バッチ更新).

(7) 低コスト高速大容量ファイル・メモリ: ドラム，ディスク，CCD メモリ，バブル・メモリ，MSS，メモリ・ハイアラーキ.

(8) データベース・マシン全体のコスト面の配慮: LSI 指向のアーキテクチャ，ファームウェアやマイクロプロセッサの活用，階層構造 DBMS.

現在各地で研究開発中のデータベース・マシンでは，まだこれらの要点を十分に検討しつくしていないというのが筆者の偽らない所見である. とくに，その中でも中央計算機とのインタフェース (たとえば，DDL, DML，さらにそれらのコンパイル機能の機能分担，コマンド・プログラムの水準など) の点ではなお一層の研究が必要と思われ，そのようにして初めて 3.1 で述べた後置プロセッサ方式の利点を活かすことができ

るものと考える．ここでくり返すまでもなく，本稿ではデータベース・システムにおける記憶構造（及びデータ構造から記憶構造への写像）を中心としたデータベース・マシンのハードウェア面を重点的に述べた．したがって，ユーザの希求する今後のデータベース・システムの実現のためには，ソフトウェア手法を中心とした情報構造及びデータ構造面の諸機能をデータベース・マシンに具備することが必須であることはいうまでもない．

最後に，本稿の用意に当たりご意見を聞かせて頂いた日本電気（株）中央研究所の真名垣昌夫氏に感謝する．

参 考 文 献

1) Proc. AFIPS NCC, Vol. 44, pp. 379～395 (1975).
2) 関野陽：非数値処理アーキテクチャ会議に出席して—DBMS のハードウェア・サポートの研究，情報処理学会データ・ベース研資料 29 (1976).
3) データ・ベース管理システムの研究動向—MIT Prof. M. M. Hammer，日本電子工業振興協会レポート 51-C-315 (1976).
4) たとえば，「情報処理」本号，石田喬也：ANSI/SPARC データ・ベース・システム概念，pp. 911～915.
5) G. J. Lipovski et al.: On Non-numeric Architecture, Computer Architecture Newsletter, 4, 1 (1975).
6) L. A. Hollaar: An Architecture for the Efficient Combining of Ordered Lists, Second Workshop on Computer Architecture for Non-numeric Processing (1976).
7) B. W. Jordan, Jr. et al.: File Operations in a Streaming Processor, ibid. (1976).
8) P. B. Berra. et al.: A Multiple Associative Memory Organization for Pipelining a Directory to a Very Large Data Base, ibid. or COMPCON 76 Spring (1976).
9) J. A. Rudolph: A Production Implementation of an Associative Array Processor-STARAN, AFIPS FJCC, pp. 229～241 (1972).
10) Y. Chu: Architecture of a Nonnumerical Accumulator, Second Workshop on Computer Architecture for Non-numeric Processing (1976).
11) IMSAI 108 Intelligent Disk System User's Manual, IMS 社 (1976).
12) たとえば Introduction to the IBM Mass Storage System, IBM 社 GA32-0028 (1974).
13) R. H. Canaday, et al.: A Back-end Computer for Data Base Management, CACM, 17, 10, pp. 575～582 (1974).
14) T. Marill, et al.: The Datacomputer—A Network Data Utility, AFIPS NCC, pp. 389～395 (1975).
15) D. R. Anderson: Data Base Processor Technology, AFIPS NCC, pp. 811～818 (1976).
16) G. P. Copeland, et al.: The Architecture of CASSM—A Cellular System for Non-numeric Processing, Proc. First Annual Symp. on Computer Architecture, pp. 121～128 (1973).
17) S. Y. W. Su, et al.: CASSM: A Cellular System for Very Large Data Bases, Proc. International Conference on Very Large Data Bases, pp. 456～472 (1975).
18) E.A. Ozkarahan, et al.: RAP—An Associative Processor for Data Base Management, AFIPS NCC, pp. 379～387 (1975).
19) E. A. Ozkarahan, et al.: Performance Evaluation of a Relational Associative Processor, Second Workshop on Computer Architecture for Non-numeric Processing (1976).
20) S. A. Schuster, et al.: A Virtual Memory System for a Relational Associative Processor, ibid. or AFIPS NCC, pp. 855～862 (1976).
21) C. S. Lin, et al.: The Design of a Rotating Associative Memory for Relational Database Applications, ACM Trans. on Database Systems, 1, 1, pp. 53～65 (1976).
22) たとえば，「情報処理」本号，植村俊亮：CODASYL 方式のデータベース・システム，pp. 892～903.
23) P. G. Comba: Needed: Distributed Control, Proc. International Conference on Very Large Data Bases, pp. 364～375 (1975).
24) R. W. Peebles, et al.: A Computer Architecture for Large (Distributed) Data Bases, ibid., pp. 405～427 (1975).
25) S. Damron, et al.: A Random Access Terabit Magnetic Memory, AFIPS FJCC, pp. 1381～1387 (1968).
26) SRI における D. Sagalowicz, E. Sacerdoti, 古川康一氏との私的会話による (1976).
27) たとえば，W. H. Kautz: Cellular Logic-in-Memory Arrays, IEEE Trans. on Computers, C-18, 8, pp. 719～727 (1969).
28) D. L. Slotnick: Logic Per Track Devices, Advances in Computers, Academic Press (1970).
29) たとえば，「情報処理」本号，穂鷹良介，渋谷政昭：データベースの関係形式，pp. 904～910.
30) 植村俊亮：データベース・マシンのアーキテクチャ，昭和 51 年電気四学会連合大会発表予定 (1976).

（昭和51年7月21日受付）
（昭和51年8月12日再受付）