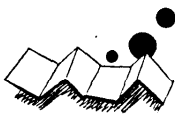


解説



連想処理とデータベース†

市川忠男† 平川正人††

1. まえがき

1962年の“フォン・ノイマンの提唱¹⁾”を契機として、計算機はそれぞれの時点での現実的な要求を踏まえて急速に発展してきたが、そこでは主として数値演算に主眼が置かれていた。そして CRAY-I や CYBER-205 などのスーパーコンピュータにその頂点を見ることが出来る。

ところが近頃では、これまでの数値処理指向型の計算機ではその有効性が十分に発揮できないような応用分野が広がりつつある。たとえば、情報化社会の促進に重要な役割を担う大規模データベースなどであり、その処理対象は非数値データにまで及んでいる。このため新しい思想に基づいた計算機の出現が必須となり、今日の記号処理研究の意義づけを与えている。

非数値処理を効率よく行うものに連想処理がある。連想処理に関する研究は古くからなされているが²⁾、近年とくにデータベースへの応用との関連のもとに著しく発展した。連想処理ではアドレスの概念がなく、すべてデータ項目の内容によってそれらが選択抽出される。さらに、連想処理は本質的に並列処理であり、データベースなどの大容量非数値データを取り扱うのに都合がよい。

一方、連想処理をハードウェア的にサポートする連想記憶は論理機能つき記憶 (logic-in-memory)³⁾ の形態をとり、ユニットの機能の画一性とシステム構成上の規則性は VLSI 技術との整合性を保障し、連想処理の将来性を明るくしている。

このような観点から、本稿では、とくにデータベースとの関連の上で連想処理方式の分類を行い、連想プロセッサとしてのそれらの実現例を示す。さらに、VLSI 技術が処理の基本アルゴリズムに及ぼす影響について考察を付す。

2. 記号処理とデータベース

2.1 技術動向

記号処理の応用分野は、知識工学、自然言語処理、数式処理、データベースなどの多岐にわたるが、これらの処理においてはパターンマッチングがその基本となる。

ところで、現在の応用コンピュータでは、それらのほとんどがノイマン型アーキテクチャの上に構築されているので、パターンマッチング処理もまたソフトウェアによって実現されている。たとえば、LISP、SNOBOL、あるいはソフトウェア後置形のデータベースマシン^{4), 20)}などに見られる通りであるが、ここでは処理速度の向上に限界がある。

他方、たとえば LISP についてみれば、LISP 言語の特性、あるいはリストデータの動的割りつけに対する詳細な評価が行われ⁵⁾⁻⁷⁾、LISP 言語自体を従来の計算機上で効率よくインプリメントする技術も整ってきてはいるが、それらは本質的な解決を与えるものとは考えられない。今日の集積回路技術の上に立てば、むしろアーキテクチャ・レベルで直接パターンマッチング処理を実現すべきであろう。LISP マシン^{8), 9)}、SNOBOL マシン、ハードウェア後置形のデータベースマシン²⁰⁾などがそれに属する。

LISP マシン や SNOBOL マシンについては、本特集号の別稿で詳しく述べられるので、以下ではデータベース応用に土俵を限定する。

2.2 フォン・ノイマン・ボトルネック

ノイマン型計算機¹⁾は、図-1 に示すように、1つの ALU、主記憶装置、2次記憶装置、および I/O プロセッサから構成される。さらにプログラムはプログラムカウンタによってシーケンシャルに実行される。

ところで、データベースにおいては、対象とするデータが膨大であるため、必然的に2次記憶装置の存在が強調され、そのため (i) I/O プロセッサの処理オーバーヘッドと (ii) ALU の処理オーバーヘッドが生じる。その上 (iii) アドレスという概念によるセマ

† Associative Processing and Databases by Tadao ICHIKAWA and Masahito HIRAKAWA (Department of Electrical Engineering, Faculty of Engineering, Hiroshima University).
†† 広島大学工学部第二類(電気系)

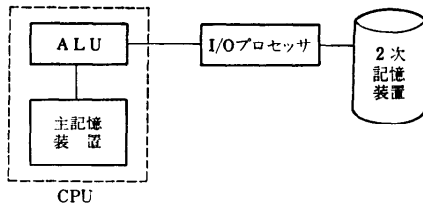


図-1 フォン・ノイマン型計算機の構成

ンティクスの汚損 (セマンティック・ギャップ)¹⁰⁾ が本質的に内在する。これらがいわゆるフォン・ノイマン・ボトルネックであり、ノイマン型アーキテクチャ上にデータベースを構築することの限界を説明している。

次に述べる連想処理、およびそれに基づく新しいアーキテクチャは、これらの問題を切り抜けるための確かな手掛りを与える。

2.3 連想処理とその効用

連想処理の歴史は古く、クライオトロン²⁾にまでさかのぼることができる。連想処理の基本は内容アドレッシング (content-addressing) であり、その実現例は内容アドレスメモリ (CAM: content addressable memory) に代表される^{11),12)}。

内容アドレッシングとは、内容によってデータをアクセスするものであり、従来のアドレス情報によるものとは質を異にする。この様子を 図-2 に示す。この場合、単に内容の一致、不一致だけでなく、greater than, less than などといった高度な評価を行うこともできる。さらに、CAM における演算形態はビット直列ワード並列であり、並列動作を基本とする。

連想処理は、その機能水準によって 図-3 に示すように3つに分類される^{3),4)}。すなわち、(i) 内容アドレス (content address) 形、(ii) 文脈アドレス (context address) 形、および (iii) 高次連想モデル形である。いずれの場合も、比較的単純な操作を基本として処理

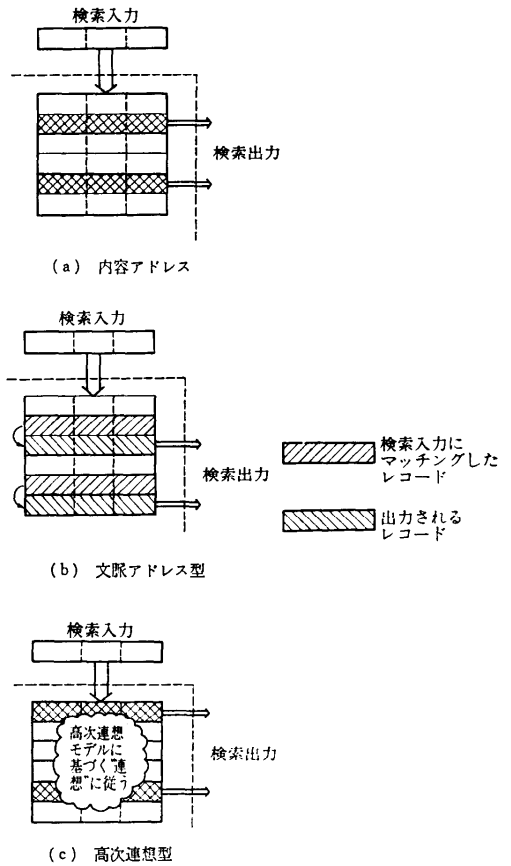


図-3 連想処理の機能水準による分類

の並列性を高めようとするものであり、とくにデータベース応用においては、検索対象を格納する2次記憶装置上でのパターンマッチングに適用した場合に効果を上げる。この点、記号処理における他の応用、すなわち自然言語処理や数式処理と大きく異なる。期待される効果は (i) サーチ処理におけるセマンティック・ギャップの減少、(ii) 処理の高速化、(iii) 2次記憶上

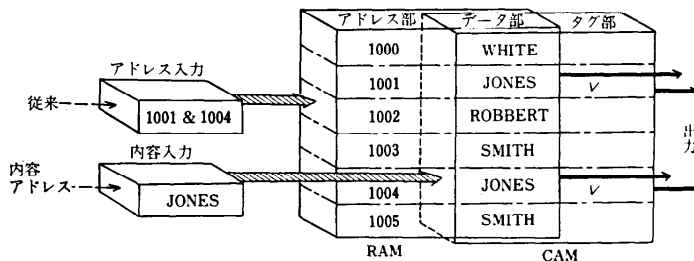


図-2 従来方式と内容アドレス方式の比較

でサーチを行うことによる I/O チャンネルの負荷の減少、および (iv) 処理の流れの単純化、などである。

3. データベースにおける連想処理

データベースへの応用にみられる連想機能を、その実現技法、ならびにロジックとメモリの構成の観点から分類、考察する。

3.1 連想機能の実現技法

連想機能は、その実現技法を尺度にとると次の3つに分類できる^{13), 14)}。

(a) 内容アドレッシングによる実現

内容アドレッシングについてはすでに述べた通りである。その性質上、以下に述べる (b), (c) の方式に比べて、(i) 2次データベースのスペースオーバーヘッドがない、(ii) したがって、データベースの変更に対する保守・管理が簡単、さらに (iii) 汎用性がある、などの利点をもっている。ただし、コスト的に高価なシステムとなる。

STARAN¹⁹⁾, DIRECT⁴¹⁾, および論理セル形のデータベースマシン¹⁷⁾などがこの範ちゅうにはいる。

(b) ハッシングによる実現

この方式では、データ変換モジュール(ハッシュ表)を介して間接的にデータベースをサーチする。すなわち、検索条件(キーデータ)を満足する可能性のあるレコードのロケーションを算出するためにハッシングを用いる。ハッシングによってデータベースをクラスタリングし、検索の場合には、その中で条件を満足するクラスタのみを参照する。これによって検索が高速化される。ただし、ハッシュ関数のとりかたによってはクラスタの大きさに不均衡を生じ、処理効率の低下を招くこともある。最近では、ハードウェアによるハッシングの試みも多い。

例として、DBC^{29)~33)}, CAFS^{37), 38)}, あるいは武蔵野通研のデータベースマシン⁴⁶⁾などがある。特に CAFS では、数種類のハッシュ関数によって同時にハッシングを行い、効率の低下をおさえるように工夫されている。

(c) 転置ファイルによる実現

転置ファイルはデータベースをアクセスするための2次データベースであり、これを介して検索条件を満足するレコードのロケーションが導かれる。これはデータベースアクセス効率の改善を目的としたものであり、連想機能の1つの実現法と見なすことができる。しかしながら、転置ファイルとデータベースとの一貫

性の保持が困難であることなどの問題が残されている。

この転置ファイルによる連想機能の実現例としては、ADABAS、データコンピュータ²³⁾, GDS²⁴⁾, などがある。また、転置ファイルの考え方は DBC における構造メモリ (Structure Memory)³⁰⁾ にも導入されており、ここでは大容量ディスクを扱うための前処理を担当している。

3.2 連想処理のシステム構成

この節では、連想処理のシステム構成を、メモリとロジックの機能水準、およびそれらのかかわりの観点から分類する。

(a) 連想チップによるもの

ロジックとメモリの双方が単一のチップ上に配置される。連想機能の実現には内容アドレッシングが基本となる。したがって、この方式の実現は LSI 技術の進展に大きく影響される。すなわち、チップ上に連想機能のすべてを埋め込むため、素子の超高集積化が要求されるわけである。また、処理が原則的に1ユニット時間内で完了するため、高速化がはかれることが特徴である。

現時点までに実用化されているものとしては、STARANにおける連想モジュール (Array Module) があげられる¹⁹⁾。これは主記憶上の連想処理ユニットであり、 $4 \times 256 \times 256$ ビットのデータに対して内容アドレッシングによる連想処理を実現する。容量の点ではまだ十分とは言えないが、将来実用規模のチップがデータベース構成要素としてコスト的に満足されるようになれば、非常に大きなインパクトを与えるだろう。

(b) 既存メモリと連想ロジックによるもの

連想機能は既存メモリ (RAM, 磁気ディスク, CCD, 磁気バブルなど) に連想ロジックを組み合わせることによって実現される。これは (a) の方式と比べると処理速度の点で劣るが、現在のハードウェア技術水準からみると、大量のデータを高速に処理することに対する現実的な手段を提供するものと言える。また、コスト的にも十分満足できる段階にある。

ここで、連想ロジックを実現する手段としては、

3.1 で述べたすべての技法が適用される。

たとえば、この形態の上で内容アドレッシングを用いるものとして、Slotnick によって提案されたモデル²⁵⁾や、論理セル形に基づいた実用化の試み^{17), 18)} などがある。また最近では、磁気バブルや CCD に連想ロジックを組み込んだインテリジェント・メモリの提

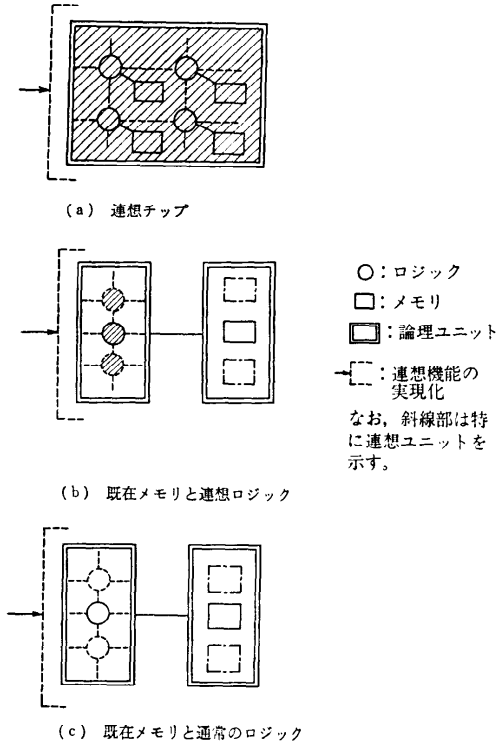


図-4 連想処理のシステム構成

案もさかんである^{51)~54)}。

次に、ハッシングを利用した例としては、前述の CAFS^{37),38)}や武蔵野通研のデータベースマシン⁴⁶⁾などがある。DBCのキーワード変換ユニット(Keyword Transformation Unit)³⁹⁾もこの範ちゅうにとらえることができる。

一方、既存メモリと連想ロジックの結合形態の上で、連想ロジックとして転置ファイルを用いるものは、インデックスプロセッサ、あるいは転置ファイルプロセッサとも呼ばれる。この例としては DBC の構造メモリ³⁰⁾や Stellohorn のシステム⁵⁵⁾などがある。

(c) 既存メモリと通常のロジックによるもの

データベース処理はソフトウェアベースで行われる。これはソフトウェア後置形のデータベースマシンに対応する。転置ファイルによる連想機能の実現が基本であるが²³⁾、ハッシングの機能をあわせ持つものもある²⁴⁾。応用分野が限定された場合にはこれを仮想的な連想プロセッサと見なすこともでき、周辺環境が規定される以前の研究開発段階で柔軟な機能を提供する。

以上3つの実現法を 図-4 に示す。

4. データベースマシンと VLSI インパクト

4.1 データベースマシンとそのアーキテクチャ

すでに、データベースマシンに関してはすぐれたサーベイがいくつかあり^{15)~22)}、いろいろな角度からデータベースマシンの分類が行われているが、本稿では記憶媒体とそれに付加するロジックの機能水準から分類を試みる。その分類手段は3章のそれに従うものとする。ただし、記憶媒体については、(i) データベース本体を記憶する大容量記憶装置 (Mass Memory: MM) と、(ii) 主記憶装置あるいは MM の検索を効率的に行わせるために用意される記憶装置 (Primary Memory: PM) とを区別する。これによって、より詳細な分類が可能となる。

これまでに提案されている主だったデータベースマシンについて分類を試みた結果を 図-5 に示す。カテゴリ1に属するマシンは、ソフトウェア後置形と呼ばれるものである。カテゴリ2からカテゴリ5はハードウェア後置形であり、カテゴリ2に属するマシンは、一般に論理セル形と呼ばれる。カテゴリ3は機能分散形、カテゴリ4は連想メモリ形に対応する¹⁵⁾。さらに、カテゴリ5を新たに設定した。これを高速連想処理形と呼ぶことにする。最近さかんに提案されているものはすべてハードウェア後置形に属するものであり、以下ではこれに土俵を限定する。

論理セル形のデータベースマシン¹⁷⁾は、大容量記憶装置のトラック (あるいはループ) ごとにロジックを付加し、大容量記憶装置上で直接検索を行わせるものである。しかも各トラック (ループ) ごとに並列処理

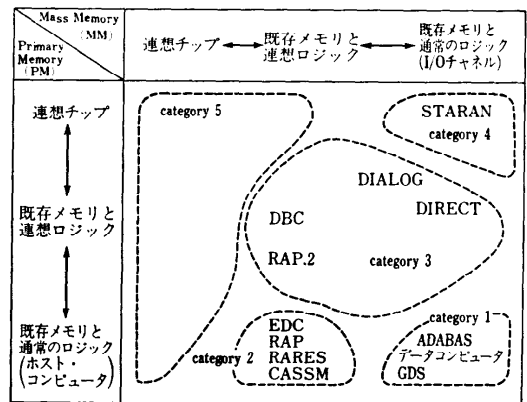


図-5 データベースコンピュータの分類

を行う。そのため、I/O チャンネルとホストコンピュータの負荷の軽減がはかれる。その反面、複雑な検索条件をもつ問合せを効率よく処理できないこと、ロジックの処理負荷の不均衡による効率の低下などといった欠点がある。その上、アーキテクチャ面での制約から、比較的小規模データベースに応用が限定される。

カテゴリ 3 に属する機能分散形のデータベースマシンは、大規模なデータベースを効率よく処理することを目標とする。そのため処理は機能ごとに分散化され、その各々で並列処理やパイプライン処理による効率改善がはかれる。このためのアーキテクチャとしては種々のものがあり、構造ループとデータループをもつ DBC^{29)~33)} や、クロスバースイッチによるマルチプロセッサ DIRECT⁴¹⁾ などがある。

連想メモリ形のデータベースマシン¹⁹⁾は、通常連想プロセッサと呼ばれているものである。ここでは、まず 2 次記憶装置から主記憶 (CAM) 装置へデータのステージングが行われる。データはバスを通して並列に転送される。処理はすべて CAM 上で行われるため、きわめて高速である。ただし、非常に高価なシステムとなること、ならびに CAM と 2 次記憶とのデータの交換がひんぱんに発生するような問合せに対しては著しく処理効率が低下すること、などが問題である。

カテゴリ 5 として新たに設定した高速連想処理形に属するデータベースマシンはまだみあたらない。しかしながら、これによれば著しい処理効率の改善が可能である。この実現化は、これからの研究にかかっているといえよう。

4.2 データベースマシンの具体例

以下では、4.1 における分類に基づいて、いくつかの具体例を紹介する。

(1) RAP

RAP (Relational Associative Processor) は、1976 年にトロント大学で試作されたデータベース用連想プロセッサであり、関係モデルをサポートする²⁶⁾。これは論理セル形のマシンであり、固定ヘッド磁気ディスクの各トラックに配置されたセルと呼ばれるプロセッサによってデータベースを操作する。ホストコンピュータから与えられた命令に従って、各セルはそれぞれ独立に動作し、関係代数演算の実行ならびにデータベースの更新操作を行う。セルの構成図を 図-6 に示す。

隣りあうセル間では直接通信が可能である。読み込

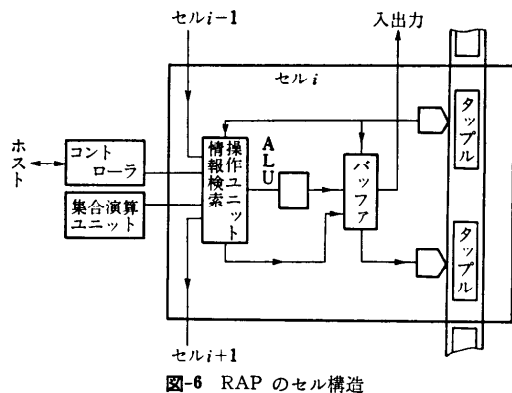


図-6 RAP のセル構造

みヘッドから読み込まれたデータは、1024 ビットのバッファ内でディスクの回転に遅れることなく処理され、データの書き込みも、書き込みヘッドから即時実行可能である。また、演算の途中結果は、各レコードの先頭につけられたマークビットと呼ばれるフィールドによって保持される。

RAP で取り扱うレコードは固定長であり、1 つのセルに複数のリレーションを格納することはできない。したがって、取り扱うことのできるタプルの大きさはバッファサイズによって制限される。また、マークビットをディスク上の各レコード内に保持するため、複数の問合せを同時に処理することもできない。さらに、複数リレーションが操作対象となる Join 演算では、その演算速度がディスクの回転数に比例して悪くなる。ただし、Selection や Restriction のような単純検索の場合には、セル数に比例した速度向上が得られる。

RAP の研究は、さらに RAP. 2²⁷⁾ や RAP. 3²⁸⁾ の開発を促した。まず、RAP から RAP. 2 への主な改良点は、(i) コントローラにミニ/マイクロコンピュータを導入したこと、(ii) 磁気ディスクを CCD に変更したこと、(iii) 命令セットを拡張したこと、である。さらに、RAP. 2 から RAP. 3 への変更点は、(i) メモリを CCD から磁気バブル(あるいは RAM) に変えたこと、(ii) セルのプロセッサを複数個 (マイクロプロセッサ) にしたこと、などである。

(2) DBC

DBC (Data Base Computer) は、オハイオ州立大学で開発が進められているデータベースマシンである^{29)~33)}。その設計にあたっては、(i) 大規模データベース (10¹⁰ バイト程度) を効率よく扱うこと、(ii) 複数のデータモデル、すなわち関係モデル、網モデル、

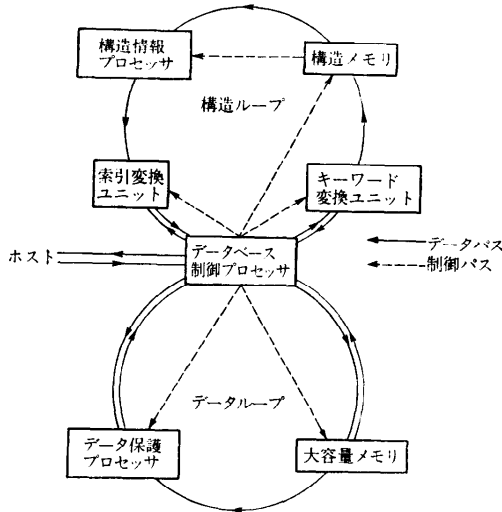


図-7 DBC の構成図

階層モデルなどをサポートすること、(iii) 現在あるいは近い将来実用化される技術のみを用いること、などが考慮された。そのため、データベース格納用2次記憶装置には可動ヘッドディスクが用いられた。また、機密保護機能をハードウェアレベルで実現している点も大きな特徴として挙げられる。

図-7 に DBC の構成を示す。データベース本体はデータループ内の大容量メモリ (Mass Memory) に格納される。可動ヘッドディスクでは、1 シリンダの各トラックごとにプロセッサが付加されており、シリンダ単位にその中の全トラックを並列に読みだすことができる。

データベースの検索にあたっては、ヘッドの移動を極力おさえるためにデータベースのクラスタリングに基づいて、その物理的構造の管理および操作を独立したプロセッサで行っている。これにより、必要最小回数のシリンダ (MAU: Minimum Access Unit) アクセスで処理が行えるようになっており、この機能をつかさどるのが構造ループである。ループの中の構造メモリ (Structure Memory) は論理セル形の連想モジュールであり、キーワードとそれに対応するインデックスの集合、すなわち転置ファイル群を格納している。さらに、構造ループでは問合せに対するセキュリティ情報が生成され、これによって不法なアクセスを禁止する機能が与えられている。

このように、DBC では、データベース処理をディレクトリ処理とデータベース本体の操作の2ステージ

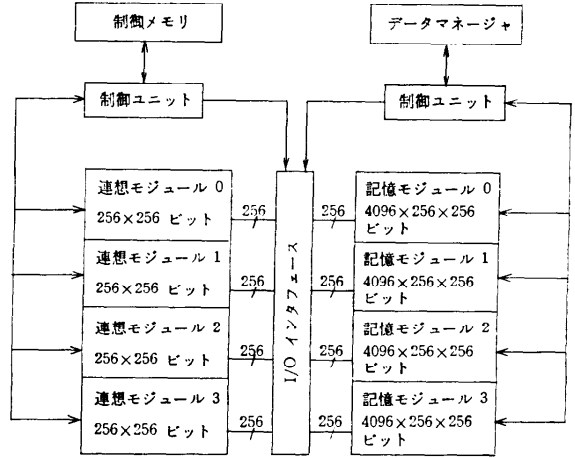


図-8 STARAN を用いたデータベースマシンの構成

に分け、機能的に分散させている。さらに、その各々をパイプライン処理することによって、大量のデータを効率よくあつかえるようにしている。ただし、あらかじめ施されているクラスタリングが検索条件にマッチしない場合には、ヘッドの移動回数が著しく増加するという欠点がある。

(3) STARAN

STARAN は非数値処理用に開発された大規模な連想プロセッサであり、実用化された数少ないシステムである³⁴⁾⁻³⁶⁾。当初は航空管制、レーダ解析、天気予報などの応用分野に使用されたが、最近では関係データベースへの応用も行われている¹⁹⁾。

STARAN を用いたデータベースマシンの構成を図-8 に示す。基本的には4つの連想モジュール (Array Module) と記憶モジュール (Storage Module) および I/O インタフェースから成る。連想モジュールは、256×256ビット構成の連想メモリである。データは2次記憶 (記憶モジュール: 4096×256×256ビット) から I/O インタフェースを介して1024ビット並列に転送され、その後、内容アドレッシングによって検索が実行される。連想モジュール上にロードされているデータの交換が起こらなければ、10⁸ バイトの全データに対して2秒以内で処理が完了する。このように、ロードされているデータのみで処理が完了する場合には、複雑な問合せも高速に実行可能である。ところが、データ交換が頻繁に発生するような問合せに対しては処理効率は劣化する。

さらに、大容量のデータベースに応用するための記憶装置の階層化も提案されている¹⁹⁾。

以上、主だったいくつかのデータベースマシンの構成ならびに特徴を紹介したが、紙面の都合から断念したものも多い。これらについては、文献³⁷⁾⁻⁵⁴⁾を参照されたい。

4.3 VLSI 化とそのアルゴリズムへの影響

データベースの処理能力の改善をめざして、トップダウン的なアプローチとボトムアップ的なアプローチの2通りがある。前者は先に述べたデータベースマシンのアーキテクチャに立脚した研究であり、後者はデータベースの基本演算レベルからの提案である。将来の技術を予想する場合に、後者のアプローチも見逃すわけにはいかない。本節では、その研究動向について述べる。

データベースの基本演算レベルからのアプローチは、サーチとソート、あるいは関係代数演算を高速に処理することを主眼とする。これは、VLSI 技術を背景としてアルゴリズムをハードウェア化するものであり、データベース専用高速演算チップの開発研究と言いかえることもできる。

これらの動作原理は、基本的に次のようである。まず、1次元あるいは2次元に配置されたセルに、データ流を多重に送り込む。セルは演算を遂行するための比較的単純な操作を行うもので、隣接セル間の通信が可能である。セル間で互いに通信を行いながら、パイプライン的に処理を実行する。セルの構造は規則的であり、VLSI 技術によるチップレベルの実現が容易である。また、 n 個のデータに対して、基本的に $O(n)$ の速度で処理が完了する。

このような構造をとるもので、最も注目を集めているものに脈動アレイ (systolic array) の研究がある^{56), 57)}。すでに、ソート、文字列比較、Join, Cartesian product, Division などの関係代数演算を実行する脈動アレイが提案されており⁵⁸⁾、これらを用いたデータベースマシンも実現性の色を濃くしている。

別のアプローチとしては、ヒープ木を用いた VLSI モジュールの提案もある⁴⁹⁾。これはソートエンジン、サーチエンジンなどと呼ばれ、データベースマシンの構成要素となる。さらに、最近ではデータ更新の処理効率改善をめざした B-tree マシンの提案も行われている⁵⁹⁾。

5. おわりに

情報処理に対するニーズが多岐にわたり、かつ高度化するにつれて、記号処理の担うべき役割は加速度的

に成長しつつある。そしてデータベース応用を想定した連想処理を例にとってみても、もっと多次元的な視野に立った高度な技術開発が要求されている。

1980年代はそのための技術的な用意が整う時期にある。VLSI 連想チップの実現が期待される。連想機能のためのアルゴリズムとデータ構造の見なおしも必要であろう。また連想処理の高機能化についても、まだ基礎的な段階ではあるが、すでに工学的な実現性が示されている^{60), 61)}。

これらがしばらくは並列的に進展し、やがて合流時期を見出したときに知識ベースの実現に貢献する。他方、その過程で、より現実的なコンピュータ技術への還元としてハードウェアとソフトウェアの役割分担の適正化研究を促進する。

参考文献

- 1) Burks, A. W., Goldstine, H. H. and von Neumann, J.: Preliminary Discussion of the Logical Design of an Electronic Computing Instrument, Part I, Datamation, Vol. 8, No. 9, pp. 24-31 (1962); Part II, Datamation Vol. 8, No. 10, pp. 36-41 (1962).
- 2) Slade, A. and McMahon, H. O.: A Cryotron Catalog Memory System, Proc. Eastern Joint Comput. Conf., pp. 115-120 (1957).
- 3) Lee, C. Y.: Interconnecting Cells, Basis for a Distributed Logic Computer, Proc. AFIPS 1962 Fall Conf., pp. 130-136 (1962).
- 4) Canaday, R. H., Harrison, R. D., Ivie, E. L., Ryder, J. L. and Wehr, L. A.: A Back-end Computer for Data Base Management, Comm. ACM, Vol. 17, No. 10 pp. 575-582 (1974).
- 5) Clark, D. W. and Green, C. C.: An Empirical Study of List Structure in Lisp, Comm. ACM, Vol. 20, No. 2, pp. 78-87 (1977).
- 6) Clark, D. W.: Measurement of Dynamic List Structure Use in Lisp, IEEE Trans. Softw. Eng., SE-5 (1979).
- 7) 山口, 島田: 仮想計算機による LISP プログラムの動的特性, 信学論, Vol. 61-D, No. 8, pp. 517-524 (1978).
- 8) Goto, E., Ida, T., Hiraki, K., Suzuki, M., and Inada, N.: Flats, A Machine for Numerical, Symbolic and Associative Computing, Proc. 6th Annual Symposium on Computer Architecture, pp. 102-110 (1980).
- 9) Ida, T.: Hashing Hardware and Its Application to Symbol Manipulation, Proc. International Workshop on High Level Language Computer Architecture, pp. 99-107 (1980).
- 10) Myers, G. J.: Advances in Computer Architec

- ture, John Wiley & Sons (1978).
- 11) Yau, S. S. and Fung, H. S.: Associative Processor Architecture A Survey, *Comput. Surv.*, Vol. 9, No. 1, pp. 3-27 (1977).
 - 12) Ramamoorthy, C. V., Turner, J. L. and Wah, B. W.: A Design of a Fast Cellular Associative Memory for Ordered Retrieval, *IEEE Trans. Comput.*, Vol. C-27, No. 9, pp. 800-815 (1978).
 - 13) 市川, 上林: 連想プロセッサ I, *信学誌*, Vol. 64, No. 6, pp. 609-614 (1981).
 - 14) 上林, 市川: 連想プロセッサ II, *信学誌*, Vol. 64, No. 7, pp. 736-742 (1981).
 - 15) Su, S. Y. W., Chang, H., Copeland, G., Fisher, P., Lowenthal, E. and Schuster, S.: Database Machines and Some Issues on DBMS Standards, *Proc. AFIPS Conf.*, Vol. 49, pp. 191-208 (1980).
 - 16) DeWitt, D. J. and Hawthorn, P. B.: A Performance Evaluation of Database Machine Architecture, *Proc. 7th VLDB*, pp. 199-213 (1981).
 - 17) Su, S. Y. W.: Cellular-Logic Devices: Concepts and Applications, *Computer*, Vol. 12, No. 3, pp. 11-25 (1979).
 - 18) Smith, D. C. P. and Smith, J. M.: Relational Data Base Machines, *Computer*, Vol. 12, No. 3, pp. 28-38 (1979).
 - 19) Berra, P. B. and Oliver, E.: The Role of Associative Array Processors in Data Base Machine Architecture, *Computer*, Vol. 12, No. 3, pp. 53-60 (1979).
 - 20) Kerr, D. S.: Data Base Machines with Large Content-Addressable Blocks and Structural Information Processors, *Computer*, Vol. 12, No. 3, pp. 64-79 (1979).
 - 21) 前川: ファイルプロセッサとデータベースマシン, *情報処理*, Vol. 18, No. 4, pp. 357-367 (1977).
 - 22) 植村: データベースマシンアーキテクチャ, *信学誌*, Vol. 62, No. 11, pp. 1289-1296 (1979).
 - 23) Marill, T. and Stern, D.: The Datacomputer-A Network Data Utility, *Proc. AFIPS Conf.*, Vol. 44, pp. 389-395 (1975).
 - 24) Hakoziaki, K., Makino, T., Mizuma, M., Umemura, M. and Hiyoshi, S.: A Conceptual Design of a Generalized Database Subsystems, *Proc. 3rd VLDB*, pp. 246-253 (1977).
 - 25) Slotnick, D. L.: Logic Per Track Device, *Advances in Computers*, Academic Press, pp. 291-296 (1970).
 - 26) Ozkarahan, E. A., Schuster, S. A. and Smith, K. C.: RAP-An Associative Processor for Data Base Management, *Proc. AFIPS Conf.*, Vol. 44, pp. 379-387 (1975).
 - 27) Schuster, S. A., Nguyen, H. B. and Ozkarahan, E. A.: RAP. 2-An Associative Processor for Databases and Its Applications, *IEEE Trans. Comput.*, Vol. C-28, No. 6, pp. 446-458 (1979).
 - 28) Oflazer, K. and Ozkarahan, E. A.: RAP. 3-A Multi-Microprocessor Cell Architecture for the RAP Data-Base Machine, *Proc. International Workshop on High-Level Language Computer Architecture*, pp. 108-119 (1980).
 - 29) Baum, R. I. and Hsiao, D. K.: Database Computers-A Step Towards Data Utilities, *IEEE Trans. Comput.*, Vol. C-25, No. 12, pp. 1254-1259 (1976).
 - 30) Hsiao, D. K., Kannan, K. and Kerr, D. S.: Structure Memory Designs for a Database Computer, *Proc. ACM 1977 Conf.*, pp. 343-350 (1977).
 - 31) Kannan, K.: The Design of a Mass Memory for a Database Computer, *Proc. 5th Annual Symposium on Computer Architecture*, pp. 44-51 (1978).
 - 32) Banerjee, J. and Hsiao, D. K.: Concepts and Capabilities of a Database Computer, *ACM Trans. on Database Systems*, Vol. 3, No. 4, pp. 347-384 (1978).
 - 33) 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).
 - 34) Rudolph, J. A.: A Production Implementation of an Associative Array Processor: STARAN, *Proc. AFIPS Conf.*, Vol. 41, pp. 229-241 (1972).
 - 35) Batcher, K. E.: STARAN Parallel Processor System Hardware, *Proc. AFIPS Conf.*, Vol. 43, pp. 405-410 (1974).
 - 36) Davis, E. W.: STARAN Parallel Processor System Software, *Proc. AFIPS Conf.*, Vol. 43, pp. 17-22 (1974).
 - 37) Babb, B.: Implementing a Relational Database by means of Specialized Hardware, *ACM Trans. on Database Systems*, Vol. 4, No. 1, pp. 1-29 (1979).
 - 38) Maller, V. A. J.: Information Retrieval Using the Content Addressable File Store, *Proc. IFIP Congress 80*, pp. 187-192 (1980).
 - 39) Su, S. Y. W., Nguyen, L. H., Emam, A. and Lipovski, G. J.: The Architectural Features and Implementation Techniques of the Multicell CASSM, *IEEE Trans. Comput.*, Vol. C-28, No. 6, pp. 430-445 (1979).
 - 40) Lin, C. S., Smith, D. C. P. and Smith, J. M.: The Design of a Rotating Associative Memory for Relational Database Applications, *ACM*

- Trans. on Database Systems, Vol. 1, No. 1, pp. 53-65 (1976).
- 41) 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).
 - 42) Wah, B. W. and Yao, S. B.: DIALOG-A Distributed Processor Organization for Database Machine, Proc. AFIPS Conf., Vol. 49, pp. 243-253 (1980).
 - 43) Umemura, S., Yuba, T., Kokubu, A., Oomoto, R. and Sugawara, Y.: The Design and Implementation of a Magnetic Bubble Database Machine, Proc. IFIP Congress 80, pp. 433-438 (1980).
 - 44) Kamibayashi, N., Kato, H., Kiyoki, Y., Ozawa, H., Seo, K. and Aiso, H.: SPIRIT: A New Relational Database Computer Employing Functional Distributed Multi-Microprocessor Configuration, Proc. 1st International Conf. on Distributed Computing Systems, pp. 757-771 (1979).
 - 45) 上林, 瀬尾: 記憶階層に関係代数演算の前処理機構を組み込んだ関係データベースマシンの基本アーキテクチャ, 信学技報 EC 81-47, (1981).
 - 46) 藤原, 中村: 関係代数演算に適した知能ディスク, 信学全大, S1-1, (1980).
 - 47) 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 81-35, (1981).
 - 48) Ahuja, S. R. and Roberts, C. S.: An Associative/Parallel Processor for Partial Match Retrieval Using Superimposed Codes, Proc. 7th Annual Symposium on Computer Architecture, pp. 218-227 (1980).
 - 49) Tanaka, Y., Nozawa, Y. and Masuyama, A.: Pipeline Searching and Sorting Modules as Components of a Dataflow Database Computer, Proc. IFIP Congress 80, pp. 427-432 (1980).
 - 50) Boral, H. and DeWitt D. J.: Design Consideration for Data-Flow Database Machines, Proc. International Conference on Management of Data, pp. 94-104 (1980).
 - 51) Edelberg, M. and Schissler, L. R.: Intelligent Memory, Proc. AFIPS Conf., Vol. 45, pp. 393-400 (1976).
 - 52) Lee, S. Y. and Chang, H.: Associative-Search Bubble Devices for Content-Addressable Memory and Array Logic, IEEE Trans. Comput., Vol. C-28, No. 9, pp. 627-636 (1979).
 - 53) Liu, J. W. S. and Jino, M.: Intelligent Magnetic Bubble Memories and Their Applications in Data Base Management Systems, IEEE Trans. Comput., Vol. C-28, No. 12, pp. 888-906 (1979).
 - 54) Doty, K. L., Greenblatt, J. D. and Su, Y. W.: Magnetic Bubble Memory Architecture for Supporting Associative Searching of Relational Databases, IEEE Trans. Comput., Vol. C-29, No. 11, pp. 957-970 (1980).
 - 55) Stellhorn, W. H.: An Inverted File Processor for Information Retrieval, IEEE Trans. Comput., Vol. C-26, No. 12, pp. 1258-1267 (1977).
 - 56) Foster, M. J. and Kung, H. T.: The Design of Special Purpose VLSI Chips, Computer, Vol. 13, No. 1, pp. 26-40 (1980).
 - 57) Kung, H. T.: Why Systolic Architecture?, Computer, Vol. 15, No. 1, pp. 37-46 (1982).
 - 58) Kung, H. T. and Lehman, P. L.: Systolic (VLSI) Arrays for Relational Database Operations, Proc. International Conference on Management of Data, pp. 105-116 (1980).
 - 59) 幅田, 田中: B-Tree マシン, 信学技報 AL 81-64 (1981).
 - 60) 市川, 坂村, 諸隈, 相磯: 連想プロセッサ ARES, 信学論 (D), Vol. J 61-D, No. 10, pp. 743-750 (1978).
 - 61) 平川, 上林, 市川: あいまいな問合せを受理する情報検索システム, 信学論 (D), Vol. J 65-D, No. 1, pp. 64-71 (1982).

(昭和 57 年 4 月 26 日受付)