

連想プロセッサ ARES
ARES-An Associative Processor

市川忠男⁺　坂村　健⁺⁺　諸隈立志⁺⁺　山石裕之⁺⁺　相磯秀夫⁺⁺
Tadao ICHIKAWA Ken SAKAMURA Tatsushi MOROKUMA Hiroyuki YAMAISHI Hideo AISOU

⁺国際電々(株)研究所 ⁺⁺慶應大学工学部
KDD Res. & Dev. Labs. Keio University

人工知能に関連した諸問題におけるセマンティクデータベースの高速サーチを目的とした連想プロセッサの原理と構成を述べる。連想の原理は、従来の誤り訂正と内容アドレシングの技法を結びつけたもので、これによって論理機械にある種の直感性に似た効果が与えられ、処理の高速化に寄与する。アーキテクチャに関しては処理の並列性に重点を置いているが、同時にコストパフォーマンスの観点から並列性の度合いに関する評価も行っている。その結果、データベースの規模が大きくなる場合にも十分対処し得る好ましいアーキテクチャとなっていることが結論される。

1. まえがき

近年の計算機アーキテクチャ技術の急速な発展はパターン認識の領域にも大きな影響を与えている。⁽¹⁾ たとえば、処理能力が増大するにつれて自然な連續発声音を認識したり、診断を助けるための医用画像データの処理や資源開発のためのリモートセンシングデータの解析など、龐大なデータの処理が要求される問題もこなせるようになってきた。これらの動きを、従来のように強い制約の中での対象物の認識率そのものを議論する立場から離れ、より高度な社会の要求に応えるための機械と人間との調和点を見出す試みとしてとらえ、それらのシステムをパターン理解システムと総称する。

さて、実際的な応用面に関して理解システムを構成しようとすると、今度は逆に問題指向型の高性能プロセッサが必要となる。たとえば、音声理解システムの構成を目的としたカーネギーメロン大学の C.mmp⁽²⁾ や Cm*⁽³⁾ その傾向がうかがえる。高度な並列性的導入によって処理能力が高められている。ところで処理内容に目を向けると、その基本的な問題は、共通概念としての裏づけを持ったセマンティクデータベー

スの高速サーチに帰着する。この点ではイメージを対象に選んだ場合も同様である。

筆者らは、このような観点から、セマンティクデータベースの高速サーチのためのマルチマイクロプロセッサの構成に関する検討を進めている。⁽⁴⁾ このシステムが最も特徴とする点は連想方式にある。この連想方式は、データ間の関連の度合いが符号間距離で表わされるように符号化された原データに、誤り訂正処理を行って個々のデータ固有の性質から離れた基本的な構造を求める、基本的な構造の上で一致度の検定を行って関連の強いデータを指定するという考えに基づいている。原データに誤り訂正を適用する操作をブルアーリング (blurring) と呼び、そうして得られたデータ空間をブルアードデータ空間 (blurred data space) と呼ぶ。このようにして、関連度の評価を直接原空間で行う場合に要求される煩雑な計算処理はブルアードデータ空間での一致検定という単純な操作に置き換える。すなわち、誤り訂正と内容アドレシングの技法を結びつけることによって論理機械にある種の直感性に似た効果を付与した点に特徴があるといえる。以後この連想プロセッサ

を "A processor, capable of Associating stored information through Relevancy Estimation" の意で **ARES** と略称する。

本文では、先ず連想方式の概略を述べ、次に **ARES** の並列処理性に重点を置いたアーキテクチャを示す。ここでは、連想出力の数を制御する適応機構と、データの深層構造を求める目的で行われるブルアーリングの階層化の実現機構に特徴がある。最後にコストパフォーマンスの観点から並列性の度合いに関する考察を付す。シミュレーションの結果から、ここで提案するアーキテクチャがデータベースの規模が大きくなる場合にも十分対処し得るものであることが結論される。

2. **ARES** における連想方式

セマンティクデータベースは、たとえばイメージを対象とする場合には、人間がそれを観察する場合に意識的、あるいは無意識に用いる類別の目安に照らしてカテゴリカルに記号化された特徴データからなり、共通概念の上にたって特徴が十分に説明できるような内容が盛られていることが好ましい。

以下では、個々のデータワード D をカテゴリ i に対応したブロック d_{si} ($1 \leq i \leq p$) を単位に式(1)のように表わすこととし、各ブロックに関する部分的記号表現の間の関連性は符号間距離 t に応づけられると仮定する。ただし、それぞれのブロック長は必ずしも等長である必要はなく、そのカテゴリに属するクラスの数とクラス間の関係表現のために選ばれる割当て符号に依存する。

$$D = (d_{s1}, d_{s2}, \dots, d_{si}, \dots, d_{sp}), 1 \leq i \leq p. \quad (1)$$

2.1 連想の原理

データワード D を式(2)に示すように長さ n のブロック d_{bj} ($1 \leq j \leq q$) に分解し、個々のブロック d_{bj} に対して長さ n 、訂正可能な誤り距離 t の誤り訂正符号を適用する。

$$D = (d_{b1}, d_{b2}, \dots, d_{bj}, \dots, d_{bq}), 1 \leq j \leq q. \quad (2)$$

このようにして導びかれた符号ベクトルを式(3)のように $e(D)$ と表わす。

$$e(D) = (e(d_{b1}), e(d_{b2}), \dots, e(d_{bj}), \dots, e(d_{bq})), 1 \leq j \leq q. \quad (3)$$

誤り訂正を経た個々の符号ベクトル $e(d_{bj})$ は d_{bj} のブルアーリングされた特徴を表わす。ブロック d_{bj} をブルアーリングブロック (blurring block) と呼び、 $e(D)$ をブルアードデータ (blurred data) と呼ぶ。 n は通常カテゴリの人為的な設定の影響をやわらげる目的でいくつかのカテゴリにまたがった記号列をカバーするように選ばれる。このようにして、 $e(D)$ は t 、あるいはそれ以下の距離で互いに関連づけられるグローバルな構造を反映する。

入力データワード Q が与えられると、それに対応した $e(Q)$ によってブルアードデータ $e(D)$ の内容アドレンジングを行う。このとき、一致度の評価は一致したブルアーリングブロックの数で行うこととし、"しきい値" θ に等しいか θ を越えた $e(D)$ が指定される。このようにして、

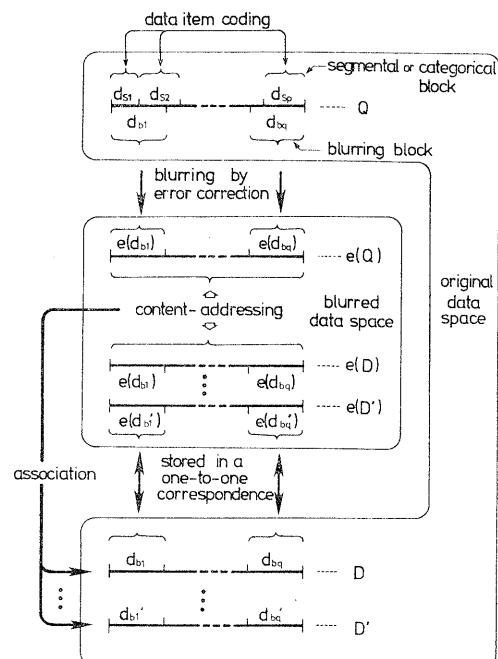


図 1 連想のプロセス

いま D と $e(D)$ が一対になって記憶されているとすると、 Q に強く関連づけられた D が煩雑な計算処理を行わずに選ばれることが分る。

ブルアードデータ $e(D)$ が格納されるデータ空間をブルアードデータ空間と呼び、ブルアードデータ空間における $e(Q)$ による内容アドレシングによって D を選定する操作を“連想”と呼ぶ。連想の過程を図 1 に示す。連想の効果は誤り訂正符号の符号長 n と訂正可能な誤り距離 t によって組織的に制御される。

次に、連想に先だってデータのよりグローバルな構造を知るために、誤り訂正によるブルアーリングの一般化を示す。

2.2 ブルアーリングの階層化

符号長 n 、情報デジット数 k 、訂正可能な誤り距離 t の符号を (n, k, t) と表わし、 (n_1, k_1, t_1) 符号によって D から求められたブルアードデータを $D_1 = e_{n_1}(D_0)$ と表わす。ただし、 $D_0 = D$ とする。さらに、データのより一般的な構造を求めたいときには、 D_1 を長さ n_2 のブルアーリングブロックに分割し、それぞれのブロックに (n_2, k_2, t_2) 符号を適用する。このようにして、内容アドレシングを働かせる対象として十分な深層構造が得られるまでブルアーリングを進める。これを式(4)に示す。ただし、 $n_s \geq n_{s-1}$ のときには $t_s \geq 2t_{s-1} + 1$ が満足されている必要がある。

$$D_0 \rightarrow D_1 = e_{n_1}(D_0) \rightarrow D_2 = e_{n_2}(D_1) \rightarrow \dots \\ \rightarrow D_s = e_{n_s}(D_{s-1}) \rightarrow \dots \quad (4)$$

原データ空間とレベル s のブルアードデータ空間をそれぞれ B_0, B_s と表わす。 B_0 のデータは直接観測されるデータ固有の物理的特徴に強く依存しているが、 B_s は多くのデータに共通な、より一般的な特徴を表わしており、誤り訂正の階層的適用が進むにつれて共通概念のレベルへと登って行く。

上記ブルアーリングには完全符号が選ばれることが好ましい。図 2 は $n_s = n_{s-1}$ ($s = 1, 2, \dots$) の場合を例に、单一ブロックについての階層的ブルアーリングの様子を示している。 B_{s-1} における半径 t_s の円の中に位置づけられたブロックは、互いに強い関連性を示すものとして、 B_s

の中では単一のブロックに統合される。

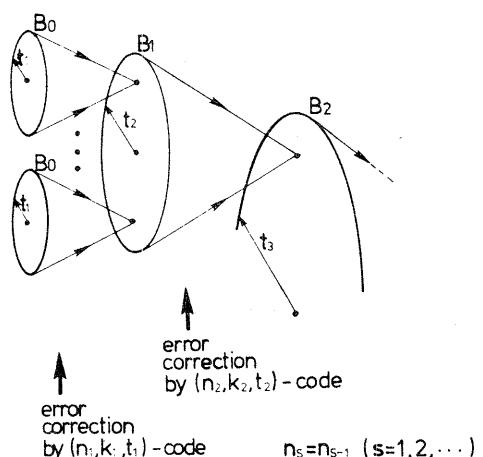


図 2 ブルアーリングのプロセス

2.3 データの深層構造に基づく連想

連想のための内容アドレシングは、データの特徴が十分要求に見合ったレベルに一般化されていれば、ブルアーリングの任意のレベルで適用可能である。 s^* を内容アドレシングが適用されるブルアーリングレベルとすると、 B_s ($0 \leq s < s^*$) の中の、指定された D_{s^*} に対応する D_s は、 D_s ($0 \leq s < s^*$) と D_{s^*} が一一の一の関係をもって記憶されていることから、ブルアーリングの過程を逆にたどらないでも求められる。

図 3 は、内容アドレシングと階層的ブルアーリングとがどのように結合されて連想を実行するかを概念的に示している。入力データワード Q_0 は B_0 の内部から選ばれることもあるし、また外部から与えられる場合もある。ヒットマークは B_{s^*} で内容アドレスされた D_{s^*} を指定するためのもので、連想出力ワードの数に制限が与えられているときにはヒットマークの数が制限 δ を越えないで、かつ最大数のヒットマークが得られるように θ が適応的に制御される。

さらに、A R E S は記憶データを修正する機能を持っている。 B_0 の中のある D_0 がレベル s^* での内容アドレシングで Q_0 によって指定された

とすると、 D_0 とそれに関連した D_s ($0 < s \leq s^*$) はそれぞれ Q_0 と Q_s ($0 < s \leq s^*$) で置き換える。 Q_0 で指定される D_0 が見つからない場合には Q_s ($0 \leq s \leq s^*$) はそのまま新たに B_s ($0 \leq s \leq s^*$) に登録される。なお、 Q_0 が既に B_0 に記憶されている 2 つのデータの重みつき平均として与えられることがある。

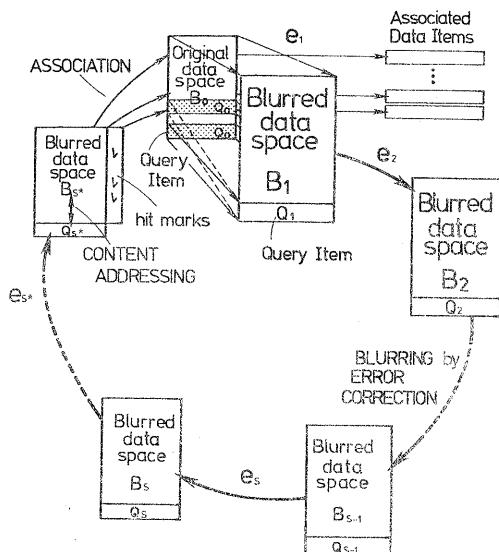


図 3 ブルアーリングとコンテントアドレッシング

3. マルチマイクロプロセッサ ARE S

前節で述べたセマンティクデータベースの連想サーチを効果的に行うために、アーキテクチャのレベルで次のような機能を用意する。

- (1) 階層的なブルアーリング機能
- (2) 内容アドレッシング機能
- (3) 連想出力ワード数の適応的制御機能

階層的なブルアーリングを行うために、多層化されたブルアードデータ空間に対する効果的なアクセスを可能とする制御回路が必要である。また、ブルアーリングのための誤り訂正符号としては、2元巡回符号が符号長選択の自由度と、誤り訂正機構の実現の容易さから多くの応用面に対して有効である。

内容アドレッシングは、図 3 に示したように、最後のブルアーリング e_{s^*} を終えたのちにブルアードデータ空間 B_{s^*} でブルアーリングブロックを

単位に行われる。マスクされていないブロックについて、 Q_{s^*} のブロックと一致したブロックの数を数え上げ、これを“しきい値” θ と比較する。 D_{s^*} の一致数が θ に等しいか、あるいは θ を越えていれば、 D_{s^*} に対応したヒットマークを 1 に設定する。ヒットマークの数が連想出力ワードの数に対する制限 δ を越えた場合には θ を 1 増やし、 θ に満たない場合には θ を 1 減らして再度内容アドレッシングを行い、 δ 以内となるべく多くのヒットマークが得られる様に θ を制御する。このようにしてヒューリスティックに選定された θ によって連想出力ワードが得られる。

上記の内容アドレッシング機能は、手書き文字認識を対象にすでに提案された单一プロセッサ ARE S⁽⁴⁾ によって実行されるものであるが、階層的なブルアーリングによってブルアード空間を効果的に形成するにはアーキテクチャの上で高い柔軟性が要求される。したがって、上記の高度の連想機能を実現するにはマルチプロセッサ構成が最も適しているといえよう。

3.1 ARE S のアーキテクチャ

図 4 にマルチマイクロプロセッサ ARE S の構成図を示す。分散処理を行なうため、マスターコントロールにすべての論理ユニットの同期をとらせる。論理ユニットは、ARE S の適用対象によって適宜選定されるセマンティクデータベースのデータ構造に適応できるよう、機能的に柔軟なマイクロプロセッサで構成される。以後、論理ユニットをセルと呼ぶ。

クラスタコントロールは各々 8 個のセルを制御し、4 個のクラスタコントロールをマスターコントロールが統轄している。したがって、最大 4 つの異なるプログラムがクラスタ別に並列に実行される。一方、すべてのクラスタが同じプログラムにゆだねられるときには 32 の並列演算がすべてのセルにわたって行われる。

マルチブルレスポンスリゾルバ (MRR) は複数入力の並列カウンタである。各 MRR-1 は接続されたセルからのヒットマークを受け、これを集計する。これがさらに MRR-2 で集計され、その結果と δ との比較にもとづいてマスターコントロールが θ の適応制御を行う。MRR の回路構成を図 5 に示す。図中の flag bit は以上

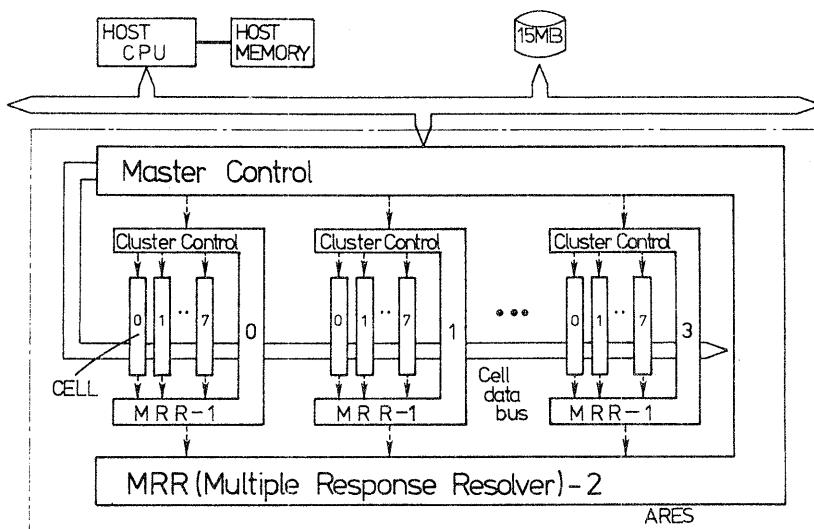


図4 ARESの構成図

の説明におけるヒットマークに相当する。Access bit に “1” がたったとき、その出力番号に対応するセルから連想出力ワードを取り出す。

図6は連想機構の核となるセルの内部構造を示している。原データとそのブルアーリングされたデータ表現は、ともに通常の内容アドレスメモリに見られる1つのセルに収容される。図中の block size と word size は、それぞれブル

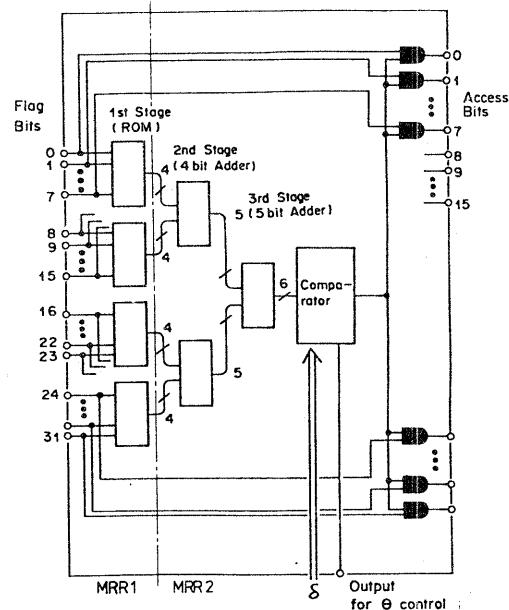


図5 マルチプルレスポンスリゾルバ

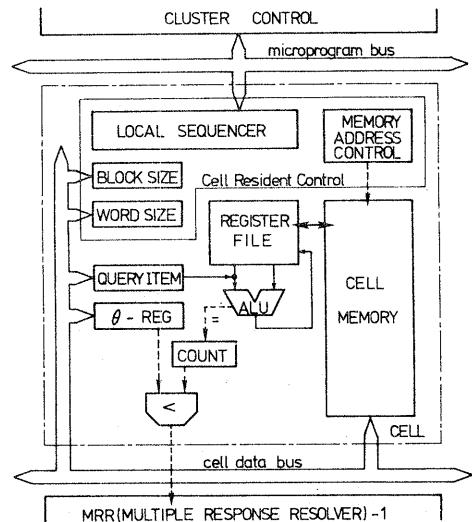


図6 セルの内部構造

アーリングの最終段でのブロック長とデータワードの長さを示している。

セルのクラスタリングによって高い信頼性と機能的柔軟性が得られた。すなわち、1つのセルやMRR-1の機能が欠けた場合でも、処理能力は若干減少するがシステムが停止してしまうことはない。また、各クラスタは独立に機能する事ができる。

3.2 ARESの機能的振舞い

セマンティクデータベースを扱うARESの機能的振舞いを以下に説明する。ホストコンピュータとしてはHP-21MXが用いられる。ホストコンピュータがマスタコントロールへグローバルなコマンドを送り、オペレーションが初期化される。マスタコントロールはこのコマンドを解析し、それに関連したサブコマンドをクラスタコントロールに与える。サブコマンドは実行されるマイクロプログラムの先頭アドレスを指す機能を持っている。マイクロプログラムはあらかじめクラスタコントロールに格納されており、クラスタコントロールは各々指定されたオペレーションを独立に実行する。

図7はブルアーリングモードにおけるブルアードデータ空間がセルの中でどのように構成され

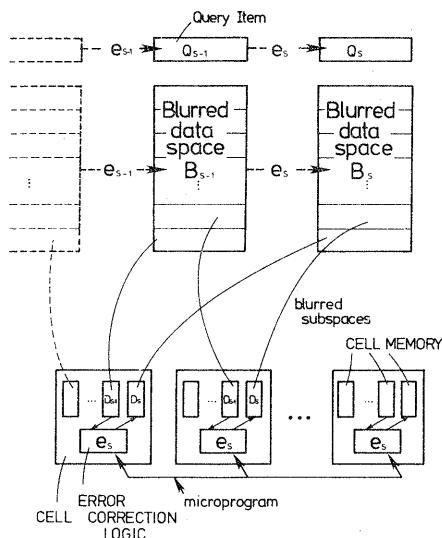


図7 ブルアーリングモードにおけるARES

るかを示している。それぞれのブルアード空間はサブスペースに分解され、空間中に含まれている各データは並列処理が可能となるようにセルメモリに分散される。ブルアーリングのための誤り訂正是図6のALUで実行され、クラスタコントロール中のマイクロプログラムで制御される。

図8はコンテンツアドレッシングモードにおけるブルアードデータ空間とセルとの関係を示している。入力データワードと δ の値はマスタコントロールによって各セルに並列に送り込まれ、セルメモリに分散されたブルアードデータ空間 B_s^* でコンテンツアドレッシングが実行される。 θ は δ のもとでMRRによってヒューリスティクに選定される。 θ のヒューリスティクな選定過程での内容アドレッシングの高速くり返しの必要性からMRRを独立のハードウェアとしたが、設計にあたっては、高速性と実装のコストとのトレードオフを考慮した。

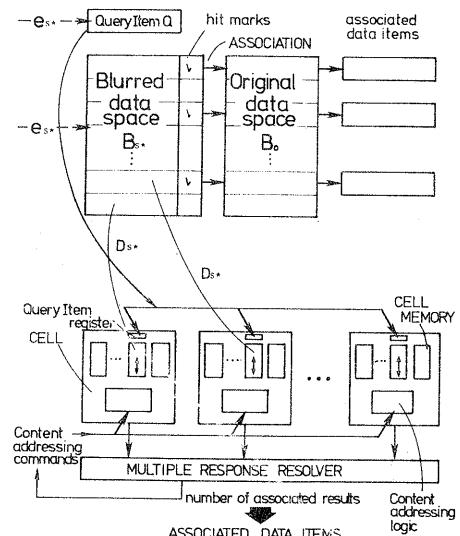


図8 コンテンツアドレッシングモードにおけるARES

3.5 最適セル数の評価

図4, 5に示されているように、これまでセル数を32個としてきた。これはここに述べるコストパフォーマンスを考慮した最適セル数の評価シミュレーションの結果によるものであ

る。シミュレーションは、簡単化のためブルアーリングは除外し、 θ のヒューリスティクな選定を含むブルードデータ空間 B_s^* での内容アドレッシングに限定する。したがって、以後 B_s^* を単にデータ空間 V と呼ぶことにする。

データ空間 V としては 10^3 , 10^5 , および 10^7 個のデータワードからなるものを想定する。各データワードの構成は 23 ビット × 9 ブロックと 7 ビット × 30 ブロックの 2 種類とする。1 ブロック中のビット数 23 と 7 は、それぞれ (23, 12, 3) ゴーレイ符号と (7, 4, 1) ハミング符号の符号長に対応している。

シミュレーションモデルを規定するパラメータとその設定値等を表 1 に示す。なお、データ空間ではビットパターンが一様に分布していると仮定し、内容アドレッシングで一致するブロックの数が θ を越えるワード数、および内容アドレッシングに要する時間をシミュレートするための乱数はそれぞれ 2 項分布、パスカル分布に従うものとする。

表 1. シミュレーションパラメータとその設定値

項目	設定値
データ空間: V	$10^3, 10^5, 10^7$ ワード
データ構造	23 ビット × 9 ブロック, 7 ビット × 30 ブロック
連想出カワード数の上限: θ	100
セルプロセッサの演算ビット数	16 ビット
セル数: N_c	1 ~ 128
グラスク数: N_{CL}	$V_c / 81^*$
セルメモリ容量: W_c	$LV / N_c \cdot 256^{**}$
メモリプロセッササイクル比: R	1.2 (Memory cycle / Processor cycle)
θの初期値: θ_0	1
θの適応化時間: T_{CL1}	$20 [pc]^{***}$
一致データワード処理時間: T_{CL2}	80 (pc)
セル価格: C_c	40,000 (yen/cell) (メモリを除く)
グラスクコントロール価格: C_{CL}	60,000 (yen/cell)
メモリ価格: C_M	16 (yen byte)

* $N_c/8$ を越えない最大の整数

** V/N_c が下さわらない 256 の最小の倍数

*** Processor cycle

データ空間 V はセルメモリに分散されるので、限定されたセル数を N_c とすると 1 セル当りのデータ空間 V_c は V/N_c となるが、セルメモリの容量を連続的に変化させるのは現実的ではないので 256 バイト単位で増設するとしてシミュレーションを行なった。また、セルの演算ビット数は 16 ビットとし、8 個のセルを 1 台のグラスクコントロールで管理することにする。

セルの処理単位が 16 ビットであるため、1 ブロックが 23 ビットのデータワードは 2 ユニット × 9 ブロック = 18 プロセッササイクルで処理される。また、ワード長が 207 ビットであるから、セルメモリには 26 バイトが必要となる。したがって、メモリサイクルとプロセッササイクルの比 R を 1.2 とすると、セルメモリの 1 つのデータワードを 1 回アドレッシングするのに要する時間 (サーチサイクルタイム) T_c は $18 + 13R = 33.6 [pc]$ となる。同様に、1 ブロック 7 ビットのデータワードでは $T_c = 4.68 [pc]$ となる。

図 9 は、 $V = 10^5$ のデータ空間に対して、グラスクコントロールでの θ の適応化に 20 [pc]、アドレッシングされた一致データワードの処理に 80 [pc] 要するとして、 θ の初期値を 1 としてセル数を 1 ~ 128 に変化させながらそれぞれ 100 回連想を行なった場合の平均実効時間を示している。図 10 は、これに式(5)で求められるコスト C を乗じたコスト時間積を表わしている。

$$C = N_c \times (C_c + C_M \times W_c) + N_{CL} \times C_{CL} \quad (5)$$

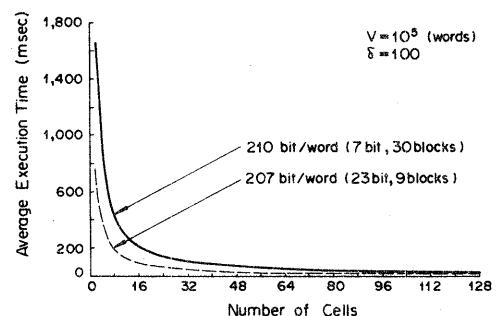


図 9 1 回の連想に要する平均実効時間

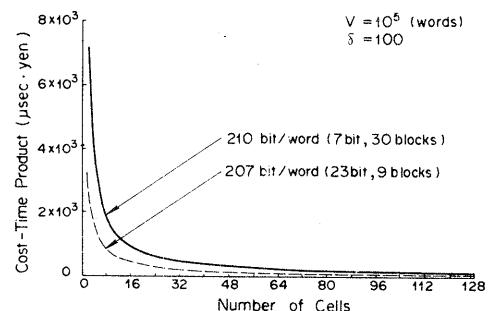


図 10 1 回の連想に要するコスト時間積

図10のコスト時間積を示す曲線の傾きがほとんど0に近くなるときのセル数が、コストパフォーマンスの観点から最適セル数であるといえる。最適セル数がデータ空間の大きさを変えたときどのように変化するかを、ブロック長23の場合を例に示したのが図11である。一点鎖線は $V = 10^3$ を基準に、最適セル数がデータ空間の大きさに比例するとした場合を示している。なお、ブロック長7の場合も、最適セル数、図の傾向ともにほど同じである。

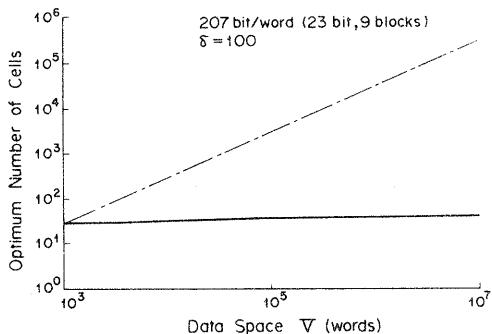


図11 データ空間の大きさに対する最適セル数

図11が示すように、データ空間の大きさが変わってもパフォーマンス／コストを最良とするセル数はほど一定で、20～40の間にある。したがってセル数を32とし、4台のクラスタコントロールでこれらを制御することとした。

コストパフォーマンスの観点から見て、 10^7 個のデータワード (2×10^3 ビット／ワード) からなる大きなデータ空間に対しても、セル数32個で十分対処しえることは、ARESのマルチプロセッサ構成が連想サーチに好ましいアーキテクチャとなっていることを説明している。

4. むすび

本稿では、パターン理解システムの構成において基本的な問題となるセマンティクデータベースの高速サーチを目的として、当初手書き文字の認識を対象として提案した連想方式⁽⁵⁾を発展させた。認識の対象を表現する個々のデータは、対象の識別の目安としてのカテゴリを単位に、物理的、ないしは概念上の特徴が符号間距離に対応するように記号化されたいくつかのブ

ロックから構成されているとする。これにより、いくつかのカテゴリーをカバーするような符号長の誤り訂正符号を用いて対象固有の性質から離れた一般的な特徴が得られ、この上で内容アドレッシングを行って関連の強いデータを選び出すというのが連想の原理である。ここで、内容アドレッシングに先立って誤り訂正によるブルアーリングを階層的に適用すれば、共通概念で裏づけられるような、対象のより深い構造を知ることができる。

上記の連想方式の一般化の上にたって、これを実現するARESのアーキテクチャを示し、その有効性を議論した。コンピュータアーキテクチャの観点から見ると、ミニコンピュータHP-21MXをホストとするマルチマイクロプロセッサ構成をとっている点に特徴がある。これにより、構造上の拡張性と、マイクロプログラムのサポートによって多様な応用面に適応する機能上の柔軟性とが得られた。さらに、連想出力ワードの数を適応的に制御する機能を有する点も特徴の一つであり、パターン認識応用に対して有効に働く。

以上、セマンティクデータベースの高速サーチのための連想プロセッサARESについて述べたが、筆者らはイメージを対象としたセマンティクデータベースの構成と、そのための解析を行う特殊計算システムについても検討を進めている。

文 献

- (1) T.Ichikawa et.al., "Architecture for pattern recognition applications," AFIPS PANELS Vol.1 (NCC'77), June 1977.
- (2) W.A.Wulf and C.G.Bell, "C.mmp - A multi-miniprocessor," Proc., FJCC '72, pp.715-777, Dec. 1972.
- (3) R.J.Swan, S.H.Fuller and D.P.Siewiorek, "Cm*-A modular, multi-microprocessor," AFIPS Conf. Proc., Vol.46 (NCC'77), pp.637-644, June '77.
- (4) T.Ichikawa, K.Sakamura and H.Aiso, "ARES - A memory, capable of associating stored information through relevancy estimation," AFIPS Conf. Proc., Vol.46 (NCC'77), pp.947-954, June 1977.
- (5) T.Ichikawa and J.Yoshida, "On-line recognition of handprinted characters with associative read-out of patterns in a memory," Proc., Second Int'l Joint Conf. on Pattern Recognition, pp.206-207, August 1974.
- (6) T.Ichikawa, K.Sakamura and H.Aiso, "A multi-microprocessor AREs with associative processing capability on semantic data bases," to be presented at NCC'78 (invited), June 1978.