

意味ネットワークマシン I X M 第 2 版の概要

樋口哲也 古谷立美 楠本博之 半田剣一 国分明男
電子技術総合研究所

連想メモリを備えた P E 6 4 台と、マーカ伝搬の並列化機能を持つ 9 台のネットワークプロセッサ (N P) から成る意味ネットワークマシン I X M のプロトタイプ第 2 版 (I X M 2) について述べる。 I X M 2 は、完全結合をベースとした再帰的な計算機間接続方式をとり、完全結合の転送能力と大規模システムへの拡張のしやすさを目的としている。意味ネットワークの基本処理のうち、集合演算と連想処理は大容量連想記憶 (最大 2 5 6 K W まで実装可能) により常に一定時間でできるため、大規模意味ネットワーク処理における計算量の急激な増加に対して有効な抑制手段となる。 I X M 2 はホストの S U N より制御され、 S U N の Quintua Prolog 上に実装した知識表現言語 I X L よりコールされる意味ネットワーク処理用の述語を全解探索で実行する。

The Development of Semantic Network Machine IXM2

Tetsuya Higuchi, Tatsumi Furuya, Hiroyuki Kusumoto
Kenichi Handa, Akio Kokubu
Electrotechnical Laboratory
1-1-4 Umezono, Tsukuba, Ibaraki 305, Japan
higuchi%etl.jp relay.cs.net

The second prototype of a semantic network machine IXM, called IXM2, is discussed. IXM2 is a multiprocessor system with large associative memories. It consists of 64 associative processors and 9 network processors. The connection structure of IXM2 is based on completely connected topology among 8 PEs in order to perform efficient data transfer. Associative memories are very effective in suppressing the explosion of the computational amount in large semantic network processing. IXM2 executes programs written in the IXL language, the superset of Prolog, so that IXM2 can be utilized as a back-end of SUN workstations.

1. はじめに

意味ネットワークは代表的な知識表現形式の一つとして知識ベース、自然言語処理、記憶モデルなど、これまでに広く用いられてきた。これは、グラフとノードによる図的表現のわかりやすさや、階層的知識の表現能力によるところが大きい。しかし、意味ネットワークの提案以来二十年近くを経ても、その利用形態はまだまだ実験的な領域を出ず、これまでに大規模な意味ネットワークにもとづくシステムが作られた例はない。

この理由として第一に挙げられるのは、ネットワークの大型化に伴って計算量が爆発的に増加する傾向があるためである。つまり、たとえ大きな意味ネットワークを構築して処理しようとしても多大な時間がかかり、とても実用にはならないことが多い。意味ネットワーク処理でよく行われるのはパターンマッチであるが、意味ネットワークが大きくなって来ると、探索すべき空間が爆発的に増大し、多少計算機単体の速度をあげたところでそれに追従できない。スーパーコンピュータを導入してもベクトル処理の効果は意味ネットワークではあまり期待できない。大規模意味ネットワークの処理を行うためには処理アルゴリズムのオーダを低減することがポイントであり、そのためには超並列計算機が必要となる。

一方、意味ネットワークの応用は多岐にわたっており、これまで提案された種々の意味ネットワークはそれぞれのニーズに特化したものとなっている。たとえばリンクの意味付けやノードの表わす概念の抽象度にしても、実に様々である。したがってそれら各種の応用を柔軟に記述できる知識表現言語を提案し、その言語で記述したプログラムを上述の並列探索が行える意味ネットワークマシンで実行できる形態が望ましい。

このような実用規模での意味ネットワーク処理を実現するために、筆者らは意味記憶システムIX(イクス)を開発している。IXは、(1)意味ネットワーク処理向き高並列マシンIXM(イクスム)と、(2)意味ネットワークに基づく知識表現言語IXL(イクスル)から成り、実用規模の意味ネットワーク処理を統一的にサポートすることを目的としている。

IXMは、大容量連想メモリを備えたマルチプロセッサシステムであり、大容量連想メモリの持つ高い並列性を意味ネットワーク処理に活かすことを狙いとしている。

現在稼働中のプロトタイプは33台のPEと大容量連想記憶(128KWまで実装可能)から成り、SUN-3/260上のQuintus Prologと接続して、基本処理機能の評価とプログラム

開発を進めている。

これと並行して、73台のPEから成るIXM第2版(IXM2)の開発を行ってきたが、このほどそのハードウェアが完成した。本稿では、意味記憶システムIXの全体を示しつつ、IXM2の概要を述べる。

2. 意味ネットワーク言語IXL

IXLはIXMマシンのプログラミング言語であり、Prologのスーパーセットとなっている。IXLは、意味ネットワーク処理用の述語(これをIXLコマンドと呼ぶ)をPrologに付加したもので、意味ネットワークに対する全解探索をPrologプログラムの中から行える(もちろんPrologもそのまま使える)。

例えば次の節が入力されたとする。

```
?-isa(canary,X),write(X),fail.
```

isa(canary,X)は、カナリアの上位概念を求めるための、意味ネットワーク処理用の述語である。これがホスト計算機上で解釈されると、IXMマシンにIXLコマンドとして渡され、その複数の解が一斉に求められる。解は各PEで見つかり次第、ホスト計算機に送られるが、ホストのPrologインタフェイスではそれら複数の解を管理し、上の節でいえば、Xにバックトラックするたびに一つづつ解を返して行くので、みかけは通常のPrologの実行とかわらない。

Prologが苦手とする集合演算も、IXMマシンでは連想メモリの利用により効率的に処理できる。

IXLは図1に示すように、ロジックの述語に似た形式のコマンド言語であり、知識ベースの記述、検索、修正の機能を持つ。例えば、知識ベースを記述するためのコマンド、link(is-a, canary, bird)は、「カナリアは鳥である。」を示す意味ネットワークを生成する。このあと、検索コマンドとしてisa(canary, X)を与えれば、解としてbirdが得られる。

IXLは、従来の意味ネットワーク言語の弱点を補

To construct a relation:	To connect nodes by a link:
assertion(R, X, Y).	link(is_a, X, Y).
property(R, X, Y).	link(not_isa, X, Y).
	link(instance_of, X, Y).
To inquire about a link:	link(not_instanceof, X, Y)
isa(X, Y).	link(a_kind_of, [X, Y, ...], Z).
instance(X, Y).	link(source, R, X).
ako(X, Y).	link(destination, R, Y).
source(R, X).	link(rule, X, ((
destination(R, Y).	asst(R, X, Y):-....
	prop(R, X, Y):-....
To inquire about a relation:	isa(X, Y):-....
asst(R, X, Y).	instance(X, Y):-....)
prop(R, X, Y).	

図1 IXLコマンド

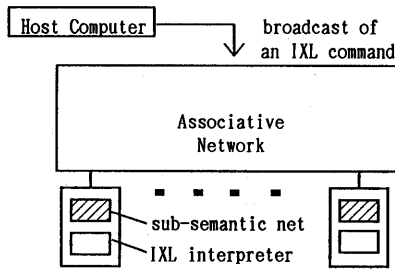


図2 IXL Mの全体構成

強し、次の3つの特徴をもっている。

- 1) 概念間の関係の意味をより正確に、かつ柔軟に記述する。
- 2) 手続き的知識をロジックの節形式で記述する。
- 3) 否定的な知識を陽に記述できる。

手続き的知識の記述が必要になるのは、意味ネットワークの表わす知識が各種の事実や階層関係など宣言的なものが主であり、これだけでは複雑な推論規則を扱えないからである。IXLではロジックの節形式で手続き的知識を表わしている。

3. 意味ネットワークマシン IXL M

3.1 全体構成

IXL Mマシンは、ホスト計算機の制御下でMIMD的に動作するマルチプロセッサであり、図2に示すように連想メモリを備えたPEと、これらを接続し、かつ意味ネットワーク処理の一部も担う連想ネットワーク (Associative Network) から構成される。

各PEには連想メモリがあり、大規模意味ネットワークを分割した“部分”意味ネットワークをそれぞれ格納する。

IXL Mマシンの実行は、ホスト計算機から発せられる一つのIXLコマンドを単位としている。たとえば、カナリアは何の下位概念かを調べるのに、ホストはisa(canary, X)というIXLコマンドを発するが、この処理が終わるまでIXL Mマシンは他のコマンドを受け付けない。従ってIXLコマンドのレベルではIXL Mマシンの挙動はSIMDである。

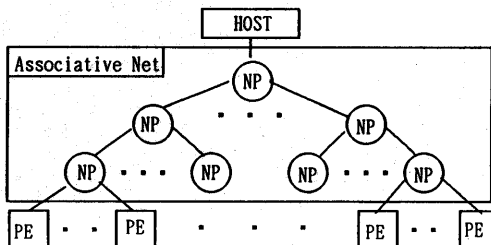


図3 木構造接続時のIXL Mの構成例

但しその一つのコマンドの実行中にIXL Mマシン内の各PEはMIMDで動作する。PEの実行はプログラムカウンタに基づく制御でなく、意味ネットワーク内を流れるマーカによって起動される非同期型制御に基づいており、並列性を生かす目的を持つ。つまり各PEは、それぞれの連想メモリが格納する意味ネットワークの内容に応じて互いにメッセージ交信しあい全く非同期的に動作して、一つのIXLコマンドの複数の解を求めて行く。

連想ネットワークはプロセッサ間交信を行うほかに、連想メモリを用いてマーカ伝搬の並列化処理を行う。連想ネットワークは、相互接続されたネットワークプロセッサ (NP) の集合であり、その接続トポロジーとして木構造をとる場合は、たとえば図3のような構成となる。NPもPEと同様に連想メモリを備えている。

3.2 連想メモリを活かした意味ネットワーク処理

IXL Mでは、各PEに連想メモリを設け、そこに大規模な意味ネットワークを分割したサブネットワーク、つまり複数のノードを格納し、連想処理、そして集合演算を一斉に施す。つまり連想メモリのワード数分の並列性が1台のPEの中に存在することになる。

連想メモリを用いた意味ネットワーク処理の例を示す。図4の意味ネットワークがあり、二つの集合と、その積集合の様子を表現しているとする。連想メモリへの格納はリンクを単位としている。図5 (a) に示すように、連想メモリ上でのリンク表現は、そのリンクによって接続されるノードの属性、そのリンクのもう一つの接続先、そのリンク名、及び意味ネットワーク処理の演算結果を保持するマーカビット領域から成る。たとえば集合AのメンバーであるA1のノードは、Aのノードをisaリンクで指しているので図5 (b)

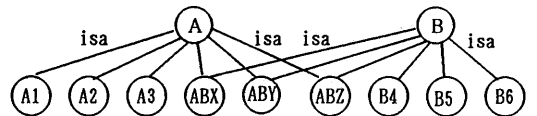


図4 二つの集合を表わす意味ネットワーク

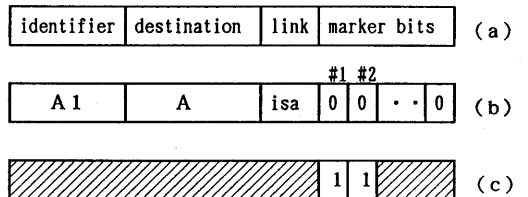


図5 連想メモリ上での意味ネットワークの表現

のようになる。

処理の例として図5の意味ネットワークにおいて二つの集合の積集合をもとめる場合を挙げる。すでに集合Aのメンバーにマーカビット1番、集合Bのメンバーにマーカビット2番がセットされているとすると、両集合の共通要素を求めるには、連想メモリの検索データレジスタを図5(c)のようにセットして検索を行えばよく、一斉に共通要素のワードにヒットフラグが立つ。つまり積集合の演算を、共通要素がいくつあっても、ただ一回の連想メモリアクセスで実現することが可能である。

これからわかるように大規模意味ネットワークでもそれが連想メモリのワード数内に収まるなら、連想・集合演算(および、特定のマーカ伝搬)は常に一定時間内で実行できることになり、意味ネットワークが大きくなったときの計算量の爆発に対して有効な手段となる。

見方を変えれば、連想メモリは各ワードが特定の機能を有するPEであり、ワード数分の並列性を持ったSIMDマシンともいえる。その場合マーカビット領域はPEのローカルメモリにたとえることができる。

4. プロトタイプシステムの概要

3に述べたアーキテクチャの検証、とくに連想ネットワークの接続トポロジーの検討を行うために、33台のCPUと連想メモリから成る意味ネットワークマシンIXMのプロトタイプを製作した。

プロトタイプは図6に示すように、1)ホスト計算機のSUN3、2)任意の接続形態を実現するための5枚のスイッチ基板、3)PEまたはNPの機能を実現する32枚のPE基板、4)ブロードキャストプロセッサ、の四つの部分から成っている。IXMの一般ユーザはSUN3上のPrologプログラムからIXMを利用する。プロトタイプの外観を図7に示す。

・PE基板

PE基板は、T800トランスピュータ、連想メモリ、SRAM、リンクアダプタ、および周辺回路から成る。

トランスピュータはプロセッササイクルが17.5MHz

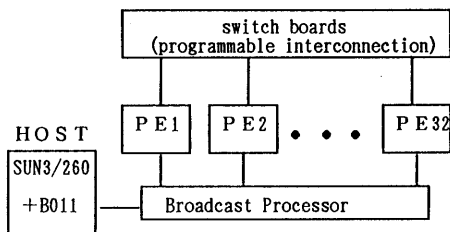


図6 プロトタイプの概要

である。内部にはサイクルタイム59nsの4KByteオンチップRAMを備えている。外部メモリは32k語 X 32ビットのSRAMで、サイクルタイムは230nsである。

トランスピュータの4本のシリアルリンクは、20Mbit/sに設定する。しかし、4本だけは連想ネットワークの種々の接続トポロジーの実験を行えないため、リンクアダプタ(C012)と呼ぶ直並列変換用のチップをT800のバスに接続して、リンク数を8つに増やしている。

連想メモリはSRAMと共にoccamのアドレス空間に割り付けられており、T800からはRAMと同じようにアクセスできる。(アクセス時間の実測平均は約450ns)。

連想メモリはPE基板当たり最大4k語まで実装可能であり、IXMマシン全体で128k語となる。この規模は連想プロセッサとして最大規模の範疇に入る。

マーカビット数が少なく済む応用では連想メモリの1語に意味ネットワークの1リンクを格納する。ただし、一つのリンクについて、それを逆方向から見たものも1リンクに数えるので、1リンクは連想メモリの2語を占める。従って、IXMプロトタイプ全体で最大64Kリンクまでの並列処理が可能である。またノード数では、最大32K個まで扱える。

・スイッチ基板

32台のトランスピュータの持つシリアルリンクの総数は128本であり、これらを5枚のスイッチ基板に分けて接続し、32PE基板間の任意の接続形態を実現で

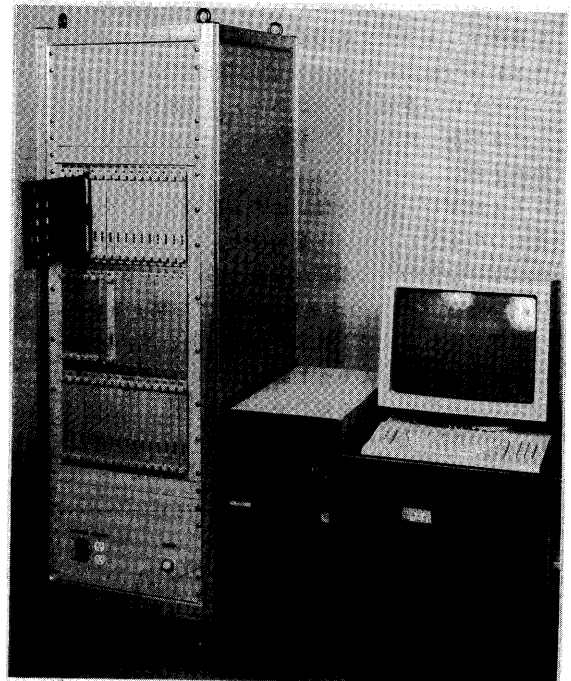


図7 IXMプロトタイプ外観

きるようにしている。スイッチ基板内のリンクスイッチLSIは、入力、出力を各32本持っており、入出力間の対応関係、つまりPE基板間の接続形態は、ホスト計算機内のトランスピュータからソフトウェアで書き換えられる。この可変な接続機構により、提案している4進木以外の構成も実現することができる。

5. IXM2の開発

5.1 目的

4で述べたIXMプロトタイプは、連想ネットワークの形態によってネットワークプロセッサ(NP)と、意味ネットワークを処理するPEの数とが変化する。このため、連想メモリを持つPE基板を種類製作し、これをNPにも使えるようにしていた。しかし、マカ伝搬の並列化などを行うNPの機能が、そのために不十分であった。

またスイッチ基板の採用により、プログラムで構成を変更することのできる相互接続網を実現したが、クロスバLSI数の制限から実現できないトポロジーが生じる、実装上リンクスピードを20Mbpsでなく10Mbpsで使わざるを得ない、などの不都合があった。

このような問題の解決と、転送能力の高い完全結合ベースの接続ネットワークでより大規模な意味ネットワークの処理実験を行いたいことから、プロトタイプ第2版としてIXM2の製作を行った。

5.2 IXM2の全体構成

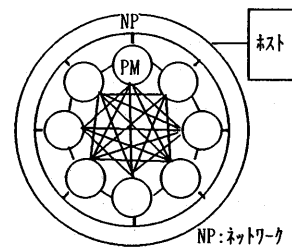
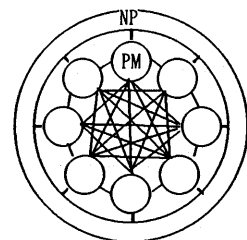


図8 全体構成の概念図



NP: ネットワークプロセッサ
PM: PE module

図9 PMの構成図

IXM2は、多数台の接続を実現するために、再帰的な接続方式を採用している。図8に全体構成の概念図を示す。ホスト計算機につながるNPの下に8つのPEモジュール(PM)が接続され、かつこれらのPM間の接続は完全結合である。

図9はPEモジュール(PM)を示している。PM内部もさらにPMから成っている。つまり両図からわかるようにPMは、その中にPMを含ん

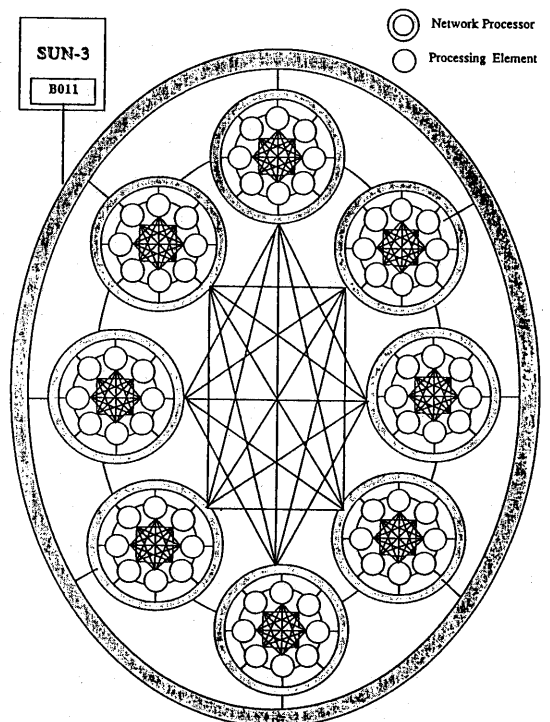


図10 IXM2の全体構成図

でいるという点で再帰的である。一つのPMは、8台のPE(最も内側なので、PMでなくPE)と、1台のNPから成る。但し、最も内側のPMは、PEボード、つまり連想メモリ付きのトランスピュータボードとなる。

今回完成したIXM2は、8つのPMと1台のNPからなる。全体の構成図を図10に示す。またその外観を図11に示す。

1つのPM内の8台のPEは、互いに完全結合されていると共に、NPに対して、それぞれ1つの接続リンクを持つ。この接続リンクは、通常のPE-NP間交信に加え、NPからPEへのブロードキャストとしても使われる

1台のNPは図10からわかるように、他の7台のNPとの

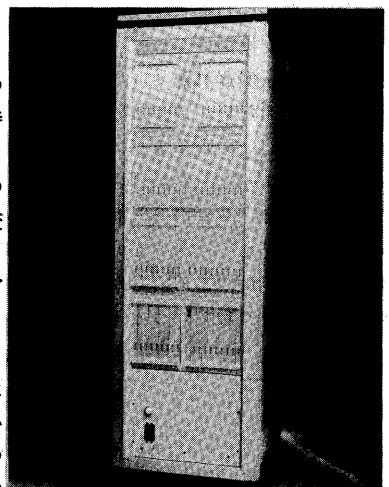


図11 IXM2の外観

完全結合を持つと共に、上位のNPへの接続を持つ。従ってNPは下位の8台のPM (PE)、同一レベルの7台のNP、及び上位のNPへの接続、の計16本の接続を持つ。しかもこの16本は、ブロードキャストとしても利用できるため、マーカ伝搬の並列化が可能である。このブロードキャストの宛先は連想メモリ

に格納しているため、転送相手を効率的に決定できる。これについては6. 3. 2で詳しく述べる。

5. 3 接続トポロジーの選択

前節に示した計算機間接続方式を考えるうえでの制約条件は、リンク接続であること、メッセージ交換であること、多数台(数百台)に対応できること、の3つとした。

一般にマルチコンピュータ(ローカルメモリを持つPEの集まり)の相互接続ネットは、両極端に共有バスと完全結合を持つスペクトラムの中で決定される。PE間接続のトポロジーとして、最も能力が高いのは完全結合であるが、コストが台数の二乗に比例するため、通常は敬遠される。

しかしリンク接続を前提に8台程度をモジュールとし、これを階層的に組み合わせ、大規模システムをつくることは、実装、コストの両面において十分に実現可能である(注)。

MIMDシステムを実現する場合、数十台までは共有メモリ型、数百台ではメッセージ交換という見方があるが、IXM2の場合はトランスピュータを採用するため台数に拘わらずメッセージ交換で行くしかない。メッセージ交換型は、iPSC、Ametekなどの例をみると、メッセージ伝搬遅れがミリ秒単位のオーダーで、決して速いとはいえない。IXMではこのあたりを完全結合で改善したい。少なくともメッセージの中継PEが無い分、シリアルリンクの遅さを補う。

完全結合は、8台のPE間を基本単位としているが、この範囲を越えるとNPを介した交信になるので効率落ちる可能性がある。しかしこの8台の空間は、意味ネットワーク数でいうと1万6千リンクであり、そ

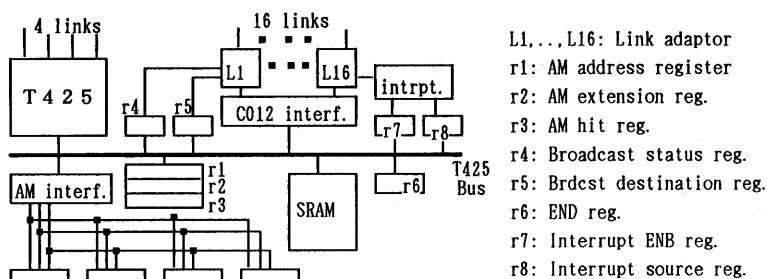


図12 NPの構成

の範囲に交信の局所性を留めることは意味ネットワークのアロケータ(意味ネットワークを各PE内の連想メモリにのせるために、大きな意味ネットワークを分割するソフト)で対応することにより、十分に可能と考えられる。

この相互接続ネットでは、完全結合の効率の良さと、大規模システムへの拡張のしやすさの両立を目標としている。

6. IXM2の構成要素

6. 1 ホスト計算機

ホスト計算機となるSUN3/260上では、Quintus Prologが走り、ここからIXLコマンドが呼び出され、IXMマシンにわたされる。実際にはPrologからUNIX上の通信プロセス(Cで記述)が呼び出され、このプロセスがSUN3のVMEバス上にあるデュアルポートメモリ(全2Mbyte)の8KbyteをIXMマシンとのデータ交信領域として使用する。

このデュアルポートメモリは、VMEバスに接続したトランスピュータボードB011内にあり、UNIX上のプロセスと、B011内のトランスピュータの両方からアクセスできる。

B011内のトランスピュータは、IXM2の最外側に位置するNP(ネットワークノード)との間にデータ転送リンクを持ち、NPに対してIXLコマンドなどIXM2への命令を送るほか、NPからはIXM2が求めた解を受け取る。

Quintus PrologへのIXLの組み込み、IXM2との交信ソフトウェアはすでに完成している。

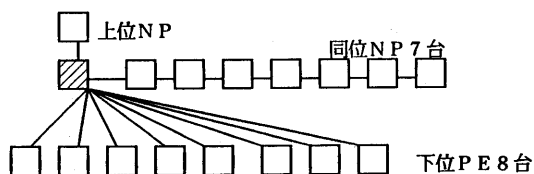


図13 NPのブロードキャスト先

注) 実際にIXM2を作ってみて、そのとおりであると感じた。ボード間の配線にマザーボードを使わず、約300本のケーブル(1ケーブルは4信号)を73台のCPUボード間に配線したが、思いのほかバックプレーンはすっきりとしている。

6.2 PE

IXM2の64台のPEは、3章のプロトタイプシステムのところで述べたPEと基本的に同一の構成である。ただし、連想メモリアンタフェイスの高速化など、若干の改良を加えている。

各PEは8本のリンクを持ち、7本が同一PM内の他の7台のPE、残り1本がNPと接続されている。

6.3 NP

NPは、PEと同様に連想メモリを持つが、PEと違い、最大16箇所までのブロードキャスト機能を持つ。この機能は、ホスト計算機からPEへのIXLコマンドの伝達のほか、後述するマーカー伝搬の並列化において効果的である。

6.3.1 NPの構成

NPの構成図を図12に示す。ブロードキャスト用に16個のリンクアダプタをトランスピュータのバスにつないでいる。図13のように16本のうち、8本はこのNPの下に位置する8台のPEにつながり、7本はこのNPと同階層に位置する7台のNPにつながり、残り1本は上位階層の1台のNPにつながる。

これらのリンク以外にトランスピュータに直結するリンクが4本あり、プログラムのロード、テストなどに用いる。連想メモリは、4チップ、2K語の容量を持つ。SRAMは128Kbyteである。

6.3.2 マーカー伝搬の並列化

マーカー伝搬は、例えばあるノードの下位概念にあたる全ノードを求める際に、そのノードからisaリンクをたどって到達できるノードに特定の識別子をたて

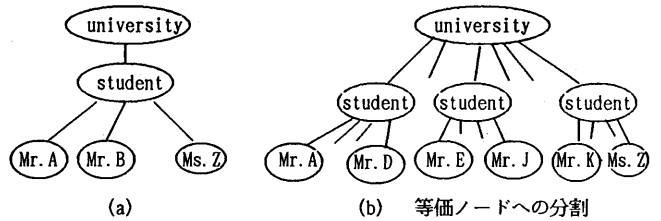


図14 等価ノード

る操作をいう（具体的には各ノードごとにマーカービットと呼ぶ複数の1ビットフラグを用意し、その何れかをセットする）。マーカー伝搬に内在する並列性は、大規模な意味ネットワークほど高くなる傾向があり、その並列性をいかすことが処理時間の短縮につながる。

IXMの場合、マーカー伝搬は、多数台のPEによる並列処理によって実現されるが、なるべく多くのPEにその処理が分配されるように支援する必要がある。このため、以下に述べるように、等価ノードとよぶノードを導入し、その効率の実現のためにNPにブロードキャスト機能を付加した。等価ノード (Equivalence Node) の概念を図14を用いて説明する。

いまuniversityのノードからマーカーを流し、大学に属する全学生をマーカー伝搬によってマークしたいとする。もしstudentのノードが1台のPEにわりつけられているとすると、そのPEは全学生に対して逐次的にマーカーを流していかなくてはならず、処理時間上のボトルネックとなりやすい。これはPEが1台しかないのだから当然である。意味ネットワークは階層的知識の記述に用いられることが多いため、このようにファンアウトの多いノードは決して特別な例ではない。

この問題の解決のためには、このようなノードを複数台のPEにわりつければよい。つまり、studentのノードを図14(b)のように複数化して別々のPE間

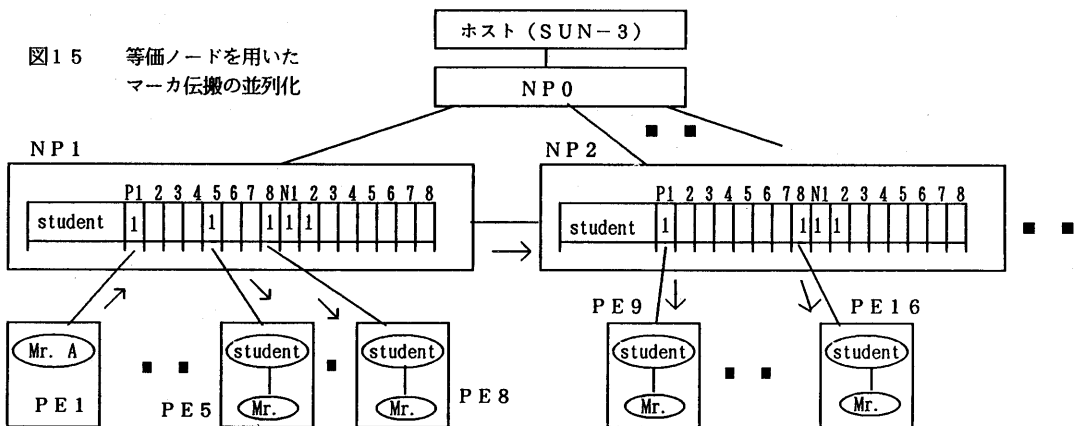


図15 等価ノードを用いたマーカー伝搬の並列化

に分散すれば、マーカの伝搬を並列に行わせることができる。このstudentノードのようなノード、つまり論理的属性は唯一つであるが物理的にはPE間に分散されているノードをIXMでは等価ノードと呼ぶ。

但し、ある等価ノードにマーカが送られてきたとき、そのマーカは複製されて、その等価ノードを含む他のPEに対しても転送されなければならない。

たとえばMr. Aによって、それが物理的につながっているstudentノードのマーカビットがセットされたならば、他のPE内のstudentノードのマーカビットもセットされなければならない。このため、連想ネットワークを形成する各ネットワークプロセッサの中には等価ノードを登録している表があり、さらにその表は連想メモリに格納されてマーカ伝搬の効率化が図られている。

一例として図15に上述のMr. Aからのマーカ伝搬の並列化の様子を示す。これは、PE1にあるMr. AのノードからPE5にあるstudentノードへマーカを送ったときに、マーカが複製されてPE8、PE9、PE16にあるstudentノードへもマーカが送られる様子を示している。つまり、Mr. Aからマーカ更新のリクエストがNP1に送られると、NP1は更新先IDであるstudentについてテーブル（連想メモリ上におかれる）をひき、studentノードの分散状況を知る。この表の内容はそのままブロードキャストの相手先を示しており、NP1はこれをブロードキャスト相手先レジスタに書き込む。やがてブロードキャストが可能になったときに割り込みがかかり、その時点でブロードキャストを行う。この例だと、PE5、PE8、NP2に対してブロードキャストが行われる。その後NP2は、PE9とPE16に対してブロードキャストを行う。

マーカ伝搬の並列化では、どれだけ多くのPEにマーカ伝搬をさせるかが重要で、意味ネットワークのアロケータの分割アルゴリズムに十分注意する必要がある。

7. おわりに

IXM2はハードウェアが完成し、現在調整を進めている。IXM2用の意味ネットワーク処理プログラムの作成も、プロトタイプ第一版用のプログラムから

の書き換えを中心に並行して進めている。

応用プログラムとしては、従来作成してきたワインの知識ベースの整備に加え、新たに植物図鑑の知識の意味ネットワーク化を検討している。また自然言語応用として、キリアンがはじめて意味ネットワークを提案したときの、辞書⁷⁾の語義文の意味ネットワーク化の検討を進めており、英単語の意味のdisambiguationを実験したい。また既に開発中の、意味ネットワークを知識ベースとする知識獲得システム⁸⁾のIXM2上への移植も予定している。

末筆ながら連想メモリでご援助を頂くNTT・LSI研究所の小倉武氏、SUN3用TDSをご提供頂いた新日鉄第一技研の福島氏、徳永氏、田内氏、相良氏、ご援助・助言を頂くインモス望月部長、ご指導頂く柏木電総研所長、棟上情報アーキテクチャ部長に感謝する。

参考文献

- 1) 小倉、他：“20Kb CAM(Content Addressable Memory) LSI”, 信学技報、CPSY87-23, pp. 31-37, 1987.
- 2) Handa, K., et al.: “Flexible semantic net for knowledge representation”, 情報処理学会論文誌、7月、1987.
- 3) Furuya, T., et al.: “Architectural evaluation of a semantic network machine,” IWDM, 1987.
- 4) 樋口、古谷、国分、楠本、半田：“並列連想記憶を用いた意味ネットワークマシン”、信学技報、EC85-55, 1986.
- 5) 半田、他：“意味記憶システムIX - 知識表現言語IXLによる関係の表現”、情報処理学会全国大会、3月、1985.
- 6) Higuchi, T. et al.: “The prototype of a semantic network machine IXM”, Intl. Conf. on Parallel Processing, Aug., 1989.
- 7) Ogden, C. K.: “The general basic English dictionary”, New York, W. W. Norton, 1942.
- 8) Handa, K., et al.: “Learning Importance of Concepts: Construction of Representative Network”, 3rd Knowledge Acquisition for Knowledge-based Systems Workshop, Nov. 1988.