計算機アーキテクチャ 73-13
(1988. 10. 31, 11. 1)

マルチPSIにおける
実験的負荷分散メカニズム

武田 保孝† 岩山 洋明† 中島 浩† 益田 嘉直†
近山 隆‡ 瀧 和男‡

†三菱電機株式会社 ‡新世代コンピュータ技術開発機構(ICOT)

現在、我々は第5世代コンピュータ・プロジェクトにおける並列ソフトウェアの研究開発環境を提供することを目的とし、逐次型推論マシンPSIのCPUを要素プロセッサとして、格子型に64台結合したマルチプロセッサシステム《マルチPSI》の開発を行っている。並列ソフトウェアの研究課題として、並列言語、並列OS、負荷分散方式等があり、何れも大規模な知識情報処理を行う上で重要なものである。

マルチPSIシステムでは、Processing Power Planeと呼ばれる仮想平面を用いる負荷分散方式をサポートするためのハードウェア機構を備えている。この負荷分散方式は、実行負荷の各要素プロセッサに対する割り付けを局所的な情報のみで動的に変更することで、負荷の均衡を得る方式である。本稿では、この負荷分散方式と、マルチPSIシステムでのハードウェア・サポートを中心として述べる。

A LOAD BALANCING MECHANISM FOR THE Multi-PSI SYSTEM

Yasutaka TAKEDA† Hiroaki IWAYAMA† Hiroshi NAKASHIMA† Kanae MASUDA†
Takashi CHIKAYAMA‡ Kazuo TAKI‡

† Mitsubishi Electric Corporation
5-1-1,Ofuna,Kamakura 247,Japan

‡ ICOT Research Center
1-4-28,Minato-ku,Tokyo 108,Japan

We are developing the Multi-PSI system which supports a real parallel execution environment for parallel software research and development. The parallel software research is important in building a parallel computer system for high performance knowledge information processing.

In large scale parallel computer systems, load balancing is an important and difficult factor of efficient utilization of their full processing power. Dynamic load balancing can be achieved by using a method based on P 3 (Processing Power Plane) model. This paper describes the load balancing method and its implementation for the Multi-PSI system.

1 はじめに

第5世代コンピュータ・プロジェクトの最終的な目標として大規模な知識情報処理を高速に実行することを目的とした、要素プロセッサ数が数百～千台規模の並列推論マシン（ＰＩＭ：Ｐａｒａｌｌｅｌ Ｉｎｆｅｒｅｎｃｅ Ｍａｃｈｉｎｅ）の開発がある[1]。

本プロジェクトを推進している新世代コンピュータ技術開発機構（略称ＩＣＯＴ）では、ＰＩＭの開発に先駆けて、並列ソフトウェアの研究開発をマルチプロセッサ上で行うことを目的とした、マルチＰＳＩシステムを開発中である（図１）[2][3]。

マルチＰＳＩシステムは、逐次型推論マシンＰＳＩのＣＰＵをＬＳＩ化により改良小型化したものを要素プロセッサとし、専用の接続ネットワーク制御機構により最大６４台までの要素プロセッサを結合したものである[4]。

要素プロセッサは、共有メモリを持たない疎結合で、格子状に接続されており、マンマシン・インタフェースとして最大4台のフロントエンド・プロセッサが接続可能である（図２）。

現在、マルチＰＳＩシステムのハードウェア開発は完了しており、本システムを用いて並列論理型言語ＫＬ１（Ｋｅｒｎｅｌ Ｌａｎｇｕａｇｅ ｖｅｒｓｉｏｎ １）の処理方式[5][6][7][8]、並列プログラムの負荷分散方式、並列処理アルゴリズムの研究ならびに並列ＯＳ（ＰＩＭＯＳ：Ｐａｒａｌｌｅｌ Ｉｎｆｅｒｅｎｃｅ Ｍａｃｈｉｎｅ Ｏｐｅｒａｔｉｎｇ Ｓｙｓｔｅｍ）の開発が行われている（図３）。

大規模なマルチプロセッサシステムでは、各要素プロセッサの実行負荷を均衡させることが重要な課題となるが、要素プロセッサ間の通信量の観点からは局所的な情報のみで負荷分散を行えるような方法が望ましい。

本稿では、まずマルチＰＳＩシステムのハードウェア構成について、要素プロセッサ間の接続ネットワークを中心に述べ、つぎにＰｒｏｃｅｓｓｉｎｇ Ｐｏｗｅｒ Ｐｌａｎｅという仮想的な平面を用い、局所的な通信のみで動的な負荷分散を行う方式とそのハードウェアサポート機構の特徴，メカニズムについて述べる[9][10]。

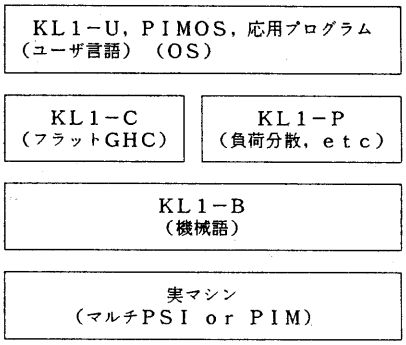


図３ 言語体系

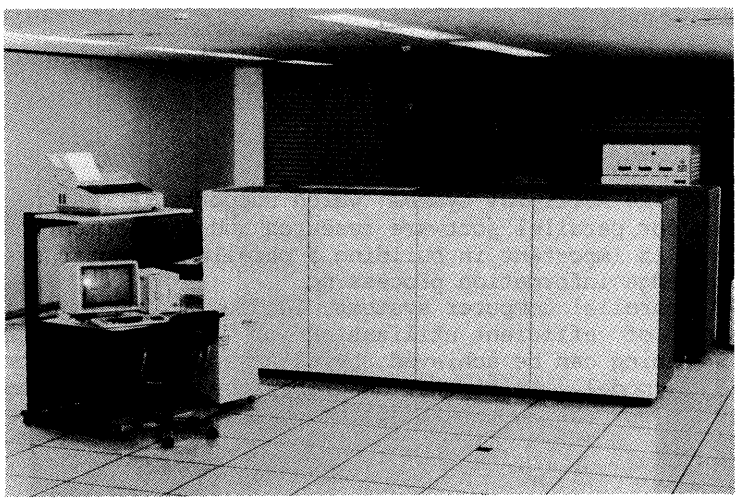


図１ マルチＰＳＩシステムの外観

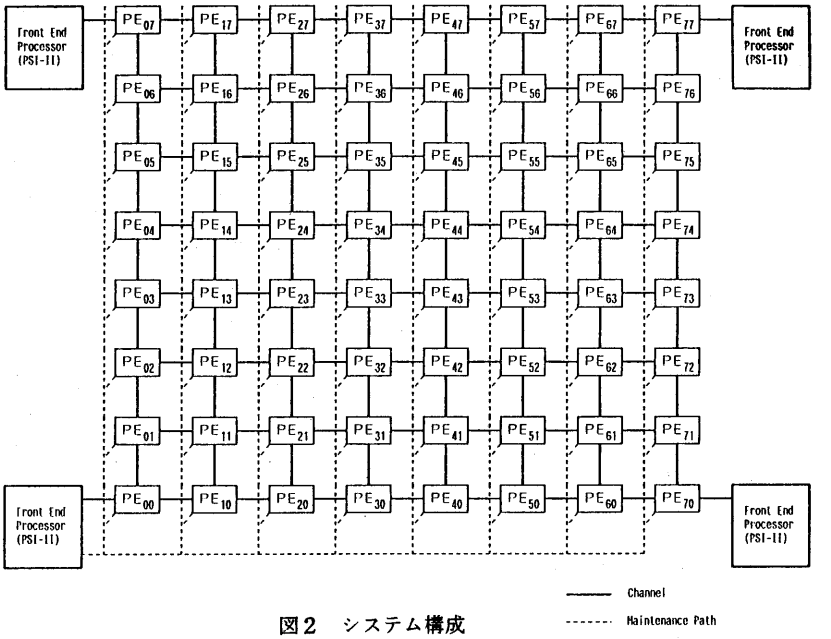


図2　システム構成

2　マルチPSIのハードウェア構成

マルチPSIシステムの要素プロセッサ（以下PE）は、図4に示す構成になっている。以下、接続制御部の機能と特徴について述べる。

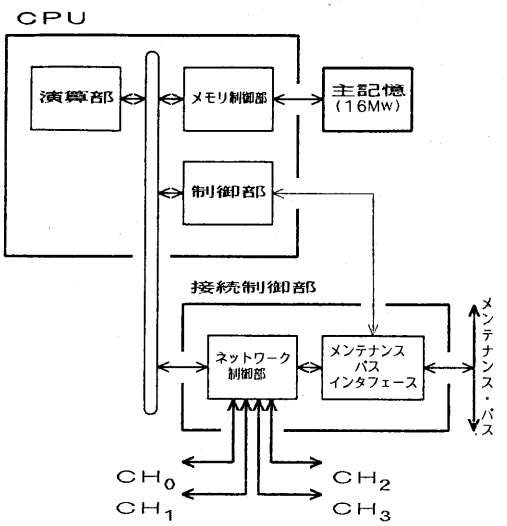


図4　要素プロセッサの構成

2．1　接続ネットワーク制御部

接続ネットワーク制御部は、図5のようにCPUの内部バスに接続されており、その内部資源はCPUの特殊レジスタとしてアクセス可能である。接続ネットワーク制御部は、隣接するPEの接続ネットワーク制御部への4本の双方向チャネル（Ch0～3）と、CPUへの双方向チャネル（Ch4）が5×4のスイッチで接続された構成となっている。

パケットの転送は、同一チャネルへの複数チャネルからの転送要求がない限り並列に行われる。各チャネルは、パリティを含めた10ビット単位で同期転送を行い、1チャネルの1方向のあたりの転送速度は5Mバイト／秒である。

Ch0～3にはパケットの転送方向を決定するためにPT（Path Table）と呼ばれるメモリが設けられており、パケットの到着時にその宛先をアドレスとしてPTの内容を参照することにより、転送先チャネルが決められる。このパケット転送方向の決定動作はハードウェアでサポートしており、CPUの動作とは無関係に自動で行われる。また、Ch0～3の出力側には転送方向の競合等で生じる、ネットワーク中のパケットの渋滞による転送速度の低下を緩和することを目的とした、48×10ビットのFIFOバッファが設けられている。

Ch4はPTを持たないため、CPUから送出されるパケットには転送方向を決定する2ビットが付加されている。また、CPUに渡すパケットは受け取ったチャネルを示す2ビットが加えられる。Ch4は入出力とも4K×12ビットのFIFOバッファを持ち、CPUから送出されるパケットはWB（Write Buffer）に書かれ、全パ

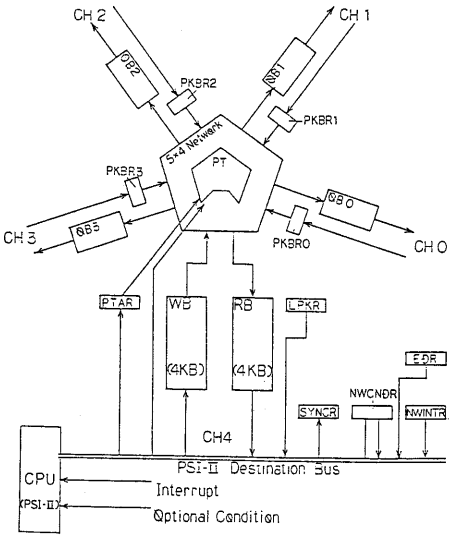


図5　接続ネットワーク制御部

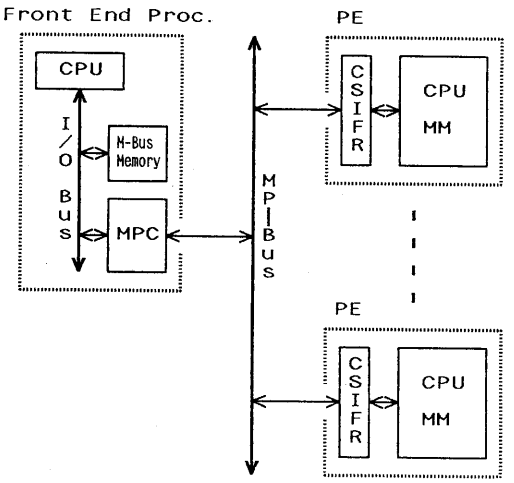


図6　メンテナンス・バス接続形態

ケットデータが揃った時点でネットワークへの送出が開始される。他ＰＥから自ＣＰＵへのパケットはＲＢ（Ｒｅａｄ　Ｂｕｆｆｅｒ）に全パケットデータが揃ってからＣＰＵに対し到着が報告される。

パケットの到着・送出及びネットワークの異常は割込みを用いてＣＰＵに報告される。

２．２　メンテナンス・バス機構

メンテナンス・バス機構は、システムの立上げ、障害管理を行う手段であり、一般の計算機におけるＳＶＰ（ＳｅｒＶｉｃｅ　Ｐｒｏｃｅｓｓｏｒ）機能に相当する。

メンテナンス・バス（以下、ＭＰ－Ｂｕｓ）は、フロントエンド・プロセッサと各ＰＥ間を１本の非同期バスでいもづる式に接続したものであり、フロントエンド・プロセッサがマスタでＰＥがスレーブであるような、単一マスタ複数スレーブ方式のバスである。各ＰＥには、ＣＳＩＦＲ（ＣｏｎＳｏｌｅ　ＩｎｔｅｒＦａｃｅ　Ｒｅｇｉｓｔｅｒ）という、ＭＰ－ＢｕｓとＣＰＵ間のインタフェース用レジスタ群が、ＭＰ－Ｂｕｓに接続されている（図６）。

フロントエンド・プロセッサから各ＰＥのＣＳＩＦＲを読み書きすることにより、各ＰＥのＣＰＵの起動、停止、ステップ実行、マイクロ命令の設定、内部バスへのアクセス、停止時の状態判断等を行うことができる。従って、一連のＣＳＩＦＲアクセスにより、マイクロプログラムで実現できるような全ての操作（任意のレジスタの参照／変更など）を実行することができる。

フロントエンド・プロセッサのＩ／Ｏバスには、ＭＰＣ（Ｍａｉｎｔｅｎａｎｃｅ　Ｐａｓｓ　Ｃｏｎｔｒｏｌｌｅｒ）という制御装置が接続されている。ＭＰＣの主な機能は、Ｍ－Ｂｕｓメモリに置かれたＭＰ－Ｂｕｓをアクセスするためのコマンド列を解釈／実行しＰＥのＣＳＩＦＲとＭ－Ｂｕｓメモリとの間のデータ転送を行うことである。コマンドは６種類用意されており、Ｍ－Ｂｕｓメモリ上には、コマンド列及びデータ格納領域として、３種類のバッファが確保されている。

フロントエンド・プロセッサからＭＰＣへのアクセスは、ＭＰＣ上にある９種類のレジスタをＩ／Ｏ命令により読み書きすることで行われる。また、コマンド列の実行終了時や、ＭＰ－Ｂｕｓの異常検出時には、ＭＰＣよりＩ／Ｏ割込みとして、フロントエンドプロセッサのＣＰＵへ報告される。

３　負荷分散メカニズム

３．１　負荷分散方式の概要

マルチＰＳＩシステム開発の目的のひとつとして、負荷分散方式の研究があげられる。この負荷分散方式は、処理能力が均一に分布した平面Ｐｒｏｃｅｓｓｉｎｇ　Ｐｏｗｅｒ　Ｐｌａｎｅ（以下Ｐ３と略す）を仮定し、動的に負荷の再分配を行う方式であり、近傍のＰＥ間の局所的な負荷の情報のみで負荷の分散を行うことを特徴としている。。

ＰＩＭのように大規模なマルチプロセッサシステムにおいてシステム全体の処理能力を効果的に使用するためには、各ＰＥの実行負荷を均衡させることが重要な課題である。

一般に画像処理等の処理・データに高い規則性のある応用では、各ＰＥの実行負荷は比較的容易に予測することができ、負荷の分配は予めプログラム中で記述することができる。しかし、知識情報処理等の処理・データ共に規則性の低い応用では、各ＰＥに対する実行時の負荷を予測することは極めて難し

い。

P3を用いる方式では、初期状態で各PEに対して適当に負荷を分配し、実行時に隣接したPE同士の局所的な負荷の不均衡に応じて、動的な負荷の再分配をすることによって各PEに対する負荷を均一にする。この負荷分散方式は以下の3点を主に考慮して提案されたものである。

(1)プログラマがマルチプロセッサシステム上で実行される並列プログラムを作成する際に、そのマルチプロセッサシステムのPE数や、PE同士の接続関係といった物理的な構成を意識しなくても良い。

(2)PE間の通信による処理のオーバヘッドを少なくするためには、通信の局所性を保つ必要がある。また、PE数の多い大規模なマルチプロセッサシステムを考えると、全てのPEの状態を監視することは極めて難しいと考えられる。従って、負荷分散の動作は局所的な負荷の状態によって行なわれ、大域的な負荷の情報や距離の隔たったPE同士の通信を必要としない。

(3)通信の局所性を保って効率的な実行をするためには、関連のある仕事を物理的に近いPEに割付けることが望ましいが、自動的に割付けを行うことは難しく、プログラマが割り付けを行うことは、前記(1)と矛盾してしまう。そこで、PE数やプロセッサ間の物理的な接続に拘らない抽象的な距離の概念を導入し、プログラム中で大雑把な割付けを陽に記述する。この抽象的な距離は、物理的なプロセッサ間の距離と正の相関を持つ。

3.2 負荷分散方式

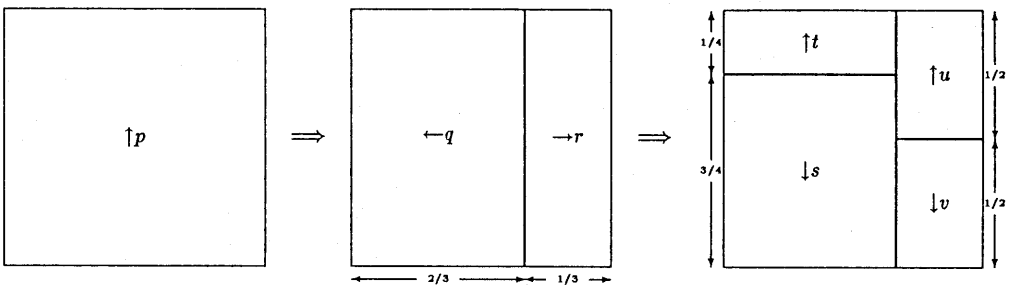
P3を用いた負荷分散方式は、P3と呼ばれる平面を仮定し、このP3上に実行すべき問題を割りつける。問題の実行は、このP3平面をPEによって構成される物理的なマルチプロセッサシステムに写像する。各PEはP3平面をPE数に応じて分割し、自PEが分担したP3の部分領域上に割付けられている問題を実行する。

Prologを基とした言語で、ある問題pを全PEで実行する場合について、P3への問題の割付けの例を考える(図7)。

PE全体はP3全体に対応するので、問題pはP3全体に割付けられる(図7a)。つぎに問題pの実行が部分問題q, rの実行に分けられ、qの実行による負荷がrの負荷の「倍程度」とすると、以下のように記述される。

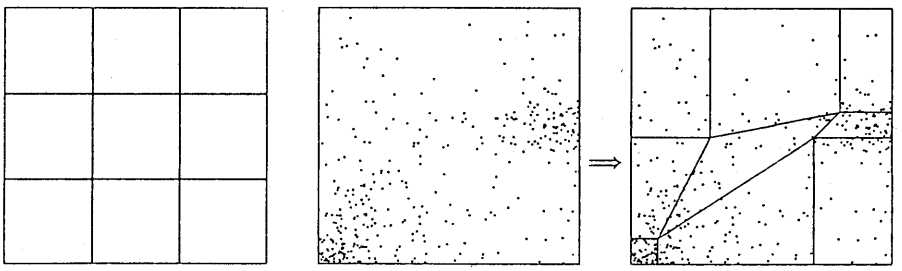
```
p  :-  ←(2×q), →r.
```

問題pの割付けられていたP3領域をプログラム中で指定された面積比で分割し、部分問題q, rを割付ける(図7b)。さらに部分問題qが部分問題s, tの実行に分けられ、部分問題tの実行が部分問題u, vに分けられるとすると、以下のように記述され、部分問題q, rの割付けられている領域を分割して部分問題s, t, u, vを割付ける(図7c)。



a. 初期状態　b. p→q, rの分割　c. q, rの分割

図7　P3の分割による負荷分散



a. 初期分配　b. 負荷の分布　c. 再分配

図8　P3の再分配による負荷の均衡

```
q  :-  ←(3×s), →t.
r  :-  ←u, →v.
```

この時、各部分問題は割付けられた領域内の適当な1座標点で実行されるとする。このようにして実行すべき問題を部分問題に分割し、各々の部分問題をP3の座標点に割付ける（図8b）。

この様な割付けを行っていくと、共通の祖先を持つ複数の部分問題は、その祖先の領域であった4角形内に含まれることになり、部分問題間の通信の局所性が保たれる。

つぎに物理的なPEによる問題の実行について説明する。初期状態においては、物理的なPEはこの部分問題が割付けられたP3平面をPEの数に応じて、例えば全PEが同じ面積になるように分割し、各PEの分担領域内の座標点に割付けられている部分問題を実行する（図8a，b）。

P3の分割で、物理的に隣接したPE同士がP3の隣り合った部分領域を分担することにより、仮想的なP3平面の上の距離をPE間の距離に対応させることができる。従って、P3の上で近い距離に割付けられた部分問題は、近いPEに割付けられる。

実行が進むことによって生じる各PE間の負荷の不均衡は、各PEが分担するP3領域を変形させることで行われる（図8c）。即ち、負荷の重いPEはその分担領域の面積を縮小させることによって実行する部分問題の数を減らし、負荷の軽いPEはその分担領域の面積を拡大することによって実行する部分問題の数を増す。

この分担領域の変形は、P3上で隣り合った領域を分担したPE同士が分担領域の境界線を移動させることによって行うことができ、境界線の移動の量は隣接するPEの間の負荷の差によって決定されるため、両PEの局所的な情報のみで負荷の均衡が得られる。従って、この局所的な負荷の調整を隣接するPE間で順番に得ていくことにより、全PEの負荷を均等化することができる。

3．3　PE間通信

各PE間で通信に用いられるパケットは、以下の2種類のものがある。

(1)通信相手であるPEが予め分かっているパケットであり、例えば、負荷調整のために隣接するPEとの間で負荷の状況を交換するパケット等である。これらのパケットは、送り先が分かっているので、送り先であるPEの物理的な番号が宛先となる。

(2)通信相手であるPEが分かっていない場合。即ち、ある部分問題を分割し、他のPEに割り付ける場合等のパケットである。部分問題をP3座標に割り付けるパケットは、その座標を受け持っているPEが、負荷の再分配によって変更されている可能性があるために送り先のPE番号を宛先とすることはできない。そのため部分問題の割付けられているP3座標を宛先とする。

これらの通信パケットは幾つかのPEのネットワーク制御部を経由して目的とするP3座標を分担しているPEに転送される。この時に、そのパケットの転送経路に当たるネットワーク制御部は、どの方向（チャネル）にパケットを転送するべきか判定しなければならない。

物理的なPE番号を宛先とするパケットについては、その転送経路を予め設定しておくことができるが、P3座標を宛先とする場合はその宛先を受け持つPEが動的に変更されるため、予め設定することはできない。通信の局所性の観点からは、全PEを監視する様な手法は避けたい。そこで、パケットの通過経路に当たるPEでの転送方向決定を局所的な情報で行うために、以下に示すような方式を用いた。

(1)図9に示すように、各PEのP3上の分担領域rの角の点から領域の各辺に対して垂線を立て、分担領域外を8つに分割する。この時、PEの分担領域の各辺e0～e3は隣接するPEの分担領域との境界であるので、物理的には隣接するPEに接続するチャネルである。

(2)図9でパケットの宛先アドレスであるP3の座標点が分割された外部領域で、PEの分担領域の辺に垂直な部分a0～a3に含まれる場合、そのパケットの宛先アドレスとそのPEとの距離はan（n＝0．．3）に接する辺en（0．．3）上の点で最小となる。従って、その辺に対応するチャネルにパケットを転送することによって、パケットの宛先アドレスと転送先のPEとの距離は、転送元のPEとの距離よりも小さくなる。

(3)図9でパケットの宛先アドレスが分割された外部領域で、PEの分担領域の角に接する部分b0,1～b3,0に含まれる場合、そのPEとパケットの宛先アドレスとの距離は分担領域の角の点で最小となる。従って、その点を共有する2辺の対応する2つのチャネルの何れにパケットを転送しても、パケットの宛先アドレスと転送先のPEとの距離は、転送元のPEとの距離

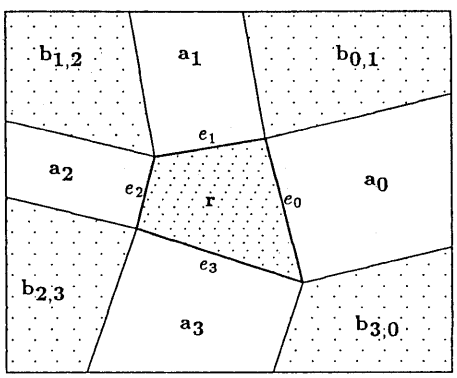


図9　幾何学的なパケット転送方向決定

よりも大きくなることはない。

上記転送ルールに従えば、パケットはＰＥ間の転送によって宛先とするＰ３座標から遠ざかることがなく、かつＰＥ数は有限であるので、いずれは宛先アドレスであるＰ３座標を受け持っているＰＥに辿り着く。

３．４　負荷分散サポート

３．３節で示したパケットの転送方向決定に従えば、パケットは必ず宛先のＰ３座標を分担するＰＥに到着する。しかし、パケットがネットワーク制御部を通過する度にその宛先アドレスから、計算によってパケットの転送先チャネルを求めることは容易でない。そこで、マルチＰＳＩでは、パケットの宛先アドレスによって予め求めた転送先チャネルをＰＴというメモリ上に保持し、パケットを受け取った際にそのメモリを宛先アドレスによって参照し、高速にパケットの転送方向を決定することとした。

ＰＴは１２８×１２８エントリの２次元のメモリであり、図９で示したパケットの転送方向決定のためのＰ３の分割をそのままＰＴの値として保持するものである。図１０に例として、転送方向決定のための分割とチャネル０のためのＰＴの値との対応を示す。

ＰＴ上では、図９で示されたどちらの方向に送っても良い部分ｂn,mは、例えば「直進を優先する」といった戦略によって、どちらか１方向への転送が示される。ＰＥの分担領域にパケットの宛先アドレスがある場合はそのパケットは自ＰＥ宛であるので、ＣＰＵに接続するチャネルに転送する。また、もともとパケットを受け取ったチャネルに転送されてしまうような部分にパケットの宛先アドレスがある場合は、最も単純なループが形成され、送り元のＰＥが上記転送方向決定ルールに従って正常に動作している状態ではありえないので、そのパケットの転送はエラーとする。

図１１にＰＴの値とパケットの転送方向を示す。上記ＰＥの分担領域の形状からパケットの転送方向を決定する方法は、平面上での直線引きと領域の塗り潰しでＰＴを作成することができ、比較的高速に行うことができる。

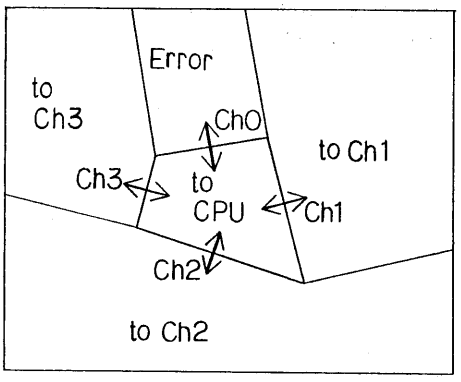


図１０　パス・テーブル

ネットワーク制御部に搭載されているＰＴは、１２８×１２８エントリしかなく実際のＰ３平面はＰＴで示されるものよりも大きい。そのため、パケットの宛先アドレスはＰ３の座標点の上位ビットのみを用いて、実際よりも粗い分解能で表されることになる。従って、ＰＥのＣＰＵに取り込まれたパケットは、各ＰＥ内で分担領域の情報を用いて、実際にそのＰＥの受け取るべきパケットか否かを判定し、隣接するＰＥの受け取るべきパケットである場合は、そのパケットを再送する。

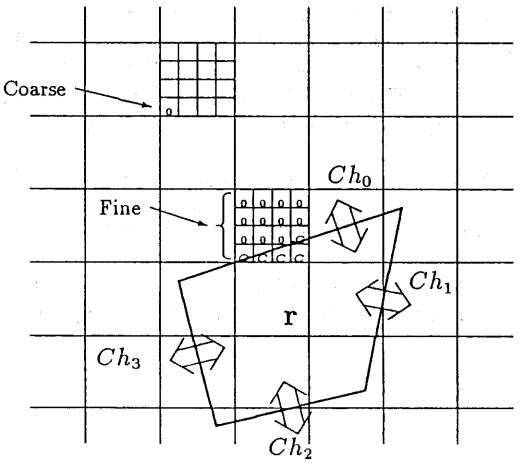
ＰＥ間の負荷調整が行われた場合、ＰＥの分担する領域の形状が変化してしまうので、その度にＰＴを書換えなければならない。ＰＴの書換えは、そのエントリ数が多いためかなりの時間を要する。ＰＴのメンテナンス時間を短縮するために、ＰＴの多段化を行った。即ち、図１２に示すようにＰＥの分担領域の近傍は詳細なＰＴ（Ｆｉｎｅ　ＰＴ）によってパケット転送方向を決定し、分担領域から遠くは粗い分解能のＰＴ（Ｃｏａｒｓｅ　ＰＴ）で表す。

Ｆｉｎｅ　ＰＴとＣｏａｒｓｅ　ＰＴは、実際のハードウェアでは入れ子になっておりＣｏａｒｓｅ　ＰＴの参照アドレスはパケットの宛先アドレスの下位ビットをマスクして表される。

ネットワーク制御部は、パケットを受け取ると、まずパケットの宛先アドレスの下位ビットをマスクしてＣｏａｒｓｅ　ＰＴを参照し、その値を読出す。

bit 2	1	0	
0	×	×	Ｃｏａｒｓｅパス・テーブル
1	×	×	Ｆｉｎｅパス・テーブル
×	0	0	to ＣＰＵ
×	0	1	to Ｃｈ（ｉ＋１）mod4
×	1	0	to Ｃｈ（ｉ＋２）mod4
×	1	1	to Ｃｈ（ｉ＋３）mod4

図１１　パス・テーブルの値とパケット転送方向



0 : *to* Ch_0　C : *to CPU*

図１２　２段パス・テーブル

Ｃｏａｒｓｅ　ＰＴから読出された値の最上位ビットが'１'である場合、宛先が決定できなかったことを示すので、ネットワーク制御部はパケットの宛先アドレスのマスクをはずして再度ＰＴを参照し、パケットの転送方向を決定する。これによって、ＰＴ上で実際のパケット転送に必要なエントリ数を減らすことができ、ＰＴ書き換えに必要な時間の短縮が行われる。

4　おわりに

マルチＰＳＩシステムにおける負荷分散方式と、ハードウェアでのサポートについて述べた。マルチＰＳＩシステムで実験する、Ｐ３を用いた負荷分散方式の特徴をまとめると以下のようになる。

(１)初期設定としてプログラマが大雑把な負荷分布を与え、実行時にシステムが自動的に負荷の再分配を行う。

(２)プログラマは要素プロセッサ数や要素プロセッサの接続形態等の物理的な構成を意識する必要がない。

(３)局所的な負荷分散を繰り返して全要素プロセッサの負荷を均等にするため、プロセッサ間通信のオーバヘッドを軽減できる。

マルチＰＳＩシステムは、現在ハードウェアの開発を終了し、ファームウェア，ソフトウェアの実機テストを行っている。
今後の課題としては

(１)負荷をどのように定義するか。一案として、各部分問題に優先順位を付加し、各要素プロセッサで単位時間に実行された部分問題の優先順位の和を、その要素プロセッサの負荷とする。

(２)負荷調整によるパステーブルの書換え中、パケットの転送をどうするか。パステーブルの書換え中は、その要素プロセッサ経由のパケット転送を停止する方法が考えられるが、この方法はネットワークの転送レートを低下させてしまう。

等があげられる。今後も上記の負荷分散方式の検討評価を継続し、マルチプロセッサ・システムにおける負荷分散方式の確立を行う予定である。

最後に、ご指導ご鞭撻いただいたＩＣＯＴ第４研究室内田俊一室長ならびに関係各位に深謝します。

＜参考文献＞

[1] 内田；並列推論マシン；人工知能学会誌Vol.2 No.4 pp450-458,1987.

[2] 武田　他；マルチＰＳＩ第２版のハードウェア構成；情報処理学会第３５回（昭和６２年後期）全国大会．

[3] K.Taki; The parallel software research and development tool: Multi-PSI system; Proceeding of France-Japan Artificial Intelligence and Computer Science Symposium 86.

[4] H.Nakashima et al; Hardware Architechture of the Sequential Inference Machine: PSI-II Proceeding of 4th Symposium on Logic Programming, 1987.

[5] M.Sato et al; KL1 Execution Model for PIM Cluster with Shared Memory; Proceeding of 4th Symposium on Logic Programing, 1987.

[6] K.Ueda; Guarded Horn Clauses; TR-103, ICOT 1985.

[7] N.Ichiyoshi et al; A Distributed Implementation of Flat GHC on the Multi-PSI; Proceeding of the 4th Symposium on Logic Programming, 1987.

[8] Y.Kimura et al;An Abstruct KL1 Machine and its Instruction Set; Proceeding of the 4th Symposium on Logic Programming, 1987.

[9] T.Chikayama; Load Balancing in Very Large Scale Multi-Processor Systems; ICOT Technical Memorandom TM-0276, 1987.

[10]Y.Takeda et al; A Load Balancing Mechanism for Large Scale Multiprocessor Systems and its Implementation; International Conference on Fifth Generation Computer Systems, 1988. (in preparation)