

TR-672

並列推論マシンPIM/cシステムの開発
—密結合負荷分散支援とファームウェア—

中川 貴之、朝家 真知子、
垂井 俊明 他（日立）

July, 1991

© 1991, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

並列推論マシン PIM/c システムの開発

—— 密結合負荷分散支援とファームウェア ——

中川貴之* 朝家真知子* 垂井俊明* 井門徳安* 杉江衛*

早木茂** 森努** 橋田亨** 今西祐之*** 佐久間良*** 小島丈一***

(* (株) 日立製作所 ** (株) 日立マイコンシステム *** ヒューマンシステム (株))

1. 概要

並列推論マシン PIM/c [1] (Parallel Inference Machine/model c) は並列論理型言語 KL1 [2] (Kernel Language 1) を高速に実行するために試作を進めている、256 プロセッサからなる並列マシンである。PIM/c のハードウェア構成は図1のように、スヌーピングキャッシュを導入した (メモリ共有) 密結合マルチと、クロスバネットワークによる (メモリ非共有) 疎結合マルチの2階層からなり、これを使い分けるように、マシン語である KL1 B [3] (KL1-Base) から考えられている。

本発表では、ハードウェアとソフトウェアの協調的な開発手法の事例として、特に、自動負

荷分散処理の高速化と、並列処理の実験を行なうための大規模ファームウェアの開発手法について報告する。

まず、PIM/c ハードウェアの特徴の一つである、KL1 ソフトウェア割り込み処理を支援するスリットチェックレジスタの、密結合マルチ自動負荷分散への導入効果について考察する。

また、このようなハードウェアの特徴を活用するにあたり、負荷分散処理のみならず通常処理の性能を引き出すことが、並列処理の隘路を見いだすために重要である。水平型マイクロマシンである PIM/c ハードウェア上に、KL1 B マシンを構成するための、大規模なマイクロプログラムの効率的な開発手法と、高速化の

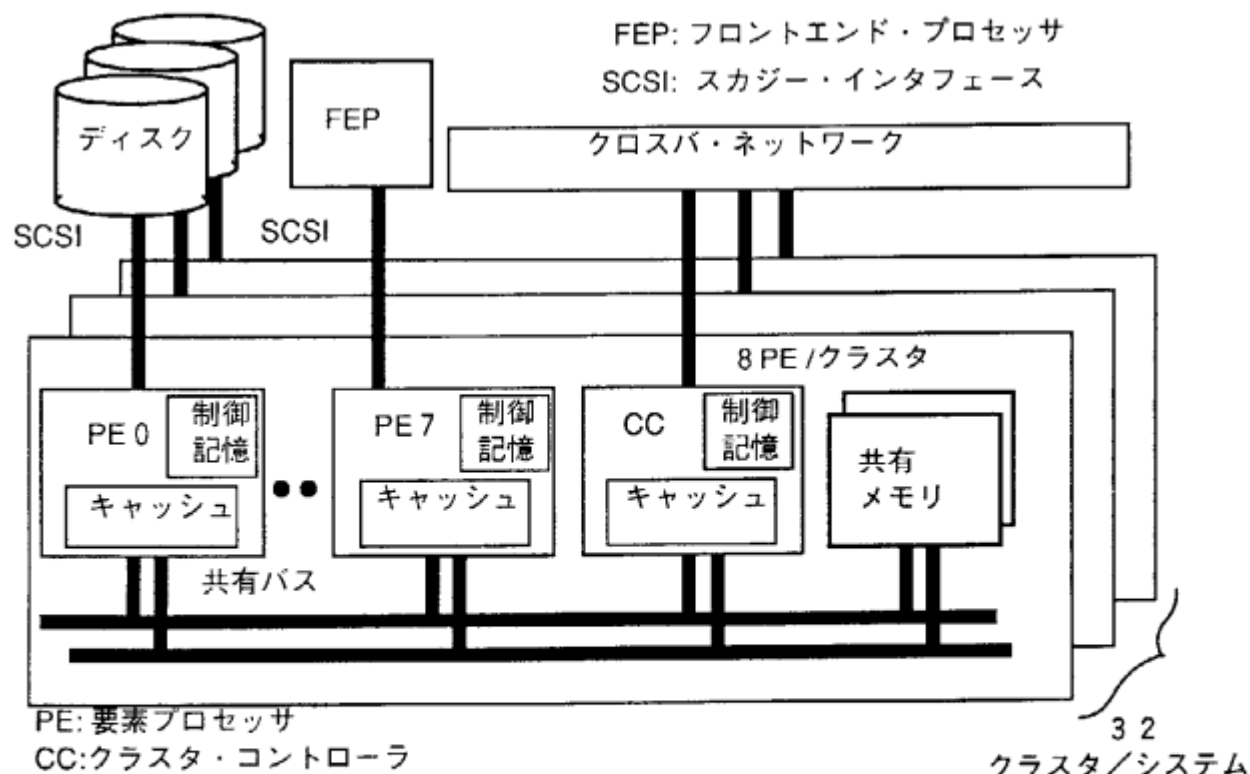


図1 PIM/c ハードウェアの構成

効果について考察する。

2. PIM/cハードウェアの特徴

PIM/cハードウェアはKL1言語で記述したプログラムを高速に実行することを目的として、開発した。PIM/cハードウェアは以下の特徴を有する。

1) 水平型マイクロアーキテクチャ

1語が104ビットからなる水平型マイクロ命令を50nsピッチで実行する。これにより分岐コストを0にできる上、レジスタ間操作とメモリアクセスおよび演算操作をオーバーラップして処理可能である。PIM/cはKL1言語の仕様とKL1専用マシンの並行開発という課題に対して、RISCによるコンパイラ主導のアプローチをとらず、多機能命令を使った、命令フェッチ回数の低減による高速化の可能性を与えている。

2) KL1にチューニングしたスヌーピングキャッシュ

共有メモリアクセスにおけるレスポンスタイムを短縮するために、可能な限りプロセッサボード上でアクセスを終了するストアバックキャッシュを実現した。共有バスのボトルネックを回避するために、KL1処理系のメモリアクセス履歴のシミュレーション結果に基づき、共有バスの使用時間が最短になるように、キャッシュプロトコルおよびセット数、キャッシュブロック長を採用した[4]。KL1プログラムの処理は、単一代入型の言語による細粒度並列処理の特徴として、書き込み率が高く、キャッシュブロックを共有する比率も高いので、密結合マルチのプロセッサ台数は10台程度を上限としている。

3) 共有バスネックを回避する2ウェイインタリーブバス

共有バススループットのボトルネックを回避し、併せて、バス競合によるレスポンスタイム

の悪化を抑えるために、PIM/cでは、共有バスおよび共有メモリをアドレスにより2ウェイにインタリーブした構成をとっている。

4) 高速なクロスバネットワーク

大規模な疎結合システムを構成するには、高スループットのネットワークが不可欠である。PIM/cでは、8台の密結合プロセッサにつき1台のクラスタコントローラを置き、更に段階的なレスポンスタイムの短縮を目指して、物理的な距離の近い同一筐体内の64台までは、クロスバスイッチ1段でメッセージ交換可能な構成を採用している。

5) 負荷分散支援スリットチェックレジスタ

密結合に於ける細粒度負荷分散のレスポンスタイムを可能な限り短縮するために、図2のように、負荷分散要求を早いもの勝ちで受け付ける、匿名通信とよぶ通信方式を高速化するレジスタSCR(Slit-Check-Register)を設けた[5]。

匿名通信は、図3のように、SCRを放送書き込みにより一斉にセットし、最初に要求をチェックしたプロセッサが一斉にリセットすることにより実現している。

スリットチェックレジスタを用いた匿名通信によれば、通常のリダクション処理の粒度を200マシンサイクルとして、相手プロセッサを指定する”狙い撃ち方式”では100サイクルを要する応答時間を、1/8の12サイクル程度に短縮する見込みである。



図2 匿名通信方式

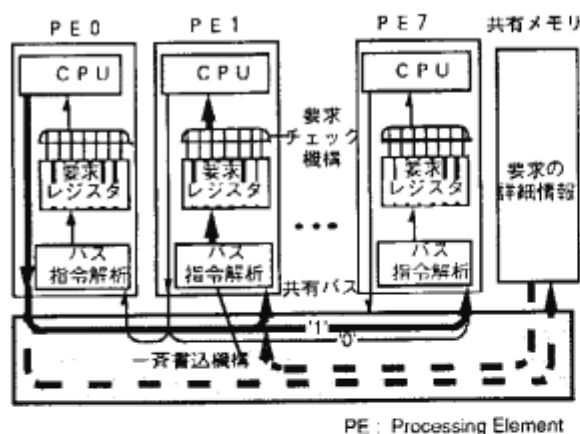


図3 密結合動的負荷分散支援ハードウェア

6) 自動負荷分散支援ネットワークレジスタ

疎結合に於ける負荷分散にあたり、正確な情報を、プロセッサに割り込まないで伝えるために、図4のように、クラスタ内の未実行の負荷量をネットワーク上に格納する負荷値レジスタCLR(Cluster-Load-Register)を設けた[6]。負荷値レジスタには最大値/最小値の検出回路をも設けてあるので複数の負荷分散方式の比較評価が可能である。

とくに、このハードウェアによる評価項目として、スマートランダム方式[7]による負荷分散アルゴリズムの評価を行なう予定である。

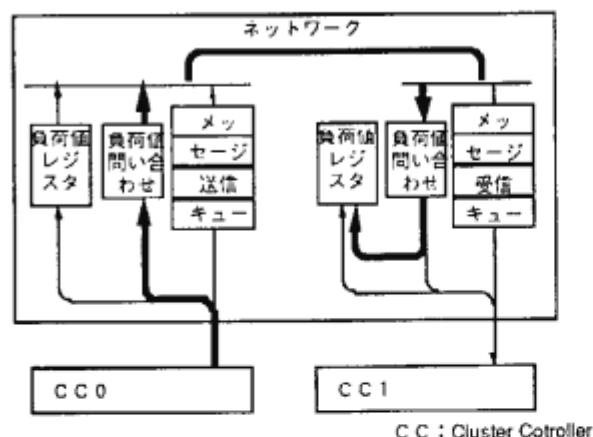


図4 疎結合動的負荷分散支援ハードウェア

3. スリットチェックレジスタの効果予測

KL1のソフトウェア割り込みは、PIMのハードウェアとKL1処理系のインタフェースである。KL1の処理では、コンテキストスイ

ッチの軽量化のために、処理を中断させることはせず、自律的に割り出す。このような割り込み処理のレスポンスを短縮するために、図3に示すように、自動負荷分散では全てのプロセッサに負荷要求を割り込みとして発行し、一番早く処理の切れ目に到達したプロセッサで、割り込みハンドラが負荷を分配する。このために、PIM/cに放送書き込み機能と高速なイベント検出機能を持つ1ビット幅のレジスタ群としてスリットチェックレジスタを実装している。

スリットチェックレジスタを用いた場合と、共有メモリのポーリングにより匿名通信を処理した場合との比較を表1に示す。PIM/cが採用する無効化タイプのキャッシュは、通常のローカリティのあるメモリアクセスに適していることが、シミュレーションにより立証されているが、イベント検出時に共有バスをアクセスすることによりレスポンスが悪いうえ、全てのキャッシュがミスヒットしてブロック転送を行なうので、システム性能の低下を引き起こす。

表1 匿名通信ハードウェアによる高速化

方式	レスポンス	共有バスアクセス数
キャッシュ	9サイクル	8回
SCR	1サイクル	2回

4. ファームウェア開発のキーポイント

上記のように、PIMは密結合と疎結合の両方の処理を実現する必要がある。しかも、並列処理の実現性を立証するには、ファームウェア上に、コンパイラやオペレーティングシステムPIMOS[8]を含む実用的なプログラミング環境を提供することが重要である。そこで効率的な実装方式が要求された。

また、並列処理の負荷分散は新たな負荷となるので、支援ハードウェアを活用して、処理のレスポンスを短く保ち、正確な情報を交換しなければ評価の有効性を失う。

KL1B命令の仕様はICOTが開発したV

PIM[9](Virtual PIM)として供給されている。VPIMは実時間ガーベジコレクションや変数の同期処理、変数のコピー管理機能を持つため、ロード/ストア命令に換算して33Kstepをこえる大規模なものである。このような大規模なシステムの構築にあたり、並列システムの実験を仕様レベルで変更容易に管理するために、ファームウェアの開発では、以下のポイントに着目している。

- 1) ハードウェア資源の有効利用
自動負荷分散支援ハードウェア、汎用レジスタ、レジスタファイル等を用途に応じて使い分ける。
- 2) VPIM変更によるチューニングの容易性
仕様レベルでの変更が容易にできて、結果を迅速に得られること。
- 3) マイクロの性能/規模の管理の容易性
実装可能な制御記憶容量を越えないためのスタティクステップ数と、ダイナミックステップ数のキーパラメタを容易に把握できる開発体制。
- 4) VPIMに付属する開発環境ツールの流用
仕様変更における単純なバグの検出が自動的に得られること。

5. 水平型マイクロマシンの性能上の隘路

PIM/cのプロセッサ[10]はタグアーキテクチャを備えた水平型マイクロマシンである。1語が104ビットからなるマイクロ命令の1ステップには、以下の処理を指定でき、処理の終了までに以下の時間を要する。

- 1) 2方向分岐処理 (1ステップ)
- 2) 多方向分岐 (2ステップ)
- 3) 定数アクセス (1ステップ)
- 4) レジスタアクセス (1ステップ)
8語のレジスタが2バンク並列にアクセス可能
- 5) レジスタファイルアクセス (2ステップ)

8語のレジスタファイルが2バンク並列にアクセス可能

6) メモリアクセス (3ステップ)

7) 演算処理 (1ステップ)

加減算とシフト演算、インクリメント演算が並列に実行可能

PIM/cでは、図5に示す多数のハードウェア資源を衝突のないように並列に活用しなければ、プロセッサ性能を引き出せず、従って、並列マシンの隘路を引き出す実験システムとは成らない。

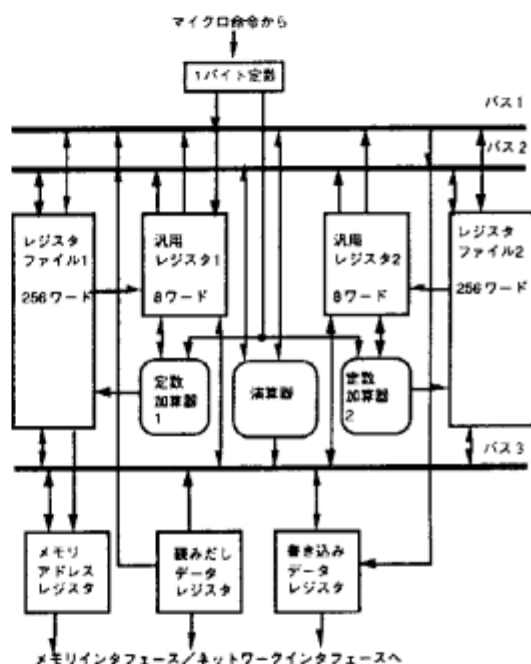


図5 PIM/cのプロセッサを構成するハードウェア資源

6. 大規模マイクロプログラムの一開発手法

以上の経緯により、PIM/cでは、マイクロプログラム第1版を自動生成ツール群を用いた自動生成により作成し、第2版を最適化ツール群を併用した半自動生成により作成する、多段階の開発手法を用いている。

マイクロプログラムの開発は、ハードウェア資源の割り付け、マイクロ命令への分解、マイ

クロプログラムのデバッグの、3段階からなる。PIM/cでは、UNIXのフィルタを用いて、機能毎の中間出力を可視化して、デバッグや性能管理を効率化している。

マイクロプログラム第1版の開発手順を図6に示す。

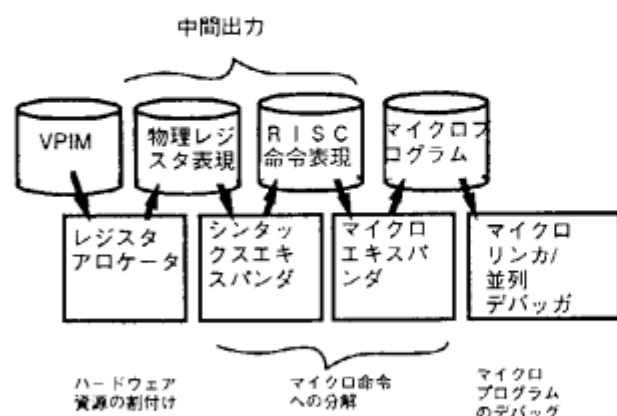


図6 マイクロプログラム第1版の開発手順

第1版に対して第2版では、性能の隘路事項を見定め、自動生成の可能性を検証するために、図7に示すように、マイクロプログラム生成の各段階に、性能上の最適化ツールと人手最適化を加えた。

マイクロプログラム第2版の開発手順は以下のとおりである。

- 1) 人手によるVPIMの最適化
- 2) レジスタおよびレジスタファイルの割付
- 3) レジスタファイルアクセスの削減
- 4) ロード/ストアレベル命令のマイクロテンプレートによる展開
- 5) 分岐処理の水平化
- 6) マイクロデバッガ上のデバッグ

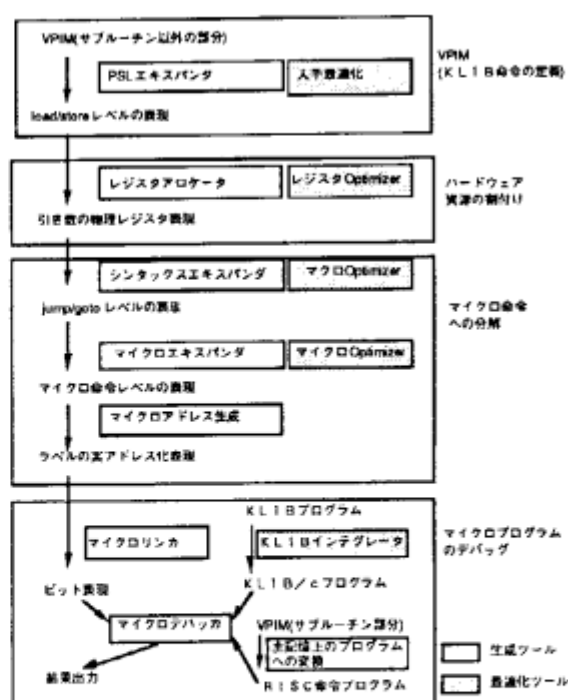


図7 マイクロプログラム第2版の開発手順

並列処理の研究には極限的な性能を引き出さねば、真の隘路は知ることができないことから、PIM/cでは、表3に示すように、自動生成したマイクロプログラムでもappendと呼ばれるプログラム実行で、目標性能の達成が可能であることを示した。

高速化項目では、特に、KL1B命令種類による多方向分岐テーブルの処理に命令本体の処理をオーバーラップさせる命令ディスパッチの高速化と、自動水平化ツールによる分岐処理の高速化の効果が大きかった。

表2 マイクロプログラムの高速化効果内訳

#	高速化項目	効果(μ Step)
0	第1版のステップ数	416
1	ディスパッチの高速化	-144
2	分岐処理の高速化	-64
3	KL1B命令の粒度の向上	-36
4	VPIMの人手最適化	-32
5	レジスタファイルアクセスの削減	-29
6	マイクロテンプレートの最適化	-11

7. KL1B命令の統合による性能向上

水平型マイクロ命令によるプロセッサ性能は、命令フェッチ処理と命令実行処理が逐次実行されるので、マイクロ命令の粒度によって決まる。KL1コンパイラは、論理的に不要な処理を削減する最適化に専念しているため、表2の項目3に示す命令統合ツールを用いた最適化が有効である。このことは、仕様レベルでの変更がこのシステム開発にとって、特に有効であることを示している。

この統合により派生したKL1BのスーパーセットをKL1B/cと呼んでいる。

8. おわりに

本稿では並列推論マシンPIM/cの、とくに自動負荷分散を目的とした、ハードウェア支援機能を紹介し、並列計算機実験システムとしての要求に柔軟に対処するためのファームウェア開発方式を提案した。

実験システムでは、追加処理のレスポンスを短く保ち、性能上の隘路を見極めることが重要であるので、水平型マイクロ命令アーキテクチャには、KL1B命令の処理の粒度を高める変更が有効であることを示した。

PIM/cのハードウェアは、現在16プロセッサの部分試作機が稼働中で、マイクロプログラム第3版のKL1処理系の実装作業を実施中である。今年度に製造する256台のシステムでは、オペレーティングシステムPIMOSを実装し、大規模なアプリケーションを実行可能とすべく作業中である。

今後の課題として、本報告による開発方式で開発したファームウェア上で、複数の負荷分散アルゴリズムを使って、実装した支援ハードウェアの効果を評価していく予定である。

9. 謝辞

本システムの開発にあたり、助言をいただいた

たICOTの瀧第1研究室長、平田主任研究員の両氏に感謝したい。なお、本研究はICOTからの委託研究の一環として、実施された。

10. 参考文献

- 1) Goto et al; "Overview of the Parallel Inference Machine Architecture(PIM)";Proc of the FGCS vol1 1988
- 2) Ueda et al; "Guarded Horn Clauses", TR209, ICOT, 1985
- 3) Kimura et al; "An Abstract KL1 Machine and its Instruction Set";Proc of the 1987 Symposium on Logic Programming
- 4) 松本ほか、"KL1のメモリ参照特性に適した並列キャッシュ機構";データフローワークショップ;1987-10
- 5) 中川ほか、"プロセッサ間ソフトウェア割り込み処理を高速化するスリットチェック機構";計算機アーキテクチャ研究会、1989-7
- 6) 井門ほか、"並列推論マシンPIM/c—負荷分散支援機構—、情報処理学会第40回全国大会、2L-4
- 7) 杉江ほか、"Load Dispatching Strategy on Parallel Inference Machines";Proc. of FGCS Vol.3 1988
- 8) Chikayama et al; "Overview of the Parallel Inference Machine Operating System(PIMOS)";Proc of the FGCS vol1 1988
- 9) 山本ほか、"並列推論マシンPIMにおける抽象機械語KL1Bの実装—高級機械語を実装するための道具立て— 並列処理に関する指宿シンポジウム、1989
- 10) 田中ほか、"並列推論マシンPIM/c—CPUについて—、情報処理学会第40回全国大会、2L-2