

並列推論マシン PIM/p のシミュレータ

三露哲司¹, 細井聡², 武井則雄¹

1: 富士通ソーシャルサイエンスラボラトリ 2: 富士通

1 はじめに

現在 ICOT では並列推論マシン PIM(Parallel Inference Machine)を開発している[3]。本報告では、PIMのモデルの一つであるPIM/pのシミュレータに関して、ポートサーバの導入によって得られた柔軟なシステム構成の実現方式を中心に報告する。PIM/pハードウェアは現在開発中であるため、UNIXワークステーション等の従来のマシン上で動作するシミュレータが必要である。本シミュレータは、PIMのシステムソフトウェアのテストや評価を行うために、PIM/pのアーキテクチャレベルのシミュレーションを行う。

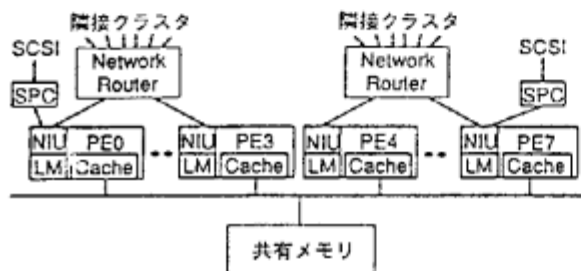


図1: クラスタの構成

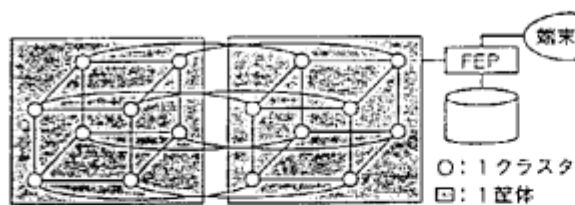


図2: PIM/p システム構成

2 PIM/p のアーキテクチャ

PIM/pは8台の要素プロセッサ(PE)と共有メモリによってクラスタを構成する(図1)[1]。

各PEはネットワークインタフェースユニット(NIU)を持つ。4つのNIUがネットワーク上の1つのノードに

接続される。このネットワーク上のノードはハイパーキューブ状に接続され、ルータの役割を果たす[2]。また、各クラスタ内の2台のPEは入出力インタフェースとしてSCSIプロトコルコントローラ(SPC)を持つ。PIM/pはSPCを介してSCSIバスによりユーザ端末やディスク等と接続される(図2)。

3 柔軟なシミュレータの実現方式

3.1 柔軟なプロセス構成

並列システムソフトウェアのデバッグや評価を行うために、本シミュレータでは、各ハードウェアチップの仕様をできるだけ忠実にシミュレートする必要があった。そこで、本シミュレータはアーキテクチャの構成要素に対応したマルチプロセスで構成することにした。具体的には、1つのプロセスで8台のPEをマシン命令レベルでシミュレートするPEプロセス、1つのプロセスで4台のNIUと1台のRTRをシミュレートするNIU=RTRプロセス、SPCプロセス、SCSIプロセス、FEPプロセスの5種類のプロセス構成とし、4.3BSD UNIXのネットワークI/Oであるソケット[4]を各プロセス間の通信に使用した。

また、ユーザが自分に必要なプロセスだけを使用するためと、マシン負荷を最小限にするために、様々な構成が可能な柔軟な作りにした。具体的には、シングルクラスタ構成でSCSIを使用しない時は、PEプロセスだけを立ち上げる。シングルPE、マルチPEの指定は、PEプロセス立ち上げ時に陽に行う。マルチクラスタ構成で使用する時は、複数のPEプロセスと各PEプロセスに接続するNIU=RTRプロセスを立ち上げる。SCSIを使用する時は、PEプロセス、NIU=RTRプロセス、SPCプロセス、SCSIプロセス、および、FEPプロセスを立ち上げる。

3.2 柔軟な使用とプロセス間通信の問題

開発効率を考慮すると、複数人がデバッグを行う必要があるので、複数台のシミュレータを同時に使用できる必要がある。同一システムを複数台使用する様なシステムでは、サーバを一つ用意し、ユーザからの要望に応じてサーバがプロセスをフォークし、ユーザはその新たに生成されたプロセスを使用する方法が一般的である。しかし、1台のシミュレータを構成する各プロセスのユーザが異なっても良いことが望まれたため、この方法は採用

Simulator of Parallel Inference Machine PIM/p
Tetsuji MITSUYU¹, Akira HOSOI², Norio TAKEI¹
1: Fujitsu Social Science Laboratory Ltd.
2: Fujitsu Ltd.

できない。この要望を満たすためには、1台のシミュレータを構成する各プロセスをその各プロセスのユーザが別々に立ち上げる必要があるからである。しかし、各プロセスを別々に立ち上げた時には、各プロセスの通信相手のポート番号を知る方法が問題となる。各プロセスのプロセス間通信をUNIXドメインとすれば、これはポート番号にユニークな名前を付ける方法の問題となり簡単に解決する。しかし、本シミュレータでは、UNIXドメインとせず、Internetドメインとした。これは、各ユーザがインターネット上の異なるマシンを使用しているためと、マルチクラスタ等の大規模な構成で複数台実行すると、マシン負荷が大きくなるので、異なるマシン上で動作可能な実現方式とするためである。

3.3 ポートサーバ

さて、Internetドメインでは、ポート番号を陽に指定する必要がある。しかし、シミュレータが固定的にポート番号を指定するようなプログラムであると、複数台のシミュレータを同時に起動した場合、ポート番号の衝突が生じる。したがって、これを避けるために、各プロセスが実行時に動的に未使用ポートを探索してそのポートを使用する必要があるが、これでは通信相手のポート番号が分からない。そこで、ポート番号を管理するプロセスとしてポートサーバを導入した。各プロセスは自分のデータとして、ユーザID、プロセスID、認識番号、ホスト名をポートサーバに登録し、通信相手のデータとして、プロセスID、認識番号、接続番号、ポート番号に登録する。ポートサーバはこれらデータの登録、参照、削除の機能を持つ。各プロセスの通信相手のポート番号は、各プロセスがポートサーバに問い合わせることにより、知ることができる。

4 並列システムのデバッグ支援機能の実現

前章では本シミュレータの柔軟な作りに関して述べたが、並列システム特有のタイミングバグの再現性確保のために、柔軟性を抑えた部分もある。本章では、シミュレータのデバッグ支援機能の中でも特に重要なタイミングバグの再現を中心に述べる。

並列システムソフトのテストでは、タイミングバグの解析のためにも、システムソフトの変更無しにそのソフトの実行を制御できることが望まれる。したがって、各プロセスのユーザインタフェースにブレーク等のコマンドを用意した。しかし、マルチプロセス構成のため、あるプロセスに対する実行制御が他プロセスの実行に影響を及ぼす場合がある。そこで、各プロセス間に割り込み機構を持たせ、コントロールメッセージ通信によって制御内容を知らせることにした。だが、これだけでは次のような場合に不十分である。例えば、あるプロセスのレ

ジスタにアクセスブレークを掛けている時に、プロセス間通信によって他プロセスがそのレジスタへデータを書き込むと、書き込まれたプロセスはブレークするが、書き込みを行ったプロセスは実行を続けてしまう。この場合、データ書き込みが起こった時点で、両プロセスが実行を一時中断することが望ましいが、前述の割り込み機構では実現できない。そこで、各プロセス間のデータ通信にはackを返すプロトコルを採用した。これにより、データ書き込みを実行したプロセスもackが返ってくるまで実行を一時中断することが可能となる。

クラスタ内で発生するタイミングバグを再現できるように、PEプロセスの実行PEの切り替えは、

あるPEが一定命令数を実行すると次のPEに実行制御を移すことで実現する。さらに、切り替え命令数を任意の値に設定できるようにしたので、様々なタイミングを生成できる。前章で述べたようなシミュレータの柔軟性を考慮して、PEプロセスには、PE台数と各PE番号指定の機能を持たせた。

5 まとめ

並列推論マシンPIM/pのシステムソフトの開発には、ハードウェアチップの仕様をシミュレートし、柔軟なシステム構成が可能なシミュレータが求められる。この要求を満たすため、アーキテクチャの構成要素に対応したプロセス構成とし、ポートサーバを導入することで、この要求が満たされることを確認した。さらに、並列システムのデバッグ支援のための機能として、実行PEの切り替えをシミュレータが行うことで、クラスタ内タイミングバグの検出に有効な方法を採用した。今後は、評価用の機能を充実させていく予定である。

謝辞

日頃、御指導御助言を頂く、龍和男室長をはじめとするICOT第1研究室PIM開発グループの皆様、ならびに富士通のPIM開発グループの皆様に感謝致します。

参考文献

- [1] 服部彰, 他: 並列型推論マシンPIM/pのアーキテクチャ, 情報学会論文誌, 30(12), 1989.
- [2] 久門耕一, 他: 並列推論マシンPIM/pのネットワーク, 第37回情報大全, 5N-3, 1988.
- [3] A.Goto, et al. Overview of the Parallel Inference Machine Architecture (PIM). In Proc. of FGCS'88, Tokyo, Japan, November.1988.
- [4] W.Richard Stevens. UNIX Network Programming Prentice Hall, Englewood Cliffs, N.J., 1990 ISBN 0-13-949876-1