

マルチPSIにおける接続ネットワークの評価

益田嘉直 岩山洋明 石塚裕一 末原義人 瀧和男
(三菱電機株式会社) (ICOT)

1. はじめに

第五世代コンピュータ・プロジェクトにおける並列推論マシンの研究開発において、並列ソフトウェアの重要性が認識されてきている。そのため、ICOTでは逐次型推論マシンPSIを複数台接続したマルチPSIシステムを第1版と第2版の2段階に分けて開発中である[1]。本稿では、現在稼働を開始しているマルチPSI第1版(PSIを要素プロセッサとして6台格子状に接続)における接続ネットワークの評価について述べる。

2. ネットワーク構成

マルチPSIは並列推論マシンに係わるソフトウェア上の研究を進めるための研究開発用マシンであり、記述言語として並列論理型言語KL1(様言語第1版)を使用する[2]。KL1は実行メカニズムの一部としてネットワーク通信をサポートする。KL1処理系に接続するネットワーク構成は図1のような階層構造となっている。第1版では処理系やNetwork HandlerはESPで書かれているが、第2版ではファームウェア化の予定である。

図1におけるNetwork Handlerの主な役割はデータ変換であり、処理系から渡されたメッセージ(オーダ)をパケット(バイト・ストリング)に成形しHandler Kernelに渡す。また、その逆の変換処理を行うことである[3],[4]。Handler Kernelでは送受信の組込述語(Firmware)が用意されており、パケットの送受信が自動的に行われる。処理系から渡されるメッセージは具体的にはthrow_goal, unify, read等の通信オーダ(全部で14種類有)となってパケットにうめ込まれネットワークを行き来する。ここでthrow_goalは負荷分散処理などの結果、他PE(要素プロセッサ)にゴールを投げる時に用いられunifyは他PEに対する単一化要求である。readは他PEに対する値の問い合わせに用いられる。図2にネットワーク評価のために用いたベンチマーク・プログラムの例を示す。本プログラムを実行させた時のメッセージ交換の模様を図3に示す(正確にはtt3a(X)というゴールを1回投げた時の通信記録)。

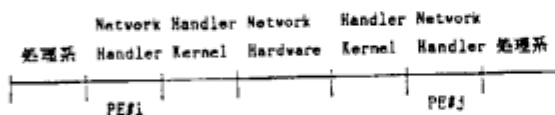


図1. ネットワークの階層構造

Network Evaluation of the Multi-PSI System
K.Masuda, H.Iwayama, H.Ishizuka, Y.Suehara, K.Taki
(Mitsubishi Electric Corporation) (ICOT)

3. 評価目標と測定方法

今回のネットワーク評価においては図1におけるNetwork Handlerから内側の階層部分の評価を中心とし、各階層での通信コスト(実行時間)を明確にし、更に「PE間通信処理対PE内処理」の性能比較を行った。この比の値はKL1プログラムを記述する場合にPE内にどの程度処理をとり込み、PE間通信を減少させる必要があるかを検討する場合に重要な値となる。

評価の方法には、各階層における基本機能、基本命令を個々に調査し、分析評価する方法と典型的なベンチマークプログラムを用いて総合的な処理性能を測定し分析評価する方法が考えられる。ここでは後者の方法による評価を中心とし、具体的にはHandler, Handler Kernel, Hardwareの各階層における通信コストを求めるために、図2の様なベンチマーク・プログラムを数種類用意し、実行時間の測定を行った。すなわち、throw_goal, unify, readといったPE間通信オーダのコストを引数タイプ及び引数値を変化させ計測した。測定に当たってはそれぞれ100回同じプログラムをループして実行し、その平均値を算出することでOSのサポートするタイマの誤差を軽減した。

```
test3_la(0) :- true | true.
test3_la(N) :- N1 := N - 1 | wait3_la(R, N1),
               call(bench_mark@t3_la,R,_).
wait3_la(success,N) :- true | test3_la(N).
t3_la :- true | alloc(5)@tt3a(X).
tt3a(X) :- true | X=atom.
```

図2. ベンチマーク・プログラム(unifyテスト)の例

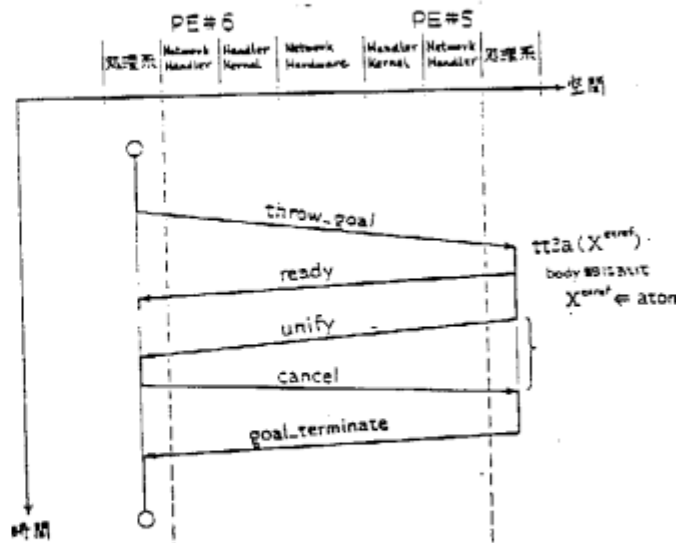


図3. メッセージ(通信オーダ)交換の概念図

4. 測定結果と考察

PE間通信オーダの中から throw_goal, unify, readを選択し、それらの通信コストを調べるために引数タイプ(atom, structure, undefined variable, integer, list)及び引数個数(1,2,5)の異なるベンチマーク・プログラムを実行させて得られた結果が図4である。図4における実行時間は100回ループの1回当たりの平均実行時間であり、pragmaをつけてゴールを他PEに投げる場合の実行時間(pragaa 有)とpragmaをつけずにゴールを自PE内で処理する場合の実行時間(pragaa 無)が示されている。ここで、pragma有の実行時間とpragma無の実行時間との差がPE間通信コストである。

ベンチマーク・プログラムではゴールを他PEに投げると1回のreductionが行われるので、「PE間通信処理 対 PE内処理」の処理時間比は以下の式より求められる。

$$\text{処理時間比} = (\text{pragma有の時間} - \text{pragma無の時間} + \text{reduction 時間}) / \text{reduction 時間}$$

但し、reductionにはtail recursiveにレジスタ上で処理が行われるものと、ゴールが一且Ready Queueにつながられてから処理されるものがあるが、ここでは後者のreduction時間(測定結果:2.25msec)を用いている。各階層における通信コストに関しては、throw_goal, unify, read等のpacketのサイズが引数のタイプにより固定しているので、Firmware(Handler Kernel)レベルの通信コストの算出が可能であり、その結果Handlerの通信コストも算出できる(表1)。

Handlerの通信コスト(H)は本来Handlerからみた通信コスト(A)からFirmwareの通信コスト(F)およびHardwareの通信コストを差し引いたものである。ここでHardwareの通信コストは環境により変動するが、ほぼFirmwareの通信コスト(F)の1/2程度である。従って、FirmwareおよびHardwareの通信コストはHandlerからみた通信コスト(A)に比べて測定誤差範囲でありHandlerの通信コスト(H)はほぼHandlerからみた通信コスト(A)とみなせる。すなわち第1版ではHandlerからみるとHandlerの実行時間(主にデータ変換)の占める割合はFirmware, Hardwareに比べて非常に高い。

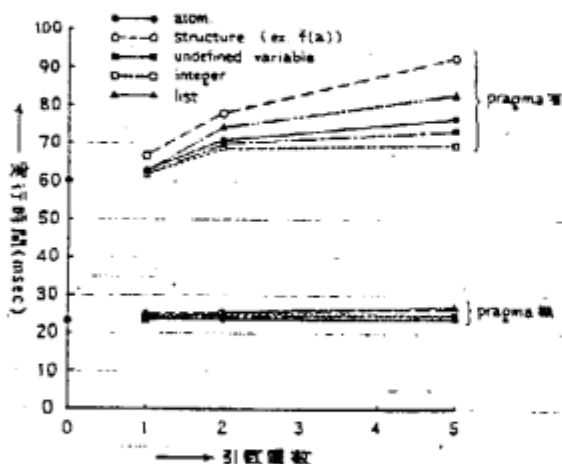


図4. 引数個数と実行時間の関係

第2版では処理系,Handlerはファームウェア化され、Network Hardwareも高速化されるけれども、Handler Kernel(Firmware)については第1版とほぼ同様の実行時間を要すると予想される。この時ネットワーク全体の通信コストは小さくなり性能向上が図られると同時に、Handlerの通信コストの占める割合は減少する。

次に、第2版における処理時間比について考察する。表1の最上段のケースを例にとり、仮に処理系,Handlerが100倍Hardwareが5倍高速化されると処理時間比は18.2倍が約28倍になる。すなわち第2版の「PE間通信処理 対 PE内処理」の処理時間比は、第1版と比べて多少増加すると予想される。この事は、3章で述べたPE間通信を減少させる必要性がそれだけ増すことを意味する。

5. おわりに

現在ICOTにおいてマルチPSI第1版上でKL1処理系の開発及びテストを行っているが、今後も評価を継続しその結果をもとにマルチPSI第2版のネットワーク並びにシステムの検討、開発を進める予定である。最後に、ご指導ご鞭撻をいただいたICOT第4研究室内田俊一室長に感謝する。

<参考文献>

- [1] 瀬和男、"並列ソフトウェア開発用マシン：マルチPSIシステム" 第32回情報処全国大会、昭和61年3月
- [2] 宮崎敏彦、他、"Multi-PSIにおけるFlat GHCの実現方式" LPC'86 pp.83~92、1986年6月 JAPAN
- [3] 木村康則、他、"マルチPSIシステムとその接続方式" 第33回情報処全国大会、昭和61年10月
- [4] 益田喜直、他、"マルチPSIのネットワーク・ハードウェア構成" 第33回情報処全国大会、昭和61年10月

表1. 測定結果一覧表 (単位 msec)

測定項目	引数 (引数タイプ)	実行時間		通信コスト(A)	F/Wの	処理時間比	
		pragaa有	pragaa無	(Handler-FW)	通信コスト (F)	(Handlerから見た通信コスト)	
throw_goal (含 ready, goal, terminate)	1	atom	62.39	23.65	38.74	0.22	18.2
		undef	61.97	24.08	37.89	0.22	17.8
		integer	61.71	24.50	37.21	0.21	17.5
		structure	66.69	25.18	41.51	0.23	19.4
		list(1)	62.78	23.18	39.60	0.23	18.6
	2	list(2)	63.84	23.91	39.93	0.23	18.7
		list(5)	66.72	25.16	41.56	0.25	19.5
		atom	70.71	23.73	46.98	0.23	21.9
		undef	69.60	24.21	45.39	0.24	21.2
		integer	68.48	24.59	43.89	0.22	20.5
	5	structure	78.09	25.48	52.61	0.25	24.4
		list(1)	74.24	25.89	48.35	0.25	22.5
		atom	76.66	23.84	52.82	0.27	24.5
		undef	73.47	24.37	49.10	0.28	22.8
		integer	69.64	24.68	44.96	0.23	21.0
unify (含 cancel)	1	structure	92.71	26.26	66.45	0.31	30.5
		list(1)	82.95	26.70	56.25	0.31	26.0
		atom	87.99	28.34	21.76	0.12	—
		undef	88.48	28.78	21.81	0.13	
		integer	88.99	29.21	21.89	0.11	
structure	89.46	29.66	21.91	0.13			
read (含 read, answer)	1	list(1)	87.18	27.53	21.76	0.13	—
		atom	101.44	70.45	30.99	0.11	
		undef	—	—	—	—	
		integer	100.31	71.10	29.21	0.10	
		structure	106.13	72.03	34.10	0.12	
list(1)	101.11	69.15	31.96	0.12	—		