PVP-SW とハイパクロスバ・ネットワークを用いた計算機の評価*

2L－1

板倉 憲一、廣野 哲、朴 泰祐、中村 宏、中澤 喜三郎†
筑波大学 電子・情報工学系‡
{itakura,hirono,taisuke,nakamura,nakazawa}@arch.is.tsukuba.ac.jp

1. はじめに

ノードプロセッサ (以下 PU) に PVP-SW [1] を用い、PU 間結合網にハイパクロスバ・ネットワーク (以下 HXB) [2] を用いた、並列計算機の性能評価をシミュレータを用いて行う。評価用問題として NAS Parallel Benchmark [3] の中から FFT(高速フーリエ変換) を用いて偏微分方程式を解く問題 (以下 FT) を用いる。

2. 評価対象の並列計算機

HXB [2] は多段のクロスバスイッチを用いた間接網であるが、各種の通信パターンに対応し、バタフライ転送や後で述べる転置転送を無衝突で転送できる [4]。

対象とする並列計算機は分散メモリ型で、メッセージパッシング方式を用いる。データの送受信は DMA によって行なわれ、1 回の転送は連続もしくは 1 定のストライドを持ったメモリ領域に対してのみ行なうことにする。

PU はメモリレイテンシを隠蔽するため、レジスタへの prefetch 機能を持った PVP-SW(Pseudo Vector Processor based on Slide-Windowed Registers) 方式 [1] を仮定する。PVP-SW によりベクトル処理が可能になるが、今回は転送を行なうことによってキャッシュが有効に働かなくなるデータについてのみベクトル化を行う。

プロセッサの処理の評価には別に開発したシミュレータを用いる。これは PA-RISC の命令セットに PVP-SW の preload 命令などを追加した命令セットをシミュレートし、実行クロック数などを計測することができる。

今回の評価での基本的な性能値は、PU の clock を 100MHz、1 回のメッセージ転送にかかる software/hardware の over head を 5μsec、転送 throughput を 200Mbyte/sec と仮定する。

3. FT の並列化アルゴリズム

3.1 問題の概要

FT [3] は、3 次元空間の偏微分方程式を数値的に解く問題である。解法としては FFT を用いることが決められており、3 次元 FFT および逆 FFT をいかに高速化するかが問題となる。問題空間は $256 \times 256 \times 128$ の倍精度複素数で与えられ、各次元は N1,N2,N3 方向と呼ばれる。初期値は乱数で与えられ、その生成は N1,N2,N3 方向の順で作られるため、PU 内での 3 次元配列は N1 方向は連続に、N2,N3 方向はそれぞれ一定のストライドでメモリに配置される。

多次元の FFT のアルゴリズムは 1 次元 FFT の繰り返しに帰着できる。各次元方向の FFT はその次元方向のデータで閉じており、空間的な並列性がある。また、1 次元 FFT はバタフライ転送により並列化できる。これを踏まえて以下の 2 つの方式を評価することにする [4]。なお、ここでは、1024 PU ($8 \times 8 \times 16$) の 3 次元 HXB について、アルゴリズムの説明を行なうが、より少ない PU 数の場合も同様のアルゴリズムを用いる。

3.2 転置方式

各次元方向の 1 次元 FFT はキャッシュ上で処理できるデータサイズである。そこで、1 方向のデータが 1PU 内で閉じかつ連続になるように mapping する。このために、3 次元 HXB を仮想的に 2 次元 HXB に展開して用いる。$8 \times 8 \times 16$ PU は 仮想 32×32 PU となり、PU (x, y, z) と 仮想 PU (m, n) の対応は以下のようにする。

$$x \times 8 \times 16 + y \times 16 + z = m \times 32 + n$$

N1 方向の FFT を行なった後、N2 方向が 1PU 内で閉じるような mapping に変えるため、データ転送を行なう。この時の mapping の変化を図 1 左に示す。このデータ転送は転置転送であり、これを 3 回 (N1→N2→N3→N1) 行なう。転置転送 1 回あたりの各 PU から転送されるデータ量は、1PU の担当する全点のデータ (128KB) であり、3 回の転置で 384KB であるが、これは多くのストライド転送に分けられ、細粒度の転送となる。例えば $N1 \rightarrow N2$ の転置では、各 PU の転送先が 32PU あり、さらに同一の転送先 PU に対して、メモリ配置の関係により転送元の PU 内で N2 方向に並んだデータしか一度に転送できないので、32 回のストライド転送を行なう。このため、合計 4096 回の転送が必要となる。また、PU 数が少ない時には N3 方向の点数が増えるためにストライド転送をする回数が増える。他の方向の転置に関して同様の転送回数が必要となり、メッセージ転送に対する over head の累積が転送時間を長くする。

3.3 固定方式

問題空間を PU 空間へ固定的に mapping し、PU 間にまたがるデータの演算はバタフライ転送によって行なう。ここで、ネットワークをから到着したデータはメモリ上に置かれるため、メモリレイテンシを隠蔽して処理を行なうことが必要となる。そこで、PU 間の FFT 処理を PVP-SW を用いてベクトル化する。

PU 内の演算はメモリ配置を変えずに行なう。この結果 N1 方向は転置方式と同様にキャッシュが有効に使える。そこで N3 方向で問題空間を分割し、さらに PU が余る時には N2 方向で分割する。この結果、各 PU は $256 \times 8 \times 1$ のデータを持ち (図 1 右)、N1,N2,N3 方向でそれぞれ 0 回、3 回、7 回のバタフライ転送を行なう。全転送量は 1280KB となるが、この転送は PU 内の全点を相手 PU に送り、それぞれ 1 回の連続領域の転送なのでメッセージ転送の over head は問題とならない。

---

*Evaluation of PVP-SW and Hyper-Crossbar Network

†Ken'ich ITAKURA, Akira HIRONO, Taisuke BOKU, Hiroshi NAKAMURA, Kisaburo NAKAZAWA

‡Institute of Information Sciences and Electronice, University of Tsukuba

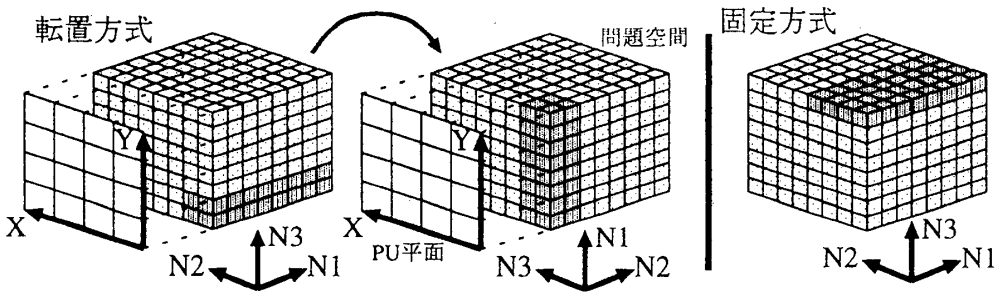

図 1:各方式の mapping

4. 評価と考察

評価の方法は、まず並列プログラムを PVM の上で作成し、内部処理のプログラムとネットワーク転送の方法を決定する。次に、内部処理のプログラムを先に述べたシミュレータにより評価し、処理時間を求める。ネットワークの転送時間は転送パターン、転送量、転送回数、およびネットワークの基本性能から算出する。2 つの方式の評価結果を表 1 に示す。なお、表 1 の exec は内部処理時間、trans はデータ転送時間、total は全処理時間を表し、括弧内は全処理時間に占める割合である。

転置方式の利点は FFT 処理がキャッシュを用いて高速に行なえる点で、欠点は転送が細粒度となり転送の over head が問題になる。これに対して、固定方式では転送は粗粒度であるが全転送量が増加する。また、処理が細粒度となりキャッシュが有効に働かない。しかし、固定方式のキャッシュの問題は、PVP-SW の機能でメモリレイテンシを隠蔽することにより解決される。

表 1 より、転置方式のデータ転送時間の割合が高く、ボトルネックとなっていることが分かる。内部処理時間は 1PU のデータサイズに比例して、PU 数が増えるとともに減少しているにもかかわらず、転送時間は 1/2 乗でしか減少しない。これは転送時間の半分以上を over head が占めているからである。これに対して、固定方式ではデータ転送の占める割合は低く、ボトルネックとなっていない。なお、1024PU のときに転送時間が比較的長いのはメッセージ長が短く over head の影響が出ているからである。2 つの方式について転送時間を比較してみると、同じ PU 数の時に固定方式の方が転送量が多いにも関わらず、短い時間で済む。

次に、図 2 に speedup のグラフを示す。これは固定方式の 64PU を 1 としてして描いたものである。2 つの方式は 256PU を境にして処理時間が逆転する。この原因は転置方式がどの方向に対しても同じ処理時間であるのに対して、固定方式では、N1 方向が転置と同じくキャッシュにより処理が行なわれ、N3 方向は PVP-SW の機能を用いたベクトル処理がなされているが、N2 方向での PU 内部がキャッシュに載らず、ボトルネックとなっている。しかし、PU 数が 512 を越えると N2 方向の内部処理もキャッシュに載るのでボトルネックが無くなり super linear の現象が起きるためである。

5. おわりに

本論文では PVP-SW と HXB を用いた並列計算機において多次元 FFT 問題を行うアルゴリズムを 2 つ示し、その評価をおこなった。PVP-SW はネットワークから受信したメモリ上にあるデータに対して、メモリレイテンシを隠蔽しベクトル処理する方法を提供する。この結果、2 つのアルゴリムの処理時間は転送パターンに大きく依存することが分かった。

謝辞

本研究に関し貴重な御意見を頂いた、筑波大学西川博昭助教授ならびに中澤研究室並列処理研究グループ諸氏に深く感謝します。なお、本研究の一部は文部省科学研究費 (奨励 (A)06780228) 及び創成的基礎研究費 (06NP0601) の補助による。

参考文献

[1] Nakamura, H. et al. “Evaluation of Pseude Vector Processor Based on Slide-Windowed Registers”, Proc. of the 27th Hawaii Int. Conf. on System Sciences, pp. 368-377 (1994)

[2] 朴 泰祐 他, “ハイパクロスバ・ネットワークにおける転送性能向上のための手法とその評価”, 並列処理シンポジウム JSPP ’94 pp. 129-136

[3] David Bailey et al. “The NAS Parallel Benchmarks ”, Report RNR-91-002 Revision 2, 22 August 1991

[4] 板倉 憲一 他 “ハイパクロスバ・ネットワークにおける NAS ベンチマークの評価”, 情処研報 94-HPC-52-20, 1994 年

表 1:各アルゴリズムの処理時間 (単位は sec)

	PU 数	exec	(%)	trans	(%)	total
転	64	5.52	(87.4)	0.79	(12.6)	6.31
	256	1.36	(80.0)	0.34	(20.0)	1.71
置	1024	0.33	(68.0)	0.16	(32.0)	0.49
	64	10.77	(96.1)	0.44	(3.9)	11.21
固	128	4.38	(94.4)	0.26	(5.6)	4.64
	256	1.65	(91.8)	0.15	(8.2)	1.80
定	512	0.49	(85.5)	0.08	(14.5)	0.57
	1024	0.24	(83.7)	0.05	(16.3)	0.29

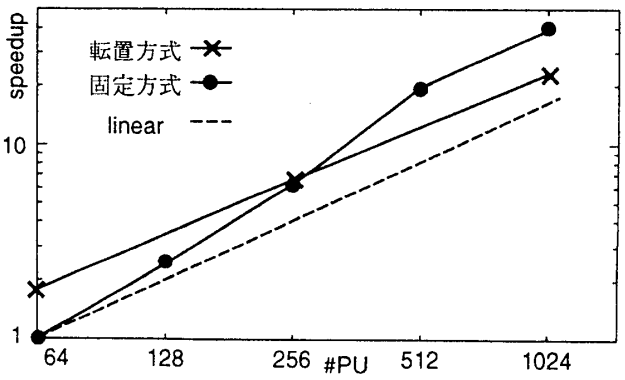

図 2:固定方式の 64PU の処理時間を 1 とした speedup